Université Toulouse III — École Doctorale Informatique et Télécommunications

Noyaux de séquences pour la vérification du locuteur par Machines à Vecteurs de Support

THÈSE

présentée et soutenue publiquement le 25 janvier 2007

pour l'obtention du

Doctorat de l'Université Toulouse III – Paul Sabatier

(spécialité informatique)

par

Jérôme Louradour

Composition du jury

Président :	Christian Wellekens
Rapporteurs :	Samy Bengio Jean-François Bonastre
Examinateur :	Jean-Paul Haton
Directeurs de thèse :	Régine André-Obrecht Khalid Daoudi

Institut de Recherche en Informatique de Toulouse — UMR 5505

Institut de Recherche en Informatique de Toulouse

À papi André,
pour toujours dans ma mémoire...

Remerciements

À ceux qui m'ont soutenu pendant les trois années consacrées à cette thèse

Merci à Khalid pour avoir éclairci mes réflexions quand les directions à prendre étaient floues et pour avoir favorisé la mise en avant de mes travaux. Merci à Régine pour avoir été particulièrement à l'écoute, pour son empathie et ses mots justes, et enfin pour avoir montré un bel exemple en s'impliquant pleinement dans la vie du laboratoire. Merci à eux deux pour avoir toujours cru en moi et merci à tous les membres du jury pour les encouragements qu'ils ont formulés lors de la soutenance qui aura été, grâce à eux, un beau point final à cette longue aventure.

Merci à mes parents, Gaëlle, Maëla, Elie, Sev, Théo... pour leur attention et leur réconfort dans les moments les plus sombres. Merci aux doctorants de l'IRIT avec qui nous nous sommes serrés les coudes : Jean-Luc, Fred, Nissou, Ivan, Tiago, Sylvie, José, Eduardo, Jean-Léon, Karine... Merci à Manu Ferragne, Luca, Jonas, Alberto, Nico... et tous les jeunes chercheurs qui m'ont motivé par leur enthousiasme lors des conférences.

À ceux qui m'ont aidé dans mon travail de thèse

Merci à Régine pour m'avoir fait partager à la fois son expérience sur la parole et ses intuitions. Merci à Khalid pour m'avoir montré ce qu'était l'efficacité au travail. Merci à Samy et Jean-François pour les remarques constructives qui m'ont permis de rendre le présent manuscript plus propre et rigoureux. Merci à Francis pour m'avoir apporté des connaissances avancées en Machine Learning. Merci à Fred, Jean-Luc, Julien et Jérôme pour tous les tuyaux en programmation, ainsi qu'à l'équipe du service informatique de l'IRIT pour leur aide et leur disponibilité. Merci (encore) à Jean-Luc pour sa générosité dans le travail et pour la collaboration avec l'INRETS. Merci à Hélène, Gaëlle et Alain pour la relecture syntaxique de la thèse. Merci à tous les conférenciers avec qui j'ai échangé oralement et par mail, en particulier JeF. Bonastre, N. Brümmer, W. Campbell, C. Fredouille, R. Kondor, J. Mariéthoz, J. Pelecanos, N. Scheffer, et V. Wan. Merci à tous ces gens pour avoir défendu à mes yeux une bonne image du monde de la recherche, fait d'entraide et de sincérité. Merci enfin aux gens qui ont contribué et contribuent encore à Linux et LaTeX.

À ceux qui ont indirectement aidé au bon déroulement de ma thèse en m'apportant un équilibre dans les à-côtés

Merci à Gaëlle pour son amour, sa fidélité, sa simplicité et son dévouement au quotidien. Merci à mes parents et mes grands-parents pour leur amour d'une autre nature mais tout aussi généreux. Merci à Maëla et Elie pour avoir été des collocataires idéaux, et à Lula et Gunga pour leur affection. Merci à eux ainsi qu'à Max, Gilles, Hugues, Rémi, Tibor, Marc, Philou, Giov&Çağla, Soussoun&Barbara, Cousines, Sev, Théo... pour m'avoir aidé à comprendre et à rester celui que je suis. Obrigado à tous les joueurs de capoeira qui m'ont fait aimer cette discipline, ainsi qu'aidé à approfondir mon apprentissage qui ne s'arrêtera jamais. Merci à Sassá et son groupe d'Itatiba pour m'avoir formidablement bien accueilli et initié, et à ceux qui m'ont ensuite accompagné et fait plaisir : Pensador, Gildas, Pauline, David, Loïc, Guilhem, Palito, Christi, Reny&René, Dirceu et ses élèves à Madrid...

Merci enfin à tous ceux que j'ai omis de citer de ne pas m'en vouloir.

Sommaire

Abréviations & Notations

Liste des abréviations utilisées

ANN	*Artificial Neural Networks.*
BIC	*Bayesian Information Criterion.*
DBN	*Dynamic Bayesian Network.*
DET	*Detection Error Trade-off (curve).*
EM	*Expectation-Maximization (algorithm).*
FSMS	*Feature Space Mahalanobis Sequence (kernel).*
FSNS	*Feature Space Normalized Sequence (kernel).*
GLDS	*Generalized Linear Discriminant Sequence (kernel).*
GLLR	*Generalized Log-Likelihood Ratio.*
GMM	*Gaussian Mixture Model.*
HMM	*Hidden Markov Model.*
ICD	*Incomplete Cholesky Decomposition.*
LDA	*Linear Discriminant Analysis.*
LLR	*Linear Logistic Regression.*
MAP	*Maximum A Posteriori.*
MFCC	*Mel Frequency Cepstral Coefficient.*
MI	*Mutual Information (kernel).*
MLP	*Multi-Layer Perceptron.*
LPCC	*Linear Predictive Coding Coefficient.*
NAP	*Nuisance Attribute Projection.*
NIST (SRE)	*National Institute of Standards and Technology (Speaker Recognition Evaluation).*
PLPC	*Perceptual Linear Prediction Coefficient.*
RBF	*Radial Basis Function.*
RKHS	*Reproducing Kernel Hilbert Space.*
ROC	*Receiver Operating Characteristic (curve).*
SMO	*Sequential Minimisation Optimisation.*
SVD	*Singular Value Decomposition.*
SVM	*Support Vector Machine.*
TOP	*Tangent vector Of Posterior log-odds (kernel).*
UBM	*Universal Background Model.*

Notations mathématiques et conventions

— Ensembles et calcul matriciel —

$\mathbb{R}$	Ensemble des réels.
$\mathbb{R}^+$	Ensemble des réels positifs ou nuls.
$\mathrm{x} \in \mathbb{X}$	Donnée appartenant à un ensemble quelconque (en caractère normal).
$\mathbf{M}^{\mathrm{T}}$	Transposée de la matrice $\mathbf{M}$ (matrices en lettres capitales et en gras).
$\mathbf{M}^{-1}$, $\mathbf{M}^{\dagger}$	Inverse de $\mathbf{M}$, pseudo-inverse de $\mathbf{M}$ (annexe A.1.1).
$\det(\mathbf{M})$, $\mathrm{tr}(\mathbf{M})$	Déterminant de $\mathbf{M}$, trace de $\mathbf{M}$.
$rang(\mathbf{M})$	Rang de $\mathbf{M}$.
$\mathbf{x} = \begin{bmatrix} x_1 \cdots x_d \end{bmatrix}^{\mathrm{T}}$	Vecteur colonne de dimension d (vecteurs en gras, scalaires en italique).
$\mathbf{I}_D$	Matrice identité de dimension D.
$\mathbf{1}_D$	Vecteur de dimension D rempli de 1.

— Données de travail —

$\mathcal{X} = \{\mathrm{x}_t\}_{t=1\cdots T_{\mathrm{X}}}$	Séquence de données, de longueur T_{X}.
$\mathbf{X} = \{\mathbf{x}_t\}$	Séquence de vecteurs.
$\mathcal{B} = \{\mathrm{b}_i\}_{i=1\cdots N}$	Ensemble de données non étiquetées (corpus du "Monde"), de taille N.
$\mathcal{A} = \{\mathrm{a}_i, \ell_i\}$	Ensemble de données étiquetées, corpus d'apprentissage.
$\mathbf{A} = \{\mathbf{a}_i, \ell_i\}$	Ensemble de vecteurs étiquetés.

— Classification, probabilités et méthodes statistiques —

ℓ	Classe, étiquette (variable aléatoire).
$\mathrm{p}(\mathrm{x}\|\ell = c)$	Probabilité d'observer x sachant que la classe est c.
$\mathrm{E}[\mathrm{x}]$	Espérance mathématique de x.
f_θ *ou* $f(\mathrm{x}\|\theta)$	Fonction discriminante paramétrée par θ.
p_θ *ou* $p(\mathrm{x}\|\theta)$	Modèle de distribution probabiliste paramétré par θ.
$\mathcal{N}_{\boldsymbol{\mu},\boldsymbol{\Sigma}}$ *ou* $\mathcal{N}(\mathbf{x}\|\boldsymbol{\mu},\boldsymbol{\Sigma})$	Gaussienne de moyenne $\boldsymbol{\mu}$ et covariance $\boldsymbol{\Sigma}$.
$\boldsymbol{\nabla}_{\boldsymbol{\theta}}\, l(\boldsymbol{\theta})\|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\mathrm{o}}}$	Gradient de $l(\boldsymbol{\theta})$ par rapport à $\boldsymbol{\theta}$ estimé en $\boldsymbol{\theta}_{\mathrm{o}}$ ($\frac{\partial}{\partial\theta_u}$ pour les dérivées partielles).

— Méthodes à noyau —

$\mathbf{x}^{\mathrm{T}}\mathbf{y}$	Produit salaire entre $\mathbf{x}$ et $\mathbf{y}$.
$k(\mathrm{x},\mathrm{y})$	Noyau entre éléments de dimension fixe.
$\kappa(\mathcal{X},\mathcal{Y})$	Noyau entre ensembles de taille variable.
$\mathbf{K}$	Matrice de Gram (§3.1.1, définition 2).
$\mathbf{K}_{\mathrm{X}}$	Matrice de Gram sur un ensemble $\mathcal{X}$, sur une séquence de vecteurs $\mathbf{X}$.
$\boldsymbol{\phi}$	Expansion correspondant à un noyau de Mercer (§3.1.2, théorème 2) : $k(\mathrm{x},\mathrm{y}) = \boldsymbol{\phi}(\mathrm{x})^{\mathrm{T}}\boldsymbol{\phi}(\mathrm{y})$.
$\boldsymbol{\psi}_{\mathrm{B}}$	Expansion empirique sur un ensemble $\mathcal{B}$ (§4.3, définition 8).
D	Dimension du *Feature Space* : $\boldsymbol{\phi}(\mathrm{x}) \in \mathbb{R}^D$.

Table des figures

Liste des tableaux

Résumé

La vérification automatique du locuteur (VAL) est une tâche de classification binaire, qui consiste à déterminer si un énoncé de parole a été prononcé ou non par un locuteur cible. Les Machines à Vecteurs de Support (SVMs) sont devenues un outil classique pour ce type de classification. Cette approche discriminante a suscité l'intérêt de nombreuses recherches en reconnaissance des formes, tant pour ses fondements théoriques solides que pour ses bonnes performances empiriques. Mais la mise en œuvre des SVMs pour la VAL en situation réelle soulèvent plusieurs problèmes relatifs aux caractéristiques propres à cette tâche. Il s'agit principalement de la taille élevée des corpus d'apprentissage et de la nature séquentielle des observations à classifier.

Cette thèse est consacrée à l'exploration des noyaux de séquences pour la classification SVM du locuteur. Nous commençons par faire un tour d'horizon des méthodes émergentes pour construire des noyaux de séquences. Ensuite nous proposons une nouvelle famille de noyaux en se basant sur une généralisation d'un noyau qui a fait ses preuves en VAL. Nous faisons l'analyse théorique et algorithmique de cette nouvelle famille avant de l'appliquer à la VAL par SVM. Après la mise en œuvre des systèmes SVMs à base des différents noyaux que nous avons étudiés, nous comparons leurs performances sur le corpus NIST SRE 2005, à partir d'un protocole de développement commun. Enfin, nous introduisons un nouveau concept pour aborder le problème de VAL, dont le principe est de déterminer si deux séquences ont été prononcées par le même locuteur. L'utilisation des SVMs pour exploiter ce concept nous amène à définir une nouvelle catégorie de noyaux : les noyaux entre paires de séquences.

Mots-clés: Vérification du locuteur, méthodes à noyau, noyaux de séquences, noyaux d'ensembles, Machines à Vecteurs de Support.

Abstract

This thesis is focused on the application of Support Vector Machines (SVM) to Automatic Text-Independent Speaker Verification. This speech processing task consists in determining whether a speech utterance was pronounced or not by a target speaker, without any constraint on the speech content. In order to apply a kernel method such as SVM to this binary classification of variable-length sequences, an appropriate approach is to use kernels that can handle sequences, and not acoustic vectors within sequences. As explained in the thesis report, both theoretical and practical reasons justify the effort of searching such kernels. The present study concentrates in exploring several aspects of kernels for sequences, and in applying them to a very large database speaker verification problem under realistic recording conditions.

After reviewing emergent methods to conceive sequence kernels and presenting them in a unified framework, we propose a new family of such kernels : the Feature Space Normalized Sequence (FSNS) kernels. These kernels are a generalization of the GLDS kernel, which is now well-known for its efficiency in speaker verification. A theoretical and algorithmic study of FSNS kernels is carried out. In particular, several forms are introduced and justified, and a sparse greedy matrix approximation method is used to suggest an efficient and suitable implementation of FSNS kernels for speaker verification.

Several SVM systems, based on different kinds of sequence kernels, are developed on a common validation corpus and compared on the NIST Speaker Recognition Evaluation 2005. These experiments, led on a large corpus of telephone conversational speech, show the relevance of the FSNS kernels. They also give an insight on how to tune sequence kernels to build robust SVM system for speaker verification. In comparison with state-of-the-art methods based on Gaussian Mixture Models (GMM) densities estimation, SVM system show competitive and better performance while the computational complexity is reduced. Finally, experiments show that fusing generative GMM approach with discriminante SVM approach improves the result, which confirms that the two approaches are complementary.

At last, a new way of addressing the speaker verification task is introduced, with the proposition of a SVM classifier whose goal is to determine whether two sequences were pronounced by the same speaker. To conceive such a system, a novel type of kernel is suggested : a kernel between pairs of sequences, inspired by promising studies on sequence processing with kernel methods. Experiments on NIST database show promising results for the new approach, which is also confirmed to be complementary with traditional approaches. We underline the fact that the new system is constructed from a single universal model, trained to be speaker-discriminante. By this way, it allows to construct really efficient approaches for automatic speaker recognition.

Keywords: Text-independent speaker verification, kernel methods, sequence kernels, kernels between sets, support vector machine.

Introduction

Contexte applicatif

Le but général du *traitement de la parole* est d'extraire automatiquement d'un signal numérique de parole des informations de haut niveau, c'est-à-dire facilement interprétables par l'être humain. Ce domaine de recherche a connu une progression rapide avec de nombreuses applications à la clé, dont la partie la plus visible a été l'apparition de logiciels de dictée vocale. Ces systèmes visent à transcrire le message linguistique véhiculé par la parole dans un code graphique qui ait un sens pour l'homme : les mots. Le signal de parole contient d'autres types d'information que ceux portant sur le message, comme l'identité de l'individu qui a prononcé le message. Cette problématique est le centre d'intérêt de la *reconnaissance du locuteur*, domaine qui connaît plusieurs sous-branches selon le contexte applicatif visé. Une formulation du problème, qui a fait l'objet d'une grande partie des recherches les plus récentes dans le domaine, est la *vérification automatique du locuteur* (VAL). Cette tâche consiste à **déterminer si un énoncé de parole a été prononcé ou non par un individu donné** que l'on désigne par "locuteur cible". Cela suppose de disposer d'un ou plusieurs autres enregistrements vocaux du locuteur cible. Les applications de la VAL sont multiples :

- *Sécurisation*
  Contrôle d'accès, pour l'entrée dans un lieu collectif (*ex :* bâtiment) ou pour la transmission d'informations personnelles (*ex :* transactions bancaires) ;
- *Criminalistique et renseignements*
  Tri automatique des communications téléphoniques pour obtenir des informations sur un individu (*ex :* recherche d'un terroriste) ;
- *Juridique*
  Expertise vocale d'un appel téléphonique pour orienter une enquête à partir de cet indice ;
- *Domotique*
  Reconnaissance automatique de l'utilisateur pour l'application de paramètres personnalisés (*ex :* réglages au volant d'une voiture.

La VAL s'inscrit dans le cadre général de la "biométrie", qui désigne l'étude d'un signal généré par l'humain pour déterminer l'identité d'un personne. Par rapport aux autres modalités de la biométrie (empreintes, ADN, etc.) le principal avantage de la parole est la facilité à recueillir des échantillons, que ce soit à l'insu ou au su du locuteur cible, grâce aux développements des technologies vocales et à la grande utilisation de la téléphonie. En contrepartie, l'inconvénient est que le signal aléatoire de parole est sujet à de nombreuses sources de variabilité extrinsèques à l'identité du locuteur qui nuisent à la robustesse des systèmes de VAL. Les principales sources de variabilité sont liées aux conditions d'enregistrement, aux états émotionnels du locuteur et

aux messages prononcés.

Pour des applications où les locuteurs cibles sont coopératifs, comme la sécurisation d'un bâtiment et la domotique, on peut envisager l'utilisation d'un mot de passe connu par le système. Ce contexte permet d'augmenter la fiabilité des systèmes de VAL, mais il est trop contraignant d'un point de vue applicatif. Dans cette étude, nous traitons le cas où aucune information a priori sur le contenu prononcé n'est disponible. Il s'agit de la VAL en mode **indépendant du texte** (IT). Un grand intérêt a été porté à ce contexte applicatif, comme en témoigne la prolifération les logiciels libres pour la VAL en mode IT[1], ainsi que la participation croissante des laboratoires aux évaluations de reconnaissance du locuteur organisées par NIST. Dans ces évaluations, la tâche de référence est la VAL en mode IT, avec, pour caractériser chaque locuteur cible, un seul enregistrement correspondant à une conversation téléphonique d'environ cinq minutes. Ce protocole expérimental servira de base à notre travail, notamment pour l'étude des complexités calculatoires.

Cadre théorique

Comme le traitement de la parole en général, la VAL est un sujet pluridisciplinaire qui utilise entre autres des éléments du traitement du signal, de la phonétique, de la linguistique, de la modélisation statistique et de l'algorithmique. Dans cette étude, nos recherches portent sur ces deux derniers aspects, regroupés dans une discipline couramment appelée "*machine learning*". Nous nous intéressons aux différentes stratégies pour "apprendre" à un ordinateur comment reconnaître un locuteur à partir de caractéristiques extraites du signal de parole.

De façon formelle, la VAL est un problème de classification binaire avec apprentissage supervisé ou semi-supervisé : le classifieur est une fonction qui renvoie une décision d'acceptation ou de rejet pour une séquence de test et un locuteur cible donné. Pour optimiser la décision, les paramètres du classifieur sont réglés à partir de données relatives au locuteur cible et de données réunissant un certain nombre d'autres locuteurs. Les séquences de parole de ce second corpus peuvent être étiquetées ou non selon les locuteurs.

De manière générale, choisir un classifieur pour un problème donné se fait en adoptant une modélisation et en réglant les paramètres libres des modèles à partir de données d'apprentissage. L'adéquation d'une modélisation et d'une technique d'apprentissage à un problème donné tient à plusieurs critères :

1. *La nature des données d'entrées.*
   La palette de choix de la modélisation varie selon que les données sont symboliques et/ou numériques. Si les données sont structurées, il faut utiliser une famille de modèles permettant d'exploiter leur structure, et une technique d'apprentissage permettant de capturer l'information discriminante contenue dans la structure.
2. *Le volume de données : nombre de paramètres, nombre d'observations en apprentissage*
   a) Dans le cas où le corpus d'apprentissage est réduit, les modèles doivent être en mesure d'extrapoler l'information manquante.
   b) Certaines techniques peuvent être trop lourdes à mettre en œuvre en "temps réel" lorsque le nombre de paramètres d'entrée et/ou de données d'apprentissage est trop

[1] [Imbiriba et al., 2004, Blouet et al., 2004, Bonastre et al., 2005]

important. L'efficacité d'une technique de classification se juge par la complexité calculatoire de l'apprentissage et plus encore de la prise de décision.

3. *Le comportement des paramètres mesurés en fonction des classes.*
   La modélisation choisie doit être en mesure de capturer correctement le "pouvoir discriminant" des paramètres d'entrée pour donner de bonnes performances.

Au regard de ces différents aspects, la tâche de VAL à partir de données téléphoniques constitue une véritable mise à l'épreuve des différentes techniques de modélisation et d'apprentissage pour la classification binaire :

1. Les entités à classer sont des séquences de longueurs variables. Dans notre étude expérimentale, les séquences sont des suites de vecteurs caractéristiques (dimension fixe) composés de paramètres acoustiques. Aussi nous nous limitons à des approches où l'ordre des vecteurs dans la séquence n'est pas pris en compte par la modélisation, comme c'est le cas dans les approches classiques pour la VAL en mode IT. Toutes les approches étudiées dans ce mémoire sont des méthodes de classification d'ensembles numériques de tailles variables. Pour faciliter la lecture, nous désignerons ces ensembles de vecteurs par "séquences", étant donné que les entrées à classer dans la VAL sont des séquences de parole.
2. a) Dans le protocole expérimental que nous envisageons, les données du locuteur cible disponibles pour apprendre le classifieur sont peu nombreuses. Elles correspondent à environ deux minutes seulement de parole pure.

   b) D'un autre coté, pour tenir compte de la richesse phonétique et linguistique, il est nécessaire de prendre en compte des corpus de parole volumineux, faisant intervenir une population variée de locuteurs.
3. Les paramètres acoustiques sur lesquels nous nous basons ont des distributions fortement multimodales. Aussi les variations de distributions selon les locuteurs sont complexes à saisir. De plus, ils sont sujets à des conditions d'observation très variables et sont corrompus par un bruit de niveau relativement élevé (enregistrements téléphoniques de basse qualité).

Notons aussi que chaque locuteur cible définit un problème de classification binaire. Les évaluations NIST mettent en jeu plusieurs centaines de locuteurs cibles et plusieurs milliers de tests avec des conditions d'enregistrement et des sujets de conversation variés. Ces évaluations à grande échelle permettent ainsi une comparaison fiable des méthodes pour la classification de séquences de longueurs variables.

Objectifs

L'objectif général de cette étude est de proposer des modélisations pour la VAL qui soient des alternatives aux approches classiques. Nous expliquons maintenant les directions que nous avons suivies.

La plupart des systèmes de VAL en mode IT sont basés sur la théorie des probabilités. Ils manipulent les séquences d'apprentissage à travers les distributions des vecteurs qu'elles contiennent. Ces distributions sont estimées par le biais de modèles statistiques paramétriques, qui ont l'avantage de permettre une extrapolation de l'information manquante et de traiter de grandes bases de données avec des ressources mémoire limitées. La méthode de référence pour la VAL en mode IT est une approche de type "*générative*" basée sur les GMMs, connu sous le sigle UBM-GMM. Pour résoudre le problème de classification, l'UBM-GMM résout un problème

intermédiaire plus général (l'estimation de densités *a priori*), contrairement aux méthodes dites "*discriminantes*" pour lesquelles la phase d'apprentissage se focalise sur les relations *entrées-sorties* des données. Notons aussi que les principes probabilistes théoriques à l'origine de l'UBM-GMM ne sont en pratique pas rigoureusement respectés, ceci afin d'améliorer les performances empiriques. De nombreux artefacts ont été proposés pour accroître le pouvoir discriminant de l'UBM-GMM, qui n'est plus purement génératif dans son implémentation courante.

Parmi les méthodes discriminantes, nous nous sommes intéressés dans cette étude à une méthode émergente : les Machines à Vecteurs de Support (SVMs). Cette méthode à noyau est maintenant bien maîtrisée pour des "petits" problèmes et a très vite connu un grand succès pour de nombreuses applications où les entités à classer sont des vecteurs de taille fixe. Sa mise en œuvre pour la VAL en mode IT est plus délicate à cause des gros volumes de données à traiter, et de la nature séquentielle de la parole. Un axe de recherche intéressant pour l'application des SVMs à la VAL est la conception et le choix de noyaux qui permettent de représenter les séquences de parole de manière adéquate dans la modélisation par SVM. Les *noyaux de séquences*, qui sont des fonctions scalaires s'appliquant à des séquences et vérifiant certaines propriétés mathématiques, font l'objet de cette étude.

Principales contributions

Les contributions de ce travail de thèse peuvent être résumées en trois points principaux :

1. L'exploration des méthodes disponibles pour concevoir des noyaux de séquences, et la comparaison empirique des noyaux de séquences en VAL, parmi ceux qui induisent une complexité calculatoire acceptable pour les protocoles d'évaluation NIST.
2. La formalisation et la mise au point d'une nouvelle famille de noyaux de séquences qui ne sont pas basés sur le cadre probabiliste. Ils sont une généralisation d'un noyau déjà existant et reconnu en VAL : le noyau GLDS. Pour réduire la complexité calculatoire initialement élevée, des approximations sont faites sans compromettre les fondements théoriques. Les nouveaux noyaux sont appliqués avec succès à la VAL en mode IT ; Les systèmes SVMs "purement discriminatifs" basés sur ces noyaux montrent des performances équivalentes aux autres systèmes basés de près ou de loin sur la modélisation probabiliste.
3. La proposition d'une nouvelle façon d'aborder le problème de VAL et d'un nouveau type de noyaux pour appliquer les SVMs dans ce nouveau contexte : les noyaux entre paires de séquences. Un tel noyau est conçu et appliqué à la VAL. Même si les performances n'égalent pas encore celles des systèmes classiques, elles sont acceptables et l'approche est prometteuse pour de futures améliorations.

Organisation du mémoire

Ce document s'organise en six chapitres. Les principales contributions de la thèse sont présentées dans les quatre derniers chapitres.

Le premier chapitre introduit des notions pour la classification de données numériques avec apprentissage supervisé, en s'intéressant tout particulièrement aux différences entre les méthodes génératives et les méthodes discriminantes. Il présente aussi la théorie de Vapnik et la formulation

des SVMs tels que nous les utilisons pour la classification en VAL. Dans le second chapitre sont décrits les systèmes classiques de VAL ainsi que quelques systèmes SVMs (non classiques) qui ont été conçus pour la VAL.

Le chapitre 3 fait un tour d'horizon des méthodes disponibles pour concevoir des noyaux de séquences, et présente les fondements mathématiques sous-jacents. Le chapitre 4 présente une nouvelle famille de noyaux de séquences, généralisant le noyau GLDS et n'utilisant pas la modélisation probabiliste. Il expose aussi une façon astucieuse de réduire à souhait la complexité calculatoire de ces noyaux. Le chapitre 5 présente la mise au point de plusieurs systèmes SVM pour la VAL, chacun étant basé sur un type de noyau particulier. Les performances empiriques sont comparées sur une évaluation NIST SRE. Enfin le chapitre 6 explore une nouvelle façon d'aborder le problème de VAL, et montre les performances d'un nouveau système SVM avec un noyau entre paires de séquences.

L'ensemble du document est conclu par une récapitulation des principales contributions de cette thèse et par la perspective de quelques perspectives de développements supplémentaires de ces travaux.

Chapitre 1

Classification supervisée de données numériques

Sommaire

DANS ce chapitre, nous introduisons les méthodes statistiques pour la classification automatique avec apprentissage supervisé. Nous commençons par présenter de manière générale deux catégories d'approches correspondant à des démarches fondamentalement différentes pour aborder le problème de classification (§1.1). La première réunit les méthodes dites "génératives", toutes dérivées de la théorie des probabilités, et parmi lesquelles on compte la plupart des approches classiques pour diverses tâches d'indexation automatique de la parole. La seconde réunit les méthodes dites "discriminantes", qui ont connu un succès croissant pour de nombreuses tâches de classification. Pour chacune de ces deux catégorie, nous donnerons des exemples de méthodes couramment utilisées (§1.2). Nous décrirons plus en détail une méthode discriminante émergente que nous appliquerons à la vérification du locuteur : les Machines à Vecteurs de Support (§1.3).

1.1 Approches génératives & approches discriminantes

Nous commençons par présenter les différences fondamentales entre deux catégories d'approches pour traiter un problème de classification avec apprentissage supervisé :

- les approches dites "génératives", ou "informatives", qui incluent l'Analyse Discriminante Linéaire, les Modèles de Mélanges de Gaussiennes, les Modèles de Markov Cachés et les Réseaux Bayésiens.
- les approches dites "discriminantes", qui incluent la méthode des *k-plus proches voisins*, la Régression Logistique, les Réseaux de Neurones et les Machines à Vecteurs de Support.

Ces deux catégories seront comparés à travers plusieurs parallèles théoriques (§1.1.1 et §1.1.2), et nous donnerons une vue d'ensemble sur la combinaison des deux types d'approches (§1.1.3).

1.1.1 Interprétation probabiliste

De façon formelle, un classifieur assigne à une observation x une étiquette $\boldsymbol{\ell} \in \{1, \cdots, n_c\}$ correspondant à une classe (n_c désigne le nombre de classes). Pour mesurer les performances d'un classifieur dans un contexte applicatif donné, il faut établir une mesure de coût des erreurs, liée à l'application. Cette mesure $\tau : (\boldsymbol{\ell}_r, \boldsymbol{\ell}_s) \in \{1, \cdots, n_c\}^2 \rightarrow \mathbb{R}^+$, peut se représenter sous forme d'une matrice liant les sorties $\boldsymbol{\ell}_s$ du classifieur et les étiquettes réelles $\boldsymbol{\ell}_r$. Un cas particulier est le coût binaire 0/1, qui vaut 0 si $\boldsymbol{\ell}_r = \boldsymbol{\ell}_s$ et 1 sinon. Cette fonction associe la même gravité à tous les types d'erreurs, et correspond à une matrice de coût dont les valeurs sont nulles sur la diagonale.

Probabilités et Classification

Selon la théorie des probabilités, chaque observation est générée par une variable aléatoire, dont la distribution peut se décomposer d'après les règles de Bayes :

$$\mathrm{p}(\mathrm{x}, \boldsymbol{\ell}) = \mathrm{p}(\boldsymbol{\ell}|\mathrm{x})\mathrm{p}(\mathrm{x}) = \mathrm{p}(\mathrm{x}|\boldsymbol{\ell})\mathrm{p}(\boldsymbol{\ell}) \tag{1.1}$$

Le but d'un classifieur est de minimiser l'espérance du coût des erreurs, désignée par "risque global". Le classifieur de Bayes idéal est alors celui qui renvoie les valeurs :

$$\boldsymbol{\ell}_r^*(\mathrm{x}) = \arg\min_{\boldsymbol{\ell}_s} \sum_{c=1}^{n_c} \tau(c, \boldsymbol{\ell}_s)\mathrm{p}(\boldsymbol{\ell} = c|\mathrm{x}) = \arg\min_{\boldsymbol{\ell}_s} \sum_{c=1}^{n_c} \tau(c, \boldsymbol{\ell}_s)\mathrm{p}(\mathrm{x}, \boldsymbol{\ell} = c) \tag{1.2}$$

Pour la fonction de coût 0/1, cela revient à choisir la classe c qui maximise la probabilité *a posteriori* $\mathrm{p}(c|\mathrm{x})$. En pratique, les véritables densités sont inconnues et l'on dispose d'observations d'apprentissage $\{\mathrm{a}_i, \boldsymbol{\ell}_i\}$ à partir desquelles on instancie des modèles paramétriques.

Approches génératives

Les approches génératives regroupent des méthodes qui utilisent les données d'apprentissage pour modéliser les densités de probabilité $\mathrm{p}(\mathrm{x}|\boldsymbol{\ell})$ de chaque classe par une famille de fonctions paramétriques. Lorsque le protocole expérimental le permet, ces méthodes peuvent aussi facilement

tenir compte des probabilités *a priori* $\mathrm{p}(\boldsymbol{\ell})$ d'apparition de chaque classe. En cas d'ignorance, il est d'usage de considérer les classes comme équiprobables *a priori*. Le terme "génératif" désigne le fait que la règle de décision, déduite d'après les relations de Bayes (1.1), soit basée sur une modélisation de la probabilité $\mathrm{p}(\mathrm{x}|\boldsymbol{\ell})$ qui "génère" les observations x pour une classe $\boldsymbol{\ell}$ donnée.

Les points importants d'un apprentissage supervisé selon le paradigme génératif sont les suivants :

1. Une famille de fonctions paramétriques est choisie pour modéliser la distribution de chaque classe : $p_\theta(\mathrm{x}|\boldsymbol{\ell} = c) = p_{\boldsymbol{\theta}_c}(\mathrm{x})$, où $\boldsymbol{\theta}_c$ est un jeu de paramètres réglé uniquement pour la classe c. On note aussi $p_\theta(\boldsymbol{\ell} = c) = p_c$ la probabilité *a priori* d'apparition de chaque classe. Les paramètres libres de la modélisation sont $\theta = \{\boldsymbol{\theta}_1, p_1, \cdots, \boldsymbol{\theta}_{n_c}, p_{n_c}\}$, ou simplement $\theta = \{\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_{n_c}\}$ si les p_c sont fixés arbitrairement (par exemple, $p_c = 1/n_c$). De tels choix font appel à des hypothèses plus ou moins réalistes.
2. Les paramètres libres θ sont réglés en maximisant la vraisemblance des données d'apprentissage étiquetées $\{\mathrm{a}_i, \boldsymbol{\ell}_i\}$:

$$\begin{aligned} \theta^* &= \arg\max_\theta \sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(\mathrm{a}_i, \boldsymbol{\ell}_i) = \arg\max_\theta \sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}\left(\mathrm{a}_i|\boldsymbol{\ell}_i\right) p(\boldsymbol{\ell} = \boldsymbol{\ell}_i) \\ &= \arg\max_\theta \sum_{c=1}^{n_c} \sum_{\{\mathrm{x}_i|\boldsymbol{\ell}_i = c\}} \log p_{\boldsymbol{\theta}_c}\left(\mathrm{x}_i\right) p_c \end{aligned} \tag{1.3}$$

3. Une règle de décision est induite d'après les relations de Bayes, conformément à (1.2).

Approches discriminantes

Les approches discriminantes regroupent une variété de méthodes statistiques qui utilisent les données d'apprentissage pour construire directement une correspondance entre les entrées X et les sorties Y. Il est souvent dit que ces méthodes modélisent directement la probabilité *a posteriori* $\mathrm{p}(\boldsymbol{\ell}|\mathrm{x})$. Mais en fait, rares sont les méthodes discriminantes (au sens de "non génératives") fondées sur la théorie probabiliste. La plupart du temps, l'accès à la probabilité *a posteriori* n'est pas trivial et nécessite des hypothèses supplémentaires à celles faites lors de l'apprentissage du classifieur discriminatif. De manière générale, nous désignons par approches "discriminantes" les méthodes qui ne s'intéressent qu'aux relations d'entrée-sortie $\mathrm{a}_i \overset{?}{\mapsto} \boldsymbol{\ell}_i$ lors de l'apprentissage.

Un exemple vectoriel

Parmi les méthodes discriminantes qui modélisent la probabilité *a posteriori* $\mathrm{p}(\boldsymbol{\ell}|\mathbf{x})$, on peut citer la Régression Linéaire Logistique (LLR). Cette méthode prédit une étiquette binaire ± 1 en supposant une loi binomiale impliquant $(d+1)$ paramètres $\theta = \{\boldsymbol{\theta}, \beta_0\}$:

$$p_\theta(\boldsymbol{\ell} = +1|\mathbf{x}) = \frac{1}{1 + e^{-(\boldsymbol{\theta}^{\mathrm{T}}\mathbf{x} + \beta_0)}} \tag{1.4}$$

Ce type de modélisation conduit à une frontière de décision linéaire d'équation $p_\theta(\boldsymbol{\ell} = +1|\mathbf{x}) = 1/2$ (ce qui implique $\boldsymbol{\theta}^{\mathrm{T}}\mathbf{x} = cte$). L'approche générative conduisant au même type de surface de

séparation est l'Analyse Discriminante Linéaire (LDA), méthode très populaire introduite pas Fisher en 1936 [Fisher, 1936]. Cette dernière suppose une distribution Gaussienne par classe, avec une matrice de covariance commune à toutes les classes :

$$p_\theta(\mathbf{x}|\boldsymbol{\ell} = c) = \frac{1}{\sqrt{(2\pi)^d \det \boldsymbol{\Sigma}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_c)^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu}_c)} \tag{1.5}$$

Dans le cas d'une classification binaire ($\boldsymbol{\ell} = \pm 1$) [Bouchard, 2005] montre que sous certaines conditions (concernant les fonctions objectifs choisies pour l'apprentissage), les deux approches ont des solutions optimales induisant les mêmes probabilités à posteriori $p(\boldsymbol{\ell}|\mathbf{x})$. [Ng et Jordan, 2001] montrent empiriquement que lorsque les données d'apprentissage sont relativement peu nombreuses, la LDA peut donner de meilleures performances que son homologue discriminatif (LLR), même dans les cas où l'hypothèse de "Gaussianité" est loin d'être vérifiée. Ceci en dit long sur la controverse à propos de la préférence génératif / discriminatif.

1.1.2 Interprétation en termes de reconnaissance des formes

Deux démarches fondamentalement différentes

La résolution du problème général d'optimisation (1.3) par les méthodes génératives, sans autre contrainte que la contrainte probabiliste $\sum_c p_c = 1$, conduit à choisir ces probabilités *a priori* en fonction seulement de la répartition des vecteurs d'apprentissage parmi les classes :

$$p_c^* = \frac{N_c}{N}$$

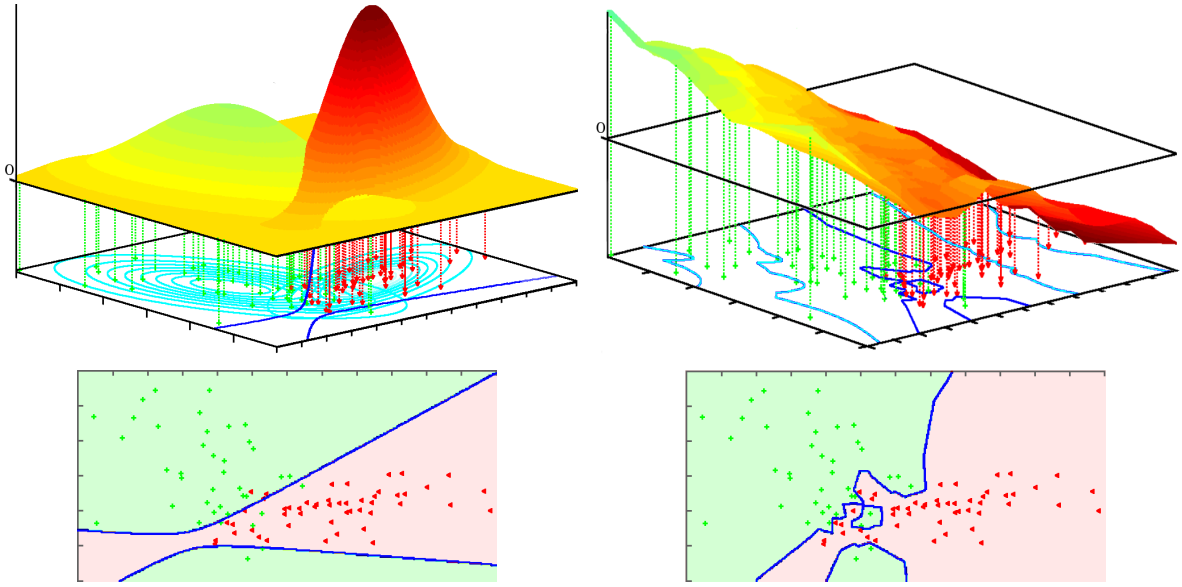
où N_c est le nombre d'entrées d'apprentissage étiquetées par c. Aussi les jeux de paramètres $\boldsymbol{\theta}_c^*$ spécifiques à chaque classe sont trouvés indépendamment les uns des autres, étant donné qu'aucune contrainte ne les relie :

$$\boldsymbol{\theta}_c^* = \arg\max_{\boldsymbol{\theta}_c} \sum_{\left\{\mathrm{a}_i | \boldsymbol{\ell}_i = c\right\}} \log p_{\boldsymbol{\theta}_c}(\mathrm{a}_i)$$

C'est ici qu'apparaît une différence fondamentale entre les méthodes génératives et les méthodes discriminantes. Lors de l'apprentissage, les méthodes génératives optimisent plusieurs fonctions objectifs, chacune faisant intervenir un sous-ensemble de vecteurs d'apprentissage correspondant à une seule et même classe. À l'opposé, les méthodes discriminantes optimisent une ou plusieurs fonction(s) objectif(s) faisant chacune intervenir plusieurs classes. En d'autres mots, les méthodes génératives exploitent l'information intraclasse et les méthodes discriminantes utilisent l'information inter-classes.

Dans le cas particulier d'une classification binaire ($n_c = 2$), un apprentissage discriminatif optimise une seule fonction objectif qui fait intervenir les deux classes, alors qu'un apprentissage génératif optimise séparément deux densités de probabilité conditionnelles. Autrement dit,

- les approches génératives optimisent les densités de probabilité de chacune des classes et induisent une frontière de décision dont l'expression analytique fait intervenir des modèles avec les paramètres choisis.
- Les approches discriminantes optimisent directement une frontière de décision.



(a) Approche générative

La distribution de chaque classe est ici supposée Gaussienne. Dans ce cas, la frontière de décision est une conique (elle est linéaire si les matrices de covariance des deux classes sont identiques)

(b) Approche discriminante

La méthode illustrée ici est celle du plus proche voisin, La fonction discriminante représentée (dont le signe reflète la décision) est la différence entre les distances aux plus proches voisins de chaque classe respective.

Fig. 1.1 - Différences Génératif / Discriminatif : Illustration des stratégies d'apprentissage (classification binaire)

La Figure 1.1 simule de telles démarches dans le cas de données vectorielles bidimensionnelles. Présentées ainsi, les méthodes discriminantes semblent plus adaptées à la tâche de classification que les méthodes génératives. C'est une raison de l'intérêt portée à ces méthodes, invoquée par [Vapnik, 1998] : il paraît en effet plus judicieux de résoudre le problème de classification directement, plutôt que de résoudre en étape intermédiaire un problème plus général (ce que font les méthodes génératives en modélisant les densités de probabilité de chaque classe).

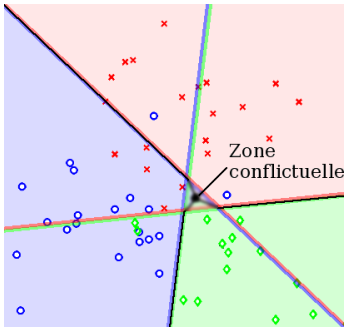
Cas multi-classes

Le cas "multi-classes" ($n_c > 2$) peut être problématique pour les approches discriminantes. Alors que certaines méthodes, comme les k-plus proches voisins, se déclinent naturellement au cas multi-classes, d'autres sont originellement basées sur des fonctions objectifs d'apprentissage biclasse. Plusieurs stratégies sont alors possibles pour généraliser ces approches discriminantes originellement conçues pour la classification binaire. Les deux principales sont les suivantes :

- Construire $n_c(n_c-1)/2$ classifieurs binaires visant à séparer chaque paire de classes, et construire un arbre de décision à partir de ces classifieurs. C'est la stratégie "un contre un" (*one against one*).
- Construire n_c classifieurs binaires visant à séparer chaque classe de toutes les autres, et combiner les sorties des différents classifieurs (typiquement, la classe choisie est celle renvoyant le plus grand score). C'est la stratégie "un contre tous" (*one against all*), la plus

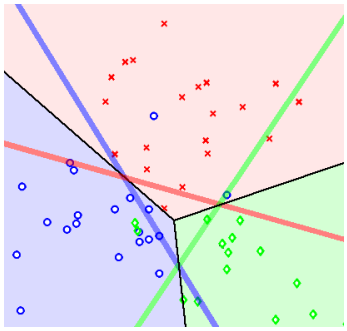
communément utilisée.

Ces deux cas sont illustrés dans Fig. 1.2 pour une approche linéaire. Le lecteur peut se référer à [Hsu et Lin, 2002] pour une étude de l'extension des Machines à Vecteurs de Support au cas multi-classes.



(a) Stratégie "un contre un"

Il existe plusieurs stratégies pour prendre une décision dans la zone conflictuelle (décision arbitraire selon la racine de l'arbre de décision / combinaison des scores)



(b) Stratégie "un contre tous"

Fig. 1.2 - Classification multi-classes : généralisation d'un classifieur binaire

1.1.3 Combinaison des approches

La combinaison d'approches génératives et discriminantes pour la classification a récemment fait l'objet de nombreuses recherches en *machine learning* [Raina et al., 2003]. Des classifieurs hybrides génératifs / discriminatifs ont été appliqués avec succès en bio-informatique [Jaakkola et Haussler, 1998], vision par ordinateur [Vasconcelos et al., 2004, Fritz et al., 2005], et traitement de l'audio [Moreno et Ho, 2003a]. L'idée est de tirer partie des avantages des deux types d'approches, en particulier :

- pour les modèles génératifs, la possibilité de traiter naturellement des séquences de taille variable (avec une robustesse aux observations aberrantes).
- Pour les modèles discriminatifs, l'adéquation du critère d'apprentissage au problème de classification (minimisation d'un terme erreur adéquat, avec garanti de généralisation).

Il y a plusieurs manières de combiner une modélisation générative (G) et une modélisation discriminante (D) :

1. Deux classifieurs G et D agissent en parallèle, et leurs résultats sont combinés.
   [Hou et Wang, 2003, Scheffer et Bonastre, 2006]
2. G est imbriqué dans D.
   [Jaakkola et Haussler, 1998, Fine et al., 2001, Liu et al., 2006]
3. D est imbriqué dans G.
   [Ganapathiraju et Picone, 2000, Campbell, 2003, Schafföner et al., 2006]

Il est difficile de prévoir quelle stratégie est *a priori* la meilleure. Généralement, combiner une démarche générative avec une démarche discriminante permet d'améliorer la robustesse, par rapport à un classifieur n'utilisant qu'un des deux paradigmes. Mais la complexité calculatoire

impliquée par le cumul des deux approches peut être fortement accrue. Elle dépend bien sûr de la formulation choisie pour combiner les deux approches.

Nous verrons dans le prochain chapitre quelques approches hybrides utilisées en reconnaissance du locuteur (§2.3). Aussi nous présenterons des noyaux construit à partir de modélisations génératives (§3.3 et §3.4), ces noyaux étant utilisés en pratique dans des approches discriminantes comme les SVMs (§1.3). Quelques-uns de ces noyaux qui présentent une complexité calculatoire raisonnable avec la modélisation GMM seront appliqués avec succès à la tâche de vérification du locuteur (§5.4). Ils montrent des niveaux de performance prometteurs.

1.2 Exemple d'approches classiques

Cette section donne quelques exemples d'approches génératives et discriminantes couramment utilisées pour la classification automatique de motifs. Le lecteur peut se référer à [Duda et al., 2000] pour un tour d'horizon plus exhaustif.

1.2.1 Mélanges de Modèles Gaussiens (GMM)

Un Mélange de Modèles Gaussiens (GMM) est une densité de probabilité vectorielle qui peut s'écrire sous la forme d'une combinaison linéaire positive de lois "Gaussiennes" :

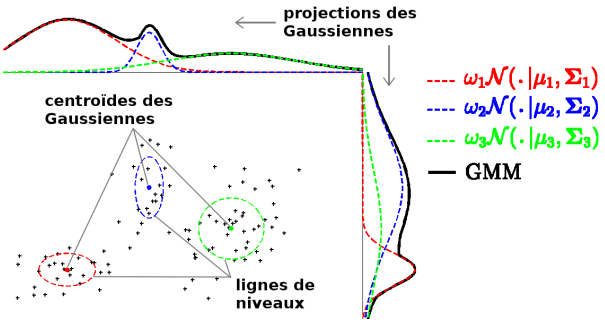
$$\begin{gathered} \forall\ \mathbf{x} \in \mathbb{R}^d\ , \qquad p(\mathbf{x}|\theta) = \sum_{g=1}^{G} \omega_g \mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\right) \\ \text{avec } \theta = \left\{\omega_g, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\right\}_g \in \left(\mathbb{R}^+ \times \mathbb{R}^d \times \mathbb{R}^{d^2}\right)^G \text{ et } \textstyle\sum_g \omega_g = 1 \end{gathered} \tag{1.6}$$

Nous rappelons que la densité de probabilité associée à la loi Gaussienne s'écrit :

$$\mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}\right) = \frac{1}{\sqrt{(2\pi)^d \det \boldsymbol{\Sigma}}} e^{-\frac{1}{2}\left(\mathbf{x}-\boldsymbol{\mu}\right)^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} \left(\mathbf{x}-\boldsymbol{\mu}\right)}$$

Si l'on note d la dimension des vecteurs d'entrée, le nombre de paramètres libres d'un GMM à G Gaussiennes est à l'origine $M = G(d^2 + d + 1)$. Mais en pratique, les matrices de covariances $\boldsymbol{\Sigma}_g$ sont souvent contraintes à être diagonales. Le nombre de paramètres libres est alors réduit à $M = G(2d + 1)$. Cela a l'avantage, en plus d'alléger fortement les calculs, de réduire la complexité de la modélisation pour éviter le sur-apprentissage. Aussi la complexité de la modélisation GMM se règle principalement avec le nombre G de Gaussiennes, par validation (croisée ou simple).

Les Gaussiennes, composantes du modèles en forme de cloches (Fig.1.3), sont souvent assimilées à des modes, qui correspondent à des amas de points d'apprentissage. Ces "*clusters*" qui peuvent être vus comme des (sous-)classes qui représentent des états acoustiques (*cluster hypothesis*). Cette façon de voir les GMMs a inspiré de nombreux travaux en vérification du locuteur (§2.2.2). Concernant l'apprentissage des GMMs, la technique de base consiste à régler les paramètres libres en recherchant à maximiser la vraisemblance des données d'apprentissage au modèle, avec des algorithmes de type EM [Dempster et al., 1977]. Le principe de ces algorithmes est d'estimer (*Expectation*) et d'optimiser (*Maximization*) itérativement la vraisemblance des données d'apprentissage aux modèles, jusqu'à atteindre une pseudo-stationnarité



Les lignes de niveau des Gaussiennes elliptiques correspondent à des covariances diagonales

Fig. 1.3 - Mélanges de Modèles Gaussiens (GMM) : illustration avec des vecteurs 2D.

(approche d'un maximum local). L'algorithme est décrit en détail dans l'annexe A.3. Soulignons que l'apprentissage par procédure EM ne garantit pas la convergence vers un optimum global. Plusieurs stratégies sont envisageables pour la phase d'initialisation. Par exemple, à partir du dictionnaire fourni par une Quantification Vectorielle [Linde et al., 1980], on peut procéder à une classification non supervisée (*clustering*) des données d'apprentissage, et estimer les poids, moyennes et covariances de chacun des *clusters*.

Les GMMs, comme les modèles probabilistes en général, peuvent être facilement étendus pour le traitement de séquences de tailles variables. Ils peuvent par exemple être incorporés dans des modèles dynamiques comme nous allons le voir en §1.2.2. En dehors de cette possibilité, les vecteurs des séquences sont supposés indépendants. Aussi l'échelle logarithmique est utilisée afin d'éviter les problèmes de précision numérique posés par la multiplication de valeurs de densité faibles :

$$\log p(\mathbf{X}|\theta) = \sum_{t=1}^{T_\mathrm{X}} \log p(\mathbf{x}_t|\theta) \tag{1.7}$$

Pour certaines d'approches où la règle de décision n'est pas directement dérivée de la stratégie Bayésienne, un facteur multiplicatif $1/T_\mathrm{X}$ est rajoutée pour normaliser vis-à-vis des longueurs des séquences. Ce sera le cas pour les systèmes UBM-GMM (§2.2.3).

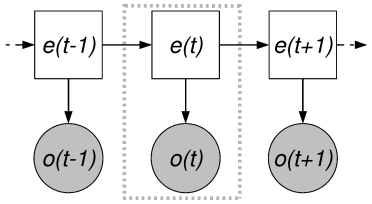
1.2.2 Modèles génératifs dynamiques

Les modèles probabilistes dynamiques sont capables de capturer les dépendances temporelles entre différentes variables aléatoires. Ils permettent donc de manipuler les séquences sans faire d'hypothèse restrictive comme l'indépendance entre les observations qui forment chaque séquence.

Modèles de Markov Cachés (HMM)

Les HMMs ont une structure composée de variables aléatoires continues qui dépendent de variables aléatoires discrètes cachées, appelées communément "états". Il est supposé que chaque état à un instant t+1 ne dépend que de l'état à l'instant t précédent. Fig.1.4 représente la

structure d'un HMM avec les représentations habituelles des Réseaux Bayésiens. Un HMM est paramétré par les probabilités initiales de chaque état, la matrice des probabilités de transition entre les états et enfin les fonctions de densité modélisant les probabilités d'émission de la variable continue selon chaque état.



carré / *cercle* : variable discrète / continue
blanc / *grisé* : variable cachée / observée
$e(t)$: État caché, à l'instant t
$o(t)$: Observation à l'instant t. Lois d'émissions $p(o(t)|e(t))$
: *slice* (structure élémentaire)

Fig. 1.4 - Modèles de Markov Cachés (HMM) : Structure.

L'algorithme de Baum-Welch [Baum et al., 1970] permet de régler les paramètres d'un HMM sur un ensemble de séquences d'apprentissage où les états cachés sont étiquetés. La probabilité d'émission d'une séquence peut être calculée efficacement avec l'algorithme *Forward*, et la séquence d'états cachés la plus probable par l'algorithme de [Viterbi, 1967].

Les HMMs ont connu un grand succès en reconnaissance de la parole (dès les années 70) et en bio-informatique (à partir des années 80). Pour la reconnaissance de la parole [Rabiner, 1989], les variables observées sont les vecteurs acoustiques et les états cachés représentent les phonèmes à reconnaître. Leur probabilité d'émission selon chaque état est couramment modélisée par un GMM.

Réseaux Bayésiens Dynamiques (DBN)

Les Réseaux Bayésiens sont des graphes orientés acycliques mettent en jeu plusieurs variables aléatoires discrètes et/ou continues qui peuvent être en pratique cachées et/ou observée. Dans les représentations courantes, les noeuds correspondent à des variables et les arcs aux dépendances probabilistes $p(X|\text{parents}(X))$. Fig.1.5 montre une telle représentation. Les GMMs y sont représentés comme des Réseaux Bayésiens où l'index des Gaussiennes sont les états cachés, et où les lois d'émission des vecteurs observés sont modélisées par des Gaussiennes. Les Réseaux Bayésiens Dynamiques (DBNs) sont des extensions de Réseaux Bayésiens pour modéliser les dépendances temporelles entre variables. Les HMMs sont un cas particulier des DBNs (Fig.1.4).

La structure d'un DBN peut être fixée arbitrairement en tenant compte de dépendances intuitées *a priori*, ou appris par divers algorithmes [Heckerman, 1995] comme l'algorithme du K2 [Cooper et Herskovits, 1992]. Une fois la structure connue, la méthode d'estimation des paramètres dépend de la présence ou non de variables cachées [Lauritzen, 1995, Naïm et al., 2004].

[Sanchez-Soto, 2005] applique les Réseaux Bayésiens pour la vérification du locuteur en mode

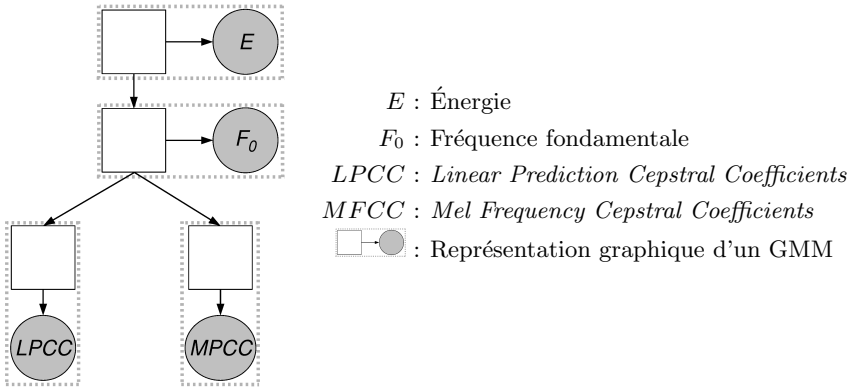


Fig. 1.5 - Réseaux Bayésiens (BN) : Exemple d'une structure utilisée en vérification du locuteur.

indépendant du texte. À cause des contraintes sur les complexités calculatoires, l'auteur se limite à modéliser des lois d'émission des vecteurs acoustiques avec des GMMs à peu de composantes Gaussiennes, comparé aux tailles des GMMs atteintes dans les systèmes UBM-GMM classiques (§2.2). La structure non dynamique qui donne les meilleures performances est celle représentée dans Fig.1.5. Elle est plus robuste que son homologue reliant les variables observées continues. Par contre, les résultats sont améliorés lorsque le réseau de la figure 1.5 est utilisé comme structure élémentaire ("*slice*") d'un Réseau Bayésien Dynamique, c'est-à-dire lorsque la dépendance entre les états discrets à l'instant t+1 et leurs homologues à l'instant t sont modélisés. D'autres structures plus simples ont été utilisés avec succès dans la vérification du locuteur par [Ezzaidi et al., 2001, Arcienega et Drygajlo, 2002].

1.2.3 Adaptation des modèles

L'adaptation désigne un ensemble de techniques qui visent à transformer un modèle de référence pour tenir compte de nouvelles données n'ayant pas servi à l'apprentissage de ce modèle. Les techniques d'adaptation sont utilisées en classification à diverses fins :

- Permettre d'apprendre un modèle complexe sur un petit ensemble de données tout en évitant le sur-apprentissage. L'idée est d'utiliser un modèle de référence qui permette d'extrapoler au mieux l'information manquante (sur les régions peu représentées par l'ensemble de données).
- Compenser l'écart (*mismatch*) entre les conditions d'apprentissage d'un modèle et les conditions d'observation de données de test.
- Tenir compte de nouvelles solutions du problème de classification pour améliorer l'adéquation des modèles.

L'adaptation est très utilisée en traitement de la parole parce que le signal présente une grande variabilité extrinsèque aux informations de haut-niveau recherchées (phonèmes, locuteur, etc.). En particulier, cette variabilité est due à la diversité des conditions d'enregistrement. Typiquement, l'adaptation s'appuie sur un modèle de référence qui a été entraîné avec une collection exhaustive de données, qui réunissent un maximum de conditions d'observation pour tenir compte des diverses sources de variabilité. Il s'agit par exemple pour la reconnaissance de parole d'un modèle "indépendant du locuteur".

De nombreuses études ont abouti sur des techniques d'adaptation robuste pour les modèles probabilistes, dont bénéficient les approches génératives. Le principe général de l'adaptation des modèles de distribution est de de contrôler l'écart entre le modèle appris sur les nouvelles données et le modèle de référence. L'adaptation peut se faire :

- en initialisant l'apprentissage avec le modèle de référence et en augmentant la vraisemblance des nouvelles données de manière itérative tout en limitant l'écart au modèle de référence [Gauvain et Lee, 1994].
- en apprenant un modèle de façon classique sur les nouvelles données et en le normalisant de manière à ce qu'il soit à une distance (§3.3.2) donnée du modèle de référence [Ben et Bimbot, 2003].

Dans de nombreuses techniques d'adaptation pour les GMMs, les vecteurs moyennes des composantes Gaussiennes du modèle de référence sont adaptés à partir de transformations de la forme :

$$\tilde{\boldsymbol{\mu}}_g = \alpha \boldsymbol{\mu}_g^* + (1-\alpha)\boldsymbol{\mu}_g^{\mathrm{o}} \tag{1.8}$$

où $\boldsymbol{\mu}_g^{\mathrm{o}}$ est la moyenne de la $g^{\text{ième}}$ composante du modèle de référence, $\boldsymbol{\mu}_g^*$ est la moyenne de la $g^{\text{ième}}$ composante déduite d'une maximisation de la probabilité *a posteriori* (MAP), et $\tilde{\boldsymbol{\mu}}_g$ est la moyenne régularisée finalement affectée. Le paramètre de régularisation $\alpha \in [0,1]$ peut être fonction de la répartition des données [Gauvain et Lee, 1994], ou alors d'une distance entre le GMM de moyennes $\{\boldsymbol{\mu}_g^*\}$ et le GMM de référence de moyennes $\{\boldsymbol{\mu}_g^{\mathrm{o}}\}$ [Bredin et al., 2006]. Il existe des formes analogues pour l'adaptation des poids et des covariances des composantes Gaussiennes, mais elles sont moins utilisées en pratique pour des questions de robustesse. Nous reviendrons sur l'adaptation pour la vérification du locuteur dans (§2.2).

1.2.4 Exemples d'approches discriminantes

Dans cette partie, nous présentons deux exemples classiques de modélisation discriminante qui ont des points communs avec les Machines à Vecteurs de Support (§1.3). Ces méthodes d'optimisation n'exploitent pas la théorie des probabilité contrairement aux modèles génératifs. Ces techniques sont bien maîtrisées pour la classification des données vectorielles de taille fixe (données statiques). En revanche, leur extension aux séquences de tailles variables n'est pas triviale.

k-*plus proches voisins*

Cette méthode discriminante base appartient à la catégorie des algorithmes graphiques et ne comporte pas d'étape d'apprentissage à proprement parler. Elle consiste à garder en mémoire tous les vecteurs d'apprentissage étiquetés et elle repose sur une mesure de distance arbitraire entre les vecteurs. En phase de test, les distances entre le vecteur à classer et tous les vecteurs d'apprentissage sont estimées et rangées en ordre décroissant. Pour la décision, on procède par vote majoritaire parmi les k vecteurs d'apprentissage les plus proches. Il peut arriver que deux classes majoritaires aient le même nombre de *plus proches voisins*. Pour résoudre ce conflit, plusieurs stratégies sont envisageables, comme par exemple choisir la classe ayant la distance moyenne la plus faible. Notons enfin que la capacité de généralisation de la modélisation est réglé via le paramètre k. Prendre un k élevé permet de lisser la modélisation et d'éviter le

sur-apprentissage. Le réglage de k, pour un bon compromis biais-variance, peut se faire par validation (croisée ou simple).

Outre sa simplicité, l'avantage de cette méthode est qu'elle peut naturellement s'appliquer au cas multi-classes même avec un nombre élevé de classes [Shakhnarovish et al., 2005]. Mais les inconvénients sont de taille :

1. Un volume important de données d'apprentissage implique une capacité des ressources mémoire nécessaires d'autant plus élevée, ainsi qu'une forte complexité calculatoire en phase de test.
2. Le renvoi d'une mesure de confiance de la décision (score) ne peut se faire que de manière arbitraire, par exemple en calculant une moyenne des distances au *k-plus proches voisins.* à partir des distances calculées. À la base, la méthode est conçue pour renvoyer une décision binaire.

Ces inconvénients sont rédhibitoires pour le traitement de la parole, où :

1. Les volumes de données nécessaires pour une bonne modélisation sont importants ;
2. Les objets à traiter sont des séquences de tailles variables.

Les *k-plus proches voisins* ont toutefois été appliqués avec succès à la vérification du locuteur, dans des protocoles faisant intervenir de petits corpus. [Bahler et al., 1994].

Réseaux de Neurones

Les Réseaux de Neurones (ANNs) conçus par [McCulloch et Pitts, 1943] sont une des premières méthodes discriminantes pour la classification et la régression à avoir connu un réel succès dans de nombreuses applications, même si beaucoup de chercheurs y voient encore une "boîte noire" qui frustre leur créativité[2]. Ils peuvent servir à la classification comme à la régression non linéaire de fonctions. Construit sur un paradigme biologique et inspiré du système nerveux humain, l'ANN est un système de "neurones formels" inter-connectés dans un réseau, et qui coopèrent pour fournir un score comme représenté dans l'exemple de Fig.1.6. Les neurones sont des

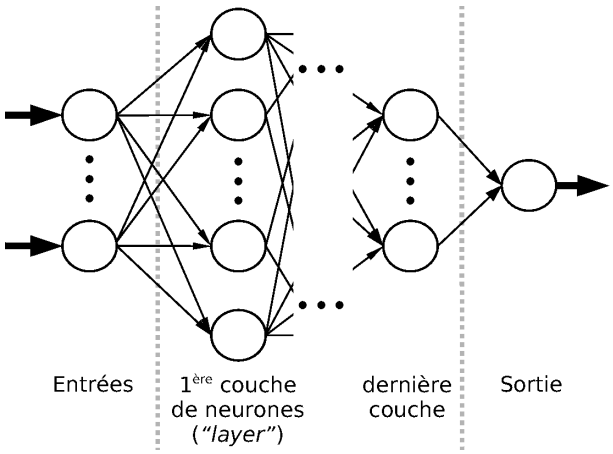


Fig. 1.6 - Réseaux de Neurones : vue simplifiée d'un réseau en couches.

[2] A.K.Dewdney a par exemple traité les réseaux de neurone de "mauvaise science" (1997).

opérateurs qui appliquent une fonction aux valeurs d'entrée qu'il reçoit d'un autre neurone (ou de l'entrée du réseau) pour renvoyer une valeur vers un autre neurone ou une sortie du réseau. Tout comme pour les Réseaux Bayésiens, la topologie d'un ANN peut être choisie de manière arbitraire ou apprise avec des algorithmes variés, comme par exemple l'apprentissage en cascade [Fahlman et Lebiere, 1990] ou l'élagage des connexions (*pruning*) [LeCun et al., 1990]. Les paramètres libres de chaque fonction de neurone sont appelés "coefficients synaptiques". Selon une méthode d'apprentissage fondée par [Hebb, 1949], ils sont réglés par itérations successives de propagations de l'information dans les neurones mettant en jeu les couples entrées-sorties d'apprentissage. On distingue deux types de fonctions synaptiques :

- les *fonctions de combinaison*, qui sont des combinaisons linéaires des entrées pour les réseaux de type MLP, où des fonctions monotones de la distance entre le vecteur d'entrée et un vecteur de référence pour les réseaux de type RBF.
- les *fonctions d'activation*, qui sont typiquement des fonctions sigmoïdes (seuillage régularisé).

Pour faire le parallèle avec les SVMs (§1.3.2), notons que les fonctions de combinaison sont analogues à des fonctions noyaux entre le vecteur formé par les paramètres d'entrée et le vecteur formé par les poids synaptiques. Aussi, les ANNs ne sont pas invariants à toutes les transformations linéaires, et leur application nécessite de normaliser les données d'entrée de façon adéquate [Sarle, 1997], comme c'est le cas avec les SVMs. Les ANNs donnent de bonnes performances pour de nombreux problèmes statiques. Ils ont été appliqués avec succès à la vérification du locuteur par [Ganchev et al., 2003] mais ne sont pas devenus populaires dans le domaine.

1.3 Machines à Vecteurs de Support (SVM)

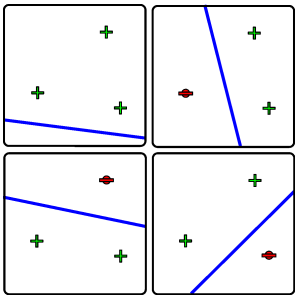
Cette section présente les Machines à Vecteurs de Support (SVMs), qui regroupent une catégorie de méthodes qui ont montré de bonnes performances empiriques pour de nombreux problèmes de classification et de régression. Le fondement théorique des SVMs vient de la théorie de Vapnik (§1.3.1) [Vapnik, 1998]. Les SVMs ont à la base été formulés pour des problèmes de classification binaire (§1.3.2) [Cortes et Vapnik, 1995, Burges, 1998] mais ont aussi été étendus pour le problème général de la régression. Leur puissance vient du critère de "marge" et de "l'astuce du noyau" (§1.3.3). À la fin de cette section, nous présenterons les précautions nécessaires à la mise en pratique des SVMs, ainsi que les limitations du fondement théorique de base qui ont été à l'origine de nombreuses recherches (§1.3.4).

1.3.1 Théorie d'apprentissage de Vapnik

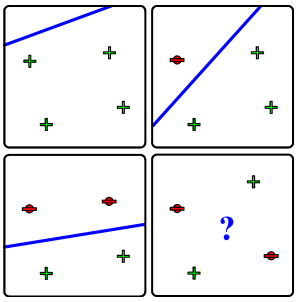
La théorie fondatrice des SVMs [Vapnik, 1998] repose sur deux critères qui permettent de juger de l'adéquation d'une méthode de classification pour un problème donné. Il s'agit de la complexité et la capacité de généralisation d'un ensemble de classifieurs, qui sont le résultat d'une modélisation et d'une technique d'apprentissage. Pour simplifier la lecture, nous parlerons par abus de langage de "classifieur" pour désigner un ensemble de classifieurs.

Complexité et Capacité d'un classifieur

La complexité d'un classifieur désigne son aptitude à pouvoir séparer deux classes selon la répartition des données dans l'espace de représentation. Une mesure de complexité classique est la dimension de Vapnik-Cervonenkis (*VC-dimension*). Elle correspond au nombre maximal N d'entités que le classifieur peut séparer quelle que soit la partition choisie pour affecter les classes 0/1 aux entités (parmi les 2^N partitions possibles). Dans le cas d'une classification de vecteurs, elle dépend de la dimension des données et de la flexibilité de la frontière de séparation. Par exemple, dans $\mathbb{R}^d$, l'ensemble des classifieurs linéaires (hyperplans séparateurs qui peuvent être caractérisés par d paramètres) a pour VC-dimension $(d+1)$. Fig.1.7 illustre le cas de $\mathbb{R}^2$ (VC-dimension égale à 3).



Si 3 vecteurs 2D sont non alignés, pour chacune des 2^3 partitions possibles, on trouve une droite qui réalise cette partition.



Pour quelque quadruplet de vecteurs 2D que ce soit, il y aura toujours une partition (comme celle illustrée en bas à gauche) qu'aucune droite ne pourra réaliser.

Fig. 1.7 - VC-dimension des classifieurs linéaires : illustration dans $\mathbb{R}^2$.

La capacité de généralisation d'un classifieur est liée à l'écart entre son taux d'erreur sur les données d'apprentissage et son taux d'erreur sur des données de test (c'est-à-dire non "vues" lors de l'apprentissage). Plus cet écart ϵ est petit, et plus la capacité du classifieur est grande. Selon Vapnik, la capacité de généralisation est fonction croissante du rapport entre le nombre de données d'apprentissage et la *VC-dimension*. Ce phénomène est simulé via l'écart ϵ dans Fig.1.8.

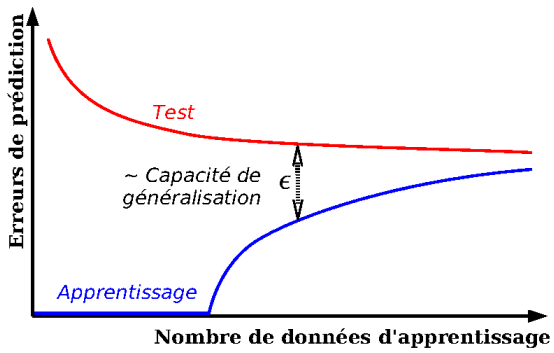


Fig. 1.8 - Capacité de généralisation d'un classifieur : illustration.

À partir d'un corpus de données étiquetées, on peut mesurer la capacité de généralisation en procédant par validation croisée (*cross-validation*). Cette procédure consiste à partitionner les données disponibles $\mathcal{A}$ en K sous-ensembles distincts $\mathcal{A}_i$. Un des sous-ensemble $\mathcal{A}_i \subset \mathcal{X}$ est

utilisé pour tester le modèle alors que les sous-ensembles complémentaires $\mathcal{A} \cap \mathcal{A}_i{}^c$ sont été utilisés pour entraîner le modèle. Cette procédure est répétée K fois et la moyenne des performances (ou une autre fonction de combinaison) est utilisée comme critère d'évaluation.

Compromis biais-variance en apprentissage

[Duda et al., 2000] ont montré que si le nombre de vecteurs d'apprentissage est inférieur au double de leur dimension d, alors il y a plus de chance que ces points soient séparables linéairement plutôt qu'ils ne le soient pas. Dans ce cas, un classifieur linéaire construit en optimisant sans contrainte les d paramètres d'un hyperplan séparateur souffre de ce que l'on appelle le sur-apprentissage (*overfitting*) : même si le taux d'erreurs sur les données d'apprentissage est quasiment nul, le taux d'erreur sur un corpus de test est généralement élevé (Fig.1.9). Cette faible capacité de généralisation est due à ce qui a été baptisé *the curse of dimensionality* par [Bellman, 1961]. Le "fléau de la dimensionalité" se réfère à la croissance exponentielle des volumes (et des distances) lorsque l'on rajoute des dimensions à un problème. Par exemple, estimer un histogramme dans un carré de coté unitaire de $\mathbb{R}^2$ à partir de 100 points est fiable, alors que l'estimation d'un histogramme dans un hyper-cube de coté unitaire dans $\mathbb{R}^{10}$ requiert un nombre minimal de points de l'ordre de 10^{10}.

Si aucune précaution n'est prise lors de l'apprentissage, les performances fournies par une modélisation ne sont pas nécessairement améliorées lorsque l'on augmente le nombre de paramètres d'entrée et/ou le nombre de paramètres libres[3] des modèles. Une manière simple d'assurer une bonne capacité de généralisation est de limiter empiriquement le nombre de paramètres libres des modèles vis-à-vis du volume des données d'apprentissage. Mais d'un autre coté, beaucoup de problèmes de classification font intervenir des données dont la répartition par classe dans l'espace de représentation a une structure complexe, qui ne peut pas être capturée par des modèles trop rigides. Seuls des classifieurs induisant des frontières de décision suffisamment souples pourront fournir des taux d'erreurs acceptables, ne serait-ce que lors de la phase d'apprentissage. En définitive, le classifieur idéal pour un problème donné doit être suffisamment complexe pour avoir un taux d'erreur d'apprentissage faible ("biais"), et avoir une bonne capacité de généralisation pour minimiser l'écart entre ce taux et le taux d'erreur sur des données de test ("variance"). L'existence d'un tel optimum est illustrée dans Fig.1.9.

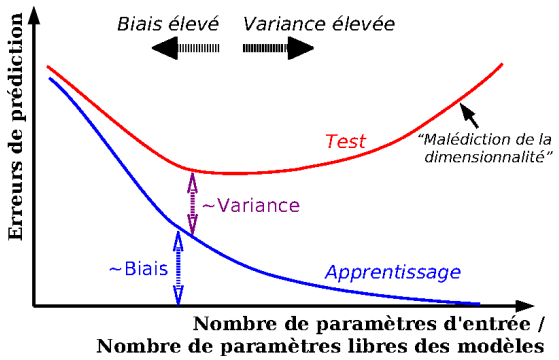


Fig. 1.9 - Biais et Variance d'un classifieur.

[3] *Paramètres libres* :Paramètres que l'on peut régler à partir les données d'apprentissage

D'après Vapnik, on peut envisager une modélisation complexe à condition de choisir comme critère d'apprentissage un terme de "risque régularisé" qui combine

1. un "risque empirique", mesure des erreurs commises par le modèle en apprentissage,
2. un terme de complexité, qui fait intervenir uniquement les paramètres du modèle.

C'est ce que font les Machines à Vecteurs Support, où la fonction objectif à minimiser est la somme pondérée du coût global des erreurs d'apprentissage et d'un critère de "marge", comme nous le décrivons dans la suite.

1.3.2 Formulation des SVMs pour la classification

Nous présentons ici les grandes lignes des SVMs tels qu'ils ont été formulés par [Cortes et Vapnik, 1995] pour un problème de classification binaire (étiquettes ± 1).

Cas linéaire

Pour permettre de mieux saisir le principe des classifieurs binaires SVMs dans le cas général, nous commençons par décrire les SVMs linéaires. Les fonctions discriminantes linéaires pour la classification de vecteurs $\mathbf{x}$ de dimension d sont des fonctions scalaires de la forme :

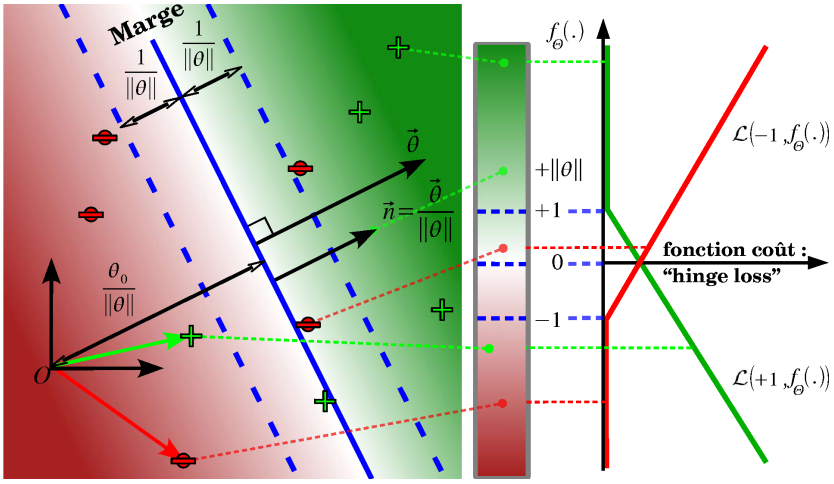
$$f_\theta(\mathbf{x}) = \boldsymbol{\theta}^{\mathrm{T}}\mathbf{x} + \beta_0 \ , \qquad (\mathbf{x}, \boldsymbol{\theta} \in \mathbb{R}^d \quad \beta_0 \in \mathbb{R}) \tag{1.9}$$

Par convention, une telle fonction induit un hyperplan séparateur d'équation $f_\theta(\mathbf{x}) = 0$. La décision du classifieur linéaire pour un vecteur $\mathbf{x}$ est prise selon la position de ce vecteur par rapport à cet hyperplan, c'est-à-dire selon le signe de la fonction discriminante. Par convention,

$$\begin{cases} f_\theta(\mathbf{x}) \geq 0 \ \rightarrow \ \ell^*(\mathbf{x}) = +1 \\ f_\theta(\mathbf{x}) < 0 \ \rightarrow \ \ell^*(\mathbf{x}) = -1 \end{cases} \tag{1.10}$$

Le gradient constant $\boldsymbol{\theta}$ de la fonction discriminante est un vecteur normal à l'hyperplan séparateur. Sa norme représente l'amplitude de la fonction discriminante à une distance unitaire de cet hyperplan. Sa direction indique le demi-espace correspondant aux valeurs positives de la fonction discriminante. Le seuil β_0 indique la position de la frontière de décision par rapport à l'origine. La "marge" est définie par la distance entre l'hyperplan d'équation $f_\theta(\mathbf{x}) = -1$ et l'hyperplan d'équation $f_\theta(\mathbf{x}) = +1$, tous deux à égale distance de l'hyperplan séparateur. Elle vaut $\frac{2}{\|\boldsymbol{\theta}\|}$ (Fig.1.10).

Supposons que les données étiquetées d'apprentissage $\{\mathbf{a}_i, \boldsymbol{\ell}_i\}$ sont linéairement séparables. Alors on peut trouver une infinité d'hyperlans séparateurs qui ne commettent aucune erreur de classification sur ces données. Nous voyons maintenant comme un critère formulé à partir de la marge permet de choisir le meilleur candidat parmi ces hyperplans. Avec la convention $\boldsymbol{\ell}_i = \pm 1$, les fonctions discriminantes qui correspondent à de tels hyperplans peuvent s'écrire telles que $\boldsymbol{\ell}_i f_\theta(\mathbf{a}_i) \geq 1$ pour tous les vecteurs d'apprentissage et $\boldsymbol{\ell}_i f_\theta(\mathbf{a}_i) = 1$ pour les vecteurs les plus près de l'hyperplan séparateur. Ces vecteurs sont ceux qui apportent le plus d'information sur la localisation de la séparation des classes : ils sont appelés "vecteurs de support". La marge est alors la distance qui sépare les vecteurs de support après une projection sur la direction perpendiculaire à l'hyperplan séparateur (Fig.1.11(a)). Le meilleur hyperplan séparateur que l'on puisse choisir



Fonction Discriminante $f_\theta(\mathbf{x}) = \boldsymbol{\theta}^{\mathrm{T}}\mathbf{x} + \beta_0$

Fig. 1.10 - Apprentissage SVM linéaire : Représentation géométrique des paramètres

est alors celui qui maximise cette marge entre les deux classes. C'est le principe de base des SVMs. Notons que la cohérence entre la définition de la marge et la notion géométrique dans le cas séparable repose sur la contrainte $\ell_i f_\theta(\mathbf{a}_i) \geq 1$. Dans le cas non séparable, nous allons voir comment cette contrainte est relâchée ("*soft margin*").

Pour un corpus d'apprentissage étiqueté $\{\mathbf{a}_i, \ell_i\}$, la fonction objectif τ à minimiser lors de l'apprentissage SVM est la somme pondérée des erreurs d'apprentissage et d'un critère de régularisation inversement proportionnel au carré de la marge :

$$\tau(\theta) = \tfrac{1}{2}\|\boldsymbol{\theta}\|^2 + C\sum_{i=1}^{N}\mathcal{L}\big(\ell_i, f_\theta(\mathbf{a}_i)\big) \tag{1.11}$$

où C permet d'ajuster le compromis biais-variance et où $\mathcal{L}$ est une fonction de coût particulière. Il existe plusieurs variantes pour cette fonction de coût [Steinwart, 2005]. La plus utilisées est connue sous le nom de "*hinge loss*" (Fig.1.10) qui conduit à la formulation des *L1*-SVM et qui s'écrit :

$$\mathcal{L}\left(\ell, f(\mathbf{x})\right) \;=\; \big(1 - \ell f\left(\mathbf{x}\right)\big)_+ \;=\; \max\left\{0\;,\; 1 - \ell f(\mathbf{x})\right\} \tag{1.12}$$

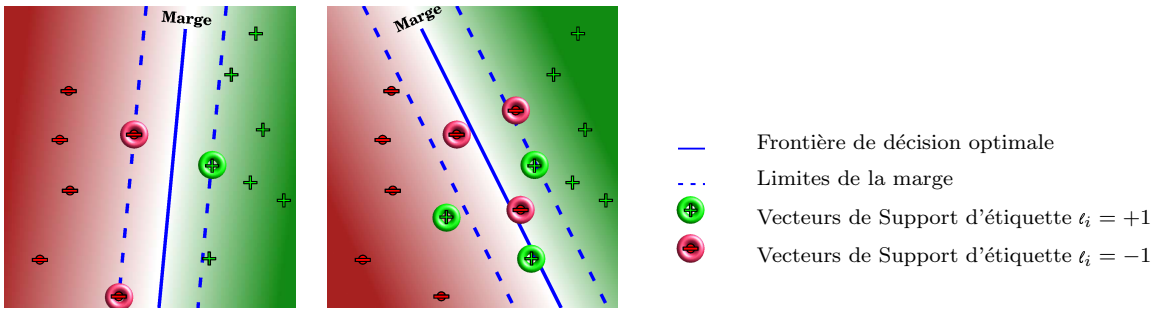
On peut montrer que la minimisation du critère (1.11) avec la fonction de coût (1.12) revient à minimiser

$$\tau'(\theta, \xi) = \tfrac{1}{2}\|\boldsymbol{\theta}\|^2 + C\sum_{i=1}^{N}\xi_i \tag{1.13}$$

sous les contraintes

$$\forall\; i \in \{1 \cdots N\} \qquad \begin{cases} \xi_i \geq 0 \\ \ell_i f_\theta\left(\mathbf{a}_i\right) \geq 1 - \xi_i \end{cases} \tag{1.14}$$

Si les données d'apprentissage sont linéairement séparables, on peut trouver une solution pour laquelle tous les ξ_i sont nuls. Les vecteurs de support sont alors ceux qui vérifient $\ell_i f_\theta\left(\mathbf{a}_i\right) = 1$ et l'hyperplan séparateur optimal est celui qui maximise la marge (Fig.1.11(a)). Dans le cas non séparable, les vecteurs de support sont les vecteurs d'apprentissage qui vérifient $\ell_i f_\theta\left(\mathbf{a}_i\right) \leq 1$. Ils correspondent aux vecteurs qui sont mal classés ou qui se situent sur l'hyperplan d'équation $f_\theta\left(\mathbf{x}\right) = \pm 1$ (par abus de langage, "sur la marge") comme illustré dans Fig.1.11(b).



(a) Cas séparable (b) Cas non séparable

Fig. 1.11 - SVM linéaire : Marge et Vecteurs de Support

Généralisation du cas linéaire

Dans le cas général, les données d'entrée appartiennent à un espace d'entrée $\mathbb{X}$ quelconque (*input space*). Pour généraliser les SVMs dans un tel contexte, il suffit d'appliquer l'algorithme linéaire dans un espace vectoriel de dimension D où seraient projetées les données. Cet espace est appelé "*Feature Space*" (espace des caractéristiques). On suppose donc une expansion non linaire sous-jacente ("*map*" ou "*embedding*" dans la littérature anglophone) :

$$\begin{cases} \mathbb{X} & \rightarrow & \mathbb{R}^D \\ \mathrm{x} & \mapsto & \boldsymbol{\Phi}(\mathrm{x}) \end{cases} \tag{1.15}$$

Si l'on considère le cas d'entrées vectorielles de dimension d ($\mathbb{X} = \mathbb{R}^d$) avec $D > d$, construire une fonction discriminante linéaire à partir d'une telle expansion revient à chercher une frontière non linéaire dans l'espace d'entrée $\mathbb{X}$, comme illustré pour un classifieur polynomial de degré 2 dans Fig.1.12. Les allures des frontières résultant d'un apprentissage SVM pour d'autres types d'expansions seront montrées en §3.2.3. L'*expansion* $\boldsymbol{\Phi}$ dans un espace de plus haute dimension $\mathbb{R}^D$ sert alors à augmenter la séparabilité des données. Accroître les caractéristiques via l'expansion $\boldsymbol{\Phi}$ revient généralement à augmenter la complexité de la modélisation, la VC-dimension des classifieurs linéaires dans le *Feature Space* étant $(D+1)$. Le critère de régularisation des SVMs (marge) permet alors d'éviter le fléau de la dimensionalité (dans le *Feature Space*), au même titre qu'il permet d'éviter le sur-apprentissage des SVMs linéaires lorsque le nombre de paramètres d'entrée est élevé.

De manière analogue au cas linéaire, la phase d'apprentissage consiste à rechercher les solutions f_θ qui minimisent une fonction objectif τ, avec les formes :

$$f_\theta(x) = \sum_{u=1}^{D} \theta_u \phi_u(x) + \beta_0 \tag{1.16}$$

$$\tau(\theta) = \tfrac{1}{2} \sum_u {\theta_u}^2 + C \sum_{i=1}^{N} \left(1 - \boldsymbol{\ell}_i f_\theta(\mathbf{a}_i)\right)_+ \tag{1.17}$$

Intéressons-nous maintenant à la fonction k de deux variables définie par :

$$k(\mathrm{x}, \mathrm{y}) = \boldsymbol{\Phi}(\mathrm{x})^{\mathrm{T}} \boldsymbol{\Phi}(\mathrm{y}) = \sum_{u=1}^{D} \phi_u(\mathrm{x}) \phi_u(\mathrm{y}) \tag{1.18}$$

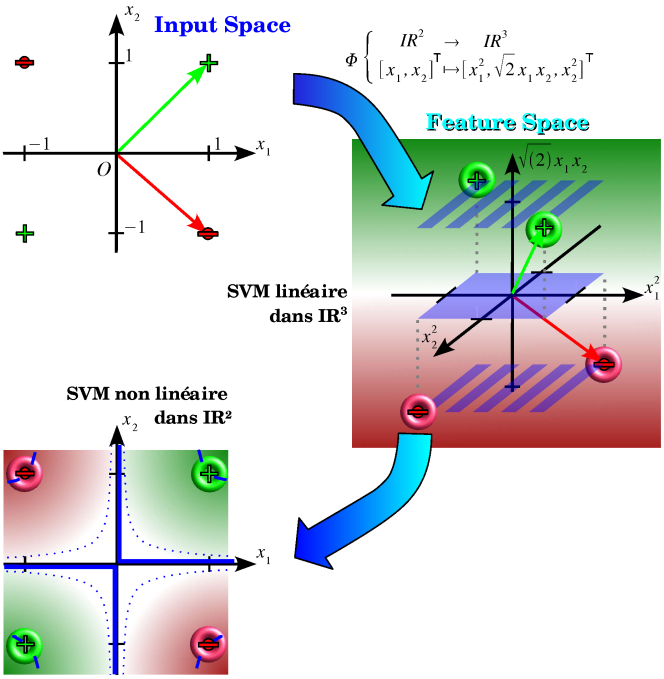


Fig. 1.12 - SVM non linéaire : Illustration du principe (cas d'école, polynomial)

Cette fonction symétrique et définie positive (§3.1 définition 3) est appelée "noyau". Elle joue un rôle fondamental dans la généralisation des SVMs. Comme nous allons le voir, la connaissance des valeurs $k(\cdot,\cdot)$ permet de faire abstraction des *expansions* $\boldsymbol{\phi}(\cdot)$ à la fois pour résoudre le problème d'optimisation des SVM et pour appliquer un classifieur SVM sur de nouvelles données. Autrement dit, la fonction noyau permet d'éviter le calcul de l'expansion dans le *Feature Space*. Avant d'en venir aux intérêt d'une telle astuce, nous présentons les notions fondamentales qui permettent de comprendre comment l'astuce du noyau permet de rendre le calcul des expansions implicite.

Définition 1 (Espace de Hilbert à Noyaux Reproduisants).

Soit $k : \mathbb{X}\times\mathbb{X} \to \mathbb{R}$ une fonction noyau symétrique et définie positive. On appelle Espace de Hilbert à Noyaux Reproduisants (RKHS) associé au noyau k, l'espace $\mathcal{H}$ des fonctions de $\mathbb{X}$ à valeurs dans $\mathbb{R}$ engendré par les fonctions du type $k(\mathrm{x}_i,\cdot)$:

$$f \in \mathcal{H} \Leftrightarrow \left[\exists\ N \leq +\infty\ ,\ \exists\ \{\mathrm{x}_i\}_{i=1\cdots N} \in \mathbb{X}^N \text{ tq } \quad \forall\ \mathrm{x} \in \mathbb{X}\ ,\ f(\mathrm{x}) = \sum_i \beta_i k(\mathrm{x}_i,\mathrm{x})\right] \tag{1.19}$$

Le produit scalaire dans cet espace $\mathcal{H}$, défini par $\left\langle f, k(\mathrm{x}_i,\cdot)\right\rangle_{\mathcal{H}} = f(\mathrm{x}_i)$, est aussi symétrique et défini positif. Le caractère "reproduisant" fait référence à la propriété $\langle k(\mathrm{x}_i,\cdot), k(\mathrm{x}_j,\cdot)\rangle_{\mathcal{H}} = k(\mathrm{x}_i,\mathrm{x}_j)$

Cette définition permet d'introduire le "théorème des représentants" qui jouent un rôle central dans les méthodes à noyaux. Il a été énoncé pour des fonctions de coût quadratique par [Kimeldorf et Wahba, 1971] et, plus tard, généralisé à des fonctions de coût quelconque par [Cox et O'Sullivan, 1990]. Ce théorème appliqué à un RKHS $\mathcal{H}$ peut se formuler comme suit.

Théorème 1 (Théorème des représentants).

Soient
- une fonction strictement monotone $\Omega : \mathbb{R}^+ \to \mathbb{R}$,
- un ensemble $\mathcal{A} = \{\mathrm{a}_i, \boldsymbol{\ell}_i\}_{i=1\cdots N}$ d'éléments étiquetés $(\mathbb{X}\times\mathbb{R})^N$,
- une fonction de coût $\mathcal{L} : \mathcal{A}\times\mathbb{R}^2 \to \mathbb{R}^+$,
- et $\mathcal{H}$ le RKHS généré par le noyau reproduisant k.

La fonction f^* dans $\mathcal{H}$ qui minimise le risque régularisé

$$f^* = \arg\min_{f\in\mathcal{H}} \left\{ \Omega\left(\|f\|_{\mathcal{H}}\right) + \sum_{i=1}^{N} \mathcal{L}\left(\mathrm{a}_i, \boldsymbol{\ell}_i, f(\mathrm{a}_i)\right) \right\} \tag{1.20}$$

admet une représentation de la forme :

$$f^*(\mathrm{x}) = \sum_{i=1}^{N} \beta_i k(\mathrm{a}_i, \mathrm{x}) + \beta_0 \ , \qquad \{\beta_i\}_{i=0\cdots N} \in \mathbb{R}^{N+1} \tag{1.21}$$

Si de plus $|\boldsymbol{\ell}_i - y_i| \mapsto \mathcal{L}\left(\mathrm{a}_i, \boldsymbol{\ell}_i, y_i\right)$ est croissante, alors elle peut s'écrire en fonction des données d'apprentissage et de coefficients positifs ("coefficients de Lagrange") :

$$f^*(\mathrm{x}) = \sum_{i=1}^{N} \alpha_i \boldsymbol{\ell}_i k(\mathrm{a}_i, \mathrm{x}) + \beta_0 \ , \qquad \{\alpha_i\}_{i=1\cdots N} \in \left(\mathbb{R}^+\right)^N \tag{1.22}$$

Le théorème des représentants garantit que les fonctions discriminantes $f(\mathbf{x})$ qui minimisent le risque régularisé peuvent se mettre sous la forme (1.22). Cela permet de reformuler le critère d'apprentissage "primal" τ (1.17) en un critère "dual" :

$$\tau_D(\alpha) = \sum_{i=1}^{N} \alpha_i - \tfrac{1}{2} \sum_{i,j} \alpha_i \alpha_j \boldsymbol{\ell}_i \boldsymbol{\ell}_j k\left(\mathrm{a}_i, \mathrm{a}_j\right) \tag{1.23}$$

Ce critère est à maximiser, sous les contraintes

$$\begin{cases} 0 \leq \alpha_1, \cdots, \alpha_N \leq C \\ \sum_{i=1}^{N} \alpha_i \boldsymbol{\ell}_i = 0 \end{cases} \tag{1.24}$$

En pratique, le problème dual est résolu par des méthodes de programmation quadratique.

La nouvelle forme (1.22) ne fait pas intervenir explicitement le calcul de l'*expansion* $\boldsymbol{\phi}(\mathbf{x})$. L'astuce du noyau présente en fait deux intérêts fondamentaux :

1. (*Entrées vectorielles*) Son expression peut souvent être simplifiée[4], afin de ne pas à calculer explicitement l'*expansion* dans le *Feature Space*, dont la dimension peut être très grande, voire infinie. Dans le premier cas l'intérêt du noyau est simplement de diminuer

[4] Dans Fig.1.12, le noyau utilisé a une forme polynomiale compacte $k(\mathbf{x},\mathbf{y}) = \left[x_1, \sqrt{2}x_1x_2, y_2\right]\left[y_1, \sqrt{2}y_1y_2, y_2\right]^{\mathrm{T}} = (x_1y_1)^2 + 2(x_1y_1)(x_2y_2) + (x_2y_2)^2 = (\mathbf{x}^{\mathrm{T}}\mathbf{y})^2$

la complexité du problème. Dans le second cas, son intérêt est plus fondamental : certaines fonctions noyau symétriques et définis positifs correspondent à des Feature Space de dimension infinie. Ils génèrent donc des classifieurs SVMs à *VC-dimension* infinie, c'est-à-dire pouvant séparer un quelconque jeu de données selon toutes les partitions binaires possibles.

2. Le concept de noyau peut être élargi à d'autres types d'objets que les entrées vectorielles. Il permet ainsi de généraliser l'algorithme SVM à tout type d'objets (symboles, séquences,...). La fonction noyau $k(\mathrm{x}, \mathrm{y})$ manipule ces objets au même titre que le produit scalaire manipule des vecteurs.

En revanche, la complexité calculatoire de (1.22) dépend du nombre de données d'apprentissage. C'est là qu'intervient la notion de vecteurs de support. Le fait que la fonction de coût (1.12) présente un palier et une discontinuité au niveau de la dérivée première rend parcimonieuse (*sparse*) la solution du problème d'optimisation. Autrement dit $\alpha_i = 0$ pour une certaine proportion des données d'apprentissage et $\alpha_i > 0$ pour les vecteurs de support. Finalement, la complexité calculatoire de (1.22) dans le cas non linéaire est liée au nombre de vecteurs de support. Le lecteur peut se référer à [Steinwart, 2003] pour une étude sur la proportion des vecteurs de support.

Extensions des SVMs

Au départ, l'algorithme SVM a été conçu pour la classification binaire ; Sa formulation connaît des variantes, selon la fonction coût $\mathcal{L}$ choisie. Les SVMs pour la classification binaire tels que nous venons de les présenter sont désignés par "C-SVM". Ils ont été étendus pour la classification multi-classes [Hsu et Lin, 2002] et pour la régression de fonctions scalaires [Gunn, 1998]. Parmi ces extensions, une reformulation des SVMs adaptée à l'estimation de densités a émergé : les SVMs ν-monoclasse (*ν-1class SVM*) [Schölkopf et al., 2000, Schölkopf et al., 2001]. Pour ces derniers, le critère à optimiser en apprentissage fait intervenir un terme supplémentaire à (1.17), incluant un autre paramètre de régularisation ν en plus du paramètre de coût habituel C. Notons que les SVMs ν-1classe peuvent être utilisés en combinaison avec les SVMs bi-classes pour réaliser une tâche de classification tout comme, de manière générale, les modèles génératifs peuvent être combinés aux modèles discriminatifs au sein de systèmes "hybrides" [Desobry et al., 2005a].

Le critère de marge des SVMs a aussi inspiré un travail dérivant le critère de marge (discriminatif) dans un cadre probabiliste. Il s'agit des *Relevance Vector Machines* [Tipping, 2000], qui s'appliquent aussi bien en classification qu'en régression et qui sont aussi utilisés pour l'estimation de densités [Dasgupta et al., 2002].

1.3.3 Puissance des SVMs

La modélisation sous-jacente aux SVMs n'est pas paramétrique, dans le sens où la fonction discriminante f recherchée s'exprime directement à partir des exemples d'apprentissage et non avec un nombre prédéfini de paramètres libres[5]. Par rapport aux modélisations génératives et aux modélisations discriminantes paramétriques (Régression Logistique), cela permet d'adapter

[5] À la retenue près que dans le cas linéaire, le nombre de paramètres libre du modèle (hyperplan séparateur) peut être réduit à la dimension du problème.

la complexité de la frontière de décision aux données d'apprentissage (à leur abondance et leur répartition).

De manière générale, l'astuce du noyau permet d'adopter des modélisations complexes, capables dans le cas vectoriel de capturer une infinité de corrélations non linéaires entre les paramètres d'entrée. D'un autre coté, le critère de marge permet de garantir une certaine capacité de généralisation. Ceci explique pourquoi il a été observé dans plusieurs cas pratiques où le corpus d'apprentissage est limité en nombre, que les SVMs donnent de meilleures performances que les approches génératives dans le traitement de la parole [Zhou et Hansen, 2003, Arias et al., 2005] comme dans d'autres applications de reconnaissance des formes [Justino et al., 2004, Mashao, 2005]. En effet, les méthodes génératives donnent de bonnes performances à condition

1. que la distribution réelle de chaque classe de vecteurs puisse être correctement capturée par la famille de fonctions choisie ;
2. que le corpus d'apprentissage soit suffisamment important et représentatif pour estimer correctement les paramètres de cette modélisation. Le nombre d'exemples nécessaires à l'apprentissage robuste d'un modèle croît de façon polynomiale avec le nombre de paramètres libres du modèle.

Le premier critère se réfère à la complexité de la modélisation (biais) et le second à la capacité de généralisation (variance). Les critères d'apprentissage des modèles génératifs ne formulant pas explicitement le compromis biais-variance, ce dernier est fait en pratique de manière empirique avec toutes les limitations que cela suppose. C'est pourquoi les approches génératives sont limitées par rapport aux SVMs dans les cas où les données d'apprentissage sont peu nombreuses, c'est-à-dire dans les cas où elles sont

– difficiles à collecter (suivi d'un locuteur qui intervient peu [Kartik et al., 2005], reconnaissance de langue rare [Schafföner et al., 2006]) ;

ou

– coûteuse à étiqueter (classification d'unités phonétiques [Juneja et Espy-Wilson, 2002, Saenko et al., 2004]).

Un point important des SVMs autre que la gestion du compromis biais-variance est la parcimonie de la solution. La plupart des méthodes non paramétriques comme les *k-plus proches voisins* retiennent tous les vecteurs d'apprentissage pour définir la frontière de décision. Si la taille du corpus d'apprentissage est trop importante, ces méthodes demandent trop de ressources mémoire et entraînent une complexité calculatoire trop élevée pour la prise de décision. Même si dans les cas extrêmes les SVMs ne font pas exception à ce constat, ils ont l'avantage de ne retenir qu'une partie des données d'apprentissage : les Vecteurs de Support, sélectionnés grâce au critère de marge et à la forme de la fonction de coût (*hinge loss*). Intuitivement, ces Vecteurs sont les données d'apprentissage qui apportent le plus d'information sur les régions conflictuelles entre les classes, où se situe la frontière de décision idéale pour la classification (en supposant que ces Vecteurs de Support ne soient pas des données aberrantes). En les choisissant automatiquement par optimisation, les SVMs font de la "sélection de données d'apprentissage" pour se concentrer sur les régions pertinentes pour le problème de classification. Par opposition, les données situées dans les régions non pertinentes induisent du bruit dans les modèles génératifs et dans les modèles discriminatifs construits à partir de la totalité du corpus d'apprentissage, ce qui nuit à la robustesse de ces classifieurs.

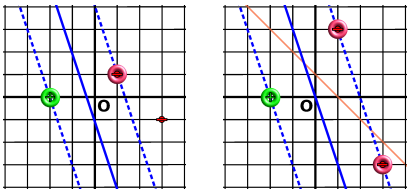
Plusieurs travaux ont montré empiriquement que les SVMs produisaient des taux de fausses alarmes plus faibles que les classifieurs génératifs dans le traitement de la pa-

role [Kartik et al., 2005, Rouas et al., 2006] comme dans d'autres domaines d'application [Fritz et al., 2005, Kanokphara et al., 2006]. Récemment, [Davenport et al., 2006] a étudié comment contrôler le taux de fausses alarmes avec les SVMs en adaptant le terme de régularisation du critère d'apprentissage.

1.3.4 Précautions et limitations

Normalisation des données

Dans le cas vectoriel, les SVMs (même linéaires) ne sont en général pas invariants aux transformations linéaires, comme le montre Fig.1.13. De façon qualitative, multiplier une caractéristique d'entrée (par une constante supérieure à 1) a pour conséquence d'augmenter l'importance de cette caractéristique dans l'apprentissage SVM et la fonction de décision.



Solutions d'un apprentissage SVM
(cas d'école, séparable)

Après avoir tripler la seconde coordonnée (axe vertical), la frontière de décision optimale (critère de marge) n'a pas subi la transformation linéaire correspondante. Le troisième vecteur est maintenant Vecteur de Support.

Les SVMs ne sont pas invariants à toutes les transformations linéaires !

Fig. 1.13 - SVM et transformations linéaires

Le tableau 1.3 récapitule dans quelles mesures les SVMs sont invariants aux transformations linéaires classiques. [Abe, 2003] fait les démonstrations de ces résultats. Les noyaux pris comme exemple et leurs paramètres (ν, p et ρ) sont définis dans la section 3.2. Le tableau montre aussi l'effet du passage d'une normalisation des caractéristiques d'entrée dans l'intervalle cible $[-1, 1]$ à une normalisation similaire dans l'intervalle $[0, 1]$. Ces deux normalisations se font par des transformations linéaires spécifiques à chaque caractéristique d'entrée et sont paramétrées en fonction de l'étendue des caractéristiques sur le corpus d'apprentissage. Elles sont classiques pour l'application des SVMs et des Réseaux de Neurones [Sarle, 1997, Imbiriba et al., 2004] et permettent d'introduire une invariance aux transformations linéaires.

La robustesse des SVMs est donc sensible à la façon de normaliser les données d'entrée, étape cruciale pour la stabilité de la méthode. A contrario, les classifieurs probabilistes (dont ceux basés sur le paradigme génératif) n'utilisent l'étape de normalisation que pour augmenter la robustesse aux changements de conditions d'observation des données [Pelecanos et Sridharan, 2001, Xiang et al., 2002]. Dans le cas où les conditions d'apprentissage et de test sont similaires (*matched*), ces classifieurs procèdent à ce que l'on peut voir comme une sélection de caractéristiques d'entrée. En effet, une caractéristique non pertinente pour un problème de classification donné n'est pas liée (au sens probabiliste) à la classe : elle a quasiment la même distribution pour chacune des classes. Son influence est alors minime lors du calcul des rapports de vraisemblance pour les classifieurs probabilistes. En cela, ces classifieurs procèdent à une "sélection des caractéristiques d'entrée" et les classifieurs GMMs sont robustes aux variables d'entrée aberrantes, contrairement aux SVMs. En fait, les SVMs linéaires supposent que

les caractéristiques d'entrée ont toutes la même importance *a priori* vis-à-vis du problème de classification, et les SVMs non linéaires font une telle hypothèse dans le *Feature Space*. Quelques techniques permettent de normaliser implicitement les données dans le *Feature Space* en modifiant la fonction noyau. Elles seront présentées en §3.1.4, et nous en proposerons une dans le chapitre 4.

Tab. 1.3 - SVMs vectoriels et transformations linéaires dans l'espace d'entrée

Noyau	**Homothétie** ($\mathbf{x}' = h\mathbf{x}$)	**Rotation**	**Translation** ($\mathbf{x}' = \mathbf{x} + \mathbf{t}$)	**Changement de normalisation** $[-1, 1] \rightarrow [0, 1]$
Produit scalaire	$C = C'/h^2$	X	O	$C = 4C'$
Neural Network	$\nu = \nu'/h^2$	X	_	$\nu \approx 4\nu'$
Polynomial	$C = C'/h^{2p}$	X	_	$C \approx 4^p C'$
RBF Gaussien	$\rho = h\rho'$	X	X	$\rho = \rho'/2$

X : Invariance du noyau
O : Invariance de l'algorithme SVM
$C = f(C')$: Transformation du paramètre de coût C à effectuer pour retomber sur la même solution
$\nu = f(\nu')$, $\rho = f(\rho')$: Transformation du paramètre de noyau pour retomber sur la même solution

Complexité calculatoire et ressources mémoire

Les algorithmes d'apprentissage SVM se déroulent à partir de la seule connaissance des valeurs de noyau $k(\mathrm{a}_i, \mathrm{a}_j)$ entre les données d'apprentissage. Ces valeurs sont habituellement mémorisées dans une matrice carrée que l'on appelle "matrice de Gram" (§3.1.1) et qui est de taille $N \times N$ en notant N le nombre de données d'apprentissage. Si le noyau est symétrique, cette matrice l'est aussi et on peut économiser l'espace mémoire en conséquence. Quoiqu'il en soit, optimiser le critère d'apprentissage dual (1.23) des SVMs par programmation quadratique requiert un espace mémoire $O(N^2)$ et une complexité calculatoire $O(N^3)$ avec les boîtes-à-outils classiques. Heureusement, de nombreux travaux ont permis de réduire cette complexité avec des algorithmes sous-optimaux mais néanmoins adaptés et performants [Collobert et al., 2002, Tsang et al., 2005]. Un des plus connu est l'algorithme SMO [Platt, 1999], dont le principe est de diviser le problème initial en une série de problèmes plus petits, et dont la complexité empirique varie entre $O(N^2)$ et $O(N^3)$. En phase de test, la complexité calculatoire pour appliquer un classifieur SVM à de nouvelles données est linéaire vis-à-vis du nombre de Vecteurs de Support. Ce nombre de Vecteurs de Support est quant à lui au pire en $O(N)$ [Steinwart, 2003].

Finalement, un des désavantages des SVMs par rapport aux méthodes paramétriques est que leur complexité est liée au volume du corpus d'apprentissage. Aussi, en présence de zones de forte densité dans l'ensemble d'apprentissage, la matrice de Gram peut être mal conditionnée (déterminant quasiment nul), ce qui menace la convergence des algorithmes de programmation quadratique et donc l'adéquation des modèles appris. L'utilisation des SVMs pour les problèmes à très grande échelle n'est donc pas triviale. Dans le traitement de la parole, les corpus d'apprentissage mêlent plusieurs heures d'enregistrement ce qui correspond à un nombre très élevé de trames sur lesquelles sont extraites les caractéristiques. Nous verrons que l'application des SVMs

sur ces caractéristiques est délicate à mettre en œuvre, et qu'il est plus judicieux de manipuler en entrée des SVMs des caractéristiques relatives à des séquences entières plutôt qu'à des trames isolées.

Adaptation des modèles

L'adaptation de modèles peut être cruciale pour améliorer la robustesse des systèmes de traitement de la parole étant donnée la variabilité des conditions d'enregistrement (§1.2.1). Le cadre probabiliste offre aux modèles génératifs la possibilité d'être naturellement adaptés sur un ensemble de données nouvelles. Adapter un modèle SVM (sans reconsidérer toutes les données ayant servi à apprendre ce modèle) est plus délicat du fait que les modèles SVMs soient non paramétriques. Cela reste un problème ouvert, considéré dans des études très récentes [Li et Bilmes, 2006].

Chapitre 2

Vérification du Locuteur & Approches classiques

Sommaire

Ce chapitre passe en revue les méthodes classiques pour la vérification du locuteur en mode "indépendant du texte". Nous rappelons que cette tâche consiste à déterminer si un extrait de parole a été prononcé ou non par un locuteur donné, sans connaissance *a priori* sur le contenu phonétique (contrairement à un scénario avec mot de passe vocal).

Nous commençons ici par présenter des généralités concernant les systèmes classiques de vérification du locuteur (§2.1) : les modules qui composent ces systèmes, la manière d'utiliser des données de développement pour régler leurs paramètres, et la mesure de leur performance. Nous décrirons ensuite la méthode de modélisation la plus communément utilisée en vérification du locuteur (§2.2). Il s'agit d'une approche générative, connue sous le sigle UBM-GMM, qui prend en entrée des vecteurs acoustiques dérivés d'une analyse spectrale du signal. Enfin, nous donnerons quelques exemples de systèmes SVM qui ont été conçus pour la vérification du locuteur (§2.3), et qui s'inspirent de près ou de loin de l'approche UBM-GMM. Ces systèmes ne sont toutefois pas reconnus comme des systèmes classiques. Trouver la meilleure façon d'appliquer les SVMs à la vérification du locuteur reste un problème ouvert, notamment à cause de la nature séquentielle de la parole. Cela fera l'objet des chapitres suivants.

2.1 Vérification du locuteur : généralités

2.1.1 Modules d'un système de vérification du locuteur

Comme le représente Fig.2.1, les systèmes de vérification du locuteur sont composés de quatre modules principaux interdépendants :

1. **Le pré-traitement** (§2.1.4). Il s'agit d'extraire du signal de parole "pur" (valeurs de l'amplitude échantillonnées à des fréquences de l'ordre de 8000 Hertz) des caractéristiques (paramètres numériques et/ou symboliques). Un bon pré-traitement fournit des paramètres dépendants des variations inter-locuteurs et peu sensibles aux variations extrinsèques à l'identité du locuteur (conditions d'enregistrement, variabilités intralocuteurs, etc.).
2. **L'apprentissage** (§2.1.5). Il s'agit d'instancier des modèles à partir de paramètres extraits de locuteurs étiquetés ou non. L'apprentissage se fait souvent par des méthodes d'"entraînement" itératives.
3. **L'attribution de scores** (§2.1.6). Ce module est étroitement lié à la façon dont ont été conçus et entraînés les modèles dans le module d'apprentissage. Alors que ce dernier s'applique à des séquences d'entraînement (*train*), l'attribution de scores s'applique à des séquences *test*. Notons que l'apprentissage peut tenir compte de plusieurs séquences *train* pour l'élaboration d'un modèle, alors que le module de *scoring* traite les séquences *test* indépendamment les unes des autres.
4. **La prise de décision** (§2.1.6). Ce petit module vient directement après l'attribution de scores. Typiquement, il s'agit de comparer les scores à un seuil (fixé lors de la phase de développement) pour renvoyer une décision binaire. Notons que l'utilité de ce module est controversée pour les applications judiciaires. En effet, dans des conditions réalistes d'utilisation (conversations téléphoniques), on ne peut pas espérer 100% de réussite. De ce fait, avoir un chiffre probabiliste en sortie peut être préférable dans le judiciaire. Le lecteur peut se référer à [Boë et al., 1999, Boë et al., 2001, Bonastre et al., 2003] pour des réflections sur la limitation des systèmes biométriques vocaux.

2.1.2 Corpus

Pour régler les paramètres d'un système (*tuning*) et évaluer ses performances, il faut plusieurs corpus de données :

(1.A) **Le corpus du Monde (*background*)** qui désigne une collecte de données de parole provenant d'une grande population de locuteurs. Il est utilisé pour régler les paramètres du système qui sont indépendants des locuteurs cibles. Par exemple le Monde sert à apprendre un modèle générique (*Universal Background Model*), ou encore à calculer des statistiques indépendantes du locuteur pour normaliser certains paramètres.

(1.B) **Le corpus d'imposteurs** qui fait intervenir des locuteurs étrangers au corpus des locuteurs cibles. Il permet de régler des paramètres spécifiques aux locuteurs cibles en apportant de l'information sur la proximité entre le locuteur cible et les autres locuteurs. Les séquences imposteurs servent à estimer les statistiques pour normaliser les scores, ou encore à apprendre des modèles discriminatifs (entrées négatives d'un SVM...). Notons que les concepts de Monde et d'imposteurs sont toutefois très proches : leurs rôles respectifs peuvent être ambigus selon l'architecture envisagée (générative / discriminante).

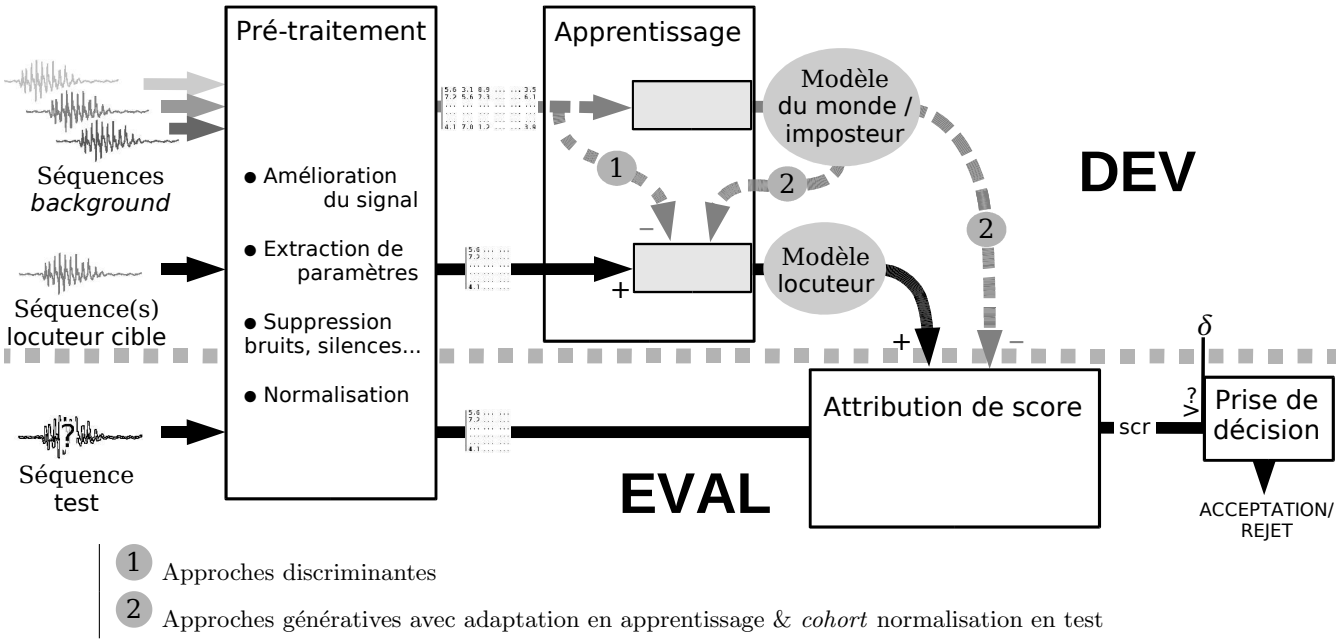


Fig. 2.1 - Diagramme général d'un système de Vérification du Locuteur

Nous discuterons de cela lorsque nous choisirons un protocole commun pour comparer équitablement les performances des différentes approches (§5.1.1).

(1.C) **Le corpus de validation** avec des séquences étiquetées par locuteur, et plusieurs séquences pour certains locuteurs. Il sert à simuler une évaluation du système pour régler empiriquement tous les paramètres que les corpus (1.A,B) ne permettent pas de régler. Ces réglages se font en optimisant la mesure de performance choisie (§2.1.3). On compte parmi les paramètres que permet de régler le corpus de validation :
- les paramètres qui déterminent le compromis biais-variance pour l'apprentissage des modèles,
- le seuil de décision,
- les paramètres d'une éventuelle fusion entre plusieurs systèmes (§2.1.7).

(2.A) **Le corpus d'apprentissage (*train*)** qui sert à apprendre les Modèles des locuteurs cibles. Il est constitué de séquences prononcées par chaque locuteur à vérifier.

(2.B) **Le corpus de test** qui comprend entre autres des séquences prononcées par des locuteurs du corpus d'apprentissage (sans quoi on ne peux pas mesurer de taux de faux rejets). Le système n'a pas accès aux étiquettes des séquences de test, qui sont simplement utilisées pour estimer les performances.

Les trois premiers corpus (1.A,B,C) constituent le **corpus de développement**. Idéalement, ils doivent correspondre à trois populations de locuteurs distinctes, pour ne pas biaiser les réglages des paramètres. Les corpus (2.A,B) constituent le **corpus de d'évaluation**, qui doit faire intervenir des locuteurs étrangers au corpus de développement pour ne pas biaiser la mesure de performance.

Notons que les protocoles [NIST SRE, 1997], ..., [NIST SRE, 2006] ne fournissent que le corpus d'évaluation (séquences *train* et *test*), ce qui est critiquable pour la comparaison de systèmes issus de laboratoires qui n'ont pas forcément tous accès aux mêmes bases de données de dévelop-

pement. Car il ne faut pas négliger que les performances sont meilleures dans les cas où il y a une certaine similarité[6] entre les bases de données utilisées pour régler le système et les séquences servant à évaluer le système. Cette adéquation des données de développement et d'évaluation est plus importante que le volume des corpus utilisés (qui doivent tout de même être suffisamment gros). Ceci explique les dégradations de performances mesurées lors des évaluations NIST où le protocole d'enregistrement des données a été significativement modifié par rapport aux années antérieures (*e.g.* [NIST SRE, 2003]→[NIST SRE, 2004]), et les améliorations notables dans les cas où les protocoles sont restés identiques (*e.g.* [NIST SRE, 2004]→[NIST SRE, 2005]).

2.1.3 Mesure des performances

Sans prendre en compte la phase finale de prise de décision binaire, on peut estimer les performances d'un système de vérification du locuteur à partir des scores de sortie, pour évaluer le prétraitement et la modélisation utilisés. La représentation la plus communément utilisée de la pertinence des scores fournis est la courbe de performance DET introduite par [Martin et al., 1997]. Les courbes DET représentent le taux de faux rejets (FR%) en fonction du taux de fausses alarmes (FA%) en échelle log-normale, comme représenté dans Fig.2.2. L'échelle logarithmique est choisie de manière à ce que deux distributions Gaussiennes $\mathcal{N}(\cdot|\boldsymbol{\mu}_{\text{loc}}, \sigma^2_{\text{loc}})$ et $\mathcal{N}(\cdot|\boldsymbol{\mu}_{\text{imp}}, \sigma^2_{\text{imp}})$ pour les scores clients/imposteurs donnent une droite, dont la pente est le rapport des variances $(-\sigma^2_{\text{imp}}/\sigma^2_{\text{loc}})$, et dont la distance à l'origine dépend de la différence des moyennes $(\boldsymbol{\mu}_{\text{loc}} - \boldsymbol{\mu}_{\text{imp}})$ normalisée en variance. Les courbes traçant FR% en fonction de FA% en échelle linéaire s'appellent les courbes ROC [Van Trees, 1968]. (elles sont d'allure parabolique).

Bien que les courbes DET permettent de se faire une idée sur les performances fournies, leur limite est qu'elles ne permettent pas forcément de comparer deux systèmes. En effet, ces courbes peuvent se chevaucher et le choix du meilleur système dépend du compromis désiré entre le taux de fausses alarmes ou le taux de faux rejets. Pour remédier à cette incertitude, il faut se limiter à un point particulier de la courbe, comme :

- Le taux d'erreurs égales (EER pour *Equal Error Rate*) qui correspond au point où les deux taux mesurés sont égaux FA% = FR% (*i.e.* en pratique, le point qui minimise l'écart |FA% − FR%|).
- Le point de fonctionnement optimal, qui correspond au minimum d'une fonction de coût (*Detection Cost Function*) de la forme

$$\text{DCF} = \tau_{\text{FR}} P_{\text{loc}} \text{FR}\% + \tau_{\text{FA}} P_{\text{imp}} \text{FA}\% \tag{2.1}$$

  où τ_{FR} (resp. τ_{FA}) et P_{loc} (resp. P_{imp}) représentent le coût estimé d'un faux rejet (resp. fausse alarme) et la probabilité a priori d'apparition d'un locuteur cible (resp. d'un imposteur). Ces paramètres sont à choisir arbitrairement pour une application et un contexte donnés.

Si le seuil de décision est fixé pour un système, alors on peut considérer le "point de fonctionnement réel" (*operating point*) pour évaluer la DCF (2.1). Typiquement, le seuil pour estimer le point de fonctionnement réel est choisi de manière à minimiser le critère DCF sur un corpus de validation. L'écart entre la valeur de DCF au seuil de décision choisi (point de fonctionnement

[6] similarités dans protocoles d'enregistrement et la population enregistrée.

réel) et le minimum de DCF (idéal) est représentatif de la variabilité des scores en fonction des conditions de test.

D'autres courbes ont été utilisées pour représenter les performances d'un système de vérification d'après les scores fournis, comme l'histogramme du HTER (*Half Total Error Rate*), qui mesure le taux d'erreur moyen $\frac{1}{2}(\mathrm{FA\%} + \mathrm{FR\%})$ en fonction du seuil de décision [Bengio et Mariéthoz, 2004]. Aussi [Brümmer et du Preez, 2005] propose une mesure pour évaluer la pertinence des score de sortie interprétés en terme de probabilités *a posteriori*, indépendamment de l'application.

2.1.4 Pré-traitement

Le module de pré-traitement est identique qu'il soit en amont d'une phase d'apprentissage ou de test. Il doit être robuste aux paramètres extrinsèques au locuteur. Les sous-modules typiques du pré-traitement sont, dans l'ordre habituel d'intervention :

1. **Amélioration du signal**. Il s'agit de modifier le signal d'entrée par l'application d'un filtre de manière à rehausser une partie de l'information pertinente amoindrie par les conditions d'enregistrement (typiquement les fréquences aiguës).
2. **Extraction de paramètres**. La plupart du temps, cette extraction se fait sur des fenêtres d'échantillonnage de taille fixe (trames). Cela permet de prendre en compte l'aspect dynamique (au moins à court terme) inhérent à la parole. Nous reviendrons ensuite sur les divers paramètres que l'on extrait habituellement.
3. **Suppression des silences**. Cette étape est couplée à l'extraction de paramètres, et sert à exclure de la modélisation des phénomènes extérieurs au parlé du locuteur (silences, bruits, ...). La plupart du temps, il s'agit d'exclure du module suivant certaines données correspondant à des temps morts en terme de parole, à partir de mesures d'énergie (amplitude du signal). Toutefois, étant donné la sensibilité de l'énergie aux conditions d'enregistrement, il convient de ne pas fixer de seuil *a priori*, mais de le choisir en tenant compte de toute une séquence (supposée contenir parole et silence). Parmi les techniques robustes, on peut citer l'apprentissage non supervisé de deux Gaussiennes sur l'énergie proposé par [Magrin-Chagnolleau et al., 2001]. Le lecteur peut se référer à [Li et al., 2002, Zilca et al., 2004] pour d'autres approches robustes.
4. **Normalisation des paramètres**. Cette étape intervient sur les données sélectionnés après retrait des silences. Nous y reviendrons à la fin de cette partie.

Paramètres extraits

Les systèmes UBM-GMM couramment utilisés en vérification du locuteur sont basés sur des paramètres acoustiques dits "cepstraux" : les MFCCs [Davis et Mermelstein, 1980], les PLPCs [Hermansky, 1990], les LPCCs [Kim et Lee, 1999]. Bien que ces paramètres aient été à la base conçus pour la reconnaissance de la parole[7] [Rabiner et Juang, 1993], ils ont montré un pouvoir discriminant pour la reconnaissance du locuteur supérieur aux autres représentations du signal de parole [Campbell, 1997, Cohen et Zigel, 2002]. Les paramètres acoustiques sont extraits sur

[7]Les coefficients cepstraux sont censés représenter la forme du conduit vocal, et ainsi présenter une forte variabilité selon le phonème prononcé.

des fenêtres glissantes de courte durée ("trames" de l'ordre de 20 ms). Leur principal défaut réside dans leur faible robustesse au bruit et aux conditions d'enregistrement [Heck et al., 2000]. Plusieurs études ont été menées pour concevoir des paramètres acoustiques robustes au canal téléphonique, comme par exemple les paramètres "RASTA" [Hermansky et Morgan, 1994]. Mais une alternative populaire consiste à normaliser les paramètres comme nous le voyons ci-après.

D'autres caractéristiques de la parole non liées au spectre contiennent de l'information sur l'identité du locuteur et sont par nature plus robustes au bruit et conditions d'enregistrement. Les paramètres utilisés en vérification du locuteur peuvent être classés en cinq niveaux d'information, dont le plus bas est le niveau acoustique. On pourrait en théorie considérer un sixième niveau "sémantique", mais aucune technique ne permet encore d'extraire ce type d'information de manière automatique.

1. **Niveau acoustique**
   Les paramètres acoustiques sont relatifs au contenu spectral du signal de parole et sont liés aux caractéristiques physiques de l'appareil vocal.
2. **Niveau prosodique** [Carey et al., 1996, Adami et al., 2003, Shriberg et al., 2005]
   La prosodie désigne les caractéristiques d'un énoncé de parole relatives à la mélodie (fréquence fondamentale), l'intonation (énergie) et le rythme (durée des unités phonétiques, des pauses).
3. **Niveau phonétique** [Andrews et al., 2002, Campbell et al., 2003b]
   Les caractéristiques phonétiques se rattachent à la façon de prononcer les différents sons identifiables d'une langue. Les paramètres phonétiques sont relatifs à la prononciation des phonèmes, aux phénomènes de coarticulation, etc.
4. **Niveau idiolectique** [Doddington, 2001]
   Les caractéristiques idiolectiques se réfèrent aux particularités langagières des individus, en particulier les mots et expressions qu'ils emploient de manière récurrente.
5. **Niveau dialogal** [Peskin et al., 2003]
   Les caractéristiques dialogales décrivent la façon dont un individu mène ses conversations, comme par exemple la fréquence et la durée moyenne des prises de parole.

Dans l'état actuel des connaissances, ces quatre derniers types de paramètres, traités avec des techniques de modélisation spécifiques, conduisent à des performances individuelles moins bonnes que celles atteintes avec les paramètres acoustiques. Toutefois, combiner les scores obtenus sur les paramètres acoustiques avec ceux obtenus sur les caractéristiques haut-niveaux permet de gagner en robustesse [Campbell et al., 2003a]. Cette amélioration, preuve d'une dé-corrélation entre les erreurs d'un système cepstral et d'un système haut-niveau, vient du fait que l'information haut-niveau est complémentaire à l'information bas-niveau cepstrale. Aussi, on peut envisager de combiner les paramètres haut-niveau avec les paramètres cepstraux dans la modélisation elle-même, comme il a été fait par [Ezzaidi et al., 2001, Arcienega et Drygajlo, 2002, Sturim et al., 2002, Klusácek et al., 2003, Nickel et al., 2004].

Normalisation des paramètres

La normalisation des paramètres acoustiques en aval du pré-traitement peut être utilisée pour augmenter la robustesse du système au canal téléphonique et réduire l'écart (*mismatch*) entre les conditions d'observation en apprentissage et celles en phase de test. Notons que si les données sont enregistrées dans les mêmes conditions alors ne pas normaliser conduit en général

à de meilleures performances, étant donné que la normalisation entraîne typiquement une perte d'information.

Les techniques de normalisation des vecteurs acoustiques couramment utilisées pour la vérification du locuteur sont :

- **La "standardisation"**, normalisation par soustraction de moyenne (CMS : "*Cepstral Mean Substraction*") suivie éventuellement d'une division par l'écart-type. Ces statistiques sont estimées sur l'intégralité de la séquence, ou sur une fenêtre glissante de taille fixe.
- **Le "*feature warping*"** [Pelecanos et Sridharan, 2001], procédure qui consiste à "faire épouser" (*warp*) localement une distribution Normale à chaque paramètre observé. Les valeurs des vecteurs paramètres normalisés sont déterminés selon le rang de chaque caractéristique (coordonnée vectorielle) sur une fenêtre centrée, glissante et de taille fixe. Cette technique sera expliquée plus en détail dans la description de nos expériences (§5.1.3).
- **Le "*short-time Gaussianization*"** [Xiang et al., 2002], utilisant le même principe que le *feature warping*, avec en amont une transformation linéaire.

Il est préférable d'appliquer ces normalisations en aval du pré-traitement, après la suppression des silences. Évaluer la robustesse d'une technique de normalisation est délicat : le choix optimal dépend en effet non seulement du protocole d'enregistrement, mais aussi de la méthode de modélisation.

2.1.5 Apprentissage

L'apprentissage peut faire appel à n'importe quelle technique de modélisation. Les critères de choix portent sur :

- Les performances fournies, qui reflètent la capacité de la modélisation à saisir le caractère discriminant des paramètres extraits ;
- La taille des modèles (capacité de mémoire requise) ;
- La complexité d'apprentissage et de test.

Ces deux derniers critères, d'ordre pratique, dépendent du protocole considéré pour l'application de vérification (longueur des séquences, nombre de séquences d'apprentissage, ...).

Les algorithmes classiques d'apprentissage des modèles génératifs utilisés en traitement de la parole (GMMs et HMMs) sont récapitulés par [Bilmes, 1997, Brugnara et De Mori, 1998, Tomasi, 2005]. Les GMMs sont de loin les modèles génératifs les plus utilisés en vérification du locuteur [Reynolds et al., 2000] (§2.2). D'autres modèles stationnaires ont été appliqués avec succès à cette application, comme les GMMs Hiérarchiques [Liu et al., 2002], les GMMs structurés phonétiquement [Faltlhauser et Ruske, 2001] basés sur une transcription en phonèmes, les *Text-Constrained* GMMs [Sturim et al., 2002] basés sur une transcription en mots, ou encore les réseaux bayésiens liant les coefficients cepstraux à des paramètres prosodiques [Arcienega et Drygajlo, 2002]. Une application à la vérification du locuteur de la technique de *boosting* [Freund et Schapire, 1999] a été proposée par [Li et al., 2003] pour augmenter le pouvoir discriminant d'un classifieur GMM par apprentissage discriminant des modèles.

2.1.6 Attribution de scores & décision

La stratégie d'attribution de scores est un module qui prend en entrée un modèle de locuteur cible et une séquence de données observées à partir d'un extrait à tester. Elle est donc intimement liée à la façon dont ont été construits les modèles : chaque type de modélisation a sa propre procédure d'attribution de scores.

Pour des applications juridiques, où il s'agit de fournir une expertise biométrique à un tribunal, le comportement des scores renvoyés a une importance. Idéalement, le score renvoyé doit correspondre à une probabilité *a posteriori* qu'un locuteur cible ait prononcé un énoncé, étant donné certaines hypothèses sur la population sondée. Insistons sur l'importance de fournir ces hypothèses avec la probabilité pour permettre une bonne interprétation de l'expertise. En effet, à titre d'exemple, alors que la probabilité de bien reconnaître l'ADN d'une personne parmi la population d'un pays approche 1, elle descend à 1/2 lorsque l'on se limite à examiner deux vrais jumeaux. En vérification du locuteur, un moyen simple de fournir une probabilité *a posteriori* est d'apprendre une fonction de régression entre les scores fournis par un système et les probabilités empiriques sur un corpus de validation. La robustesse d'une telle approche est bien entendu intimement liée à la représentativité de ce corpus utilisé pour régler les paramètres de régression.

Amélioration de la robustesse des scores

En mode "indépendant du texte", beaucoup de systèmes de vérification du locuteur attribuent un score à une séquence en faisant une simple moyenne des scores obtenus par les trames de la séquence. [Besacier et Bonastre, 1998, Chen, 2003] ont montrer que le "*score pruning*" pouvait permettre d'améliorer la robustesse du module d'attribution de score à une séquence. Cette méthode consiste à rejeter les trames du signal de parole produisant des scores marginaux, afin d'écarter de la phase de décision les observations aberrantes qui n'auraient pas été retirées lors du pré-traitement (retrait des silences). Plusieurs critères sont envisageables pour cette sélection de l'information basée sur les scores : très faible vraisemblance au modèle génératif "du monde", scores extrêmes, etc.

On peut généraliser cette démarche et utiliser des critères de plus haut-niveau que la distribution empirique des scores. Ainsi, quelques mesures ont été proposés pour estimer la pertinence *a priori* d'une observation pour le problème de vérification du locuteur [Garcia-Romero et al., 2004, Louradour et al., 2004, Louradour et al., 2005]. Ces mesures de "pouvoir discriminant" affectées à chaque trame de la séquence permettent

- soit de restreindre la sélection des trames lors du pré-traitement pour rendre plus efficace l'attribution de score à une séquence,
- soit de pondérer les scores des trames de manière à ce que leur combinaison linéaire améliore empiriquement les performances.

Les améliorations de performance observées par une pondération des scores des trames sont toutefois très limitées.

Normalisation des scores

Les méthodes de normalisation de scores les plus couramment utilisées sont la "Z-Norm" et la "T-Norm" [Auckenthaler et al., 2000]. Ces normalisations supposent que les scores imposteurs suivent une distribution normale, pour un locuteur cible ou une séquence test donnée. Elles transforment les scores selon la loi suivante :

$$\text{score}(\text{tst}|\text{loc}) \mapsto \widetilde{\text{score}}(\text{tst}|\text{loc}) = \frac{\text{score}(\text{tst}|\text{loc}) - \mu_{\text{imp}}}{\sigma_{\text{imp}}} \tag{2.2}$$

Dans le cas de la Z-Norm, μ_{imp} et σ_{imp} sont la moyenne et l'écart-type des scores obtenus par le locuteur cible sur un ensemble de séquences de développement (considérés comme des tests imposteurs). Dans le cas de la T-Norm, ce sont la moyenne et l'écart-type de scores obtenus par la séquence test sur un ensemble de modèles locuteur de développement (supposées correspondre à des modèles d'imposteurs). En pratique, ces normalisations ont pour effet général une rotation sur les courbes de performance DET [Auckenthaler et al., 2000]. Il a été observé [Barras et Gauvain, 2003] que la T-Norm est préférable quand on privilégie un taux de fausses alarmes FA% bas, et que la Z-Norm est préférable quand on préfère un taux de faux rejets FR% bas (Fig.2.2).

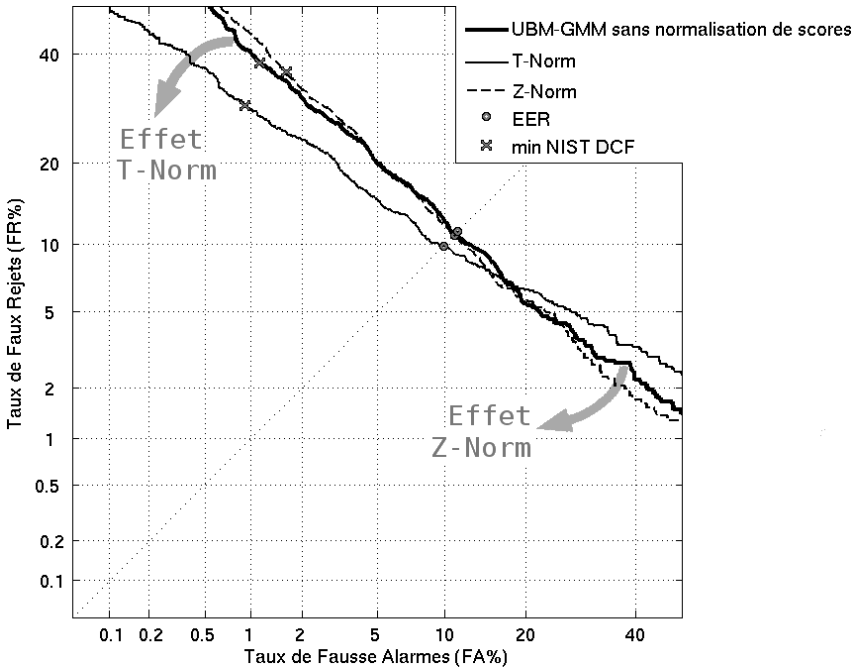


Fig. 2.2 - Courbes DET : Effets des normalisations de score (système UBM-GMM).

Ces normalisations sont d'autant plus performantes que les imposteurs ayant servi aux calcul des statistiques $\{\mu_{\text{imp}}, \sigma_{\text{imp}}\}$ sont proches du locuteur cible. En effet, le pouvoir discriminant d'un système UBM-GMM peut être renforcé par une sélection des séquences imposteurs spécifique à chaque locuteur cible [Sturim et Reynolds, 2005]. D'autres techniques de normalisation ont été conçues, comme la D-Norm [Ben et al., 2002] basée sur la divergence de Kullback-Leibler, ou encore la ZT-Norm, Z-Norm suivie d'une T-Norm (utilisée par le Laboratoire d'Informatique d'Avignon dans les dernières évaluations NIST).

Choix du seuil de décision

La décision d'acceptation ou de rejet d'une séquence test se fait généralement par comparaison du score final à un seuil. Si l'on dispose d'un corpus de développement étiqueté, on peut fixer ce seuil pour optimiser la mesure de performance (2.1) considérée pour l'application visée. Sinon, on peut procéder par validation croisée sur les données d'apprentissage.

Si l'on dispose de beaucoup de données d'apprentissage pour un locuteur, on peut même envisager une approche individuelle (ou lieu d'une approche globale), c'est-à-dire choisir un seuil de décision spécifique au locuteur. Mais les diverses normalisations appliquées aux scores ("*cohort normalisation*" dans le rapport de vraisemblance, et Z- ou T-Norm) rendent légitime une approche globale, qui est non seulement moins lourde à mettre en œuvre, mais qui permet aussi de rendre cohérente la courbe DET pour une visibilité des performances. En effet, dans le cas où le seuil de décision est variable selon les locuteurs, le point de fonctionnement réel situe généralement hors de la courbe DET.

2.1.7 Fusion de plusieurs systèmes

Divers systèmes de vérification du locuteur peuvent être construits à partir de la mesure de diverses caractéristiques du signal de parole, de plus ou moins haut niveaux (§2.1.4). Combiner ces différentes sources d'information peut conduire à des performances supérieures à celles que l'on obtient en les utilisant séparément [Campbell et al., 2003c]. On peut aussi gagner à fusionner des systèmes de vérification du locuteur basés sur les mêmes paramètres, mais adoptant des approches différentes. La fusion permet aussi de concevoir des systèmes biométriques multimodaux, en combinant par exemple la modalité vocale avec la signature en-ligne [Fuentes et al., 2002] ou encore l'image du visage [Chetty et Wagner, 2005].

Comme il peut être laborieux de combiner les différents paramètres à l'intérieur même de la modélisation, une alternative simple est de concevoir plusieurs systèmes et de fusionner les résultats. La fusion de plusieurs systèmes a alors lieu juste avant la prise de décision : il s'agit de combiner les scores. Mais soulignons que dans les cas où les différents paramètres peuvent être combinés dans la modélisation de manière naturelle, cette alternative semble préférable comme il a été rapporté par [Conrad et Paliwal, 2001] qui opèrent simplement une concaténation de paramètres acoustiques. Récemment, [Dean et al., 2006] a aussi conçu un système où les sources d'information audio et vidéo d'un enregistrement audiovisuel sont combinées à l'intérieur d'un même modèle HMM. La combinaison des sources d'information à l'intérieur des modèles reste un problème ouvert.

Même si la fusion par combinaison de scores présente un enjeu important pour la robustesse des systèmes, elle n'a fait l'objet de recherches que depuis récemment. Les techniques les plus classiques conçues consistent à d'abord normaliser les scores (Z/T-Norm) de manière à ce qu'ils soient comparables en ordre de grandeur, et ensuite à combiner les scores normalisés avec :

- une combinaison linéaire dont les poids peuvent être choisis soit de manière empirique par validation croisée [Scheffer et Bonastre, 2006], soit en utilisant une méthode d'optimisation plus sophistiquée comme les SVMs linéaires [Garcia-Romero et al., 2003] ou la régression linéaire logistique [Brümmer, 2005a].

ou

– une combinaison non linéaire réglée par exemple avec un réseau de neurones [Campbell et al., 2004, El Hannani et Petrovska-Delacrétaz, 2005].

Les techniques de fusion de base ont été améliorées pour adapter la fusion au locuteur cible [Fiérrez-Aguilar et al., 2006] et/ou aux données des test [Mak et al., 2003, Cheung et al., 2005] selon des critères variés. Par exemple [Solewicz, 2005] utilise des informations sur les conditions d'enregistrement et le style de conversation pour opérer une sélection parmi plusieurs fusions apprises. Une approche plus flexible a été développée par [Richiardi et al., 2006] qui utilise le cadre probabiliste pour combiner les scores de différents experts, en tenant compte non seulement de la fiabilité des experts (distribution des scores sur un corpus de validation), mais aussi de diverses mesures de qualité du signal. Enfin, les théories de la logique floue, propice à la fusion d'information en prenant compte de notions d'incertitude, ont été appliquées à la fusion pour la vérification du locuteur par [Lau et al., 2004].

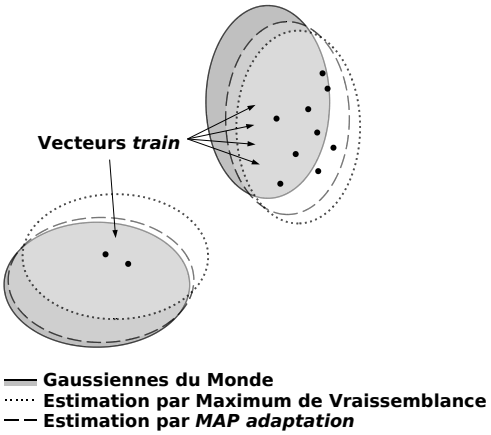
2.2 Système UBM-GMM

2.2.1 Principe général

Au centre de l'approche générative "UBM-GMM", classique pour la vérification du locuteur en mode "indépendant du texte" [Reynolds et al., 2000], une notion fondamentale joue un rôle à la fois lors de l'apprentissage et lors de l'attribution de scores : le "Modèle du Monde" (UBM). Il s'agit d'un modèle de la parole indépendant du locuteur, représentant la répartition *a priori* des données acoustiques d'entrée. Sa forme paramétrique est un Mélange de Modèles Gaussiens (GMM).

La technique d'apprentissage des modèles GMMs pour les locuteurs cibles est l'adaptation MAP de l'UBM, qui a été mentionnée en §1.2.3. Les choix techniques couramment utilisés en vérification du locuteur seront présentés dans la partie suivante (§2.2.2). Fig.2.3 représente le principe général de l'adaptation MAP. Il s'agit d'initialiser l'apprentissage avec l'UBM puis de modifier les moyennes des Gaussiennes, en une fois ou par itérations successives, de manière à les déplacer à mi-chemin entre leur estimation *a priori* (UBM) et leur estimation par maximum de vraisemblance *a posteriori*. Cette technique, contrairement à un apprentissage direct par maximum de vraisemblance, permet d'éviter le sur-apprentissage, ou encore de combler l'information manquante en extrapolant les régions de l'espace d'entrée mal représentés par les paroles disponibles du locuteur. En effet, étant donné la richesse du langage parlé, il peut arriver que certains phonèmes n'apparaissent pas dans le contenu prononcé par un locuteur cible. Faire une adaptation à partir de l'UBM permet d'extrapoler la distribution dans les régions acoustiques correspondantes, afin de ne pas sous-évaluer les scores des séquences de test où seraient prononcés ces phonèmes.

En plus de ce premier intérêt que l'on peut voir comme un "lissage" des modèles locuteurs, le Modèle du Monde sert aussi à représenter les imposteurs en phase de test UBM-GMM. Comme nous le verrons (§2.2.3) la vraisemblance d'une séquence test à l'UBM est vue comme sa probabilité d'émission dans le cas où l'on suppose un enregistrement d'un imposteur.



Cas de Mixtures à 2 Gaussiennes
bidimensionnelles à covariances diagonales
(adaptation des vecteurs moyennes seulement)

Fig. 2.3 - Simulation d'adaptation MAP

2.2.2 L'apprentissage et ses artefacts

Covariance et sur-apprentissage

Les systèmes UBM-GMM estiment des Mélanges de Modèles Gaussiens dont les matrices de covariance sont diagonales. Cela présente d'abord l'intérêt de réduire considérablement les complexités de calcul (estimation d'une Gaussienne en $O(d)$ au lieu de $O(d^2)$). De plus cela permet de réduire la complexité des modèles de manière à éviter le sur-apprentissage. En effet, prendre des matrices de covariances pleines impliquerait un nombre nettement plus élevé de paramètres libres à régler, qui nécessiterait un nombre bien plus important de données d'apprentissage pour la robustesse de l'estimation. De manière générale, la matrice de covariance a une influence importante sur l'estimation des probabilités, ce qui la rend d'autant plus délicate à estimer. Nous verrons ci-après qu'elle est estimée une fois pour toutes sur les données du Monde (riches en information) et laissée fixe pour tous les autres modèles appris, afin de gagner en robustesse.

Une autre technique est appliquée pour adapter la complexité des modèles au traitement de la parole : le "*variance flooring*" [Melin et al., 1998]. Il s'agit d'éviter que les variances atteignent des valeurs trop faibles, en introduisant un seuil minimal dans les itérations de l'algorithme EM. Cela permet d'améliorer les performances avec des modèles comportant un nombre élevé de Gaussiennes (typiquement 2048/4096) [Bonastre et al., 2004]. Sans précaution de ce genre, les performantes décroissent à partir d'un nombre moyen de Gaussienne de l'ordre de 256/512, car au delà, les Gaussiennes apprises dans le Mélange représentent des phénomènes trop singuliers. Le *variance flooring* est donc un moyen empirique de contrôler la capacité de généralisation des GMMs.

Adaptation

Nous rappelons que le but de l'adaptation est de combler l'information manquante dans l'ensemble des observations disponibles pour un locuteur cible (phonèmes peu ou pas présents dans le contenu prononcé), tout en utilisant des modèles suffisamment complexes pour capturer la distribution fortement multimodale des paramètres acoustiques extraits (§1.2.3). Elle consiste en pratique à initialiser l'apprentissage des Modèles Locuteur avec le Modèle du Monde (UBM supposé appris de manière robuste) et à limiter l'écart entre le modèle appris et l'UBM en intervenant dans les itérations d'apprentissage.

La technique la plus répandue en traitement de la parole est l'adaptation MAP, que l'on doit à [Gauvain et Lee, 1994] et qui est devenue la technique de référence dans la vérification du locuteur [Reynolds et al., 2000]. Dans cette approche, seules les moyennes des Gaussiennes de l'UBM sont adaptées en appliquant l'équation (1.8) qui remplace la seconde étape de l'itération EM classique (*Maximization*). Le paramètre d'adaptation α dépend de l'occupation de chaque Gaussienne par les données d'apprentissage. Si l'on note $\mathbf{a}_t$ les vecteurs d'apprentissage et g l'indice des Gaussiennes dans les GMMs, alors ce coefficient est donné par

$$\begin{aligned} \alpha &= \frac{\sum_t p(g|\mathbf{a}_t)}{\sum_t p(g|\mathbf{a}_t) + r} \\ \text{avec } p(g|\mathbf{a}_t) &= \frac{\omega_g p(\mathbf{a}_t|\mathcal{N}_g)}{p(\mathbf{a}_t|\text{GMM})} = \frac{\omega_g \mathcal{N}(\mathbf{a}_t|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)}{\sum_h \omega_h \mathcal{N}(\mathbf{a}_t|\boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h)} \end{aligned} \tag{2.3}$$

où r est un paramètre fixé arbitrairement ("*relevance factor*"). Les poids et la matrice covariance restent quant à eux fixés aux valeurs du Modèle du Monde. En effet, adapter les poids à une séquence prononcé par un locuteur cible, selon le taux d'occupation de chaque Gaussienne, rendrait le modèle trop dépendant du contenu textuel (phonèmes utilisés). Ensuite, adapter les covariances risquerait d'induire un sur-apprentissage (variances trop faibles) dans le cas où le nombre d'observations est limité.

D'autres techniques d'adaptation ont été développées [Mengusoglu, 2003]. Le lecteur peut se référer aux travaux de [Woodland, 1999, Kunn et al., 2000] pour un aperçu de celles utilisées pour le locuteur ainsi que leur étude théorique. [Mariéthoz et Bengio, 2002] compare les performances des diverses techniques d'adaptation pour la tâche de vérification. Enfin, [Siohan et al., 1999] développe une technique d'adaptation analogue à l'adaptation MAP des GMMs pour les Modèles de Markov Cachés.

Cluster hypothesis

La *cluster hypothesis* désigne la vision multi-classes des GMMs : chaque Gaussienne est vue comme une classe acoustique (*cluster*). La *cluster hypothesis* intervient dans les techniques d'apprentissage classiques des GMMs (EM et adaptation MAP). En effet, à chaque itération de ces algorithmes, les probabilités *a posteriori* des composantes Gaussiennes sont manipulées ($p(g|\mathbf{x})$ dans l'équation 2.3).

La *cluster hypothesis* a inspiré les premiers travaux sur l'adaptation de modèles et de paramètres Par exemple, deux stratégies ont été élaborées pour améliorer la robustesse des systèmes UBM-GMM, en considérant les GMMs comme un mélange de *clusters* et en supposant que les

vecteurs de chaque *cluster* soient transformés de manière déterministe lors d'un changement de conditions d'enregistrement :

- La transformation de modèles par translation des Gaussiennes, technique connue sous le nom de *Speaker Model Synthesis* [Heck et Weintraub, 1997] ;
- La transformation de paramètres par translation selon la Gaussienne la plus probable, connue sous le nom de *Feature Mapping* [Reynolds, 2003].

Dans la première approche, chaque type de conditions d'enregistrement conduit à l'estimation d'un UBM et d'un GMM pour le locuteur cible. Pour attribuer un score à une séquence de test, la condition d'enregistrement de cette dernière est automatiquement reconnue pour choisir le couple UBM-GMM adéquat. Dans la seconde approche, un seul GMM est appris par locuteur, adapté d'un UBM indépendant de la condition d'enregistrement. Mais à la différence de l'approche classique, ces modèles prennent en entrée des paramètres normalisés par une astuce relative à la *cluster hypothesis*. Dans les deux cas, l'idée est de simuler un rapprochement entre les conditions d'enregistrement de la(es) séquence(s) ayant servi à l'apprentissage du modèle du locuteur cible et de la séquence test. Les transformations supposées pour les GMMs lors d'un changement de conditions sont typiquement des translations des moyennes des Gaussiennes. Cette hypothèse est analogue à une approximation linéaire par morceaux pour la régression d'une fonction réelle. Elle laisse inchangée les poids et les covariances des Gaussiennes, tout comme l'adaptation MAP dans l'apprentissage des GMMs pour la vérification du locuteur. Précisons qu'il est aussi légitime et envisageable d'adapter en plus les matrices de covariances.

Note sur la prise en compte de la dynamique

Notons que contrairement à la reconnaissance de la parole où les HMMs constituent la modélisation de référence, ces modèles qui prennent en compte des probabilités de transitions entre états sont très peu utilisés en vérification du locuteur ([Qing et Chen, 2006] rapporte des performances moins bonnes), du moins en mode "indépendant du texte". La popularité des GMMs, HMMs à un état qui supposent les observations indépendantes, vient du fait que l'on cherche à concevoir des systèmes peu sensibles au contenu phonétique prononcé, et en particulier à l'ordre des phonèmes. Un locuteur doit être reconnu de la même manière qu'il dise "$A-B$" ou "$B-A$". En mode "dépendant du texte", on préférera des approches vraiment séquentielles, avec alignement dynamique ou modèles dynamiques, comme il a par exemple été fait par [Noda et al., 1998].

On pourrait alors croire que les GMMs ne capturent pas l'aspect dynamique de la parole (évolution des fréquences). Or cet aspect est primordial et ne doit pas être mis à l'écart pour la reconnaissance du locuteur. En fait, la dynamique *à court terme* est habituellement prise en compte dans le pré-traitement, où les dérivées sont concaténées aux vecteurs acoustiques classiques. À notre connaissance, aucune méthode pour prendre en compte la dynamique à long terme des paramètres cepstraux n'a encore été appliquée avec succès à la vérification du locuteur.

2.2.3 Rapport de vraisemblance

Le score habituellement utilisé dans les systèmes UBM-GMM est le "log-rapport de vraisemblance" moyen (*average log-likelihood ratio*) [Reynolds, 1995], donné par la formule :

$$\text{score}_{\text{loc}}(\mathbf{X}) = \tfrac{1}{T_{\mathbf{X}}}\big(\log p(\mathbf{X}|\boldsymbol{\theta}_{\text{loc}}) - \log p(\mathbf{X}|\boldsymbol{\theta}_{\text{UBM}})\big) \tag{2.4}$$

où $p(\cdot|\boldsymbol{\theta}_{\text{UBM}})$ et $p(\cdot|\boldsymbol{\theta}_{\text{loc}})$ représentent les vraisemblances au Modèle du Monde (hypothèse de refus) et au Modèle du Locuteur cible (hypothèse d'acceptation), qui se calculent selon l'équation (1.7) de §1.2.1. L'échelle logarithmique permet d'éviter les problèmes de précision numérique (la densité de probabilité d'une séquence étant une multiplication de densités de probabilités très faibles). La normalisation par la longueur T_{X} de la séquence sert à assurer la cohérence du seuil de décision. Le terme impliquant $\boldsymbol{\theta}_{\text{UBM}}$ peut être vu comme une normalisation de score (vraisemblance) par rapport à un corpus d'imposteurs ("*cohort*" ou *background*). On peut alors jouer sur la sélection de ce corpus d'imposteurs [Rosenberg et al., 1992] pour améliorer les performances. Le lecteur peut se référer à [Reynolds, 1997] pour une comparaison des diverses méthodes de calcul de rapport de vraisemblance.

Pour des soucis d'efficacité mais aussi pour améliorer la robustesse, les vraisemblances des vecteurs d'observations sont couramment estimées par "*N-best scoring*" (où typiquement $N = 10$). Il s'agit d'estimer pour chaque vecteur $\mathbf{x}_t$ d'une séquence *test* :

$$p_{Nbest}(\mathbf{x}_t|\left\{\omega_g^{\text{UBM}}, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g^{\text{UBM}}\right\}) = \sum_{i=1}^{N} \omega_{best(i)}^{\text{UBM}} \mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_{best(i)}, \boldsymbol{\Sigma}_{best(i)}^{\text{UBM}})$$

$$\text{au lieu de} \quad \sum_{g=1}^{G} \omega_g^{\text{UBM}} \mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g^{\text{UBM}})$$

où les Gaussiennes sont rangées par ordre décroissant de vraisemblance du vecteur observé au Modèle du Monde : $\omega_{best(1)}^{\text{UBM}} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{best(1)}^{\text{UBM}}, \boldsymbol{\Sigma}_{best(1)}^{\text{UBM}}) > \cdots > \omega_{best(N)}^{\text{UBM}} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{best(N)}^{\text{UBM}}, \boldsymbol{\Sigma}_{best(N)}^{\text{UBM}}) > \cdots > \omega_{best(G)}^{\text{UBM}} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{best(G)}^{\text{UBM}}, \boldsymbol{\Sigma}_{best(G)}^{\text{UBM}})$. Étant donné un GMM adapté de l'UBM (tel le GMM du locuteur cible), le calcul des vraisemblances se fait de manière analogue en ne considérant que les N Gaussiennes dérivées des N "meilleures" Gaussiennes de l'UBM. Cette astuce permet alors, une fois calculée la vraisemblance à l'UBM, d'accélérer l'estimation de la vraisemblance aux autres modèles dérivés du Modèle du Monde. Ceci allège considérablement les calculs de normalisation des scores par T-Norm (§2.1.6). Sans cette astuce, un tel post-traitement des scores serait trop coûteux pour être applicable dans les protocoles d'évaluation NIST.

2.3 Systèmes SVMs

Les SVMs sont réputés pour fournir de bonnes performances dans les problèmes de classification binaire et sont devenus de plus en plus populaires dans de nombreux domaines, dont le traitement de la parole. Étant donné que la vérification du locuteur est par essence une tâche de classification binaire, on pourrait s'attendre à ce que les SVMs soient une méthode de modélisation classique dans le domaine. Or aucun système SVM de vérification du locuteur ne peut être encore admis comme système SVM de référence, contrairement au système UBM-GMM sur des paramètres cepstraux qui est la base de la modélisation générative des systèmes implémentés dans la plupart des laboratoires. Toutefois, la communauté de reconnaissance du locuteur commence à adopter des SVMs à noyaux de séquences. Le noyau GLDS [Campbell, 2001] est par exemple reconnu pour son efficacité et sa simplicité d'implémentation. Aussi un noyau récemment proposé par [Campbell et al., 2006b], basé une modélisation GMMs des distributions sur les séquences, a connu un certain succès lors de la dernière évaluation NIST SRE 2006.

Nous passerons plus tard en revue les noyaux de séquences (§3.5.2) et nous appliquerons dans

nos expériences de vérification du locuteur les SVMs à noyaux de séquences dont la complexité calculatoire est acceptable pour les protocoles d'évaluation NIST (chapitre 5). Dans cette section, nous évoquons deux catégories de systèmes SVMs pour la vérification du locuteur qui ne sont pas basés sur les noyaux de séquences. La partie §2.3.1 décrit une approche triviale du problème avec des noyaux calculés sur les vecteurs acoustiques, qui a été à l'origine de la première application des SVMs à la vérification du locuteur [Schmidt et Gish, 1996]. Cette approche souffre de plusieurs problèmes dont nous discuterons. La partie §2.3.2 donne un aperçu des approches SVM où les noyaux manipulent les scores de vraisemblance GMM. Il s'agit de méthodes de post-traitement pour la modélisation UBM-GMM. En cela, elles ne constituent pas une véritable alternative aux systèmes UBM-GMM.

2.3.1 Extension de l'approche vectorielle SVM

La première tentative d'appliquer les SVMs à la reconnaissance du locuteur, par [Schmidt et Gish, 1996] pour une tâche d'identification, consistait à le faire dans le cadre vectoriel au niveau des trames de parole. Lors de l'apprentissage, les entrées sont les vecteurs extraits du signal de parole (les trames et non les séquences). En phase de test, le score d'une séquence est une moyenne des scores obtenus pour chaque vecteur de la séquence, de manière analogue à l'approche GMM (où les scores sont des log-vraisemblances). Mais agir ainsi présente plusieurs inconvénients :

1. Impossibilité d'exploiter de gros corpus de développement, à cause de la complexité de l'algorithme d'apprentissage. Ce problème est rapporté par [Imbiriba et al., 2004], et étudié plus en détail par [Staroniewicz et Majewski, 2004].
2. Inadéquation du critère d'apprentissage. En effet, le but recherché est de classer des séquences et non des trames isolées de parole. Les scores SVMs des trames ne correspondant pas à des probabilités, on ne sait pas déterminer la manière optimale de les combiner. Calculer la moyenne est une option de facilité par défaut.
3. Manque de robustesse, à cause du faible pouvoir discriminant des vecteurs pris isolément. En pratique, ces vecteurs représentent seulement environ 20ms de parole et sont corrompus par le bruit. on peut prévoir un manque de robustesse des algorithmes d'apprentissage "locaux" [Bengio et LeCun, 2006] comme les SVMs avec un noyau RBF Gaussien. La "localité" de ces algorithmes connote leur incapacité à modéliser le signal (ici, séquence de parole) dans sa globalité.

L'utilisation de noyaux de séquences permet de surmonter ces difficultés. Un avantage purement pratique de tels noyaux est la réduction du nombre des entrées dans l'algorithme d'apprentissage SVM. Par exemple, manipuler des séquences de 2 minutes diminue le nombre d'exemples d'apprentissage d'environ 10^4 (nombre de vecteurs par séquence). Aussi, un avantage fondamental des noyaux de séquence est la possibilité de formuler un critère d'apprentissage adéquat au problème de classification de séquences. Enfin, les noyaux de séquences sont aptes à capturer des informations relatives à la distribution des vecteurs acoustiques dans la séquence, typiquement plus robuste au bruit et aux observations aberrantes. Tout cela explique pourquoi les résultats des SVMs vectoriels en reconnaissance du locuteur sont nettement moins bons que ceux des SVMs à noyaux de séquences, comme constaté par [Wan et Renals, 2002].

Quelques études ont été menées pour surmonter les problèmes rencontrés avec les SVMs à noyaux vectoriels. Par exemple, pour limiter la complexité en apprentissage, on peut envisager

de procéder à une Quantification Vectorielle des données de développement, comme suggère [Wan et Renals, 2002]. Le dictionnaire obtenu par quantification vectorielle est utilisé comme corpus d'apprentissage au lieu du corpus volumineux de départ. Mais cette réduction d'information est purement arbitraire et n'a rien d'optimal étant donné qu'elle ne tient pas compte de la fonction noyau choisie : on n'a aucune maîtrise sur les approximations faites.

Une approche plus élégante a été formulée par [Lei et al., 2005], où l'idée d'utiliser des "mélanges de SVMs" proposée par [Collobert et al., 2002] est reprise et adaptée à la vérification du locuteur, en s'inspirant de la méthodologie UBM-GMM. En apprentissage, les données sont classées en utilisant un critère de proximité aux vecteurs d'un dictionnaire obtenu par Quantification Vectorielle non supervisée. Chaque sous-ensemble de données (*cluster*) est alors utilisé pour apprendre un modèle SVM. Les trames d'une séquence test sont classées de même manière analogue et un score leur ait affecté en utilisant le modèle SVM adéquat. Une fonction sigmoïde logit est ensuite appliquée aux scores SVM des trames de manière à les ajuster empiriquement à des probabilités *a posteriori* selon l'étude menée par [Platt, 2000]. Cela justifie le moyennage des log-scores obtenus par les vecteurs d'une séquence testée. Les astuces conçues par [Lei et al., 2005] pour surmonter les problèmes de complexité sont inspirées de la méthode UBM-GMM. Dans la partie expérimentale (§5.2.2), nous décrirons plus en détail le système de mélanges de SVMs vectoriels et nous l'appliquerons à un protocole d'évaluation NIST. Nous constaterons que les performances sont visiblement moins bonnes que celles des SVMs à noyaux de séquences, alors que la complexité calculatoire reste élevée.

2.3.2 Post-traitement des scores GMMs par les SVMs

D'autres travaux en vérification du locuteur ont porté sur l'amélioration de la prise de décision UBM-GMM grâce à une modélisation SVM. Il s'agit de manière générale d'entraîner des modèles SVM sur des paramètres construits à partir des vraisemblances aux GMMs. Une première idée lancée par [Bengio et Mariéthoz, 2001] consiste à incorporer les scores fournis par les GMMs dans un SVM. D'après les règles de Bayes, l'équation (2.4) est optimale tant que le locuteur cible et les imposteurs (représentés par l'UBM) sont bien modélisés. Or l'estimation des vraisemblances n'est pas parfaite en pratique. [Bengio et Mariéthoz, 2001] propose alors d'augmenter le rapport de log-vraisemblances avec la pondération suivante :

$$\text{score}_{\text{loc}}(\mathbf{X}) = a \log p(\mathbf{X}|\boldsymbol{\theta}_{\text{loc}}) - b \log p(\mathbf{X}|\boldsymbol{\theta}_{\text{UBM}}) + c \tag{2.5}$$

où a, b et c sont des paramètres ajustables. Ces paramètres sont appris par un SVM linéaire sur un corpus de validation. Les vecteurs 2-dimensionnels à classer dans les SVMs sont composés des log-vraisemblances aux modèles du locuteur et du monde, $p(\mathbf{X}|\boldsymbol{\theta}_{\text{loc}})$ et $\log p(\mathbf{X}|\boldsymbol{\theta}_{\text{UBM}})$. Le concept peut être généralisé en utilisant un noyau non linéaire dans le SVM (ce qui induit une autre forme que (2.5) pour la fonction score apprise). Dans [Bengio et Mariéthoz, 2001], on peut voir qu'un noyau Gaussien avec une forte zone d'influence (§3.2.2, paramètre ρ) améliore légèrement les performances.

Dans le cas où l'on dispose suffisamment de données d'apprentissage par locuteur cible pour se permettre de faire de la validation croisée, [Le et Bengio, 2003] rapporte que les performances peuvent être améliorées en rendant spécifique à chaque locuteur cible l'apprentissage SVM, c'est-à-dire en limitant les tests positifs à des tests factices impliquant le modèle du locuteur cible, dans l'ensemble de validation. Pour un noyau linéaire, ceci revient à construire des fonctions

scores de forme spécifique à chaque locuteur :

$$\text{score}_{\text{loc}}(\mathbf{X}) = a_{\text{loc}} \log p(\mathbf{X}|\boldsymbol{\theta}_{\text{loc}}) - b_{\text{loc}} \log p(\mathbf{X}|\boldsymbol{\theta}_{\text{UBM}}) + c_{\text{loc}} \tag{2.6}$$

[Liu et al., 2006] généralise le travail de [Bengio et Mariéthoz, 2001] dans le cas d'un 1-*best scoring* (§2.2.3). La forme particulière des 1-*best scores* lorsque les matrices de covariance sont diagonales peut se développer sous forme de combinaison linéaire de distributions Gaussiennes mono-dimensionnelles $\mathcal{N}_u$, relatives à chacun des paramètres d'entrée indexés par u

$$\begin{aligned} \text{score}_{\text{loc}}(\mathbf{X}) &= \sum_t \left[\log \mathcal{N}(\mathbf{x}_t|\boldsymbol{\theta}_{\text{loc}}^{\text{1best(t)}}) - \log \mathcal{N}(\mathbf{x}_t|\boldsymbol{\theta}_{\text{UBM}}^{\text{1best}(t)}) \right] \\ &= \sum_{u=1}^{d} \sum_t \left[\log \mathcal{N}_u(x_{t,u}|\theta_{\text{loc},u}^{\text{1best(t)}}) - \log \mathcal{N}_u(x_{t,u}|\theta_{\text{UBM},u}^{\text{1best}(t)}) \right] \end{aligned} \tag{2.7}$$

où l'on note $\mathbf{x}_t = [x_{t,1} \cdots x_{t,d}]^{\text{T}}$ les vecteurs d'entrée. Basé sur l'observation que des classifieurs 1-*best scoring* UBM-GMM appris respectivement sur chaque paramètre d'entrée x_u promettent des performances différentes, l'idée est d'apprendre les coefficients $\{a_u, b_u, c_u\}_{u=1\cdots d}$ pour chaque paramètre d'entrée (jouant des rôles analogues à $\{a, b, c\}$ dans l'équation 2.5). D'autres approches intuitives basées sur le post-traitement des 1-*best scores* avec la modélisation SVM ont été proposées par [Fine et al., 2001, Kharroubi et al., 2001]

Chapitre 3

Noyaux de vecteurs et de séquences

Sommaire

DANS ce chapitre, nous présentons le concept de noyaux, à la base d'un groupe de méthode appelées "méthodes à noyau" (*kernel methods*), parmi lesquelles on compte les SVMs (§1.3). Nous donnons les éléments essentiels qui permettent de construire et choisir un noyau pour un problème de classification. Nous nous intéressons particulièrement à la classification de données numériques, c'est-à-dire de vecteurs et séquences de vecteurs.

De manière sommaire, l'astuce du noyau permet d'appliquer une grande famille d'algorithmes à tout type de données (numériques, symboliques, structurées, etc.) en tenant compte de relations complexes entre les caractéristiques mesurées. Pour la classification de données numériques, il s'agit par exemple d'exploiter les corrélations non linéaires entre les composantes d'entrée pour mieux capturer leur pouvoir discriminant. Notons que l'astuce du noyau ne sert pas seulement aux problèmes de classification ou de régression, comme c'est le cas avec les SVMs. Elle peut s'appliquer dans le cadre général de l'analyse de données et permet par exemple d'analyser des corrélations non-linéaires, et les dépendances entre caractéristiques d'entrée [Gretton et al., 2005]. Ainsi ont été proposés avec l'astuce du noyau des généralisations de l'Analyse en Composantes Principales [Schölkopf et Smola, 2002, Mika et al., 1999] et de l'Analyse en Composante Indépendantes [Bach et Jordan, 2002],

Dans ce chapitre, le concept de noyau est d'abord introduit avec les principales notions mathématiques auxquelles il renvoie de manière générale (§3.1). Nous donnons ensuite des exemples de fonctions noyaux qui s'appliquent aux vecteurs de dimension fixe et les éléments qui permettent de régler leurs paramètres en fonction de la complexité de modélisation désirée (§3.2). Avant d'introduire les différents noyaux conçus pour les séquences de vecteurs, nous présenterons deux familles de noyaux qui peuvent être appliqués aux séquences de tailles variables grâce à leur fondement probabiliste. La première rassemble plusieurs noyaux qui permettent de mesurer la similarité entre les fonctions de distribution de probabilité (§3.3). La seconde réunit les "noyaux d'Information Mutuelle" (§3.4) qui peuvent s'appliquer à n'importe quel type de données du moment que l'on sait estimer des distributions de probabilité. Les noyaux de séquences envisageables pour la vérification du locuteur sont finalement récapitulés (§3.5) : ils s'appuient sur les trois catégories de noyaux décrits en §3.2, §3.3 et §3.4.

3.1 Généralités sur les noyaux

3.1.1 L'astuce du noyau

En pratique, l'*astuce du noyau* consiste à réécrire un algorithme où toutes les relations entre données d'entrée peuvent s'écrire sous forme de produits scalaires, en remplaçant ce produit scalaire par une fonction scalaire de deux variables ("noyau"). L'astuce du noyau permet ainsi de généraliser un algorithme linéaire manipulant des vecteurs :

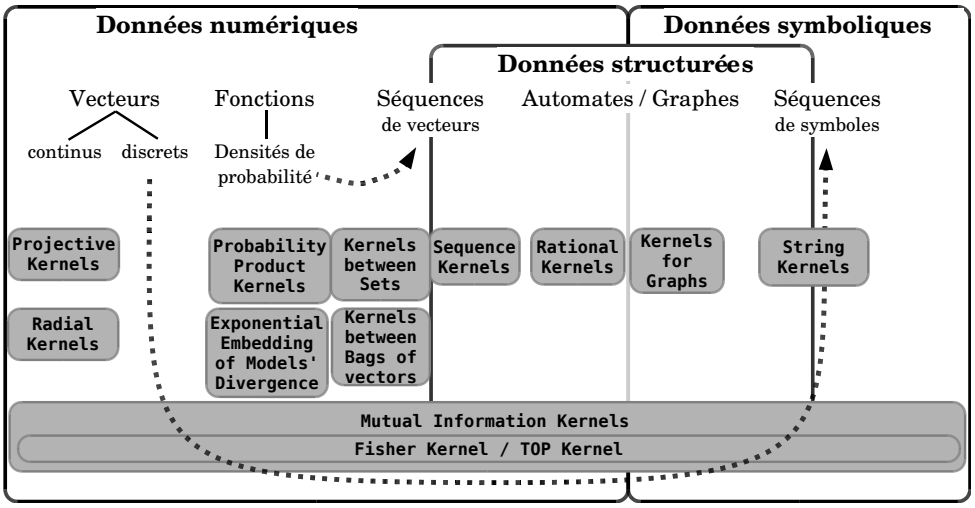
1. pour traiter les vecteurs de façon non linéaire (parce que les données présentent des non-linéarités qu'il est utile d'exploiter pour le problème visé) ;

ou

2. pour manipuler d'autres types d'objets que les vecteurs.

Concernant le second point, l'astuce du noyau permet par exemple traiter des objets symboliques [Saunders, 2002] comme les chaînes de caractères [Lodhi et al., 2002] ou les séquences de protéines [Leslie et al., 2002b], ou encore des objets structurés [Gärtner, 2003] comme les graphes [Kashima et al., 2004] ou les automates [Cortes et al., 2003, Cortes et al., 2004]. Nous verrons aussi dans ce chapitre que l'on peut manipuler directement des séquences de tailles variables (ordonnées ou non) ainsi que des modèles statistiques. Le tableau 3.1 liste les noms de noyaux couramment utilisés en anglais selon le type d'objets manipulés. Par exemple, pour les noyaux de séquences de données discrètes, il est de coutume de parler de "*String Kernels*", en bioinformatique [Leslie et al., 2002a], catégorisation de texte [Lodhi et al., 2002] et reconnaissance de la parole [Goddard et al., 2003].

Tab. 3.1 - Liste de noyaux, selon objets manipulés en entrée.



D'un point de vue qualitatif, le noyau peut être vu une *mesure de similarité*, qui permet de comparer deux objets d'un même type. Pour appliquer les méthodes à noyau sur un ensemble de données, il suffit en pratique de connaître les valeurs de noyaux pour tous les couples de cet ensemble. Par exemple, pour dérouler l'algorithme d'apprentissage SVM, il suffit de connaître les valeurs de noyaux estimées sur le corpus d'apprentissage (§1.3.2) Ces valeurs sont habituellement mémorisées dans une matrice carrée : la "matrice de Gram".

Définition 2 (Matrice de Gram).
Soient une fonction noyau $k : \mathbb{X}^2 \to \mathbb{R}$ et un ensemble de données $\mathcal{A} = \{\mathrm{a}_i\}_{i=1\cdots N}$ de taille N. La "matrice de Gram" $\mathbf{K}_\mathcal{A}$ est définie par la matrice carrée $N \times N$ contenant les valeurs du noyau sur les couples :

$$(\mathbf{K}_\mathcal{A})_{i,j} = k(\mathrm{a}_i, \mathrm{a}_j) \tag{3.1}$$

Dans toute la suite, nous notons k le noyau et $\mathbb{X}$ l'espace d'entrée.

3.1.2 Propriétés mathématiques

Tout comme le produit scalaire, on peut attendre de la fonction noyau qu'elle soit définie positive (définition 3). Avant d'en venir aux avantages d'une telle propriété mathématique, nous rappelons sa définition.

Définition 3 ((Semi-)défini-positivité).
Une fonction scalaire $k : \mathbb{X}^2 \to \mathbb{R}$ est "semi-définie positive" si et seulement si

$$\forall\ (\psi : \mathbb{X} \to \mathbb{R}) \neq \mathbf{0}\ , \qquad \iint_{\mathbb{X}^2} k(\mathrm{x}, \mathrm{y})\psi(\mathrm{x})\psi(\mathrm{y})\,\mathrm{dx}\,\mathrm{dy} \geq 0$$

Elle est "définie positive" (propriété plus forte) *ssi* elle vérifie la même propriété avec une inégalité stricte ('$>$' au lieu des '$\geq$').
Une matrice carrée $\mathbf{K}$ de taille $N \times N$ est "semi-définie positive" (resp. "définie positive") *ssi* pour tout vecteur colonne $\boldsymbol{\psi} \in \mathbb{R}^N$ non nul, $\boldsymbol{\psi}^\mathrm{T}\mathbf{K}\boldsymbol{\psi} \geq 0$ (resp. $\boldsymbol{\psi}^\mathrm{T}\mathbf{K}\boldsymbol{\psi} > 0$).

Les fonctions scalaires symétriques et définies positives, que l'on désigne souvent simplement par "noyaux", sont plus précisément des "noyaux de Mercer"[8]. Cette expression vient de ce que l'on appelle "théorème de Mercer" [Shawe-Taylor et Cristianini, 2004], qui est à rigoureusement parler un corollaire du théorème de Mercer.

.../...

[8] On trouve aussi le terme de "*covariance kernels*" [Seeger, 2002a]

Théorème 2 (Théorème de Mercer).

Si un noyau $k : \mathbb{X}^2 \to \mathbb{R}$ *est symétrique et semi-défini positif, alors il admet un développement de la forme*

$$\forall\ (\mathrm{x},\mathrm{y}) \in \mathbb{X}^2\ , \qquad k(\mathrm{x},\mathrm{y}) = \sum_{u=1}^{D} \phi_u(\mathrm{x})\phi_u(\mathrm{y})$$

où $D \leq +\infty$ *et où les* D *fonctions scalaires* $\phi_u : \mathbb{X} \to \mathbb{R}$ *peuvent être choisies parmi une famille orthonormée. Les conditions de symétrie et de semi-défini-positivité sont nécessaires et suffisantes.*

Autrement dit, si $k : \mathbb{X}^2 \to \mathbb{R}$ est défini positif, alors il peut s'exprimer comme un produit scalaire dans un espace vectoriel où sont projetées les données, qui est appelé "*Feature Space*". Inversement, si l'on définit une correspondance entre des données d'entrée et un espace vectoriel, alors le produit scalaire dans cet espace vectoriel sera un noyau défini positif. L'*expansion* correspondant à un noyau est une fonction $\boldsymbol{\phi} : \mathbb{X} \to \mathbb{R}^D$ telle que

$$\forall\ (\mathrm{x},\mathrm{y}) \in \mathbb{X}^2\ ,\ k(\mathrm{x},\mathrm{y}) = \boldsymbol{\phi}(\mathrm{x})^{\mathrm{T}}\boldsymbol{\phi}(\mathrm{y}) \tag{3.2}$$

Notons que les données d'entrée projetées dans le *Feature Space* $\mathbb{R}^D$ gisent sur un sous-ensemble ouvert de $\mathbb{R}^D$ qui est une variété de Riemann. A contrario, les éléments du *Feature Space* $\mathbb{R}^D$ n'admettent pas forcément de pré-image dans l'espace d'entrée [Mika et al., 1999]. Fig.3.1 représente la variété correspondant à des données d'entrée bidimensionnelles et à un noyau polynomial (§3.2.1) de degré deux : $k(\mathbf{x},\mathbf{y}) = (\mathbf{x}^{\mathrm{T}}\mathbf{y})^2$.

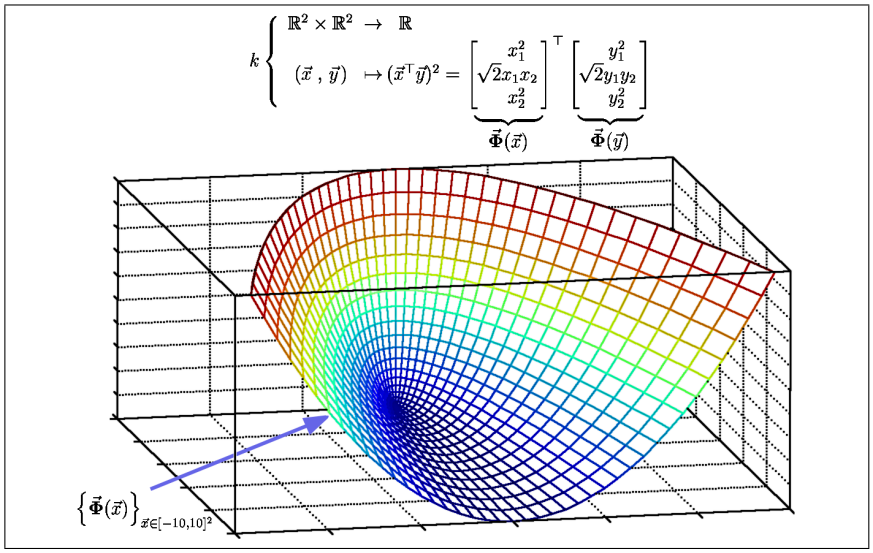


Fig. 3.1 - Variété de Riemann correspondant à un noyau polynomial.

La VC-dimension (§1.3.1) des fonctions discriminantes formées à partir d'un noyau correspondant à un *Feature Space* de dimension D est généralement $(D+1)$. Pour certaines fonctions noyau symétrique et définies positives, D peut être infini et $\boldsymbol{\phi}$ n'a pas toujours une approximation analytique connue. En fait, le théorème de Mercer est un théorème d'existence. Il ne fournit

par contre aucun moyen de déduire l'expansion $\boldsymbol{\phi}$ sous-jacente à un noyau k donné. Ainsi, il existe des noyaux pour lesquels aucune approximation analytique n'a encore été trouvée. C'est le cas du noyau Gaussien (§3.2.2), pour lequel il n'a que récemment été proposée une expression analytique de l'expansion dans le cas de données uni-dimensionnelles [Xu et al., 2006]. Cette approximation se résume au développement limité suivant :

$$\forall\ (x,y) \in \mathbb{R}^2, \quad \boldsymbol{\phi}^D(x)^{\mathrm{T}} \boldsymbol{\phi}^D(y) \xrightarrow{D \to +\infty} e^{-\frac{(x-y)^2}{2\rho^2}}$$

$$\text{avec} \quad \boldsymbol{\phi}^D \left\{ \begin{array}{ccc} \mathbb{R} & \to & \mathbb{R}^D \\ x & \mapsto & e^{-\frac{x^2}{2\rho^2}} \left[1, \frac{x}{\rho}, \cdots, \frac{x^D}{D!\rho^D}\right]^{\mathrm{T}} \end{array} \right.$$

Revenons maintenant à la matrice de Gram (définition 2), notion centrale des méthodes à noyau. Il est facile de constater que les propriétés d'un noyau k se répercutent sur la matrice de Gram $\mathbf{K}_{\mathcal{A}}$ quelque soit l'ensemble de donné $\mathcal{A}$ considéré :

- Si la fonction noyau k est symétrique, alors la matrice de Gram $\mathbf{K}_{\mathcal{A}}$ est symétrique.
- Si k est définie positive, alors la matrice de $\mathbf{K}_{\mathcal{A}}$ est définie positive.

Inversement, si la matrice de Gram sur des données d'apprentissages est symétrique et définie positive, alors la propriété de Mercer est vérifiée *sur ces données d'apprentissage* (même si l'on a aucune garanti sur le défini-positivité du noyau) :

$$\mathbf{K}_{\mathcal{A}}\ \textit{sym.déf-pos.} \Rightarrow \left[\exists\ \{\phi_u\}_{u \in \mathbb{N}^+}\ \ \text{tq} \quad \forall\ (\mathrm{a}_i, \mathrm{a}_j) \in \mathcal{A}^2\ ,\ k(\mathrm{a}_i, \mathrm{a}_j) = \sum_{u=1}^{+\infty} \phi_u(\mathrm{a}_i)\phi_u(\mathrm{a}_j) \right]$$

La propriété de défini-positivité d'un noyau entraîne plusieurs propriétés qui aident à la résolution d'une grande famille d'algorithme d'optimisation, comme l'inégalité de Cauchy-Schwartz :

$$\forall\ (\mathrm{x}, \mathrm{y}) \in \mathbb{X}^2\ , \qquad k(\mathrm{x}, \mathrm{y}) \leq \sqrt{k(\mathrm{x}, \mathrm{x}) k(\mathrm{y}, \mathrm{y})}$$

De manière générale, la symétrie et la défini-positivité d'une fonction noyau présentent plusieurs avantages :

1. *Avantage pratique* - Ces propriétés permettent la convergence d'algorithmes basés sur une programmation quadratique. Pour les SVMs par exemple (§1.3), une matrice de Gram définie positive garantit la convergence de l'algorithme d'apprentissage (quadratique) vers une solution optimale et unique. Dans le cas contraire, l'algorithme d'apprentissage peut converger vers une solution localement mais pas globalement optimale ou, pire, ne pas converger (problèmes numériques).
2. *Avantage théorique* - En considérant l'expansion sous-jacente $\boldsymbol{\phi}$ dont l'existence est garantie par le théorème de Mercer, on peut interpréter les algorithmes à noyau comme des algorithmes linéaires dans le *Feature Space*. La matrice de Gram encode alors les positions relatives d'un ensemble de données dans le *Feature Space*.

Notons que des exemples pratiques ont montré qu'un SVM avec un noyau non défini positif pouvait donner de bonnes performances [Bahlmann et al., 2002, DeCoste et Schölkopf, 2002, Wallraven et al., 2003, Boughorbel et al., 2004, Mariéthoz et Bengio, 2006][9]. Plusieurs travaux ont été menés sur l'étude théorique des noyaux non définis positifs. [Boughorbel et al., 2004]

[9] Bien que [Wallraven et al., 2003] prétende que le noyau conçu soit défini positif, une partie des auteurs ont rectifié cette incorrection dans un article postérieur [Fritz et al., 2005].

propose une étude statistique de la défini-positivité de la matrice de Gram lorsque le noyau n'est pas défini positif. D'autres études plus théoriques fournissent des interprétations permettant de mieux les comprendre les noyaux non défini positifs :

- [Ong et al., 2004] propose une généralisation des RKHS : les "*Reproducing Kernel Kreĭn Spaces*". Dans ce type d'espace, il s'agit de "stabiliser" une fonction de coût, au lieu de la minimiser comme dans un apprentissage SVM classique. La résolution s'appuie alors sur une généralisation du théorème 1 des représentants.
- [Haasdonk, 2005] introduit une généralisation du concept de *Feature Space* vectoriel : les espaces "pseudo-euclidiens", qui mettent en jeu des données complexes (avec une partie imaginaire). L'auteur propose une méthode de résolution du problème d'apprentissage SVM en conséquence de cette nouvelle approche.

3.1.3 Noyau & complexité

Choix du noyau

À l'origine, le choix du noyau est arbitraire et fait partie intégrante du choix de la représentation des données. Pour un SVM par exemple, il est de coutume de choisir un noyau, puis d'utiliser les données pour choisir un modèle dans le RKHS engendré par ce noyau (définition 1) en fonction du critère de compromis biais-variance. Indépendamment de ce critère d'apprentissage, le noyau incorpore une notion de complexité / capacité de modélisation étant donné qu'il définit l'espace dans lequel on recherche les solutions En pratique l'on ne sait pas encore bien déterminer "quel noyau choisir ?" à partir de données.

[Lanckriet et al., 2002] propose une méthode justifiée théoriquement, qui consiste à construire le noyau en fonction de données étiquetées, pour un problème de classification supervisée. La limitation pratique, assez rédhibitoire dans le cas d'un espace d'entrée continu, est que l'on ne peut pas étiqueter des entrées qui n'auraient pas été observées en apprentissage. En fait, dans la méthode proposée, l'apprentissage du modèle de classification revient à apprendre des correspondances entrées $\leftrightarrow$ sorties, avec toutes les entrées dont on dispose en apprentissage (une partie pouvant ne pas être étiquetées). Ces correspondances induisent un noyau défini positif, dont on ne peut toutefois pas trouver une expression analytique. Dans le cadre d'une application réaliste où les données sont numériques, les exemples à classer auront peu de chance de coïncider avec des observations d'apprentissage ; À moins de faire des approximations au plus proche voisin (ce qui alourdit l'algorithme et menace le pouvoir de généralisation du modèle), il faut réitérer la phase d'apprentissage pour chaque donnée de test.

Nous verrons plus tard que les noyaux d'Information Mutuelle (§3.4) permettent de construire de manière élégante des noyaux à partir d'un corpus de données non étiquetées (ou plus exactement à partir de leur densité *a priori*, estimée par un modèle paramétrique). Aussi, au lieu de s'attacher à construire des mesures de similarité adaptées aux données, d'autres travaux s'intéressent à projeter et/ou normaliser les données de départ, de manière à ce que la distance euclidienne dans le nouvel espace soit adaptée aux données [Peltonen et al., 2004].

Quoiqu'il en soit, même si la forme paramétrique du noyau est choisie arbitrairement, le choix des paramètres du noyau se fait souvent par validation croisée, tout comme le choix du paramètre C de compromis biais-variance pour l'apprentissage SVM (§1.3). Nous rappelons

que la validation croisée consiste à séparer l'intégralité du corpus d'apprentissage $\mathcal{A} = \{\mathrm{a}_i, \boldsymbol{\ell}_i\}$ en n partie complémentaires $\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_n = \mathcal{A}$. Pour plusieurs jeux de paramètres choisis itérativement, on mesure la moyenne des performances du SVM obtenues en entraînant sur le corpus formé par la réunion de $(n-1)$ parties $\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_{k-1} \cup \mathcal{A}_{k+1} \cup \cdots \cup \mathcal{A}_n$ ($k \in \{1 \cdots n\}$), et en testant sur la partie complémentaire $\mathcal{A}_k$. On choisit finalement le jeu de paramètres qui fournit les meilleures performances. À partir de ce jeu empiriquement optimal, on entraîne un modèle SVM sur toutes les données du corpus d'apprentissage $\mathcal{A}$.

Spectre d'un noyau

La complexité d'un noyau se manifeste dans son "spectre", défini ci-dessous.

Définition 4 (Spectre d'un noyau / d'une matrice symétrique).

Les "valeurs propres" d'un noyau de Mercer k associées à une probabilité *a priori* $\mathrm{p}(\mathrm{x})$ sur les données, sont les scalaires λ_i qui vérifient :

$$\exists\ \psi_i : \mathbb{X} \to \mathbb{R} \text{ avec } \int_{\mathbb{X}} \psi_i(\mathrm{x})^2\,\mathrm{dx} = 1 \text{ tq} \quad \forall\ \mathrm{y} \in \mathbb{X}\ ,\ \int_{\mathbb{X}} k(\mathrm{x},\mathrm{y})\mathrm{p}(\mathrm{x})\psi_i(\mathrm{x})\,\mathrm{dx} = \lambda_i\ \psi_i(\mathrm{y})$$

Les fonctions ψ_i (de norme unitaire) sont appelées "fonctions propres" (*eigenfunctions*) associées aux valeurs propres λ_i correspondantes.

Le "spectre" d'un noyau désigne alors la liste (infinie) des valeurs propres λ_i rangées par ordre décroissant $\{\lambda_i\}_{i=1\cdots\infty}$ ($\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_\infty$).

Les valeurs propres d'une matrice $\mathbf{K}$ symétrique sont réelles et positives si la matrice est définie positive. Si N est la taille de la matrice, il s'agit des scalaires λ_i^* qui permettent la Décomposition en Valeurs Singulières :

$$\mathbf{K} = \mathbf{V} \begin{bmatrix} \lambda_1^* & & & \\ & \lambda_2^* & & (0) \\ & & \ddots & \\ (0) & & & \lambda_N^* \end{bmatrix} \mathbf{V}^{\mathrm{T}} \tag{3.3}$$

où $\mathbf{V}$ est une matrice orthonormale ($\mathbf{V}\mathbf{V}^{\mathrm{T}} = \mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{I}_N$).

De manière analogue au cas des fonctions, les vecteurs colonnes de $\mathbf{V}$ intervenant dans cette décomposition sont les "vecteurs propres" (*eigenvectors*) de $\mathbf{K}$.

Le spectre de la matrice $\mathbf{K}$ est défini comme la liste des valeurs propres rangées par ordre décroissant $\{\lambda_i^*\}_{i=1\cdots N}$ ($\lambda_1^* \geq \lambda_2^* \geq \cdots \geq \lambda_N^*$).

Plutôt que d'estimer le spectre d'une fonction noyau, il est pratique de calculer le spectre d'une matrice de Gram sur un ensemble de données. Cela permet de ne pas avoir à modéliser la distribution *a priori* $p(\mathrm{x})$ des données. La relation entre les valeurs propres d'un noyau et les valeurs propres d'une matrice de Gram est donnée par le théorème suivant (théorème 3.4 de [Baker, 1977]) :

Théorème 3 (Relations entre valeurs propres de noyau / matrice de Gram).
Considérons un ensemble $\mathcal{A}$ de N données indépendantes et générées par une variable aléatoire issue d'une probabilité $p(\mathrm{x})$. Alors avec les mêmes notations que la définition 4, *les valeurs propres du noyau k peuvent être estimées empiriquement à partir des valeurs propres de la matrice de Gram selon la relation de convergence :*

$$\frac{1}{N}\lambda_i^* \xrightarrow{N\to\infty} \lambda_i \tag{3.4}$$

Pour un ensemble de données, **plus un noyau est complexe et moins le spectre de la matrice de Gram décroît rapidement**. Cela sera observé plus tard (§3.2, Figs. 3.2 et 3.3) lorsque nous représenterons des spectres de matrice de Gram en échelle logarithmique[10] $\frac{1}{N}\lambda_i^*$, le rapport $\frac{1}{N}$ étant affecté d'après (3.4). Pour vérifier ce résultat qualitatif, intéressons-nous à deux cas extrêmes :

- Le noyau constant $k(\mathrm{x},\mathrm{y}) = 1$ est le noyau le moins complexe que l'on puisse concevoir. Il revient à projeter toutes les données en un seul point (dans le *Feature Space*) et rend la tâche de classification impossible à cause de cette représentation trop naïve. Pour tout ensemble de données, la matrice de Gram correspondant à ce noyau est alors la matrice remplie de uns, de rang unitaire. Si N désigne la taille de la matrice de Gram, le spectre de cette matrice est $\{N, 0, 0, \cdots\}$ (décroissance abrupte).
- Le noyau le plus complexe envisageable est le noyau défini par la fonction de Dirac :

$$k_\delta(\mathrm{x},\mathrm{y}) = \begin{cases} 1 & \text{si } \mathrm{x} = \mathrm{y} \\ 0 & \text{sinon} \end{cases} \tag{3.5}$$

  Ce noyau binaire correspond à un *Feature Space* de dimension infini où toutes les données sont orthogonales ; il permet ainsi de séparer tout jeu de données étiquetées ± 1. Pour tout ensemble de données de taille N, la matrice de Gram correspondant à ce noyau est la matrice identité ($\mathbf{K} = \mathbf{I}_N$) de rang plein N, et de spectre constant $\{1, \cdots, 1\}$.

En définitive, pour bien établir le lien entre la décroissance du spectre et la complexité du noyau à partir de ces deux exemples, on dispose aussi des propriétés d'algèbre linéaire suivantes, auxquelles nous ferons référence plus tard. Le lecteur peut se référer à [Bousquet et Hermann, 2003] pour un plus grand approfondissement sur les liens théoriques entre spectre et complexité.

Théorème 4 (Algèbre linéaire).
Si $\mathbf{K}$ est un matrice carrée de valeurs propres $\lambda_1^*, \cdots, \lambda_N^*$, alors :

- *La somme des valeurs propres de* $\mathbf{K}$ *est égale à la trace de* $\mathbf{K}$. $\sum_{i=1}^N \lambda_i^* = \mathrm{tr}\,(\mathbf{K})$
- *Le produit des valeurs propres de* $\mathbf{K}$ *est égal au déterminant de* $\mathbf{K}$. $\prod_{i=1}^N \lambda_i^* = \det(\mathbf{K})$

Si en plus $\mathbf{K}$ *est semi-définie positive (resp. définie positive), toutes ses valeurs propres sont toutes positives ou nulles (resp. strictement positives). Aussi la somme des valeurs propres (trace) est une norme pour les matrices semi-définies positives.*

[10] L'échelle logarithme entre en jeu pour tenir compte du fait qu'une valeur propre λ_i^* nulle pose un problème numérique dans les algorithmes à noyau ; et aussi parce que le logarithme du spectre des noyaux Gaussiens, avec une distribution normale pour probabilité *a priori*, est linéaire [Shawe-Taylor et al., 2005]

3.1.4 Noyau & Normalisation

Nous avons vu que les algorithmes à noyau comme les SVMs étaient sensibles aux transformations linéaires (§1.3.4). Et ceci même avec un noyau linéaire, qui, bien qu'invariant aux rotations, est sensible aux changements d'échelle sur certaines dimensions d'entrée. Par exemple, si les vecteurs d'entrée $[x_1, x_2]^{\mathrm{T}}$ sont bidimensionnels, alors après une transformation linéaire $[x_1, x_2]^{\mathrm{T}} \mapsto [x_1, hx_2]^{\mathrm{T}}$, le produit scalaire sera h^2 fois plus sensible à des variations de la seconde composante x_2. En règle générale, plus une variable d'entrée x_u a une variance élevée, et plus elle aura d'influence sur les variations du produit scalaire. Cela peut avoir un effet bénéfique sur les performances d'un SVM linéaire si cette variable d'entrée a un pouvoir discriminant plus fort que les autres pour le problème de classification envisagé. Mais en règle générale, on ignore *a priori* le pouvoir discriminant relatif de chaque variable d'entrée, et on préfère normaliser les variables d'entrées, de manière à rendre unitaire leur variance.

La normalisation des données est plus délicate avec des noyaux non linéaires, pour lesquels on ne maîtrise forcément pas les caractéristiques du *Feature Space*. Par exemple, des problèmes numériques surviennent avec certains noyaux mal réglés, lorsque certaines composantes ϕ_u du *Feature Space* varient avec une trop grande amplitude par rapport aux autres composantes. Ces problèmes se traduisent par la singularité de la matrice de Gram (problèmes de précision numérique) ou par une représentation non adéquate des données (risque de surapprentissage). Pour les éviter, l'idéal est de faire une normalisation (implicite) dans le *Feature Space*, ce qui revient en pratique à modifier la fonction noyau plutôt que les données d'entrée.

Une manière classique de normaliser les noyaux est de centrer les données dans le *Feature Space*. Cela permet d'éviter certains problèmes de précision numérique lors de l'apprentissage d'un SVM. En effet, si l'origine du *Feature Space* est trop éloignée de l'enveloppe convexe des *expansions* des données d'apprentissage, alors les produits scalaires entre ces *expansions* présentent des variations trop faibles, et on se rapproche d'un noyau constant (perte d'information). Le centrage dans le *Feature Space* peut se faire de manière implicite, en remplaçant la matrice de Gram $\mathbf{K}$ par la matrice de Gram "centrée" $\widetilde{\mathbf{K}}$:

$$\widetilde{\mathbf{K}} = \mathbf{K} - \tfrac{1}{N}\left(\mathbf{JK} + \mathbf{KJ}\right) + \tfrac{1}{N^2}\mathbf{JKJ} \qquad \text{où } \mathbf{J} = \mathbf{1}_N\mathbf{1}_N^{\mathrm{T}} \text{ est la matrice } N\times N \text{ remplie de 1} \tag{3.6}$$

Pour un corpus d'apprentissage étiqueté, [Meila, 2003] propose d'équilibrer le centrage de la matrice de Gram en tenant compte des proportions de données selon les étiquettes. Dans le chapitre 4, nous montrerons comment opérer implicitement une normalisation de [Mahalanobis, 1936] dans le *Feature Space* à partir des valeurs de noyau sur un ensemble de données.

Aussi, on peut normaliser tout noyau indépendamment des données d'apprentissage, par exemple de la manière intrinsèque suivante [Wan, 2003, Goddard et al., 2004] :

$$k \xmapsto{h\in\mathbb{R}^+} \mathring{k}(\mathrm{x},\mathrm{y}) = \frac{k(\mathrm{x},\mathrm{y}) + h^2}{\sqrt{\left(k(\mathrm{x},\mathrm{x}) + h^2\right)\left(k(\mathrm{y},\mathrm{y}) + h^2\right)}} \tag{3.7}$$

Une telle normalisation peut se faire sans manipuler l'expression analytique du noyau, en modifiant simplement la matrice de Gram. Si l'on note $\mathbf{K} = (K_{ij})_{(i,j)\in\{1,\cdots,N\}}$ cette matrice, alors

remplacer k par $\mathring{k}$ revient par exemple pour $h = 0$ à remplacer $\mathbf{K}$ par $\mathring{\mathbf{K}}$, avec

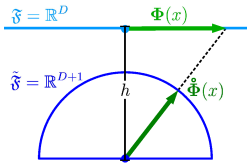
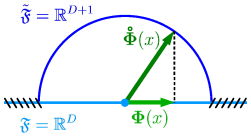
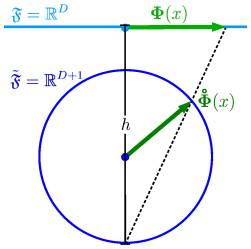
$$\begin{cases} \forall\ i = 1 \cdots N\ , & \mathring{K}_{ii} = 1 \\ \forall\ \{i,j\}\ \text{tq} \quad i \neq j\ , & \mathring{K}_{ij} = \dfrac{K_{ij}}{\sqrt{K_{ii} K_{jj}}} \end{cases} \tag{3.8}$$

Dans le *Feature Space*, cela revient à projeter les points sur une demi-sphère de rayon unitaire. En effet, la propriété de Mercer est conservée par cette normalisation, et si l'on note $\mathring{\boldsymbol{\Phi}}$ l'*expansion* correspondant au noyau normalisé $\mathring{k}$, alors :

$$\forall\ \mathrm{x} \in \mathbb{X}\ , \qquad \|\mathring{\boldsymbol{\Phi}}(\mathrm{x})\| = \sqrt{\mathring{k}(\mathrm{x},\mathrm{x})} = 1 \tag{3.9}$$

D'autres projections sphériques sont envisageables comme indiqué dans le tableau 3.8. La topologie sphérique des données dans le *Feature Space*, induite par les noyaux normalisés sphériquement, a inspiré plusieurs travaux de recherche [Desobry et al., 2005b]. D'un point de vue numérique, pour un algorithme comme l'apprentissage SVM, une normalisation sphérique permet de bien conditionner la matrice de Gram, ce qui permet en pratique une meilleure convergence des algorithmes d'apprentissage.

Tab. 3.8 - Normalisation sphérique de noyaux : Expressions et Illustrations

Expression analytique du noyau normalisé $\mathring{k}(\mathrm{x},\mathrm{y})$	**Projection équivalente $\mathring{\boldsymbol{\Phi}}(\mathrm{x})$**	**Illustration géométrique (dans le *Feature Space*)**
$\dfrac{k(\mathrm{x},\mathrm{y})+h^2}{\sqrt{(k(\mathrm{x},\mathrm{x})+h^2)(k(\mathrm{y},\mathrm{y})+h^2)}}$	$\dfrac{1}{\sqrt{\|\boldsymbol{\Phi}(\mathrm{x})\|^2+h^2}} \begin{bmatrix} \boldsymbol{\Phi}(\mathrm{x}) \\ h \end{bmatrix}$	*Projection gnomonique* 
pour $\|\boldsymbol{\Phi}(\mathrm{x})\| \leq 1$, $k(\mathrm{x},\mathrm{y}) + \sqrt{(1-k(\mathrm{x},\mathrm{x}))\,(1-k(\mathrm{y},\mathrm{y}))}$	$\begin{bmatrix} \boldsymbol{\Phi}(\mathrm{x}) \\ \sqrt{1-\boldsymbol{\Phi}(\mathrm{x})^{\mathrm{T}}\boldsymbol{\Phi}(\mathrm{x})} \end{bmatrix}$	*Projection orthographique* 
$\dfrac{k(\mathrm{x},\mathrm{x})k(\mathrm{y},\mathrm{y})-h^2(k(\mathrm{x},\mathrm{x})+k(\mathrm{y},\mathrm{y})-4k(\mathrm{x},\mathrm{y}))+h^4}{(k(\mathrm{x},\mathrm{x})+h^2)(k(\mathrm{y},\mathrm{y})+h^2)}$	$\begin{bmatrix} \dfrac{2h\boldsymbol{\Phi}(\mathrm{x})}{h^2+\|\boldsymbol{\Phi}(\mathrm{x})\|^2} \\ \dfrac{h^2-\|\boldsymbol{\Phi}(\mathrm{x})\|^2}{h^2+\|\boldsymbol{\Phi}(\mathrm{x})\|^2} \end{bmatrix}$	*Projection stéréographique* 

3.1.5 Combinaison de noyaux

La combinaison de noyaux peut désigner deux choses :

- *Combinaison de valeurs (de noyaux)* - Cela peut servir à combiner plusieurs types d'informations (numérique / symbolique) à l'intérieur de la modélisation. Pour les SVMs par exemple, il en général préférable de regrouper toutes les caractéristiques mesurées au sein d'un seul et même critère d'apprentissage, plutôt que de combiner les sorties de plusieurs SVMs.
- *Combinaison de fonctions (noyaux)* - Cela sert en général à augmenter la complexité de la modélisation sur un type de donnée.

Dans tous les cas, il est préférable que la combinaison de noyaux conserve les conditions de Mercer.

Combinaison de valeurs de noyaux

Concernant la combinaison de valeurs de noyaux, on dispose des résultats suivants [Schölkopf et Smola, 2002] :

Théorème 5.
La combinaison linéaire positive de plusieurs noyaux de Mercer est un noyau de Mercer.
$k(\mathrm{x},\mathrm{y}) = \sum_i \alpha_i k_i(\mathrm{x},\mathrm{y})\ , \qquad \alpha_i > 0$
Le produit de noyaux de Mercer est un noyau de Mercer.
$k(\mathrm{x},\mathrm{y}) = \prod_i k_i(\mathrm{x},\mathrm{y})$

Le choix des paramètres de combinaison optimaux (*e.g.* coefficients α_i d'une combinaison linéaire) peut se faire selon plusieurs méthodes, comme le *boosting* [Crammer et al., 2003] ou l'optimisation de critères comme des mesures d'alignement [Cristianini et al., 2002, Kandola et al., 2002, Pothin et Richard, 2005] ou encore le "*class separability criterion*" [Wang et Luk Chan, 2002].

Combinaison de fonctions noyaux

Pour la combinaison des fonctions noyaux, on dispose du résultat suivant :

Théorème 6.
Si $\begin{cases} k(\mathrm{x},\mathrm{y}) \text{ est un noyau de Mercer, et} \\ k'(\mathbf{x},\mathbf{y}) = l\left[\ \mathbf{x}^{\mathrm{T}}\mathbf{x}\ ,\ \mathbf{x}^{\mathrm{T}}\mathbf{y}\ ,\ \mathbf{y}^{\mathrm{T}}\mathbf{y}\ \right] \text{ est un noyau de Mercer vectoriel (§3.2)} \end{cases}$
alors $l\left[\ k(\mathrm{x},\mathrm{x})\ ,\ k(\mathrm{x},\mathrm{y})\ ,\ k(\mathrm{y},\mathrm{y})\ \right]$ *est un noyau de Mercer.*

Preuve :
En effet, en considérant des noyaux vérifiant la propriété de Mercer :

$$\text{Si } \begin{cases} k(\mathrm{x},\mathrm{y}) = \boldsymbol{\phi}(\mathrm{x})^{\mathrm{T}}\boldsymbol{\phi}(\mathrm{y}) \ , & \boldsymbol{\phi} : \mathbb{X} \to \mathbb{R}^D \\ k'(\mathbf{x},\mathbf{y}) = \boldsymbol{\phi}'(\mathbf{x})^{\mathrm{T}}\boldsymbol{\phi}'(\mathbf{y}) \ , & \boldsymbol{\phi}' : \mathbb{R}^D \to \mathbb{R}^{D'} \end{cases}$$

alors on peut considérer le produit scalaire des compositions $\big(\boldsymbol{\phi}' \circ \boldsymbol{\phi}(\mathrm{x})\big)^{\mathrm{T}} \big(\boldsymbol{\phi}' \circ \boldsymbol{\phi}(\mathrm{y})\big)$

■

Le théorème 6 implique aussi que l'on puisse créer à partir d'un noyau une infinité de noyaux résultant de combinaisons polynomiales, résultat qui se généralise dans la formulation suivante [Tan et Wang, 2004] :

Théorème 7.
La combinaison linéaire des puissances d'un noyau de Mercer est un noyau de Mercer.
$k(\mathrm{x},\mathrm{y}) = \sum_{i=0}^{\infty} \alpha_i k_1(\mathrm{x},\mathrm{y})^i \ , \qquad \alpha_i > 0$

Ce type de combinaison suggérée par le théorème 6 est aussi couramment employé pour combiner un noyau Gaussien avec un noyau quelconque, selon :

$$\begin{array}{lcccl} \textit{Noyau} & & \textit{Distance} & & \textit{Noyau Radial} \\ k(\mathrm{x},\mathrm{y}) & \mapsto & \sqrt{\begin{array}{r} k(\mathrm{x},\mathrm{x}) - 2k(\mathrm{x},\mathrm{y}) \\ +k(\mathrm{y},\mathrm{y}) \end{array}} & \xrightarrow{\gamma>0} & k(\mathrm{x},\mathrm{y}) = e^{-\gamma\left(\ k(\mathrm{x},\mathrm{x}) - 2k(\mathrm{x},\mathrm{y}) \ + k(\mathrm{y},\mathrm{y})\ \right)} \end{array}$$

Cette astuce est connue sous le nom de "*Exponential Embedding*" [Seeger, 2002a]. Elle peut être étendue pour une distance quelconque (dans les cas où l'on connaît la distance mais pas le produit scalaire correspondant), selon le passage :

$$\begin{array}{ccc} \textit{Distance} & & \textit{Noyau Radial} \\ d(\mathrm{x},\mathrm{y}) & \xrightarrow{\gamma>0} & k(\mathrm{x},\mathrm{y}) = e^{-\gamma d(\mathrm{x},\mathrm{y})^2} \end{array} \tag{3.10}$$

Le théorème suivant [Berg et al., 1984, théorème 3.2.2] que l'on doit aux fondements théoriques de [Schoenberg, 1938], nous donne la condition pour que le noyau défini par (3.10) vérifie les conditions de Mercer.

Théorème 8.
La fonction $(\mathrm{x},\mathrm{y}) \mapsto \exp(-\gamma\delta(\mathrm{x},\mathrm{y}))$ *est défini positif pour tout* $\gamma > 0$ *si et seulement si la fonction* $\delta : \mathbb{X}\times\mathbb{X} \to \mathbb{R}$ *est définie négative.*

La distance euclidienne et la distance euclidienne au carré sont des exemples de fonctions définies négatives. Une matrice de taille $N\times N$ correspondant à une distance définie négative a (N-1) valeurs propres négatives. Nous renvoyons le lecteur à [Camastra, 2004] pour plus de détails sur les conditions de défini-négativité. Le théorème 8 reste valable lorsque l'on remplace $\delta \mapsto \exp(-\gamma\delta)$ par une fonction *absolument décroissante* (§3.2, théorème 9).

3.2 Noyaux entre vecteurs

Les noyaux "projectifs" sont ceux qui peuvent s'exprimer en formulant toutes les opérations qui impliquent les variables d'entrée $\mathbf{x}$ et $\mathbf{y}$ sous forme du produit scalaire $\mathbf{x}^{\mathrm{T}}\mathbf{y}$. Les noyaux "radiaux" (aussi appelés "noyaux métriques" [Chan et al., 2004]) sont ceux qui s'expriment comme une fonction monotone des distances entre vecteurs $\|\mathbf{x}-\mathbf{y}\|$. Nous rappelons que la distance euclidienne associée au produit scalaire est donnée par :

$$\|\mathbf{x}-\mathbf{y}\|^2 = \mathbf{x}^{\mathrm{T}}\mathbf{x} - 2\mathbf{x}^{\mathrm{T}}\mathbf{y} + \mathbf{y}^{\mathrm{T}}\mathbf{y} \tag{3.11}$$

La condition pour qu'un noyau radial soit défini positif est donnée par le théorème de [Schoenberg, 1938], qui a été étendu par [Micchelli, 1986b] avec la formulation :

Théorème 9 (Schoenberg).
Une fonction $f : \mathbb{R}^+ \to \mathbb{R}$ est dite "absolument décroissante" si elle est indéfiniment dérivable, et si

$$\forall\ (n,t) \in \mathbb{N}\times\mathbb{R}^+\ ,\ \begin{cases} f^{(n)}(t) > 0 \text{ si } n \text{ est pair} \\ f^{(n)}(t) < 0 \text{ si } n \text{ est impair} \end{cases}$$

Si f est absolument décroissante, alors $k(\mathbf{x},\mathbf{y}) = f\left(\|\mathbf{x}-\mathbf{y}\|^2\right)$ est un noyau de Mercer.

Par essence, les noyaux projectifs et les noyaux radiaux sont invariants aux rotations. Par contre, seuls les noyaux radiaux sont invariants aux translations, et aucun de ces noyaux n'est invariant aux homothéties. Enfin, notons que tous les noyaux cités dans cette partie vérifient les conditions de Mercer.

3.2.1 Noyaux projectifs

Noyaux linéaires

Le produit scalaire est le noyau le plus basique. Il peut être généralisé en incorporant une matrice de transformation linéaire des données dans l'espace de départ. On rencontre dans la littérature le terme de "noyau linéaire généralisé" [Hatch et Stolcke, 2006]. L'expression mathématique est de la forme :

$$k^{\mathrm{lin}}(\mathbf{x},\mathbf{y}) = \mathbf{x}^{\mathrm{T}}\ \mathbf{R}\ \mathbf{y} \tag{3.12}$$

où $\mathbf{R}$ est une matrice définie positive, comme par exemple l'inverse d'une matrice de covariance. Une telle normalisation réduit les corrélations entre variables d'entrée et rend les variances unitaires (pour les nouvelles variables normalisés $\mathbf{R}^{1/2}\mathbf{x}$). Elle permet d'introduire une invariance aux transformations linéaires, pour une meilleure stabilité des méthodes à noyau (§1.3.4).

Noyaux polynomiaux

L'expression générale des noyaux polynomiaux de degré p positif[11] est donnée par :

$$k^p(\mathbf{x},\mathbf{y}) = (\delta + \nu\ \mathbf{x}^{\mathrm{T}}\mathbf{y})^p \qquad \text{avec} \quad \delta \geq 0\ ,\ \nu > 0 \tag{3.13}$$

Le noyau polynomial peut s'écrire comme un produit scalaire dans un *Feature Space* de dimension $D = \frac{(d+p)!}{d!\ p!}$ si $\delta \neq 0$, ou $D = \frac{(d+p-1)!}{(d-1)!\ p!}$ si $\delta = 0$ [Burges, 1998]. En notant $\mathbf{x} = \begin{bmatrix} x_1 \cdots x_d \end{bmatrix}^{\mathrm{T}}$, il peut en effet s'écrire

$$k^p(\mathbf{x},\mathbf{y}) = \sum_{u=1}^{D} \phi_u(\mathbf{x})\phi_u(\mathbf{y})$$

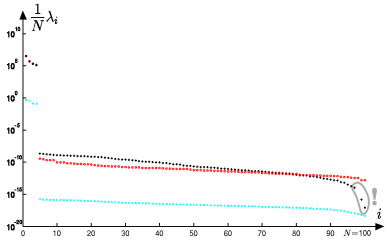
où les composantes $\phi_u(\cdot)$ sont de la forme

$$\phi_u(\mathbf{x}) = \delta^{\frac{p-q}{2}} \nu^{\frac{q}{2}} \sqrt{\tbinom{p}{q}\ \mathrm{C}^q_{\{q_i\}}} \left({x_1}^{q_1}\ {x_2}^{q_2}\ \cdots\ {x_d}^{q_d} \right) \tag{3.14}$$

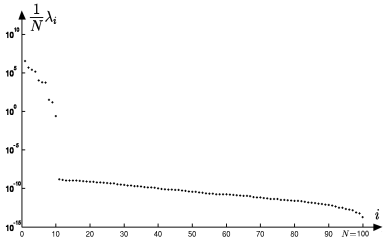
$$\text{avec} \left\{ \begin{array}{rclcrcl} q & = & \sum_{i=1}^{d} q_i \leq p & (q_i \geq 0) & & & \\ \tbinom{p}{q} & = & \frac{p!}{q!(p-q)!} & & \mathrm{C}^q_{\{q_i\}} & = & \frac{q!}{q_1!q_2!\cdots q_d!} \end{array} \right.$$

Si $\delta = 0$, la projection $\boldsymbol{\Phi}$ n'encode que les monomes $({x_1}^{q_1} x_2{}^{q_2} \cdots {x_d}^{q_d})$ de degré $q = p$, c'est-à-dire pour lesquels $\sum_{i=1}^{d} q_i = p$ et $\phi_u(\mathbf{x}) = \nu^{\frac{p}{2}} \sqrt{\mathrm{C}^p_{\{q_i\}}} \left({x_1}^{q_1} x_2{}^{q_2} \cdots {x_d}^{q_d} \right)$.

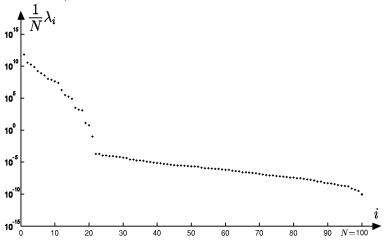
Autrement, plus δ est élevé, et plus le noyau polynomial donne de l'importance aux monomes de degrés faibles. En règle générale, prendre un δ non nul et/ou augmenter le degré p revient à augmenter la dimension du Feature Space et ainsi la complexité du noyau. L'accroissement de



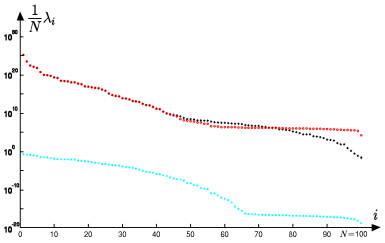
(a) Noyau polynomial $(\mathbf{x}^{\mathrm{T}}\mathbf{y})^3$ standard (noir), et normalisé avec | le centrage (3.6) (rouge), la projection sphérique (3.7) (courbe du bas).



(b) Noyau polynomial $(1 + \mathbf{x}^{\mathrm{T}}\mathbf{y})^3$.



(c) Noyau polynomial $(1 + \mathbf{x}^{\mathrm{T}}\mathbf{y})^5$.



(d) Noyau polynomial $(1 + \mathbf{x}^{\mathrm{T}}\mathbf{y})^{10}$. Mêmes courbes que dans (a).

Fig. 3.2 - Spectre de la matrice de Gram : noyaux polynomiaux.

[11] On peut aussi prendre $p < 0$: on parle alors de "Noyau de Hardy" (et on doit avoir $\delta, \nu < 0$ pour que le noyau vérifie les conditions de Mercer).

la complexité est visible sur Fig.3.2, qui représente les allures des spectres (§3.1.3) de quelques noyaux polynomiaux pour des données bidimensionnelles générées artificiellement.

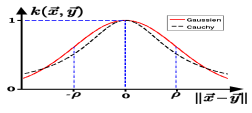
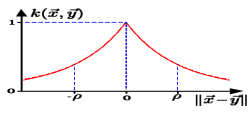
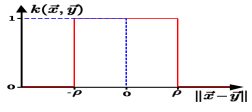
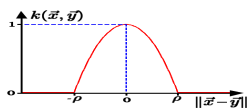
Les noyaux polynomiaux, comme les noyaux projectifs en général, peuvent gagner à utiliser une normalisation sphérique (3.7). En effet, à moins que les données d'entrée soient comprises dans un intervalle borné du type $[-1,1]^d$ les valeurs $k^p(\mathbf{x},\mathbf{x})$ peuvent varier dans une gamme de valeurs extrêmement large. Ceci pose problème dans les méthodes à noyau : la matrice de Gram a un spectre trop large en amplitude, et on dit qu'elle est "mal conditionnée" (*ill-conditioned*). Les effets des normalisations de noyau sur le spectre de la matrice de Gram sont illustrés dans Fig.3.2.

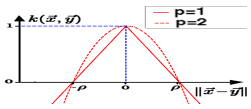
3.2.2 Noyaux radiaux

Les noyaux radiaux correspondent à des fonctions représentantes $k(\mathrm{x}_0,\cdot)$ qui sont des fonctions centrées en x_0, avec un maximal local en ce centre. La plupart des noyaux radiaux sont bornés dans $[0,1]$, avec une valeur nulle ou pratiquement nulle atteinte lorsque deux points sont suffisamment éloignés (dans ce cas les noyaux radiaux non bornés tendent vers $-\infty$). La distance marquant la zone d'influence de chaque vecteur x_0 est paramétrée par un facteur positif ρ (homogène à une distance entre vecteurs). Le tableau 3.15 donne une liste de noyau radiaux. Le noyau Gaussien est le noyau radial de loin le plus populaire. Il est désigné par abus de langage par le sigle RBF (*Radial Basis Function*) et est souvent paramétré dans la littérature par $\gamma = \frac{1}{2\rho^2}$.

Les noyaux radiaux correspondent à un *Feature Space* de dimension $D = +\infty$, et permettent aux SVMs de construire des frontières particulièrement complexes. Notons aussi que ces noyaux ne nécessitent pas de normalisation sphérique étant donné qu'ils vérifient déjà $k(\mathrm{x},\mathrm{y}) = cte$.

Tab. 3.15 - Noyaux Radiaux : Expressions et allures

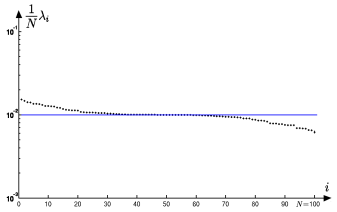
Noyau Gaussien Noyau de Cauchy	$k(\mathbf{x},\mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\rho^2}}$ $k(\mathbf{x},\mathbf{y}) = \frac{1}{1+\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\rho^2}}$	
Noyau de Laplace	$k(\mathbf{x},\mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|}{\rho}}$	
Noyau radial uniforme (binaire)	$k(\mathbf{x},\mathbf{y}) = \mathbf{1}_{\|\mathbf{x}-\mathbf{y}\|\leq\rho}$	
Noyau d'Epanechnikov	$k(\mathbf{x},\mathbf{y}) = \left\|\rho^2 - \|\mathbf{x}-\mathbf{y}\|^2\right\|_+$	
.../...	.../...	.../...

.../...	.../...	.../...
Noyau triangulaire [Berg et al., 1984]	$k(\mathbf{x},\mathbf{y}) = 1 - \left(\frac{\|\mathbf{x}-\mathbf{y}\|}{\rho}\right)^p \ , p \leq 2$	

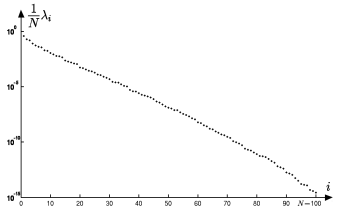
Lorsque la zone d'influence induite par le noyau diminue (c'est-à-dire que ρ diminue), les SVMs à noyaux radiaux ont un comportement proche des *k-plus proches voisins* (§1.2.4) si la zone d'influence est petite (par rapport à l'étalement des données). De manière générale, la complexité des modèles obtenus avec des noyaux radiaux croît lorsque ρ diminue. Dans Fig.3.3, on peut vérifier que plus ρ est élevé, plus le spectre décroît rapidement (§3.1.3). Les valeurs extrêmes $\rho \to \infty$ et $\rho \to 0$ correspondent respectivement à un noyau constant et à un noyau de Dirac (3.5), et la valeur idéale est un compromis entre les deux. [Schölkopf et al., 1999] recommande de prendre ρ de l'ordre de

$$\rho \approx \rho_0 = \sqrt{d\overline{\sigma}} \tag{3.15}$$

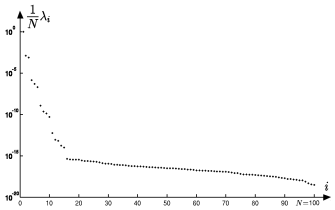
où d est la dimension des vecteurs d'entrées, et $\overline{\sigma} = \frac{1}{d}\sum_{u=1}^{d}\left(\mathrm{E}[x_u^2] - \mathrm{E}[x_u]^2\right)$ est la moyenne quadratique des écarts-types de chaque composante vectorielle. Si toutes les variables d'entrée x_u ont à peu près la même importance pour le problème de classification, alors une des meilleures stratégies est de normaliser les données avec un tenseur de Mahalanobis, de manière à avoir une variance unitaire sur chaque composante, et de prendre $\rho = \sqrt{d}$.



(a) Valeur faible $\rho = \rho_0/10$. La droite bleue représente la courbe théorique du noyau de Dirac (3.5).



(b) Valeur recommandé $\rho = \rho_0$ (3.15). Le spectre décroît exponentiellement.



(c) Valeur élevée $\rho = 30\rho_0$. Le noyau est peu complexe. (problèmes numériques possibles)

Fig. 3.3 - Spectre de la matrice de Gram : noyaux Gaussiens.

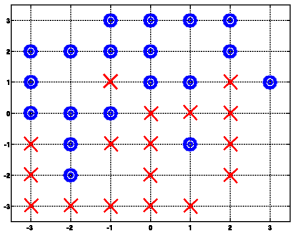
3.2.3 Forme des modèles SVM

Dans les pages qui suivent, nous illustrons les types de frontière que construisent les classifieurs SVM à noyau vectoriel sur un cas d'école bidimensionnel représenté en Fig.3.4(a). Les valeurs des fonctions discriminantes apprises par minimisation du risque régularisé sont représentées par les intensités de couleur. La frontière de décision est représentée par une courbe noire.

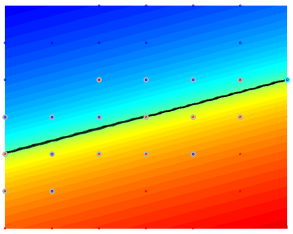
Fig.3.4 montre l'allure des frontières de décisions obtenues par apprentissage SVM avec les noyaux polynomiaux. On peut constater que même si augmenter le degré du polynôme permet de complexifier la modélisation, le comportement de la frontière devient très instable à partir d'un certain degré. Non seulement des effets de bord apparaissent, mais en plus les fonctions

discriminantes atteignent des valeurs très faibles autour des vecteurs d'apprentissage. La prise de décision souffre de l'imprécision numérique apportée par ce second phénomène (mauvaise localisation de la frontière).

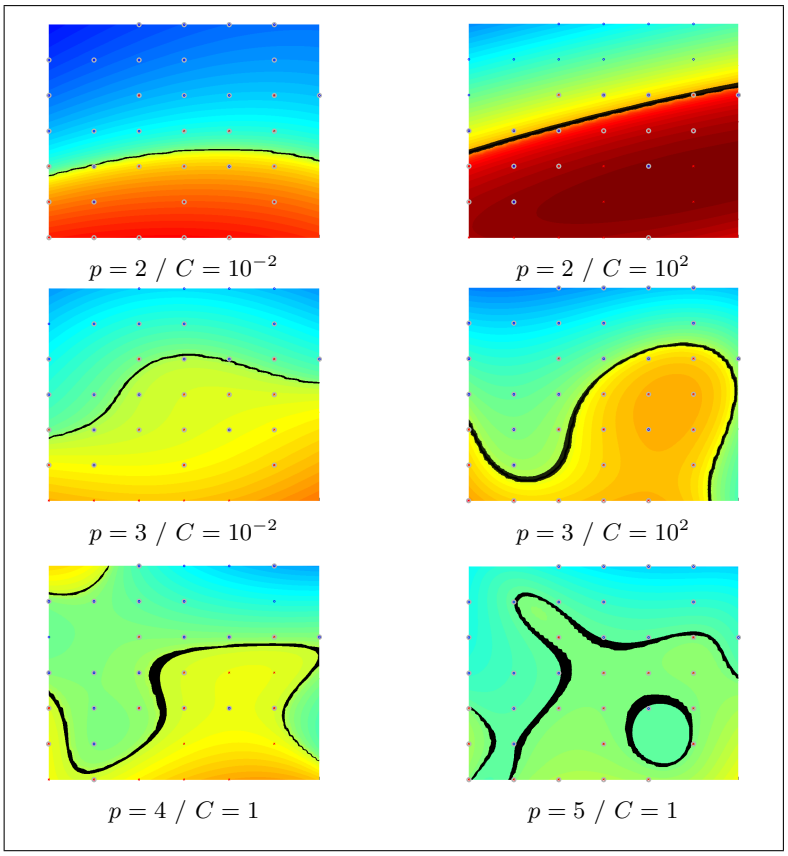
Fig.3.5 représente l'allure des solutions d'un SVM obtenus avec les noyaux RBF Gaussiens. L'effet du sur-apprentissage par les noyaux radiaux lorsque la zone d'influence ρ est trop faible est visible dans Fig.3.5(b) (en bas à droite). Lorsqu'au contraire ρ est élevé, la frontière de décision devient quasiment linéaire (même figure, en bas à gauche). Fig.3.5(a) illustre comment la fonction discriminante est construite à partir du noyau et des vecteurs de support.



(a) Disposition des vecteurs 2D d'entrée pour les SVMs.

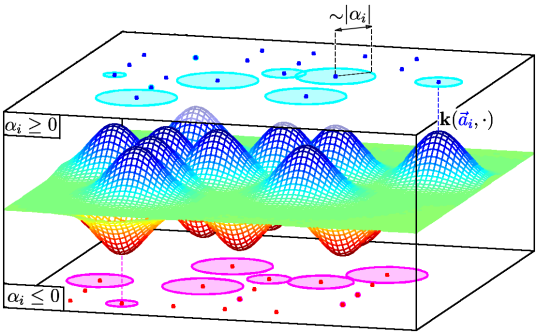


(b) Noyau linéaire $k(\mathbf{x},\mathbf{y}) = \mathbf{x}^\top\mathbf{y}$, appris avec $C = 1$.



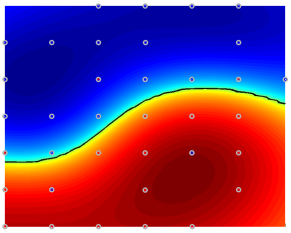
(c) Noyaux polynomiaux $k^p(\mathbf{x},\mathbf{y}) = (1 + \mathbf{x}^\top\mathbf{y})^p$

Fig. 3.4 - Allure des modèles SVM vectoriels : noyaux projectifs.

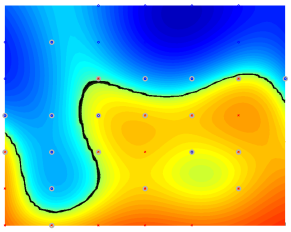


(a) Fonctions "représentantes" $k(\mathbf{a}_i, \cdot)$ et illustration des poids de Lagrange β_i associés, correspondant aux Vecteurs de Supports, avec $\rho = \rho_0$ / $C = 10^2$

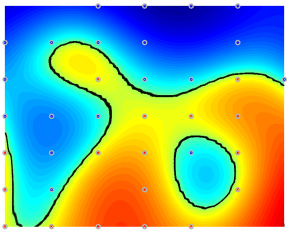
$\downarrow$



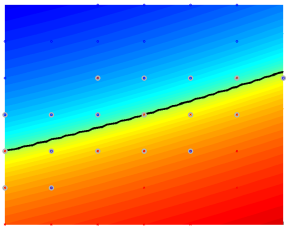
$\rho = \rho_0$ / $C = 1$



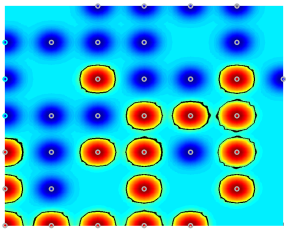
$\rho = \rho_0$ / $C = 10^2$



$\rho = \rho_0$ / $C = 10^4$



$\rho = 100\rho_0$ / $C = 10^6$



$\rho = \frac{\rho_0}{10}$ / $C = 10^{-6}$

Même avec un C élevé, la frontière est pratiquement linéaire (manque de complexité pour capturer la partition).

Même avec un C faible, la frontière est trop complexe (sur-apprentissage). Les vecteurs positifs (bleus) étant majoritaires, ils l'emportent sur les zones d'incertitude.

(b) Modèles SVM appris avec un noyau Gaussien $k(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\rho^2}}$

Fig. 3.5 - Allure des modèles SVM vectoriels : noyaux Gaussiens.

3.3 Noyaux entre densités de probabilité

Les noyaux entre densités de probabilité permettent d'étendre les algorithmes comme les SVMs pour traiter d'autres types d'entrée que les vecteurs de taille fixe. Ces fonctions s'appliquent à des distributions de données et permettent en pratique de manipuler des ensembles de tailles variables, pour divers types d'applications :

1. *Ensembles finis de données symboliques*
   Application directe à des histogrammes de couleurs pour la classification d'image [Chapelle et al., 1999], ou de mots pour l'indexation de textes [Lafferty et Lebanon, 2004].
2. *Ensembles finis de données numériques*
   Dans ce cas, on entraîne des modèles paramétriques pour caractériser ces ensembles. Dans le cas de la vérification du locuteur en mode *indépendant du texte* et d'autres applications en indexation d'images et de vidéos, il s'agit de séquences de tailles variables dont l'ordre n'importe pas pour le problème de classification [Kondor et Jebara, 2003, Jebara et Kondor, 2003, Moreno et Ho, 2003a, Bredin et al., 2006].
3. *Ensembles infinis*
   Du moment que l'on choisit une mesure (qui peut être la mesure uniforme), on peut définir des noyaux entre ensembles topologiques quelconques.

3.3.1 Noyaux de produit de probabilités

Les "noyaux de produit de probabilités" (*Probability Product Kernel*) sont définis selon [Jebara et Kondor, 2003] par la forme :

$$\kappa_p^{\text{pp}}(p_1, p_2) = \int_{\mathbb{X}} p_1(\mathrm{x})^p p_2(\mathrm{x})^p \,\mathrm{dx} \tag{3.16}$$

où p est un degré strictement positif. Parmi ces noyaux, les plus communément utilisés sont

– le noyau de corrélation ("*correlation kernel*" aussi connu sous le nom de "*expected likelihood kernel*") pour $p = 1$:

$$\kappa_1^{\text{pp}}(p_1, p_2) = \int_{\mathbb{X}} p_1(\mathrm{x}) p_2(\mathrm{x}) \,\mathrm{dx} \tag{3.17}$$

– le noyau de Bhattacharyya pour $p = 1/2$:

$$\kappa_{\text{B}}^{\text{pp}}(p_1, p_2) = \int_{\mathbb{X}} \sqrt{p_1(\mathrm{x}) p_2(\mathrm{x})} \,\mathrm{dx} \tag{3.18}$$

Ce noyau est considéré comme le produit scalaire de référence pour les fonctions positives, parmi lesquelles les densités de probabilités sont toutes de norme unitaire.

Régularité

Tout comme les noyaux vectoriels polynomiaux, le noyau de corrélation ($p = 1$) prend des valeurs dans un large intervalle. La grande variabilité des valeurs de noyau induit des matrices de Gram souvent mal conditionnées [Chan et al., 2004]. Par conséquent, les normalisations sphériques (§3.1.4) ont tendance à améliorer la robustesse d'un classifieur SVM basé sur de tels noyaux, comme nous le constaterons dans nos expériences de vérification du locuteur (§5.4.1).

En fait, en faisant varier p, on peut en quelque sorte lisser le noyau et remédier à ce problème de conditionnement de la matrice de Gram. Dans les cas extrêmes, lorsque $p \to 0$ alors la matrice de Gram se rapproche de la matrice pleine de 1, et lorsque $p \to +\infty$ elle se rapproche de la matrice identité.

Calcul

Pour les distributions exponentielles[12], ces noyaux admettent une expression analytique simple [Jebara et al., 2004] et aucune technique d'approximation d'intégrale n'est requise. À titre d'exemple, si l'on considère deux distributions Gaussiennes d-dimensionnelles ayant même covariance isotropique $\mathbf{\Sigma} = \sigma^2 \mathbf{I}_d$, l'expression d'un noyau de produit de probabilités est relativement simple :

$$\begin{gathered} \kappa_p^{\text{pp}}\left(\mathcal{N}_{\boldsymbol{\mu}_1,\sigma^2} \| \mathcal{N}_{\boldsymbol{\mu}_2,\sigma^2}\right) = k_{\text{o}} \times e^{-\frac{\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|^2}{4\sigma^2/p}} \\ \text{où } k_{\text{o}} = (2p)^{-d/2}(2\pi\sigma^2)^{(1-2p)d/2} \end{gathered} \tag{3.19}$$

Pour les GMMs, l'expression analytique des noyaux de produit de probabilités dans le cas où $p = 1$ est donnée dans [Lyu, 2005]. Elle sera adaptée pour la vérification du locuteur aux cas des GMMs adaptés d'un même modèle (§5.4.1).

3.3.2 Noyaux à partir de divergences entre distributions

Un grand nombre de mesures sont disponibles pour mesurer l'écart entre deux distributions. La table 3.16 liste les principales. À partir de ces mesures de divergences, il est possible de concevoir des noyaux, de la même manière que les noyaux radiaux sont construits à partir d'une distance (§3.2.2). Typiquement, l'analogue d'un noyau vectoriel Gaussien avec l'astuce de l'*Exponential Embedding* (§3.1.5 équation 3.10) :

$$\begin{array}{ccc} \textit{Divergence} & & \textit{Noyau} \\ \mathcal{D}(p_1, p_2) & \overset{\gamma>0}{\longmapsto} & \kappa(p_1, p_2) = e^{-\gamma \mathcal{D}(p_1,p_2)^2} \end{array} \tag{3.20}$$

Les quatre premières mesures listées dans Tab.3.16 sont nommées "divergences" parce qu'elle ne correspondent pas à des fonctions définies négatives, contrairement à la distance euclidienne. Ce n'est pas le cas des autres mesures de distances entre distributions, pour lesquelles le noyau construit à partir de l'*Exponential Embedding* (3.20) vérifie les conditions de Mercer (§3.1.5, théorème 8).

.../...

[12] Une famille de distributions est dite exponentielle si ses membres peuvent s'exprimer sous la forme $p(\mathrm{x}) = \exp(\mathcal{A}(x) + \boldsymbol{\theta}^{\mathrm{T}}\mathcal{T}(x) - \mathcal{K}(\boldsymbol{\theta}))$. Parmi les distributions exponentielles, on compte les distributions Gaussiennes, et, pour les variables aléatoires discrètes, les distributions de Poisson et Bernouilli.

Tab. 3.16 - Distances Probabilistes : Expressions dans le cas général

Divergence de Kullback [Cover et Thomas, 1991] (ou entropie relative)	$\mathcal{D}^{\mathrm{KL}}(p_1 \Vert p_2) = \int_{\mathbb{X}} p_1(\mathrm{x}) \log \frac{p_1(\mathrm{x})}{p_2(\mathrm{x})} \,\mathrm{dx}$ (3.21)
Divergence de Kullback symétrique	$\widetilde{\mathcal{D}}^{\mathrm{KL}}(p_1, p_2) = \mathcal{D}^{\mathrm{KL}}(p_1 \Vert p_2) + \mathcal{D}^{\mathrm{KL}}(p_2 \Vert p_1) = \int_{\mathbb{X}} [p_1(\mathrm{x}) - p_2(\mathrm{x})] \log \frac{p_1(\mathrm{x})}{p_2(\mathrm{x})} \,\mathrm{dx}$ (3.22)
Divergence de Jensen-Shannon [Lin, 1991]	$\mathcal{D}^{\mathrm{JS}}(p_1 \Vert p_2) = H[\alpha_1 p_1 + \alpha_2 p_2] - \alpha_1 H[p_1] - \alpha_2 H[p_2]$ avec $H[p] = \int_{\mathbb{X}} p(\mathrm{x}) \log p(\mathrm{x}) \,\mathrm{dx}$
Divergence de [Rényi, 1960]	$\mathcal{D}_p^{\mathrm{R}}(p_1 \Vert p_2) = \frac{1}{p-1} \log \int_{\mathbb{X}} p_1(\mathrm{x})^p p_2(\mathrm{x})^{1-p} \,\mathrm{dx}$
Distance de [Bhattacharyya, 1943]	$\mathcal{D}^{\mathrm{B}}(p_1, p_2) = -\log \int_{\mathbb{X}} \sqrt{p_1(\mathrm{x}) p_2(\mathrm{x})} \,\mathrm{dx}$ (3.23)
Distance de [Chernoff, 1952], généralisation (*) de (3.23)	$\mathcal{D}^{\mathrm{C}}(p_1, p_2) = -\log \int_{\mathbb{X}} [p_1(\mathrm{x})]^{\alpha_1} [p_2(\mathrm{x})]^{\alpha_2} \,\mathrm{dx}$
Distance de Hellinger (ou distance de [Matusita, 1955])	$\mathcal{D}^{\mathrm{H}}(p_1, p_2) = \sqrt{\int_{\mathbb{X}} \left[\sqrt{p_1(\mathrm{x})} - \sqrt{p_2(\mathrm{x})}\right]^2 \,\mathrm{dx}}$
Distance de Patrick-Fisher [Patrick et Fisher, 1969] (**)	$\mathcal{D}^{\mathrm{PF}}(p_1, p_2) = \sqrt{\int_{\mathbb{X}} [\pi_1 p_1(\mathrm{x}) - \pi_2 p_2(\mathrm{x})]^2 \,\mathrm{dx}}$
Distance de Kolmogorov [Adhikara et Joshi, 1956] (**)	$\mathcal{D}^{\mathrm{K}}(p_1, p_2) = \int_{\mathbb{X}} \lvert \pi_1 p_1(\mathrm{x}) - \pi_2 p_2(\mathrm{x}) \rvert \,\mathrm{dx}$ (3.24)
Distance de Lissack-Fu [Lissack et Fu, 1976], généralisation (*) de (3.24) (**)	$\mathcal{D}^{\mathrm{LF}}(p_1, p_2) = \int_{\mathbb{X}} \lvert \pi_1 p_1(\mathrm{x}) - \pi_2 p_2(\mathrm{x}) \rvert^{\alpha_1} [\pi_1 p_1(\mathrm{x}) + \pi_2 p_2(\mathrm{x})]^{\alpha_2} \,\mathrm{dx}$ (3.25)
(*) $0 < \alpha_1, \alpha 2 < 1$ et $\alpha_1 + \alpha 2 = 1$	
(**) $0 < \pi_1, \pi_2 < 1$ et $\pi_1 + \pi_2 = 1$. Les paramètres π_1 et π_2 représentent des probabilités *a priori*.	

Notons que certaines distances citées dans la table ci-dessus sont liées, comme la divergence de Jensen-Shannon, qui estime dans quelle mesure deux ensembles de réalisations sont issus de la même source de distribution, peut s'exprimer en fonction de la divergence de Kullback, pour $\alpha_1 = \alpha_2 = 1/2$:

$$\mathcal{D}^{\mathrm{JS}}(p_1 \| p_2) = \tfrac{1}{2}\left(\mathcal{D}^{\mathrm{KL}}(p_1 \| p) + \mathcal{D}^{\mathrm{KL}}(p_2 \| p)\right) \quad \text{où } p = \tfrac{1}{2}\left(p_1 + p_2\right)$$

Aussi la distance de Hellinger et la distance de Bhattacharyya sont liées par la relation :

$$\mathcal{D}^{\mathrm{H}}(p_1, p_2) = \sqrt{2\left(1 - e^{-\mathcal{D}^{\mathrm{B}}(p_1,p_2)}\right)}$$

Dans le cas général, le calcul des distances probabilistes n'est pas trivial, et requiert un algorithme d'approximation d'intégrale comme l'algorithme de Monte Carlo [Moreno et Ho, 2003b]. Toutefois elles ont des expressions analytiques simples pour les familles de distributions exponentielles. Les expressions des principales distances entre deux modèles Gaussiens sont listées dans Tab.3.17. Dans le cas particulier où p_1 et p_2 sont des distributions Gaussiennes de même covariance $\mathbf{\Sigma}$, alors la divergence de Kullback symétrique (et la distance de Bhattacharyya modulo un facteur multiplicatif) se réduit à une distance de Mahalanobis au carré entre les vecteurs moyennes [Mahalanobis, 1936] :

$$\begin{aligned} \text{si} \quad & \mathcal{N}_i(\mathbf{x}) = \tfrac{1}{\sqrt{(2\pi)^d \det \mathbf{\Sigma}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^{\mathrm{T}} \mathbf{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu}_i)} \\ \text{alors} \quad & \widetilde{\mathcal{D}}^{\mathrm{KL}}(\mathcal{N}_1, \mathcal{N}_2) = 8\ \mathcal{D}^{\mathrm{B}}(p_1, p_2) = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^{\mathrm{T}}\, \mathbf{\Sigma}^{-1}\, (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \end{aligned} \tag{3.26}$$

Tab. 3.17 - Distances Probabilistes : Expressions analytiques dans le cas Gaussien

Divergence de Kullback symétrique	$\widetilde{\mathcal{D}}^{\mathrm{KL}}(\mathcal{N}_{\boldsymbol{\mu}_1,\mathbf{\Sigma}_1}, \mathcal{N}_{\boldsymbol{\mu}_2,\mathbf{\Sigma}_2}) = \begin{array}{l} \frac{1}{2}\left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right)^{\mathrm{T}} \left(\mathbf{\Sigma}_1^{-1} + \mathbf{\Sigma}_2^{-1}\right) \left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right) \\ + \frac{1}{2}\mathrm{tr}\left(\mathbf{\Sigma}_1 \mathbf{\Sigma}_2^{-1} + \mathbf{\Sigma}_2 \mathbf{\Sigma}_1^{-1} - 2\mathbf{I}_d\right) \end{array}$
Distance de Bhattacharyya	$\mathcal{D}^{\mathrm{B}}(\mathcal{N}_{\boldsymbol{\mu}_1,\mathbf{\Sigma}_1}, \mathcal{N}_{\boldsymbol{\mu}_2,\mathbf{\Sigma}_2}) = \begin{array}{l} \frac{1}{4}\left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right)^{\mathrm{T}} \left(\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2\right)^{-1} \left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right) \\ + \frac{1}{2}\log \frac{\det(\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2)}{2\sqrt{\det \mathbf{\Sigma}_1 \mathbf{\Sigma}_2}} \end{array}$
Distance de Chernoff	$\mathcal{D}^{\mathrm{C}}(\mathcal{N}_{\boldsymbol{\mu}_1,\mathbf{\Sigma}_1}, \mathcal{N}_{\boldsymbol{\mu}_2,\mathbf{\Sigma}_2}) = \begin{array}{l} \frac{\alpha_1\alpha_2}{2}\left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right)^{\mathrm{T}} \left({}_{\alpha_1}\mathbf{\Sigma}_1 + {}_{\alpha_2}\mathbf{\Sigma}_2\right)^{-1} \left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right) \\ + \frac{1}{2}\log \frac{\det(\alpha_1\mathbf{\Sigma}_1 + \alpha_2\mathbf{\Sigma}_2)}{(\det \mathbf{\Sigma}_1)^{\alpha_1} (\det \mathbf{\Sigma}_2)^{\alpha_2}} \end{array}$
Distance de P.-Fisher	$\mathcal{D}^{\mathrm{PF}}(\mathcal{N}_{\boldsymbol{\mu}_1,\mathbf{\Sigma}_1}, \mathcal{N}_{\boldsymbol{\mu}_2,\mathbf{\Sigma}_2}) = \begin{array}{l} \frac{1}{2\sqrt{(2\pi)^d}}\left((\det \mathbf{\Sigma}_1)^{-1/2} + (\det \mathbf{\Sigma}_2)^{-1/2}\right) \\ - \frac{2}{\sqrt{(2\pi)^d \det(\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2)}} e^{-(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^{\mathrm{T}} (\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2)^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)} \end{array}$

3.3.3 Noyaux dérivés de métriques Hilbertiennes

[Hein et Bousquet, 2004] étendent les métriques Hilbertiennes semi-homogènes aux densités de probabilités. Ces métriques Hilbertiennes sont basées sur des distances définies négatives pour les réels positifs, paramétrées par $\alpha \in [1, +\infty]$ et $\beta \in [-\infty, -1] \cup [\frac{1}{2}, \alpha]$ selon la formule :

$$\forall \{\mathrm{x}, \mathrm{y}\} \in (\mathbb{R}^+) , \qquad d^2_{\alpha,\beta}(\mathrm{x}, \mathrm{y}) = \frac{2^{\frac{1}{\beta}}(\mathrm{x}^\alpha + \mathrm{y}^\alpha)^{\frac{1}{\alpha}} - 2^{\frac{1}{\alpha}}(\mathrm{x}^\beta + \mathrm{y}^\beta)^{\frac{1}{\beta}}}{2^{\frac{1}{\alpha}} - 2^{\frac{1}{\beta}}} \tag{3.27}$$

Pour les densités de probabilités, à valeurs réelles positives, cette notion est généralisée pour donner la distance au carré entre densités :

$$\mathcal{D}^2_{\alpha,\beta}(p_1, p_2) = \int_{\mathbb{X}} d^2_{\alpha,\beta}\left(p_1(\mathrm{x}), p_2(\mathrm{y})\right) \, \mathrm{dx} \tag{3.28}$$

On peut alors utiliser cette nouvelle mesure de distance avec un *Exponential Embedding* comme dans la sous-section précédente (§3.3.2). On peut aussi concevoir des noyaux de type "projectif", en remarquant que pour certains choix de *alpha* et β, on peut déterminer une expression analytique du produit scalaire correspondant à la distance Hilbertienne. Le tableau 3.18 montre des exemples simples. On peut y voir que le noyau de corrélation (§3.3.1) est un cas particulier pour $\alpha = 1/2$ et $\beta = 1$.

Tab. 3.18 - Distances Hilbertiennes entre densités de probabilité et produits scalaires correspondant

(α, β)	$\mathcal{D}^2_{\alpha,\beta}(p_1, p_2)$ $= \kappa_{\alpha,\beta}(p_1,p_1) - 2\kappa_{\alpha,\beta}(p_1,p_2) + \kappa_{\alpha,\beta}(p_2,p_2)$	$\kappa_{\alpha,\beta}(p_1, p_2)$
$= (1, -1)$	$\propto \int_{\mathbb{X}} \frac{\left(p_1(\mathrm{x}) - p_2(\mathrm{x})\right)^2}{p_1(\mathrm{x}) + p_2(\mathrm{x})} \, \mathrm{dx}$	$\propto \int_{\mathbb{X}} \frac{p_1(\mathrm{x}) p_2(\mathrm{x})}{p_1(\mathrm{x}) + p_2(\mathrm{x})} \, \mathrm{dx}$
$= (\frac{1}{2}, 1)$	$\propto \int_{\mathbb{X}} \left(\sqrt{p_1(\mathrm{x})} - \sqrt{p_2(\mathrm{x})}\right)^2 \, \mathrm{dx}$	$\propto \int_{\mathbb{X}} \sqrt{p_1(\mathrm{x}) p_2(\mathrm{x})} \, \mathrm{dx}$
$= (+\infty, 1)$	$\propto \int_{\mathbb{X}} \lvert p_1(\mathrm{x}) - p_2(\mathrm{x}) \rvert \, \mathrm{dx}$	$\propto \int_{\mathbb{X}} \min\left[p_1(\mathrm{x}), p_2(\mathrm{x})\right] \, \mathrm{dx}$
$\rightarrow (1, 1^-)$	$\propto \int_{\mathbb{X}} p_1(\mathrm{x}) \log\left[\frac{2p_1(\mathrm{x})}{p_1(\mathrm{x}) + p_2(\mathrm{x})}\right] + p_2(\mathrm{x}) \log\left[\frac{2p_2(\mathrm{x})}{p_1(\mathrm{x}) + p_2(\mathrm{x})}\right] \mathrm{dx}$	$\propto -\int_{\mathbb{X}} p_1(\mathrm{x}) \log\left[\frac{p_1(\mathrm{x})}{p_1(\mathrm{x}) + p_2(\mathrm{x})}\right] + p_2(\mathrm{x}) \log\left[\frac{p_2(\mathrm{x})}{p_1(\mathrm{x}) + p_2(\mathrm{x})}\right] \mathrm{dx}$

De manière générale, on dispose aussi du résultat suivant [Hein et Bousquet, 2004].

.../...

Théorème 10.
Si k est un noyau *p.s.d.* 1-homogène sur $\mathbb{R}^+$, c'est-à-dire qu'il vérifie :

$$\forall\, c \in \mathbb{R}^+ \,,\, \forall\, (\mathrm{x}, \mathrm{y}) \in (\mathbb{R}^+)^2 \,, \qquad k(c\mathrm{x}, c\mathrm{y}) = ck(\mathrm{x}, \mathrm{y})$$

Alors *la fonction* κ *défini par :*

$$\kappa(p_1, p_2) = \int_{\mathbb{X}} k\left(p_1(\mathrm{x}), p_2(\mathrm{x})\right)\, \mathrm{dx} \tag{3.29}$$

est un noyau de Mercer.

Malheureusement, les noyaux dérivés de métriques Hilbertiennes ne sont à l'heure actuelle que des outils théoriques. Aucun travaux n'a encore porté sur l'application de ces noyaux à des modèles complexes de distribution de probabilité.

3.4 Noyaux d'Information Mutuelle

Les noyaux d'Information Mutuelle (MI) [Seeger, 2002a] sont construits à partir d'une distribution *a priori* des données. Pour la classification, ils s'appliquent donc dans un contexte d'apprentissage semi-supervisé, où l'on dispose d'une base de données étiquetées et d'une base de données $\mathcal{B}$ non étiquetée mêlant toutes les classes ("données du Monde"). Les noyaux MI peuvent s'appliquer à n'importe quel type de données, du moment où l'on sait définir une mesure de probabilité sur ce type de données (car ces noyaux sont construits à partir de densités de probabilité).

L'idée sous-jacente aux noyaux MI est de construire un noyau adapté à la distribution des données, c'est-à-dire une mesure de similarité qui tienne compte de la façon dont les données sont réparties dans l'espace d'entrée. La meilleure manière de tenir compte de la distribution des données du monde est de synthétiser l'information par des modèles probabilistes paramétriques, profitant ainsi de leur capacité de généralisation (robustesse aux observations aberrantes, interpolation dans les zones où l'information disponible en apprentissage est incomplète). Dans le cas où les entrées sont des séquences de taille variable, nous verrons (§3.4.3) que le noyau MI, sous certaines hypothèses et avec quelques approximations, est équivalent à une forme exponentielle du noyau de Fisher conçu par [Jaakkola et Haussler, 1998].

3.4.1 Expression générale

Considérons une famille de modèles probabilistes $p(\cdot|\theta)$ paramétrés par θ. Étant donnée une distribution sur les paramètres (*mediator distribution*) $P_{med}(\theta)$, on considère la quantité :

$$Q(\mathrm{x}, \mathrm{y}) = \int P_{med}(\theta) p(\mathrm{x}|\theta) p(\mathrm{y}|\theta)\, d\theta \tag{3.30}$$

Alors l'information mutuelle (*Mutual Information score*) entre x et y est définie par

$$MI(\mathrm{x},\mathrm{y}) = \log \frac{Q(\mathrm{x},\mathrm{y})}{\int_{\mathbb{X}} Q(\mathrm{x},\mathrm{z})\,\mathrm{dz} \int_{\mathbb{X}} Q(\mathrm{y},\mathrm{z})\,\mathrm{dz}} \tag{3.31}$$

Concrètement, ce score mesure la similarité entre les échantillons x et y vis-à-vis du processus génératif modélisé par la distribution $P_{med}(\theta)$: c'est la quantité d'information qu'ils partagent via la variable "médiatrice" θ de loi $P_{med}(\theta)$. Afin de concevoir un noyau de Mercer à partir du score d'Information Mutuelle MI, qui n'est pas défini positif, l'astuce du *Exponential Embedding* (§3.1.5 équation 3.10) est appliquée sur la distance formée à partir de MI :

$$\begin{aligned} k^{\mathrm{MI}}(\mathrm{x},\mathrm{y}) &= e^{-\frac{1}{2}(MI(\mathrm{x},\mathrm{x})-2MI(\mathrm{x},\mathrm{y})+MI(\mathrm{y},\mathrm{y}))} \\ &= \frac{Q(\mathrm{x},\mathrm{y})}{\sqrt{Q(\mathrm{x},\mathrm{x})Q(\mathrm{y},\mathrm{y})}} \end{aligned} \tag{3.32}$$

C'est l'expression générale d'un noyau MI. [Seeger, 2002b] montre qu'un tel noyau vérifie bien les conditions de Mercer. Toute la problématique réside alors dans :

- le choix de $P_{med}(\theta)$;
- l'estimation des intégrales mises en jeu, qui pour des modèles $p(\cdot|\theta)$ non exponentiels, ne peut pas se faire de manière analytique exacte.

Choix classiques

En ce qui concerne le choix de la distribution des paramètres du modèle, [Seeger, 2002a] préconise la technique générale du *Model-Trust Scaling*, qui consiste à prendre

$$P^{\alpha}_{med}(\theta) = p(\mathcal{B}|\theta)^{\alpha} P(\theta)\ , \qquad \alpha \in [0,1] \tag{3.33}$$

où $\mathcal{B}$ représente les données d'apprentissage non étiquetées. Les cas extrêmes $\alpha = 0$ et $\alpha = 1$ correspondent respectivement à l'*a priori* $P(\theta)$ et à la probabilité a posteriori $P(\theta|\mathcal{B})$. Dans le premier cas, l'information du monde n'est pas exploitée (ce qui présente peu d'intérêt). Dans le second cas, si l'on ne donne aucune restriction aux paramètres θ, on risque le sur-apprentissage (noyau trop complexe, "collant" trop aux données du monde). Le paramètre α sert donc à doser l'influence de l'information contenue dans $\mathcal{B}$.

Usuellement, la puissance α est prise proportionnelle à l'inverse du nombre de données d'apprentissage N dans $\mathcal{B} = \{\mathrm{b}_i\}_{i=1\cdots N}$. En effet, si les données d'apprentissage sont supposées être des réalisations indépendantes, alors la probabilité jointe est décroissante exponentiellement en N :

$$p(\mathcal{B}|\theta) = \prod_{i=1}^{N} p(\mathrm{b}_i|\theta) = O(\mathrm{p}^N)\ , \qquad \text{avec } \mathrm{p} < 1$$

ce qui justifie que l'on prenne

$$\alpha \propto \tfrac{1}{N} \tag{3.34}$$

Cas particuliers

Dans le cas d'entrées vectorielles $\mathbf{x} \in \mathbb{R}^d$, considérons que la densité de probabilité sur ces entrées est une loi normale de covariance diagonale et d'écart-type ρ :

$$p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\theta}, \rho^2 \mathbf{I}_d)\ , \qquad \boldsymbol{\theta} \in \mathbb{R}^d \text{ (moyenne)} \tag{3.35}$$

Si l'on suppose de plus que $P(\boldsymbol{\theta})$ est constante, alors on peut déduire la probabilité *a posteriori* résultante

$$P^{\lambda}_{med}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\overline{\mathrm{b}}, \rho^2 \frac{\lambda}{N}\mathbf{I}_d) \ , \qquad \lambda \in [0, N] \tag{3.36}$$

Sous ces hypothèses, on peut montrer que le noyau MI résultant devient le noyau RBF Gaussien [Seeger, 2002b][13] :

$$\begin{aligned} k^{\mathrm{MI}}(\mathbf{x}, \mathbf{y}) &= e^{-\frac{1}{2}\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\rho_0^2}} \\ \text{avec} \qquad \rho_0 &= \rho\sqrt{2 + \tfrac{N}{\lambda}} \end{aligned} \tag{3.37}$$

3.4.2 Cas des mélanges de modèles

Considérons un mélange de modèle

$$p(\mathrm{x}|\theta, \pi) = \sum_g \pi_g p(\mathrm{x}|\boldsymbol{\theta}_g) \tag{3.38}$$

Alors l'information mutuelle peut être généralisée à partir de l'expression

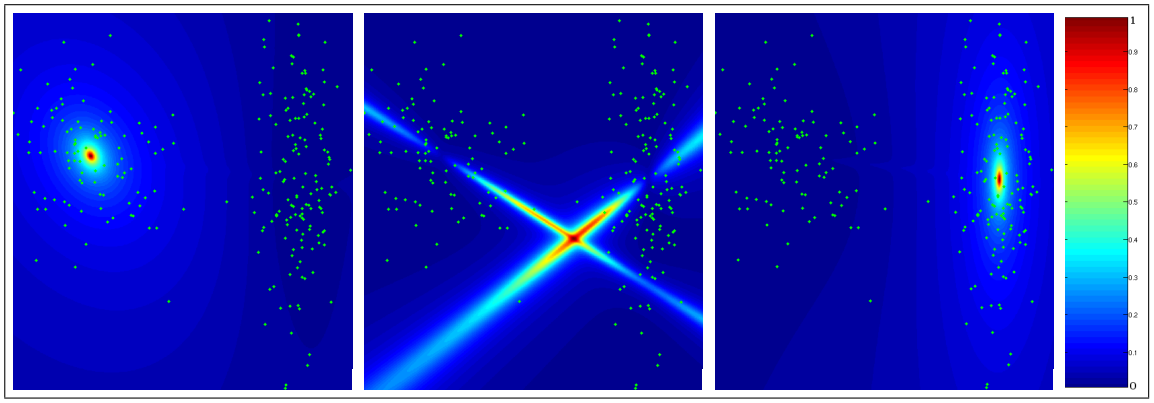
$$Q(\mathrm{x}, \mathrm{y}) = \sum_{g,h} \omega_{g,h} \int P^{\alpha}_{med}(\boldsymbol{\theta}) p(\mathrm{x}|\boldsymbol{\theta}_g) p(\mathrm{y}|\boldsymbol{\theta}_h) \, \mathrm{d}\boldsymbol{\theta} \tag{3.39}$$

où $\Omega = (\omega_{g,h})$ est une matrice symétrique à valeurs positives et strictement positives sur la diagonale. Si $\omega_{g,h} = \pi_g \pi_h$, alors on retombe sur l'expression exacte du noyau MI calculé à partir de $p(\cdot|\theta, \pi)$.

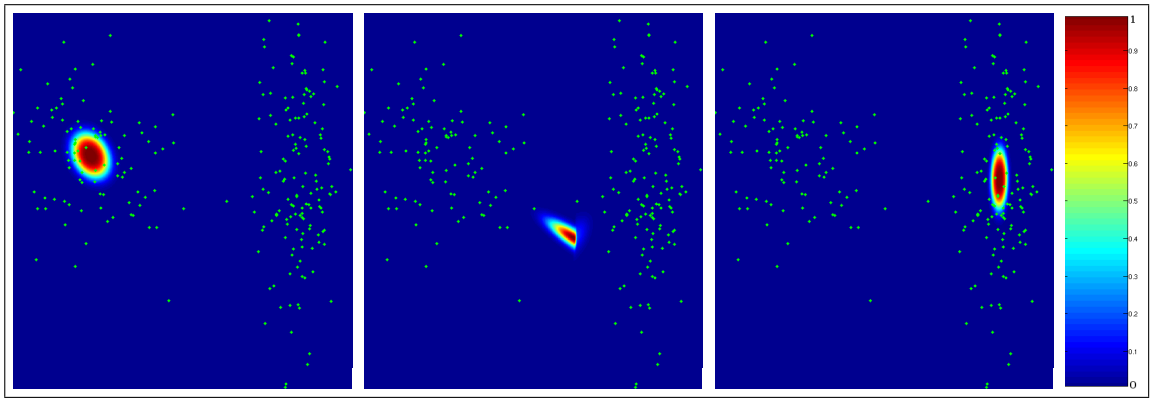
L'intérêt de cette généralisation est en fait de prendre Ω diagonale, c'est-à-dire de prendre $\omega_{g,g} = \pi_g^2$ et $\omega_{g,h} = 0$ pour $g \neq h$. Le noyau MI résultant est alors une combinaison linéaire normalisée des noyaux MI sur chaque composante du modèle. Dans le cas des GMMs, ceci permet d'obtenir des expressions analytiques simples à calculer tout en encodant la *cluster hypothesis* (§1.2.1) à savoir : pour α variant dans un intervalle raisonnable, $k^{\mathrm{MI}}(\mathrm{x}, \mathrm{y})$ est relativement petit si x et y appartiennent à différents *clusters* (modes), et inversement. Ce phénomène est visible dans Fig.3.6 qui représente les valeurs de $k(\mathrm{x_o}, \cdot)$ pour un noyau MI k correspondant à une modélisation bi-Gaussienne réglée sur deux amas de vecteurs 2-D générés artificiellement. Cette mesure de similarité dépend de la position relative de $\mathrm{x_o}$ par rapport aux deux *clusters* (gauche/droite) que modélisent les Gaussiennes.

[13] En fait le résultat indiqué ici est différent de celui affiché par [Seeger, 2002b], ce dernier n'étant d'ailleurs pas homogène. Pour permettre au lecteur de vérifier l'exactitude de la formule proposée, nous donnons l'expression intermédiaire de l'Information Mutuelle à une constante multiplicative près :

$$\begin{gathered} Q(\mathrm{x}, \mathrm{y}) \propto \exp\left[-\tfrac{1}{2\rho^2}\left(\mathrm{x}^2 + \mathrm{y}^2 + \tfrac{N}{\lambda}\overline{\mathrm{b}}^2 - \left(2 + \tfrac{N}{\lambda}\right) f(\mathrm{x}, \mathrm{y})^2\right)\right] \\ \text{avec } f(\mathrm{x}, \mathrm{y}) = \tfrac{\mathrm{x}+\mathrm{y}+\frac{N}{\lambda}\overline{\mathrm{b}}}{2+\frac{N}{\lambda}} \end{gathered}$$



(a) Valeurs du noyau $\mathbf{x} \mapsto k^{\mathrm{MI}}(\mathbf{x}_0, \mathbf{x})$ avec un mélange de 2 modèles Gaussiens estimés à partir de données du monde bidimensionnelles (verts), sur différents vecteurs $\mathbf{x}_0$ (centroïdes rouges) et $\boxed{\alpha \to 0}$



(b) Même chose avec $\boxed{\alpha \to 1}$

Fig. 3.6 - Noyau d'Information Mutuelle : Allure générale dans le cas d'un GMM à 2 Gaussiennes, pour trois vecteurs.

3.4.3 Noyau de Fisher

Définition

Considérons un modèle $p(\cdot|\boldsymbol{\theta}_{\mathrm{o}})$ appris sur les données du mondes avec un critère *MAP* :

$$\boldsymbol{\theta}_{\mathrm{o}} = \arg\max_{\boldsymbol{\theta}} p(\mathcal{B}|\boldsymbol{\theta}) \tag{3.40}$$

Le noyau de Fisher [Jaakkola et Haussler, 1998] entre deux séquences s'écrit alors

$$\kappa^{\mathrm{Fsh}}(\mathcal{X}, \mathcal{Y}) = \boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \mathcal{X})^{\mathrm{T}}\ \mathbf{I}^{\mathrm{Fsh}\,-1}\ \boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \mathcal{Y}) \tag{3.41}$$

$$\text{où} \left\{ \begin{array}{rcl} \boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \mathcal{X}) & = & \boldsymbol{\nabla}_{\boldsymbol{\theta}} \log p(\mathcal{X}|\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\mathrm{o}}} \\ \\ \mathbf{I}^{\mathrm{Fsh}} & = & \mathrm{E}\left[\ \boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \mathcal{X})\ \boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \mathcal{X})^{\mathrm{T}}\ \right] \\ & & = \int \boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \mathcal{X}) \boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \mathcal{X})^{\mathrm{T}} p(\mathcal{X}|\boldsymbol{\theta}_{\mathrm{o}})\, \mathrm{d}\mathcal{X} \end{array} \right. \tag{3.42}$$

– L'*expansion* $\boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \mathcal{X})$ représente le gradient de la log-vraisemblance $p(\mathcal{X}|\boldsymbol{\theta})$ par rapport aux paramètres du modèle et au jeu optimal $\boldsymbol{\theta}_{\mathrm{o}}$. Il décrit comment les paramètres $\boldsymbol{\theta}_{\mathrm{o}}$ contribuent à la génération de la séquence X. Il est souvent désigné par "*Fisher mapping*" ou "*Fisher score*" (et le *Feature Space* par "*Fisher score-space*" [Smith et Gales, 2002]).
– La matrice $\mathbf{I}^{\mathrm{Fsh}}$ encode les seconds moments des *expansions* $\boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \mathcal{X})$. Il s'agit de la "matrice d'Information de Fisher".

Si l'on considère que les données du monde $\mathcal{B} = \{\mathrm{b}_i\}_{i=1\cdots N}$ sont des observations indépendantes et générées par une même variable aléatoire issue d'une distribution $p(\cdot|\boldsymbol{\theta}_{\mathrm{o}})$, alors on peut estimer la matrice d'Information de Fisher sur le monde en considérant les échantillons comme des séquences ("*sample approximation*") :

$$\mathbf{I}^{\mathrm{Fsh}*} = \tfrac{1}{N} \sum_{i=1}^{N} \boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \{\mathrm{b}_i\})\ \boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \{\mathrm{b}_i\})^{\mathrm{T}} \tag{3.43}$$

Si de plus on considère que les éléments $\{\mathrm{x}_t\}_{t=1\cdots T}$ de $\mathcal{X}$ sont issus d'observations *i.i.d*, alors $\boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \mathcal{X}) = \sum_{t=1}^{T} \log p\left(\mathrm{x}_t|\boldsymbol{\theta}\right)$, et on prend en pratique

$$\boldsymbol{\delta}^*(\boldsymbol{\theta}_{\mathrm{o}}, \{\underbrace{\mathrm{x}_1, \cdots, \mathrm{x}_T}_{i.i.d.}\}) = \tfrac{1}{T} \sum_{t=1}^{T} \boldsymbol{\nabla}_{\boldsymbol{\theta}} \log p\left(\mathrm{x}_t|\boldsymbol{\theta}\right)|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\mathrm{o}}} \tag{3.44}$$

La normalisation par la longueur de la séquence T [Smith et Gales, 2002] sert à rendre le noyau "invariant" à la longueur. Le facteur $\frac{1}{T}$ n'a pas de signification bayésienne, et est rajouté de manière empirique.

L'expression de l'*expansion* de Fisher pour les GMMs sera donnée lorsque nous décrirons la mise en œuvre de la méthode pour la vérification du locuteur (§5.4.3). L'expression pour les HMMs est donnée par [Smith et al., 2001].

Soulignons qu'en pratique, l'estimation de la matrice de normalisation $\mathbf{I}^{\mathrm{Fsh}}$ (ou $\mathbf{I}^{\mathrm{Fsh}*}$) peut être laborieuse lorsque le nombre M de paramètres du modèle est trop élevé[14]. Il est alors d'usage de faire omission de la matrice $\mathbf{I}^{\mathrm{Fsh}}$, en la remplacent par la matrice identité $\mathbf{I}_N$.

[Jaakkola et Haussler, 1998] donne une justification théorique du noyau de Fisher, et [Tsuda et al., 2004] en dit plus long sur la légitimité et la puissance théorique du noyau. Mais dans ce qui suit, nous pouvons voir que le noyau de Fisher est lié à une approximation du noyau d'Information Mutuelle.

Noyau de Fisher et Noyau MI

[Seeger, 2002b] montre que le noyau de Fisher peut s'écrire comme un score d'Information Mutuelle (§3.4.1), auquel on a appliqué plusieurs hypothèses simplificatrices, qui rendent ce score nécessairement défini positif. Comme la démonstration de ce rapport technique n'a pas fait l'objet de publication, nous donnons ici les éléments qui permettent d'exprimer le score d'Information Mutelle (3.31) comme un noyau de Fisher.

[14] Dans ce cas, le nombre de données nécessaire à une estimation fiable de la matrice des moments de second ordre $\mathbf{I}^{\mathrm{Fsh}}$, est trop élevé (problème de collection de données et/ou de complexité calculatoire).

- L'expression de la distribution $P^{\alpha}_{med}(\boldsymbol{\theta})$ (3.33), est simplifiée par l'approximation *MAP* (voir [Kaas et Raftery, 1995] pour plus de détails) qui, comme l'approximation de Laplace, revient à approximer une densité de probabilité par une Gaussienne centrée en le maximum de cette probabilité. L'unique différence avec l'approximation de Laplace est la constante de normalisation de la fonction qui approxime la densité de probabilité. L'approximation *MAP* s'écrit en considérant le maximum local $\boldsymbol{\theta}_{\mathrm{o}}$ défini en (3.40) :

$$\begin{aligned} P(\boldsymbol{\theta}|\mathcal{B}) &= \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\theta}_{\mathrm{o}}, \mathbf{H}^{-1}) \\ \Rightarrow P^{\alpha}_{med}(\boldsymbol{\theta}) &= \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\theta}_{\mathrm{o}}, \mathbf{H}^{-1})^{\alpha} p(\mathcal{B})^{\alpha} P(\boldsymbol{\theta})^{1-\alpha} \end{aligned} \tag{3.45}$$

où la covariance de la Gaussienne servant à l'approximation est la matrice Hessienne $\mathbf{H}$ de la probabilité jointe, dont les valeurs sont :

$$H_{u,v} = -\left.\frac{\partial^2}{\partial\theta_u \partial\theta_v} \log P(\boldsymbol{\theta}|\mathcal{B})\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\mathrm{o}}} = -\left.\frac{\partial^2}{\partial\theta_u \partial\theta_v} \log p(\mathcal{B}|\boldsymbol{\theta}) P(\boldsymbol{\theta})\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\mathrm{o}}} \tag{3.46}$$

- La probabilité $p(\mathrm{x}|\boldsymbol{\theta})$ est approchée par un développement limité du logarithme au premier ordre, par rapport à $\boldsymbol{\theta}$ (ce qui revient à supposer que quel que soit x, elle est linéaire en $\boldsymbol{\theta}$ dans un voisinage de $\boldsymbol{\theta}_{\mathrm{o}}$) :

$$\begin{aligned} \log p(\mathrm{x}|\boldsymbol{\theta}) &\approx \log p(\mathrm{x}|\boldsymbol{\theta}_{\mathrm{o}}) + \boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \{x\})^{\mathrm{T}}(\boldsymbol{\theta} - \boldsymbol{\theta}_{\mathrm{o}}) \\ \Rightarrow p(\mathrm{x}|\boldsymbol{\theta}) &\approx p(\mathrm{x}|\boldsymbol{\theta}_{\mathrm{o}}) e^{\boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \{x\})^{\mathrm{T}}(\boldsymbol{\theta} - \boldsymbol{\theta}_{\mathrm{o}})} \end{aligned} \tag{3.47}$$

où la dérivée première de l'approximation linéaire est le gradient

$$\boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \{x\}) = \boldsymbol{\nabla}_{\boldsymbol{\theta}} \log p(\mathrm{x}|\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\mathrm{o}}} \tag{3.48}$$

Cette approximation se généralise pour une séquence $\mathcal{X}$ d'observations *i.i.d.* :

$$p(\mathcal{X}|\boldsymbol{\theta}) \approx p(\mathcal{X}|\boldsymbol{\theta}_{\mathrm{o}}) e^{\boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \mathcal{X})^{\mathrm{T}}(\boldsymbol{\theta} - \boldsymbol{\theta}_{\mathrm{o}})} \tag{3.49}$$

- La probabilité *a priori* sur la distribution des paramètres est fixée

$$P(\boldsymbol{\theta}) = p_{\boldsymbol{\theta}} \tag{3.50}$$

Avec en plus l'hypothèse précédente (3.47) de linéarité pour $p(\mathrm{x}|\boldsymbol{\theta})$, on peut montrer que :

$$H_{u,v} = -\left.\frac{\partial^2}{\partial\boldsymbol{\theta}_u \partial\boldsymbol{\theta}_v} \log p(\mathcal{B}|\boldsymbol{\theta})\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\mathrm{o}}} = \boldsymbol{\delta}_u(\boldsymbol{\theta}_{\mathrm{o}}, \mathcal{B}) \boldsymbol{\delta}_v(\boldsymbol{\theta}_{\mathrm{o}}, \mathcal{B}) \tag{3.51}$$

On identifie ainsi

$$\mathbf{H} = {}_N\mathbf{I}^{\mathrm{Fsh}*} \tag{3.52}$$

Ces approximations permettent d'aboutir à l'expression analytique suivante pour l'Information Mutuelle[15] (mêmes notations que dans §3.4.1) :

$$\begin{aligned} Q(\mathcal{X}, \mathcal{Y}) &\approx cte(\mathcal{X}, \mathcal{Y})\ e^{\frac{1}{2}(\boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \mathcal{X}) + \boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \mathcal{Y}))^{\mathrm{T}}\ \left(\alpha N \mathbf{I}^{\mathrm{Fsh}*}\right)^{-1}\ (\boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \mathcal{X}) + \boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \mathcal{Y}))} \\ MI(\mathcal{X}, \mathcal{Y}) &\approx \boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \mathcal{X})^{\mathrm{T}}\ (\alpha N \mathbf{I}^{\mathrm{Fsh}*})^{-1}\ \boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \mathcal{Y}) \\ &= \kappa^{\mathrm{Fsh}}(\mathcal{X}, \mathcal{Y}) \quad \text{pour } \alpha = \tfrac{1}{N} \end{aligned} \tag{3.53}$$

[15] en prenant $cte(\mathcal{X}, \mathcal{Y}) = \frac{(2\pi)^{M\frac{1-\alpha}{2}}}{\sqrt{\alpha}\left(N \det \mathbf{I}^{\mathrm{Fsh}*}\right)^{\frac{1+\alpha}{2}}} p(\mathcal{B})^{\alpha} p_{\boldsymbol{\theta}}^{1-\alpha}\ p(\mathcal{X}|\boldsymbol{\theta}_{\mathrm{o}}) p(\mathcal{Y}|\boldsymbol{\theta}_{\mathrm{o}})$

On reconnaît ainsi dans cette approximation de l'Information Mutuelle le noyau de Fisher proposé par [Jaakkola et Haussler, 1998]. Cette expression garanti d'ailleurs que le noyau soit défini positif (il est écrit explicitement comme un produit scalaire dans un espace de dimension M fixe où sont projetées les séquences).

Finalement, le noyau MI correspondant aux hypothèses 3.45),(3.47) et (3.50) est l'*Exponential Embedding* du noyau de Fisher :

$$\kappa^{\mathrm{MI}}(\mathcal{X},\mathcal{Y}) \approx e^{-\frac{1}{2}\left(\kappa^{\mathrm{Fsh}}(\mathcal{X},\mathcal{X})-2\kappa^{\mathrm{Fsh}}(\mathcal{X},\mathcal{Y})+\kappa^{\mathrm{Fsh}}(\mathcal{Y},\mathcal{Y})\right)} \quad \text{pour } \alpha = \tfrac{1}{N} \tag{3.54}$$

3.4.4 Noyau TOP

Les noyaux TOP (*Tangent vector Of Posterior log-odds*) ont été introduits par [Tsuda et al., 2002], pour les problèmes de classification binaires (deux étiquettes possibles $\boldsymbol{\ell} = \pm 1$). Ils ont été utilisés à la même période par [Smith et Gales, 2002] en traitement de la parole sans être nommés en tant que tels. Les auteurs parlent de "*Log-Likelihood Ratio score-space*") et ne donnent pas de justification théorique véritable.

Les noyaux TOP sont construits de manière similaire au noyau de Fisher, à la différence près que les probabilités *a posteriori* sont prises en compte, au lieu des probabilités *a priori*. Au lieu d'un logarithme de densité $\log p(\mathcal{X}|\boldsymbol{\theta})$, on considère ici la différence des log-probabilités *a posteriori* relatives à chaque classe ("*log-odds of a probabilistic model*" [Devroye et al., 1996]) :

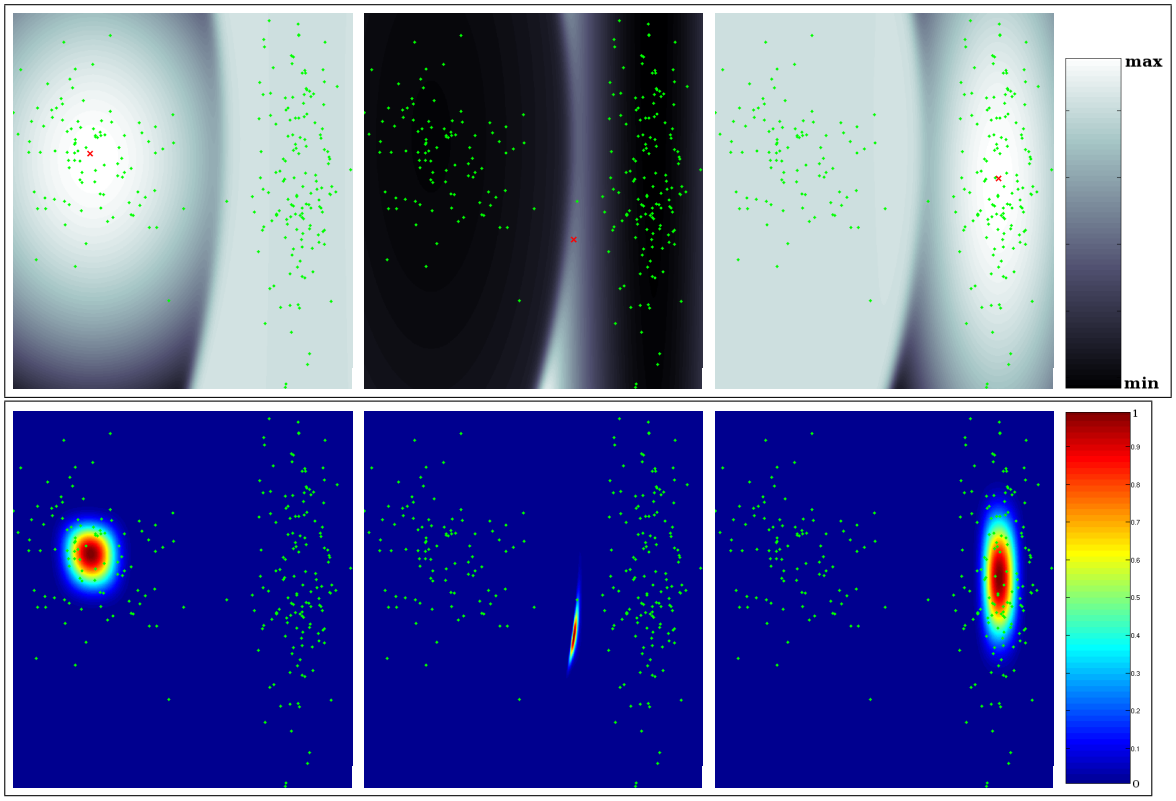
$$\boldsymbol{\nu}(\mathcal{X}|\boldsymbol{\theta}) = \log p(\boldsymbol{\ell}=+1|\mathcal{X},\boldsymbol{\theta}^{+1}) - \log p(\boldsymbol{\ell}=-1|\mathcal{X},\boldsymbol{\theta}^{-1}) \tag{3.55}$$

Alors que pour le noyau de Fisher les paramètres de la densité *a priori* sont estimés (par maximum de vraisemblance) sur des données $\{\mathrm{b}_i\}$ *non* étiquetées, le choix des paramètres $\boldsymbol{\theta}_{\mathrm{o}} = \left\{\boldsymbol{\theta}^{+1},\boldsymbol{\theta}^{-1}\right\}$ des densités *a posteriori* doit se faire sur un corpus étiqueté $\left\{\mathrm{b}_i,\boldsymbol{\ell}_i\right\}_{i=1\cdots N}$. Le noyau TOP est alors donné par le produit scalaire entre gradients des log-rapports de vraisemblance en $\boldsymbol{\theta}_{\mathrm{o}}$:

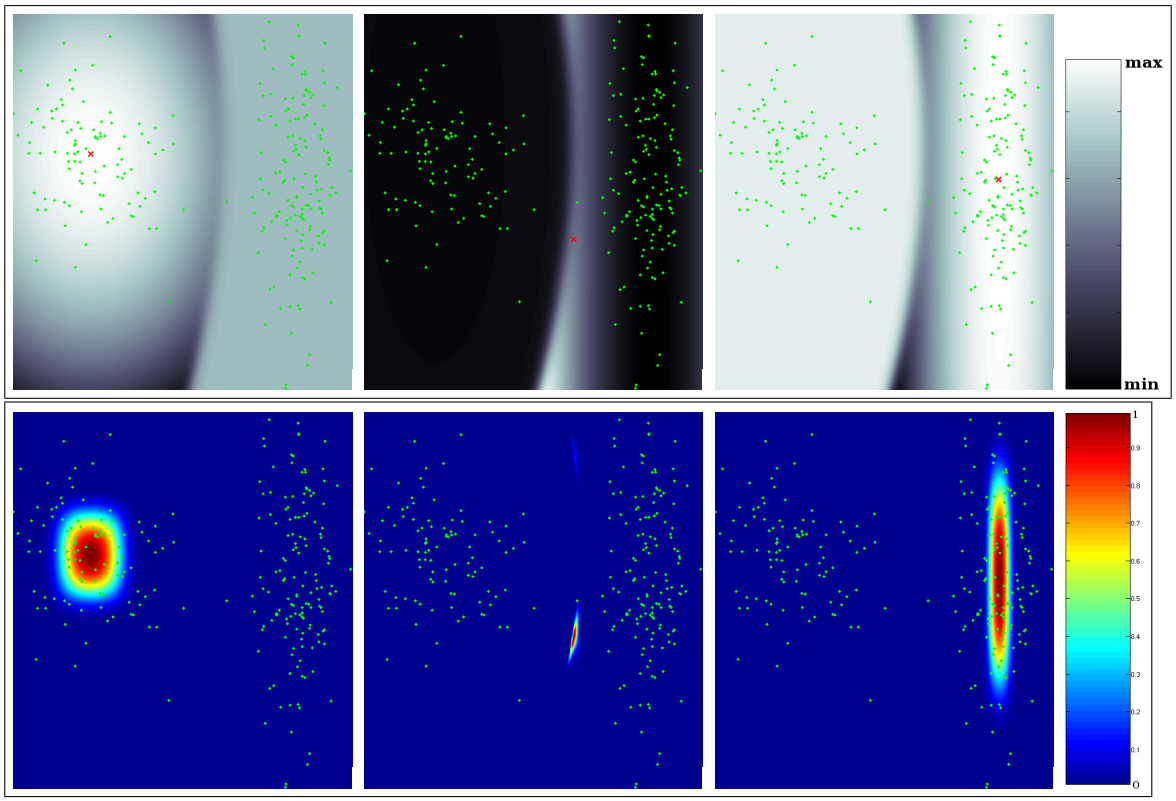
$$\kappa^{\mathrm{TOP}}(\mathcal{X},\mathcal{Y}) = \left(\boldsymbol{\nabla}_{\boldsymbol{\theta}}\boldsymbol{\nu}(\mathcal{X}|\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\mathrm{o}}}\right)^{\mathrm{T}} \left(\boldsymbol{\nabla}_{\boldsymbol{\theta}}\boldsymbol{\nu}(\mathcal{Y}|\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\mathrm{o}}}\right) \tag{3.56}$$

Étant donné certaines mesures de performance définies par [Tsuda et al., 2002], ce noyau a un meilleur taux de convergence (théorique) vers de faibles taux d'erreur. Les résultats expérimentaux montrent de meilleures performances pour le noyau TOP, en comparaison avec le noyau de Fisher. Une extension du noyau TOP au cas multi-classes (avec typiquement un nombre élevé de classes) est développée dans le travail de [Titov et Henderson, 2005], où le cadre probabiliste du noyau est adapté à la classification.

La limitation des noyaux TOP est d'ordre pratique : la construction du noyau nécessite des données de développement étiquetées, idéalement différentes de celles utilisées pour apprendre le modèle discriminant (*e.g.* exemples positifs et négatifs d'un SVM).



(a) Valeurs du noyau $\mathbf{x} \mapsto k^{\text{Fsh}}(\mathbf{x}_0, \mathbf{x})$ (haut)
et du noyau radial exponentiel $\mathbf{x} \mapsto \exp\left[-1/2k^{\text{Fsh}}(\mathbf{x}_0, \mathbf{x}_0) + k^{\text{Fsh}}(\mathbf{x}_0, \mathbf{x}) - 1/2k^{\text{Fsh}}(\mathbf{x}, \mathbf{x})\right]$ (bas)
avec un mélange de 2 modèles Gaussiens estimé à partir de données bidimensionnelles (points verts),
sur différents vecteurs $\mathbf{x}_0$ (croix rouges)



(b) Même chose en faisant omission de la matrice de normalisation du second ordre $\mathbf{I}^{\text{Fsh}}(= \mathbf{I}_N)$

Fig. 3.7 - Noyau de Fisher : Allure

3.5 Noyaux entre séquences de vecteurs pour la vérification du locuteur

Dans cette section, nous passons en revue les noyaux entre séquences de vecteurs, adéquats pour la classification de séquences. Nous nous intéressons d'abord aux noyaux entre "paquets de vecteurs", invariants aux permutations des vecteurs à l'intérieur des séquences (§3.5.1,§3.5.2). Une dernière partie sera consacrée aux noyaux entre séquences ordonnées (§3.5.3). Nous rappelons que l'importance que nous attachons aux noyaux entre paquets de vecteurs vient du cadre applicatif visé par notre étude. Pour la vérification automatique du locuteur en mode "indépendant du texte", les approches invariantes à l'ordre des vecteurs acoustiques dans les séquences, comme la modélisation GMM (§2.2), permettent de distinguer les locuteurs de manière robuste.

3.5.1 Combinaison de noyaux vectoriels

Une façon naturelle d'étendre un noyau vectoriel k à deux séquences $\mathcal{X} = \{\mathrm{x}_t\}_{t=1\cdots T_\mathrm{X}}$ et $\mathcal{Y} = \{\mathrm{y}_t\}_{t=1\cdots T_\mathrm{Y}}$ est de considérer une combinaison linéaire de noyaux entre éléments inter-séquences :

$$\kappa(\mathcal{X},\mathcal{Y}) = \sum_{t=1}^{T_\mathrm{X}} \sum_{s=1}^{T_\mathrm{Y}} \Omega_{t,s}^{(\mathrm{X},\mathrm{Y})} k(\mathrm{x}_t, \mathrm{y}_s) \tag{3.57}$$

Cette extension est valable pour tout type d'éléments constituant les ensembles. Toutefois, pour simplifier la lecture, nous parlerons de vecteurs pour désigner ces éléments, et de noyau vectoriel pour le noyau k qui manipule ces éléments.

Noyau linéaire entre *expansions* de séquences

Il est facile de constater que si le noyau vectoriel vérifie la condition de Mercer $k(\mathrm{x},\mathrm{y}) = \boldsymbol{\phi}(\mathrm{x})^\mathrm{T}\boldsymbol{\phi}(\mathrm{y})$, et si les poids peuvent se factoriser sous la forme $\Omega_{t,s}^{(\mathrm{X},\mathrm{Y})} = \omega_t^{(\mathrm{X})}\omega_s^{(\mathrm{Y})}$, alors le noyau de séquences κ vérifie à son tour les conditions de Mercer. Il peut s'écrire comme un produit scalaire entre *expansions* de séquences :

$$\kappa(\mathcal{X},\mathcal{Y}) = \underbrace{\sum_{t=1}^{T_\mathrm{X}} \omega_t^{(\mathrm{X})} \boldsymbol{\phi}(\mathrm{x}_t)^\mathrm{T}}_{\overline{\boldsymbol{\phi}}(\mathcal{X})^\mathrm{T}} \underbrace{\sum_{s=1}^{T_\mathrm{Y}} \omega_s^{(\mathrm{Y})} \boldsymbol{\phi}(\mathrm{y}_s)}_{\overline{\boldsymbol{\phi}}(\mathcal{Y})} \tag{3.58}$$

Un choix trivial est par exemple de prendre les poids fixes pour chaque séquence en appliquant une normalisation par la longueur des séquences : $\omega_t^{(\mathrm{X})} = 1/T_\mathrm{X}$. On aboutit alors au noyau moyenne :

$$\kappa(\mathcal{X},\mathcal{Y}) = \tfrac{1}{T_\mathrm{X}T_\mathrm{Y}} \sum_{t=1}^{T_\mathrm{X}} \sum_{s=1}^{T_\mathrm{Y}} k(\mathrm{x}_t, \mathrm{y}_s) \tag{3.59}$$

Le noyau GLDS conçu par [Campbell et al., 2006a], et ses généralisations par [Louradour et Daoudi, 2006] font partie de ces noyaux de Mercer. Nous y reviendrons plus en détail dans le chapitre 4, principale contribution de cette thèse.

Si l'on considère un noyau de Mercer symétrique correspondant à une *expansion* sous-jacente $\boldsymbol{\Phi}$ calculable (dimension finie), le tableau 3.20 liste les complexités calculatoires impliquées par la forme "*kernélisée*" (3.57) et par la forme (3.58) lorsqu'elles sont utilisées dans un classifieur SVM. Même si, de prime abord, le calcul du noyau entre deux séquences est plus complexe avec la seconde forme (produit scalaire entre *expansions* de séquences), Tab.3.20 montre que cette seconde forme est préférable sous tout point de vue pour traiter un problème réel de classification de séquences. En pratique, il est moins coûteux de calculer et de garder en mémoire les *expansions* (taille fixe) de séquences, plutôt que de garder en mémoire les séquences et de calculer les noyaux vectoriels inter-séquences. En particulier, la phase de test d'un SVM linéaire à partir d'*expansions* de séquences peut être rendue très efficace si les modèles sont compactés pour ne calculer qu'un seul produit scalaire lors de l'attribution de scores SVM, d'après :

$$f_{\boldsymbol{\theta}_{\text{loc}}}(\mathcal{X}) = \overline{\boldsymbol{\Phi}}(\mathcal{X})^{\text{T}} \underbrace{\left(\sum_i \alpha_i \overline{\boldsymbol{\Phi}}(\mathcal{A}_{\text{loc}}) - \sum_j \alpha_j \overline{\boldsymbol{\Phi}}(\mathcal{A}_{\text{imp}}) \right)}_{\boldsymbol{\theta}_{\text{loc}} \in \mathbb{R}^D} + \boldsymbol{\theta}_{\text{loc},0} \tag{3.60}$$

Tab. 3.20 - Complexité de calcul des noyaux de séquences par combinaison de noyaux vectoriels

	forme (3.57)		forme (3.58)	
Complexité du calcul du noyau entre 2 séquences	$O(T^2 d)$		$O(DTd)$	
	$T^2(d+1)$ opérations	$6{\times}10^8$	$2D(Td+1)$ opérations	$12{\times}10^8$
Apprentissage Calcul de la matrice de Gram	$O(S^2 T^2 d)$		$O(DSTd)$	
	$\frac{1}{2}S(S+1)T^2\,(d+1)$ opérations	$3{\times}10^{14}$	$DS(Td+\frac{1}{2}(S+1))$ opérations	$6{\times}10^{11}$
Compression des modèles (équation 3.60)	X		$O(n_{\text{o}} D)$	
		–	$2n_{\text{o}}D$ opérations	$5{\times}10^6$
Mémoire Taille des modèles SVM	$O(n_{\text{o}} Td)$		$O(D)$	
	$n_{\text{o}}\,(Td+1)+1$ réels	$6{\times}10^7$	$D+1$ réels	$5{\times}10^3$
Phase de test Calcul d'un score SVM	$O(n_{\text{o}} T^2 d)$		$O(DTd)$	
	$n_{\text{o}}\left(T^2(d+1)+1\right)+1$ opérations	$3{\times}10^{11}$	$D(Td+1)+1$ opérations	$6{\times}10^9$

Notations :
d : dimension d'entrée (et ordre de grandeur de la complexité de k)
D : taille de l'*expansion* $\boldsymbol{\Phi}$
T : taille des séquences (d'apprentissage / de test)
S : nombre de séquences d'apprentissage
n_{o} : nombre de "séquences de support" dans les SVMs
Valeurs prises pour les estimations (en bas à droite) **:**
$d = 25$; $D = T = 5000$; $S = 1000$; $n_{\text{o}} = 500$

Le fait de calculer des *expansions* de séquences présente un autre intérêt pratique : il permet d'appliquer des méthodes linéaires de sélection / construction de caractéristiques (*feature*

selection). Par exemple, une Analyse en Composantes Principales ou une Analyse Factorielle Discriminante permet de réduire la dimension D du problème. [Solomonoff et al., 2004] ont par exemple développé une méthode pour sélectionner les composantes de l'*expansion* qui ont le meilleur pouvoir discriminant pour la reconnaissance du locuteur (*i.e.* qui présentent une forte variabilité aux variations inter-locuteurs / une faible variabilité aux variations des conditions d'enregistrement) : la méthode de compensation de canal NAP. Cette technique consiste à calculer un noyau linéaire généralisé entre les maps $\overline{\mathbf{\Phi}}$ qui minimise un certain critère sur les séquences d'apprentissage $\{\mathcal{A}_i\}_{i=1\cdots N}$:

$$\kappa(\mathcal{X},\mathcal{Y}) = \overline{\mathbf{\Phi}}(\mathcal{X})^{\mathrm{T}} \left(\mathbf{I_D} - \mathbf{v}^{*\mathrm{T}}\mathbf{v}^*\right) \overline{\mathbf{\Phi}}(\mathcal{Y})$$

$$\text{avec } \mathbf{v}^* = \arg \min_{\substack{\mathbf{v}\in\mathbb{R}^D \\ \|\mathbf{v}\|_2=1}} \sum_{i=1}^{N}\sum_{j=i+1}^{N} W_{i,j} \left\| \left(\mathbf{I_D} - \mathbf{v}^{*\mathrm{T}}\mathbf{v}^*\right) \left(\overline{\mathbf{\Phi}}(\mathcal{A}_i) - \overline{\mathbf{\Phi}}(\mathcal{A}_j)\right)\right\|_2^2 \tag{3.61}$$

où $\|\cdot\|_2$ est la norme euclidienne et où les valeurs de la matrice $W_{i,j}$ sont des poids réels choisis arbitrairement pour minimiser et/ou maximiser les variances selon certains critères. Typiquement, $W_{i,j} > 0$ si les séquences d'apprentissage $\mathcal{A}_i$ et $\mathcal{A}_j$ ont été produites par le même locuteur et/ou les mêmes conditions d'enregistrement, et $W_{i,j} = 0$ sinon.

Local kernels

Les *Local Kernels* sont des noyaux initialement conçus pour la classification de séquences vidéos par [Wallraven et al., 2003]. La motivation de tels noyaux émerge du fait que les noyaux de la forme (3.58) comparent via le noyau k des vecteurs $(\mathrm{x}_t, \mathrm{y}_s)$ qui n'ont rien à voir d'un point de vue sémantique. L'idée sous-jacente est alors de ne faire intervenir dans le calcul du noyau de séquences que les couples d'objets qui sont les plus ressemblants, afin de filtrer l'information non pertinente. En pratique, cette sélection de couple est opérée par l'opérateur 'maximum' (au lieu de l'opérateur 'moyenne'). Les *Local Kernels* ont été introduits en vérification du locuteur par [Mariéthoz et Bengio, 2006], qui implémente la quantité suivante :

$$\kappa(\mathcal{X},\mathcal{Y}) = \tfrac{1}{T_\mathrm{X}} \sum_{t=1}^{T_\mathrm{X}} \max_s k(\mathrm{x}_t, \mathrm{y}_s) + \tfrac{1}{T_\mathrm{Y}} \sum_{s=1}^{T_\mathrm{Y}} \max_t k(\mathrm{x}_t, \mathrm{y}_s) \tag{3.62}$$

Autrement dit, cela revient à prendre dans (3.57) :

$$\Omega^{(\mathrm{X},\mathrm{Y})}_{t^*,s^*} = \begin{cases} 1/T_\mathrm{X} + 1/T_\mathrm{Y} & \text{si } s^* = \arg\max_s(k(\mathrm{x}_{t^*}, \mathrm{y}_s)) \text{ et } t^* = \arg\max_t(k(\mathrm{x}_t, \mathrm{y}_{s^*})) \\ 1/T_\mathrm{X} & \text{si } s^* = \arg\max_s(k(\mathrm{x}_{t^*}, \mathrm{y}_s)) \text{ et } t^* \neq \arg\max_t(k(\mathrm{x}_t, \mathrm{y}_{s^*})) \\ 1/T_\mathrm{Y} & \text{si } s^* \neq \arg\max_s(k(\mathrm{x}_{t^*}, \mathrm{y}_s)) \text{ et } t^* = \arg\max_t(k(\mathrm{x}_t, \mathrm{y}_{s^*})) \\ 0 & \text{sinon} \end{cases}$$

Les noyaux ainsi formulés présentent deux inconvénients :

- (*Théorique*) Ils ne vérifient pas les conditions de Mercer.
- (*Pratique*) Ils ne peuvent être estimés que par la forme (3.57), ce qui d'après le tableau 3.20 introduit des complexités calculatoires élevées. La mise en œuvre d'un tel noyau est délicate dans le cadre des évaluations NIST SRE (qui mettent en jeu des ordres de grandeurs proches de celles mentionnées dans Tab.3.20).

Toutefois, la non défini-positivité des noyaux de séquences ainsi conçus n'empêche pas d'observer de bonnes performances en pratique. Le succès des *Local Kernels* peut se prêter à deux interprétations :

- *Comparaison d'informations pertinentes issues d'un même processus.* En traitement de la parole, on peut par exemple pressentir que comparer les observations issues d'un phonème avec celles issues d'un phonème totalement différent ne fera qu'introduire des termes résidus parasites pour un problème de reconnaissance du locuteur.
- *Rejet de l'information non caractéristique (observations aberrantes)* pour une focalisation sur l'information pertinente. En effet, une observation non habituelle dans une séquence aura toutes les chances d'être éloignée des observations d'une autre séquence, et sera rejetée pour l'estimation du noyau avec cette séquence.

Si cette seconde interprétation s'avère plus vraie que la première, alors une étape de sélection des observations de la séquence en fin de pré-traitement (spécifique à chaque séquence et non à chaque paire de séquences) est préférable. Une telle démarche permet d'obtenir facilement un noyau vérifiant les conditions de Mercer.

3.5.2 Noyaux construits sur les densités de probabilité

Étant donné que la modélisation générative par GMM bénéficie d'une longue expérience en vérification du locuteur (§2.2), il semble intéressant de vouloir la combiner avec une méthode discriminante à noyau (§1.1.3). Nous avons vu que plusieurs techniques permettent d'intégrer les modèles probabilistes pour construire des noyaux (§3.3 et §3.4). Nous listons dans cette partie les approches de ce type qui ont été développées notamment pour la vérification du locuteur.

Application des noyaux entre densités de probabilités

Fig.3.8 représente la démarche utilisée en vérification du locuteur pour appliquer les noyaux entre densités de probabilité aux séquences acoustiques, fidèlement à la stratégie UBM-GMM. Cette approche a d'abord été introduite par [Ho et Moreno, 2004] en utilisant le noyau radial

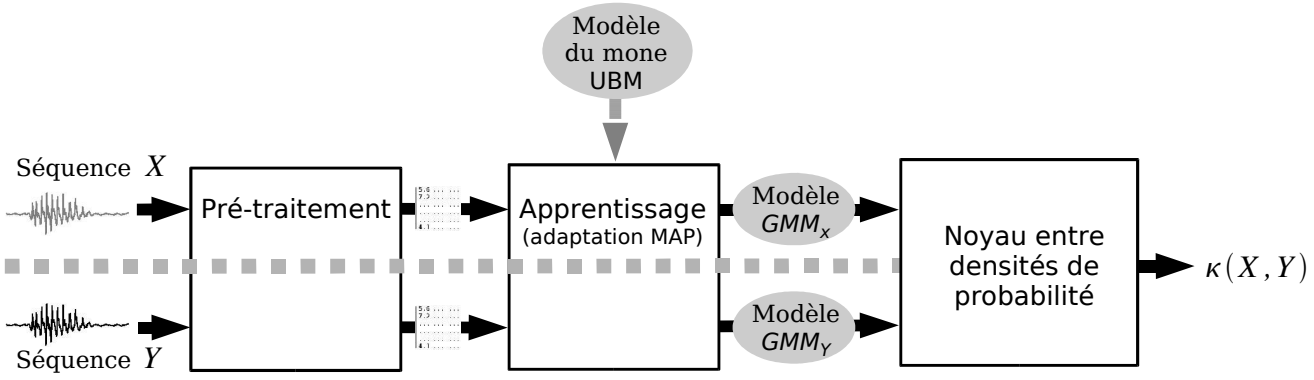


Fig. 3.8 - Schéma de l'utilisation d'un noyau entre distributions pour la vérification du locuteur.

formé à partir de la divergence de Kullback symétrique (3.22). La principale limitation d'un tel

noyau, en plus de ne pas vérifier les conditions de Mercer, est que son estimation fait appel à la méthode de Monte Carlo pour l'approximation d'intégrale, qui manque de robustesse dans le cas des GMMs.

D'autres approches ultérieures ont exploité une particularité de la modélisation UBM-GMM, qui est que les poids et matrices de covariance sont identiques pour tous les GMMs appris. Ceci permet de simplifier le calcul des divergences de Kullback entre composantes Gaussiennes (équation 3.26). Dans ce contexte, il est tentant de construire une distance entre GMMs à partir de distances de Mahalanobis entre les vecteurs moyennes des composantes Gaussiennes qui se correspondent (*i.e.* issues de l'adaptation d'une même composante de l'UBM). Le noyau de [Ho et Moreno, 2004] a ainsi été simplifié en un noyau ayant une forme analytique simple pour les GMMs adaptés [Dehak et Chollet, 2006]. De la distance entre GMMs utilisée par [Dehak et Chollet, 2006], [Campbell et al., 2006b] ont dérivé un noyau correspondant, de type "projectif", dont nous donnons l'expression ci-dessous (3.63). Ces deux travaux sont basés sur une approximation de la divergence de Kullback symétrique entre GMMs, d'après la majoration formulée par [Do, 2003] :

$$0 \leq \widetilde{\mathcal{D}}^{\mathrm{KL}}\big(\underbrace{\{\omega_g, \boldsymbol{\mu}_{X,g}, \boldsymbol{\Sigma}_g\}}_{\boldsymbol{\theta}_\mathrm{X}}, \underbrace{\{\omega_g, \boldsymbol{\mu}_{Y,g}, \boldsymbol{\Sigma}_g\}}_{\boldsymbol{\theta}_\mathrm{Y}}\big) \leq \underbrace{\tfrac{1}{2}\sum_{g=1}^{G} \omega_g \left(\boldsymbol{\mu}_{X,g} - \boldsymbol{\mu}_{Y,g}\right)^{\mathrm{T}} {\boldsymbol{\Sigma}_g}^{-1} \left(\boldsymbol{\mu}_{X,g} - \boldsymbol{\mu}_{Y,g}\right)}_{\mathcal{D}^{\mathrm{gm}}\left(\boldsymbol{\theta}_\mathrm{X}, \boldsymbol{\theta}_\mathrm{Y}\right)^2}$$

La distance $\mathcal{D}^{\mathrm{gm}}$ n'est autre qu'une somme quadratique de distances de Mahalanobis. Elle induit un noyau de Mercer dont on peut facilement trouver une expression analytique faisant intervenir un produit scalaire entre *expansions* de modèles GMMs de taille dG ("supervecteurs GMM") :

$$\kappa^{\mathrm{gm}}\left(\boldsymbol{\theta}_\mathrm{X}, \boldsymbol{\theta}_\mathrm{Y}\right) = \sum_{g=1}^{G} \left(\sqrt{\omega_g}{\boldsymbol{\Sigma}_g}^{-\frac{1}{2}} \boldsymbol{\mu}_{X,g}\right)^{\mathrm{T}} \left(\sqrt{\omega_g}{\boldsymbol{\Sigma}_g}^{-\frac{1}{2}} \boldsymbol{\mu}_{Y,g}\right) \tag{3.63}$$

Le fait de pouvoir écrire le noyau comme un produit scalaire entre *expansions* de séquence d'après cette forme fait entrer le noyau dans la catégories des noyaux de la forme (3.58) avec les avantages que cela comporte : complexité calculatoire réduite (Tab.3.20) et possibilité d'appliquer la méthode de compensation NAP (équation 3.61).

Noyau de Fisher et Noyau TOP

Le noyau de TOP (§3.4.4) a été utilisé conformément à la stratégie UBM-GMM pour la vérification du locuteur par [Wan et Renals, 2004], qui désigne le noyau par "noyau de Fisher". Pour gagner en performance, l'expansion considérée pour le calcul du noyau est :

$$U(\{\boldsymbol{\theta}^*_{\mathrm{loc}}, \boldsymbol{\theta}^*_{\mathrm{UBM}}\}, \mathcal{X}) = \begin{bmatrix} \log p(\mathcal{X}|\boldsymbol{\theta}^*_{\mathrm{loc}}) - \log p(\mathcal{X}|\boldsymbol{\theta}^*_{\mathrm{UBM}}) \\ \boldsymbol{\nabla}_{\{\boldsymbol{\mu}_g\}} \log p(\mathcal{X}|\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*_{\mathrm{loc}}} \\ -\boldsymbol{\nabla}_{\{\boldsymbol{\mu}_g\}} \log p(\mathcal{X}|\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*_{\mathrm{UBM}}} \end{bmatrix} \tag{3.64}$$

Mais cette technique reste lourde en complexité de calcul pour n'améliorer que de peu les performances des systèmes GMM classiques. Notons toutefois que le noyau préconisé par

[Jaakkola et Haussler, 1998] donne d'aussi bonnes performances, avec une *expansion* de Fisher de taille dG au lieu de $(2dG+1)$, et indépendant du locuteur cible :

$$U(\boldsymbol{\theta}^*_{\text{UBM}}, \mathcal{X}) = \left[\left. \boldsymbol{\nabla}_{\{\boldsymbol{\mu}_g\}_{g=1\cdots G}} \log p(\mathcal{X}|\boldsymbol{\theta}) \right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*_{\text{UBM}}} \right] \tag{3.65}$$

Par rapport au noyau TOP, ce noyau présente l'avantage d'être indépendant du locuteur cible, ce qui est un plus lorsque l'on impose à un système un seul seuil de décision pour tous les locuteurs. [Scheffer et Bonastre, 2006] atteignent par exemple de bonnes performances avec un noyau utilisant une formulation très proche de ce noyau de Fisher, inspiré pour sa conception du travail de [Campbell et al., 2003b]. Aucune tentative d'appliquer efficacement d'autres noyaux MI que le noyau de Fisher n'a encore abouti en vérification du locuteur.

3.5.3 Noyaux entre séquences ordonnées

Même si les noyaux de séquences qui tiennent compte de l'ordre des vecteurs ne sont pas considérés pour la vérification du locuteur en mode "indépendant du texte", ils peuvent être intéressant en mode "dépendant du texte". Aucun noyau entre séquences ordonnées n'a été appliqué dans ce contexte à l'heure actuelle. Nous suggérons ici des perspectives basées sur des travaux relatifs à d'autres applications.

Noyaux à partir des densités de probabilité

Une première alternative est de construire un noyau entre HMMs (au lieu de GMMs comme en §3.5.2), à partir de distances entre HMMs comme celles proposées par [Lyngs et al., 1999, Pedersen et Lyngsø, 2001]. Il est aussi envisageable de construire un noyau de Fisher à partir des HMMs, comme il a été fait par [Kersting et Gärtner, 2004] pour des séquences logiques.

Noyaux à partir d'alignement dynamique

Enfin, des noyaux peuvent être construits à partir de l'*Exponential Embedding* de distances par alignement dynamique, comme il a été fait par [Shimodaira et al., 2001, Wan et Carmichael, 2005]. Nous rappelons que la distance cumulée par DTW ("*Dynamic Time Warping*") entre deux séquences de vecteurs $\mathcal{X} = \{\mathrm{x}_t\}_{t=1\cdots T_\mathrm{X}}$ et $\mathcal{Y} = \{\mathrm{y}_s\}_{s=1\cdots T_\mathrm{Y}}$ est la valeur $\mathcal{D}_{(T_\mathrm{X},T_\mathrm{Y})}(\mathcal{X},\mathcal{Y})$ construite en minimisant itérativement :

$$\forall\ (t,s) \in \{1\cdots T_\mathrm{X}\}\times\{1\cdots T_\mathrm{Y}\}\ , \qquad \mathcal{D}_{(t,s)}(\mathcal{X},\mathcal{Y}) = \min \left\{ \begin{array}{c} \mathcal{D}_{(t-1,s)}(\mathcal{X},\mathcal{Y}) + d(\mathrm{x}_t,\mathrm{y}_s) \\ \mathcal{D}_{(t-1,s-1)}(\mathcal{X},\mathcal{Y}) + r.d(\mathrm{x}_t,\mathrm{y}_s) \\ \mathcal{D}_{(t,s-1)}(\mathcal{X},\mathcal{Y}) + d(\mathrm{x}_t,\mathrm{y}_s) \end{array} \right\}$$

avec typiquement $r = 2$ ou $\sqrt{2}$, et où d est une mesure distance entre les vecteurs x_t et y_s. À partir de cette distance, [Shimodaira et al., 2001] dérivent un noyau $\kappa_{(T_\mathrm{X},T_\mathrm{Y})}(\mathcal{X},\mathcal{Y})$ construit en maximisant itérativement :

$$\forall\ (t,s) \in \{1\cdots T_\mathrm{X}\}\times\{1\cdots T_\mathrm{Y}\}\ , \qquad \kappa_{(t,s)}(\mathcal{X},\mathcal{Y}) = \max \left\{ \begin{array}{c} \kappa_{(t-1,s)}(\mathcal{X},\mathcal{Y}) + {\mathrm{x}_t}^\mathrm{T}\mathrm{y}_s \\ \kappa_{(t-1,s-1)}(\mathcal{X},\mathcal{Y}) + r.{\mathrm{x}_t}^\mathrm{T}\mathrm{y}_s \\ \kappa_{(t,s-1)}(\mathcal{X},\mathcal{Y}) + {\mathrm{x}_t}^\mathrm{T}\mathrm{y}_s \end{array} \right\}$$

[Wan et Carmichael, 2005] quant à eux calculent un *Exponential Embedding* de la distance cumulée DTW. Dans les deux cas, le noyau construit ne vérifie pas les conditions de Mercer, à cause de l'opérateur 'maximum', tout comme pour les *Local Kernels* (§3.5.1). Pour combler cette lacune, Wan introduit un artifice pour normaliser les valeurs de noyau. Même si cette astuce ne garantit pas les conditions de Mercer, elle fait gagner en stabilité.

Chapitre 4

Nouveau noyau de séquences pour la vérification du locuteur

Sommaire

DANS ce chapitre, nous proposons un nouveau noyau de séquences, pour des séquences non ordonnées de tailles variables. Même si nous appliquons ensuite ce nouveau noyau de séquences à des séquences de vecteurs, il peut être étendu pour n'importe quel type de données du moment qu'un noyau (mesure de similarité) est défini entre deux objets.

Ce nouveau noyau est issu d'un noyau de séquences antérieurement proposé par [Campbell, 2001] : le noyau GLDS (§4.1), qui a montré de bonnes performances lors des évaluations NIST SRE pour une très bonne efficacité et une certaine simplicité d'implémentation. Ce noyau est basé d'une part sur une expansion formée de monomes impliquant les composantes des entrées vectorielles et, d'autre part, sur une normalisation de ces expansions à partir de statistiques estimées sur un corpus de données non étiquetées (corpus du Monde). Mais à cause de la contrainte de complexité calculatoire, les monomes de l'expansion ne sont en pratique implémentés que jusqu'à un degré limité (inférieur ou égal à 3). Cela limite la capacité de la modélisation à saisir le pouvoir discriminant des paramètres extraits, et empêche de généraliser la méthode à des entrées non-vectorielles. Un des objectifs de ce travail est de généraliser le noyau GLDS pour une expansion sous-jacente quelconque, de dimension éventuellement infinie comme c'est le cas avec les noyaux RBF qui ont été utilisés avec succès dans diverses applications [Shawe-Taylor et Cristianini, 2004]. Nous définissons ainsi une famille de noyaux de séquences (§4.2) dont le GLDS est un cas particulier (avec un noyau polynomial). Nous donnons plusieurs interprétations de ces noyaux qui seront désignés par FSNS.

L'efficacité calculatoire du noyau GLDS vient du fait que les séquences sont explicitement projetées dans un espace de dimension fixe, via une expansion moyenne. Ceci permet d'appliquer les algorithmes d'apprentissage et de test avec un nombre d'opérations linéaire par rapport à l'ordre de grandeur des longueurs T des séquences. Dans les cas où l'expansion ne peut pas être calculée explicitement, la complexité des noyaux de séquences tels qu'ils sont formulés en §4.2 devient $O(T^2)$ au lieu de $O(T)$ (Tab.3.20). Ce niveau de complexité est rédhibitoire lorsque l'on veut traiter une multitude de séquences relativement longues, comme c'est le cas dans les évaluations NIST de reconnaissance du locuteur, où T est de l'ordre de 10^4. Nous proposons de réduire cette complexité en deux temps. Nous présentons d'abord une formulation "duale" des noyaux FSNS (§4.3) où l'expansion n'a pas à être calculée explicitement et où la complexité est pourtant linéaire par rapport à T. Étant donné qu'après une telle astuce, la complexité reste encore trop élevée pour la vérification du locuteur, nous proposons de la réduire en faisant quelques approximations (§4.3). Ces approximations font intervenir une méthode d'optimisation appelée "Décomposition de Cholesky Incomplète".

4.1 Le noyau GLDS

4.1.1 Définition

Le noyau GLDS conçu par [Campbell, 2002] (plus tard mieux détaillé dans [Campbell et al., 2006a]) est basé sur une "expansion polynomiale" de vecteurs. Cette expansion est formée des monomes entre composantes des vecteurs d'entrée, jusqu'à un degré p fixé. Nous la noterons $\boldsymbol{\phi}^p$ dans cette section.

Si $\mathbf{x} = [\ x_1 \cdots x_d\ ]^{\mathrm{T}}$ désigne un vecteur d'entrée, alors ces monomes sont de la forme $(x_1^{q_1} x_2^{q_2} \cdots x_d^{q_d})$. La somme $(q_1 + \cdots + q_d)$ des degrés (positifs ou nuls) de chaque monome est inférieure ou égale à p. Par exemple pour des vecteurs bidimensionnels ($d = 2$) notés $\mathbf{x} = [\ x_1\ x_2\ ]^{\mathrm{T}}$, l'expansion polynomiale de degré $p = 2$ est : $\boldsymbol{\phi}^2(\mathbf{x}) = [\ x_1\ \ x_2\ \ x_1^2\ \ x_1x_2\ \ x_2^2\ ]^{\mathrm{T}}$.

Le noyau entre deux séquences $\mathbf{X} = \{\mathbf{x}_t\}_{t=1\cdots T_{\mathrm{X}}}$ et $\mathbf{Y} = \{\mathbf{y}_s\}_{s=1\cdots T_{\mathrm{Y}}}$ est le noyau linéaire généralisé entre les moyennes des expansions :

$$\kappa^{\mathrm{GLDS}}(\mathbf{X}, \mathbf{Y}) = \left(\tfrac{1}{T_{\mathrm{X}}} \sum_{t=1}^{T_{\mathrm{X}}} \boldsymbol{\phi}^p(\mathbf{x}_t) \right)^{\mathrm{T}} \mathbf{S}_{\mathrm{B}}{}^{-1} \left(\tfrac{1}{T_{\mathrm{Y}}} \sum_{s=1}^{T_{\mathrm{Y}}} \boldsymbol{\phi}^p(\mathbf{y}_s) \right) \tag{4.1}$$

où $\mathbf{S}_{\mathrm{B}}$ est la matrice des second moments (non centrés) des expansions polynomiales $\boldsymbol{\phi}^p$, estimés sur un corpus de vecteurs non étiquetées $\mathbf{B} = \{\mathbf{b}_i\}_{i=1\cdots N}$ ("*Background*", corpus du Monde) :

$$\mathbf{S}_{\mathrm{B}} = \tfrac{1}{N} \sum_{i=1}^{N} \boldsymbol{\phi}^p(\mathbf{b}_i) \boldsymbol{\phi}^p(\mathbf{b}_i)^{\mathrm{T}} \tag{4.2}$$

Notons que pour réduire la complexité de calcul, on peut ne calculer que les valeurs diagonales de cette matrice, ce qui est souvent fait en pratique. Cela revient à négliger les corrélations entre composantes de l'expansion polynomiale $\boldsymbol{\phi}^p$ (mais pas les corrélations entre composantes des vecteurs d'entrée, qui sont elles codées d'une certaine manière dans l'expansion). En prenant une matrice de normalisation $\mathbf{S}_{\mathrm{B}}$ diagonale, la complexité du calcul du noyau GLDS (4.1), devient $O(D)$ au lieu de $O(D^2)$, où D est la dimension de l'expansion polynomiale.

Avec une dimension d'entrée d, l'expansion de degré p est de taille $D = \frac{(d+p)!}{d!\ p!}$. Le tableau suivant montre les valeurs des tailles D des expansions polynomiales selon la dimension d d'entrée et le degré polynomial p maximal :

	$p = 2$	$p = 3$	$p = 4$	$p = 5$
$d = 15$	D =136	816	3 876	15 504
$d = 25$	351	3 276	23 751	142 506
$d = 30$	666	8 436	82 251	658 008

La croissance exponentielle de cette dimension avec le degré explique pourquoi en pratique, le noyau GLDS n'est implémenté qu'avec $p = 3$. Ce manque de flexibilité est un frein à la fois pour les performances de la modélisation et pour l'utilisation du noyau dans d'autres contextes :

1. La complexité de la modélisation a une influence directe sur la complexité calculatoire du noyau via la taille D des *expansions*. De ce fait, la formulation initiale du noyau GLDS

ne permet pas d'étendre aux séquences les noyaux polynomiaux de degrés élevés, ni les noyaux radiaux comme le noyau RBF Gaussien.

2. Le noyau GLDS ne peut pas s'appliquer à d'autres types d'objets que les séquences de vecteurs (*e.g.* séquences de symboles, de distributions, etc.), à moins de passer par une *expansion* vectorielle des séquences.

4.1.2 Fondements théoriques

[Campbell, 2001] présente le noyau GLDS comme le résultat approché d'une procédure d'apprentissage sur une séquence (*train*) et d'attribution de scores sur une autre (*test*), comme illustré dans Fig.4.1. La procédure classique *train/test* permet en effet de renvoyer une mesure de similarité, qui peut faire guise de noyau tant qu'elle est définie positive. La même "philosophie" de conception de noyau a été réutilisée plus tard par [Campbell et al., 2003b] pour concevoir un noyau entre séquences de phonèmes. Dans ce second travail, l'élément changeant par rapport à la conception du noyau GLDS est la forme du modèle utilisée dans la procédure *train* (modèle génératif de *n-grams* au lieu d'un modèle discriminatif vectoriel).

Nous présentons maintenant un développement mathématique, un peu modifié par rapport aux formulations de [Campbell, 2001], qui conduit à l'interprétation *train/test* du noyau GLDS.

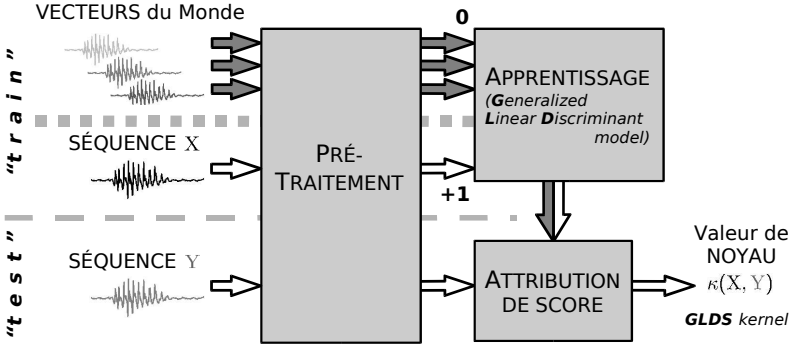


Fig. 4.1 - Conception du noyau GLDS : illustration de la démarche *train/test*.

• (*train*) Apprentissage sur une séquence X

Les vecteurs du Monde $\{\mathbf{b}_i\}_{i=1\cdots N}$ représentent un ensemble d'observations produites par des imposteurs. [Campbell et Assaleh, 1999] formulent le problème d'apprentissage d'un classifieur polynomial sur les vecteurs $\{\mathbf{x}_t\}_{t=1\cdots T_\mathrm{X}}$ sur une séquence $\mathbf{X}$ prononcée par un seul et même locuteur. Un tel apprentissage par "moindres carrés" pondérés consiste à chercher une fonction $f(\cdot|\boldsymbol{\theta}_\mathrm{X}) : \mathbb{R}^d \to \mathbb{R}$ (paramétrée par $\boldsymbol{\theta}_\mathrm{X}$) qui minimise l'écart entre la valeur renvoyée et la valeur 0/1 recherchée :

$$\boldsymbol{\theta}_\mathrm{X} = \arg\min_{\boldsymbol{\theta}} \left\{ \tfrac{1}{N} \sum_{i=1}^{N} \big(f(\mathbf{b}_i|\boldsymbol{\theta})\big)^2 + \tfrac{1}{T_\mathrm{X}} \sum_{t=1}^{T_\mathrm{X}} \big(1 - f(\mathbf{x}_t|\boldsymbol{\theta})\big)^2 \right\} \tag{4.3}$$

Pour les classifieurs polynomiaux, les fonctions $f(\cdot|\boldsymbol{\theta}_\mathrm{X})$ sont recherchées parmi les fonctions

("G*eneralized* L*inear* D*iscriminant*") de la forme :

$$f(\cdot|\boldsymbol{\theta}) = \boldsymbol{\phi}^p(\cdot)^{\mathrm{T}}\boldsymbol{\theta} \ , \qquad \boldsymbol{\theta} \in \mathbb{R}^D \tag{4.4}$$

Il est facile de montrer par la méthode classique des équations normales que la minimisation de (4.3) sous la contrainte (4.4) revient à choisir :

$$\boldsymbol{\theta}_{\mathrm{X}} = {\mathbf{S}_{\mathrm{X}}}^{-1} \left(\tfrac{1}{T_{\mathrm{X}}} \sum_{t=1}^{T_{\mathrm{X}}} \boldsymbol{\phi}^p(\mathbf{x}_t) \right) \tag{4.5}$$

$$\text{où} \qquad \mathbf{S}_{\mathrm{X}} = \underbrace{\tfrac{1}{N} \sum_{i=1}^{N} \boldsymbol{\phi}^p(\mathbf{b}_i)\boldsymbol{\phi}^p(\mathbf{b}_i)^{\mathrm{T}}}_{\mathbf{S}_{\mathrm{B}}} + \tfrac{1}{T_{\mathrm{X}}} \sum_{t=1}^{T_{\mathrm{X}}} \boldsymbol{\phi}^p(\mathbf{x}_t)\boldsymbol{\phi}^p(\mathbf{x}_t)^{\mathrm{T}} \tag{4.6}$$

Si $\mathbf{S}_{\mathrm{X}}$ n'est pas inversible, la solution $\boldsymbol{\theta}_{\mathrm{X}}$ s'exprime de manière analogue en considérant la pseudo-inverse ${\mathbf{S}_{\mathrm{X}}}^{\dagger}$ (*cf.* annexe A.1.2) au lieu de l'inverse. Une condition nécessaire pour que $\mathbf{S}_{\mathrm{X}}$ (resp. $\mathbf{S}_{\mathrm{B}}$) soit inversible est que le nombre de vecteurs d'apprentissage $N + T_{\mathrm{X}}$ (resp. N) soit supérieur ou égal à la dimension D des expansions polynomiales. Dans notre application, N est de l'ordre de 10^6 et D de l'ordre de 10^4, et il n'y a pas de problème d'inversion.

• (*test*) Attribution de scores à une séquence Y

En supposant les vecteurs d'une autre séquence $\{\mathbf{y}_t\}_{t=1\cdots T_{\mathrm{Y}}}$ indépendants, le score attribué à cette séquence $\mathbf{Y}$ selon le modèle (4.5) appris sur $\mathbf{X}$ est la moyenne de valeurs de la fonction discriminante généralisée :

$$\begin{aligned} \mathrm{score}(\mathbf{Y}|\mathbf{X}) &= \tfrac{1}{T_{\mathrm{Y}}} \sum_{s=1}^{T_{\mathrm{Y}}} f(\mathbf{y}_s|\boldsymbol{\theta}_{\mathrm{X}}) \\ &= \left(\tfrac{1}{T_{\mathrm{X}}} \sum_{t=1}^{T_{\mathrm{X}}} \boldsymbol{\phi}^p(\mathbf{x}_t) \right)^{\mathrm{T}} {\mathbf{S}_{\mathrm{X}}}^{-1} \left(\tfrac{1}{T_{\mathrm{Y}}} \sum_{s=1}^{T_{\mathrm{Y}}} \boldsymbol{\phi}^p(\mathbf{y}_s) \right) \end{aligned} \tag{4.7}$$

• Approximation

On obtient l'expression du GLDS (4.1) en négligeant le second terme de (4.6), c'est-à-dire en prenant $\mathbf{S}_{\mathrm{X}} = \mathbf{S}_{\mathrm{B}}$, où la matrice des seconds moments empiriques $\mathbf{S}_{\mathrm{B}}$ est indépendante de la séquence $\mathbf{X}$. Le bien-fondé de cette approximation tient au fait que les données du Monde sont représentatives de tous les locuteurs, y compris le locuteur intervenant dans la séquence $\mathbf{X}$. Elles permettent en effet d'estimer des statistiques "*speaker-independent*". Cet artifice, en plus de réduire les complexités de calcul, permet de garantir les conditions de Mercer : le noyau est symétrique et défini positif, vu qu'il s'écrit explicitement comme un produit scalaire entre *expansions* normalisées de séquences.

4.1.3 Une première extension du noyau GLDS

Nous présentons maintenant une première approche pour généraliser le noyau GLDS à une *expansion* $\boldsymbol{\phi}$ quelconque, en suivant la philosophie de ce dernier, c'est-à-dire en reprenant les arguments théorique de Campbell que nous venons de présenter [Louradour et Daoudi, 2005b, Louradour et Daoudi, 2005c, Louradour et Daoudi, 2005a].

L'idée est de reproduire le schéma *train/test* proposé par [Campbell, 2001] avec des fonctions de modélisation discriminante plus complexes que les fonctions polynomiales. Ces expressions sont toujours de la forme

$$f(\cdot|\boldsymbol{\theta}) = \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}(\cdot)\ , \qquad \boldsymbol{\theta} \in \mathbb{R}^D \tag{4.8}$$

mais l'*expansion* $\boldsymbol{\phi}$ peut être de dimension infini ($D = \infty$). Elle définit un noyau de Mercer selon :

$$k(\mathbf{x}, \mathbf{y}) = \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{y})$$

Pour pouvoir exploiter l'astuce du noyau au travers du noyau GLDS, il faut arriver à une formulation où l'*expansion* $\boldsymbol{\phi}$ n'intervient qu'implicitement via la fonction noyau k.

• (*train*) Apprentissage sur une séquence X

D'après le théorème 1 des représentants, la solution du problème d'apprentissage (4.3) peut s'écrire sous la forme

$$f(\mathbf{z}|\boldsymbol{\beta}_{\mathrm{X}}) = \sum_{t=1}^{T_{\mathrm{X}}} \beta_t^{(+)} k(\mathbf{x}_t, \mathbf{z}) + \sum_{i=1}^{N} \beta_i^{(-)} k(\mathbf{b}_i, \mathbf{z}) \tag{4.9}$$

Au lieu des D variables de $\boldsymbol{\theta}_{\mathrm{X}}$ intervenant dans la forme primale (4.8), cette nouvelle forme fait intervenir N variables "duales", que nous représentons par la colonne :

$$\boldsymbol{\beta}_{\mathrm{X}} = \left[\ \beta_1^{(+)}\ \cdots\ \beta_{T_{\mathrm{X}}}^{(+)}\ \ \beta_1^{(-)}\ \cdots\ \beta_N^{(-)}\ \right]^{\mathrm{T}} \tag{4.10}$$

Pour la clarté des expressions qui vont suivre, nous passons maintenant à une écriture matricielle. Nous considérons la matrice de Gram sur l'ensemble des vecteurs d'apprentissage organisé comme suit :

$$\mathbf{K} = \left[\begin{array}{c|c} \mathbf{K}_{\mathrm{X}} & \mathbf{K}_{\mathrm{X}\times\mathrm{B}} \\ \hline \mathbf{K}_{\mathrm{X}\times\mathrm{B}}{}^{\mathrm{T}} & \mathbf{K}_{\mathrm{B}} \end{array}\right] \tag{4.11}$$

$$\text{avec } \left\{\begin{array}{rcll} \left(\mathbf{K}_{\mathrm{X}}\right)_{t,s} & = & k(\mathbf{x}_t, \mathbf{x}_s)\ , & \{t,s\} \in [1\cdots T_{\mathrm{X}}]^2 \\ \left(\mathbf{K}_{\mathrm{X}\times\mathrm{B}}\right)_{t,i} & = & k(\mathbf{x}_t, \mathbf{b}_i)\ , & \{t,i\} \in [1\cdots T_{\mathrm{X}}]\times[1\cdots N] \\ \left(\mathbf{K}_{\mathrm{B}}\right)_{i,j} & = & k(\mathbf{b}_i, \mathbf{b}_j)\ , & \{i,j\} \in [1\cdots N]^2 \end{array}\right. \tag{4.12}$$

Aussi nous notons :

$$\mathbf{D} = \mathrm{diag}\Big(\ \underbrace{\tfrac{1}{T_{\mathrm{X}}}\cdots\tfrac{1}{T_{\mathrm{X}}}}_{T_{\mathrm{X}}\text{ termes}}\ \Big|\ \underbrace{\tfrac{1}{N}\cdots\tfrac{1}{N}}_{N\text{ termes}}\ \Big) \quad \text{(matrice de pondération)}$$

$$\mathbf{s} = \Big[\ \overbrace{1\ \cdots\ 1}\ \Big|\ \overbrace{0\ \cdots\ 0}\ \Big]^{\mathrm{T}} \quad \text{(vecteur des sorties désirées)}$$

Ces notations nous permettent de reformuler le problème d'apprentissage (4.3) sous une forme matricielle impliquant les variables duales :

$$\boldsymbol{\beta}_{\mathrm{X}} = \arg\min_{\boldsymbol{\beta}} \left\{(\mathbf{s} - \mathbf{K}\boldsymbol{\beta})^{\mathrm{T}}\,\mathbf{D}\,(\mathbf{s} - \mathbf{K}\boldsymbol{\beta})\right\} \tag{4.13}$$

La solution à ce problème d'optimisation par la méthode des équations normales est :

$$\boldsymbol{\beta}_{\mathrm{X}} = \left(\mathbf{K}^{\mathrm{T}}\mathbf{D}\mathbf{K}\right)^{\dagger}\mathbf{K}^{\mathrm{T}}\mathbf{D}\mathbf{s} \tag{4.14}$$

où $\mathbf{M}^{\dagger}$ désigne la pseudo-inverse de $\mathbf{M}$ (annexe A.1.2). Afin de simplifier cette expression, nous introduisons l'application suivante (sur laquelle nous reviendrons plus tard) :

$$\boldsymbol{\psi}(\mathbf{z}) = \begin{bmatrix} \overline{\boldsymbol{\psi}}_{\mathrm{X}}(\mathbf{z}) \\ \overline{\boldsymbol{\psi}}_{\mathrm{B}}(\mathbf{z}) \end{bmatrix} \tag{4.15}$$

$$\text{avec} \quad \begin{cases} \boldsymbol{\psi}_{\mathrm{X}}(\mathbf{z}) = \begin{bmatrix} k(\mathrm{x}_1, \mathbf{z}) & \cdots & k(\mathrm{x}_{T_{\mathrm{X}}}, \mathbf{z}) \end{bmatrix}^{\mathrm{T}} \\ \boldsymbol{\psi}_{\mathrm{B}}(\mathbf{z}) = \begin{bmatrix} k(\mathrm{b}_1, \mathbf{z}) & \cdots & k(\mathrm{b}_N, \mathbf{z}) \end{bmatrix}^{\mathrm{T}} \end{cases} \tag{4.16}$$

Si le noyau est symétrique ($\mathbf{K} = \mathbf{K}^{\mathrm{T}}$), alors la solution (4.14) peut finalement s'écrire :

$$\begin{aligned} \boldsymbol{\beta}_{\mathrm{X}} &= \mathbf{R}_{\mathrm{X}}{}^{\dagger}\left(\tfrac{1}{T_{\mathrm{X}}}\textstyle\sum_{t=1}^{T_{\mathrm{X}}}\boldsymbol{\psi}(\mathbf{x}_t)\right) \\ \text{où} \qquad \mathbf{R}_{\mathrm{X}} &= \tfrac{1}{N}\displaystyle\sum_{i=1}^{N}\boldsymbol{\psi}(\mathbf{b}_i)\boldsymbol{\psi}(\mathbf{b}_i)^{\mathrm{T}} + \tfrac{1}{T_{\mathrm{X}}}\sum_{t=1}^{T_{\mathrm{X}}}\boldsymbol{\psi}(\mathbf{x}_t)\boldsymbol{\psi}(\mathbf{x}_t)^{\mathrm{T}} \end{aligned} \tag{4.17}$$

• (*test*) Attribution de scores à une séquence Y

Le score attribué à une séquence $\mathbf{Y}$ selon la fonction discriminante (4.5) appris sur $\mathbf{X}$, la moyenne :

$$\begin{aligned} \mathrm{score}(\mathbf{Y}|\mathbf{X}) &= \tfrac{1}{T_{\mathrm{Y}}}\displaystyle\sum_{s=1}^{T_{\mathrm{Y}}} f(\mathbf{y}_s|\boldsymbol{\beta}_{\mathrm{X}}) && (4.18) \\ &= \left(\tfrac{1}{T_{\mathrm{X}}}\displaystyle\sum_{t=1}^{T_{\mathrm{X}}}\boldsymbol{\psi}(\mathbf{x}_t)\right)^{\mathrm{T}} \mathbf{R}_{\mathrm{X}}{}^{\dagger} \left(\tfrac{1}{T_{\mathrm{Y}}}\displaystyle\sum_{s=1}^{T_{\mathrm{Y}}}\boldsymbol{\psi}(\mathbf{y}_s)\right) && (4.19) \end{aligned}$$

• Approximations

Faire la même approximation que pour le GLDS reviendrait à remplacer la matrice $\mathbf{R}_{\mathrm{X}}$ par la matrice des seconds moments des $\boldsymbol{\psi}(\cdot)$ estimée sur les vecteurs du Monde seulement :

$$\mathbf{R}_{\mathrm{X}} \approx \tfrac{1}{N}\sum_{i=1}^{N}\boldsymbol{\psi}(\mathbf{b}_i)\boldsymbol{\psi}(\mathbf{b}_i)^{\mathrm{T}} \tag{4.20}$$

Mais cette approximation ne permet pas d'aboutir à un noyau de Mercer, à cause de la dépendance de la fonction $\boldsymbol{\psi}$ (4.15) aux vecteurs de la séquence $\mathbf{X}$. Aussi, la complexité calculatoire de la quantité (4.19) est $O\left((N+T_{\mathrm{X}})^2\right)$, ce qui est rédhibitoire pour l'application visée, N étant de l'ordre de 10^6.

Une façon simple de pallier ces deux problèmes est de restreindre la forme duale de la fonction discriminante recherchée pendant la procédure *train*. Au lieu d'utiliser le résultat du théorème des représentants, nous considérons les fonctions paramétriques de la forme

$$f^*(\mathbf{z}|\boldsymbol{\beta}_{\mathrm{X}}) = \sum_{i=1}^{m}\beta_i k(\mathbf{c}_i, \mathbf{z}) \tag{4.21}$$

où $\mathbf{C} = \{\mathbf{c}_i\}_{i=1\cdots m}$ est un ensemble de vecteurs choisis indépendamment de $\mathbf{X}$, et de taille raisonnable $m \ll N$. Les prochaines sections donneront les éléments théoriques permettant de choisir le dictionnaire $\mathbf{C}$ de manière judicieuse. Contentons-nous pour l'instant de pressentir que les vecteurs de $\mathbf{C}$ doivent être représentatifs des observations du monde réel, et que la taille m règle la complexité de la modélisation.

Répercuter la nouvelle forme de la fonction discriminante (4.21) dans les équations (4.10) à (4.17) amène à trouver une solution analogue au problème d'apprentissage, qui s'écrit

$$\begin{aligned} \boldsymbol{\beta}_{\mathrm{X}} &= \mathbf{R}_{\mathrm{X}}^{*\,\dagger} \left(\tfrac{1}{T_{\mathrm{X}}} \sum_{t=1}^{T_{\mathrm{X}}} \boldsymbol{\psi}_{\mathrm{C}}(\mathbf{x}_t) \right) \\ \text{où} \quad \boldsymbol{\psi}_{\mathrm{C}}(\mathbf{z}) &= \left[\; k(\mathbf{c}_1, \mathbf{z}) \; \cdots \; k(\mathbf{c}_m, \mathbf{z}) \; \right]^{\mathrm{T}} \\ \text{et} \quad \mathbf{R}_{\mathrm{X}}^{*} &= \tfrac{1}{N} \sum_{i=1}^{N} \boldsymbol{\psi}_{\mathrm{C}}(\mathbf{b}_i) \boldsymbol{\psi}_{\mathrm{C}}(\mathbf{b}_i)^{\mathrm{T}} + \tfrac{1}{T_{\mathrm{X}}} \sum_{t=1}^{T_{\mathrm{X}}} \boldsymbol{\psi}_{\mathrm{C}}(\mathbf{x}_t) \boldsymbol{\psi}_{\mathrm{C}}(\mathbf{x}_t)^{\mathrm{T}} \end{aligned}$$

En faisant l'approximation

$$\mathbf{R}_{\mathrm{X}}^{*} \approx \mathbf{R}_{\mathrm{B}}^{*} = \tfrac{1}{N} \sum_{i=1}^{N} \boldsymbol{\psi}_{\mathrm{C}}(\mathbf{b}_i) \boldsymbol{\psi}_{\mathrm{C}}(\mathbf{b}_i)^{\mathrm{T}}$$

on retrouve un score qui vérifie les conditions de Mercer et qui ainsi peut faire guise de noyau :

$$\mathrm{score}^{*}(\mathbf{Y}|\mathbf{X}) = \mathrm{score}^{*}(\mathbf{X}|\mathbf{Y}) = \left(\tfrac{1}{T_{\mathrm{X}}} \sum_{t=1}^{T_{\mathrm{X}}} \boldsymbol{\psi}_{\mathrm{C}}(\mathbf{x}_t) \right)^{\mathrm{T}} \mathbf{R}_{\mathrm{B}}^{*\,\dagger} \left(\tfrac{1}{T_{\mathrm{Y}}} \sum_{s=1}^{T_{\mathrm{Y}}} \boldsymbol{\psi}_{\mathrm{C}}(\mathbf{y}_s) \right) \tag{4.22}$$

Notons que dans le cas où $\mathbf{C} = \mathbf{B}$, la matrice $\mathbf{R}_{\mathrm{B}}^{*}$ est égale à $\frac{1}{N}{\mathbf{K}_{\mathrm{B}}}^2$, où $\mathbf{K}_{\mathrm{B}}$ est la matrice de Gram sur les vecteurs du Monde définie en (4.12).

Un tel développement théorique est maladroit à cause des approximation faites *ad hoc*. Ainsi formulée, la généralisation du noyau GLDS ne fournit pas de critère clairement justifié pour le choix du dictionnaire $\mathcal{C}$ qui permet de réduire la complexité. C'est pourquoi à partir de la section suivante nous proposons une autre interprétation du noyau GLDS qui amènera par la suite à introduire de manière naturelle une forme analogue à (4.22) [Louradour et al., 2006b, Louradour et al., 2006a].

4.2 Généralisation du noyau GLDS

Notations

A partir de maintenant, nous considérons une généralisation *directe* du noyau GLDS (4.1) pour une *expansion* $\boldsymbol{\phi}$ quelconque, défini sur un type de données quelconque. Nous partons de la forme finale du noyau et nous expliquerons plus tard comment cette forme peut être sujette à d'autres interprétations que celle présentée en §4.1.2. Dans la suite du chapitre, nous noterons :

- $\mathbb{X}$ l'espace d'entrée.
- $\mathcal{X}$, $\mathcal{Y}$ les séquences de données d'entrée, de longueurs variables.
- x, y (lettres normales) les données qui constituent les séquences, à valeurs dans $\mathbb{X}$.
- $\mathcal{B} = \{\mathrm{b}_i \in \mathbb{X},\ i = 1 \cdots N\}$ un corpus de données non étiquetées (corpus du Monde).
- $\boldsymbol{\phi} : \mathbb{X} \to \mathbb{R}^D$ une *expansion* dans un *Feature Space* de taille $D \leq +\infty$, définissant un noyau de Mercer sur $(\mathbb{X}\times\mathbb{X})$, $k(\mathrm{x},\mathrm{y}) = \boldsymbol{\phi}(\mathrm{x})^{\mathrm{T}}\boldsymbol{\phi}(\mathrm{y})$.

Pour simplifier la lecture, les données manipulées (x, y, b_i) seront appelées "vecteurs" étant donné qu'elles seront vectorielles dans l'application visée ($\mathbb{X} = \mathbb{R}^d$). Toutefois, soulignons que tous les développements théoriques qui suivent peuvent être appliqués à un espace d'entrée $\mathbb{X}$ quelconque. Les données traitées peuvent être structurées, numériques et/ou symboliques.

4.2.1 Noyaux FSNS

Nous commençons ici par poser les définitions des noyaux auxquels nous nous intéressons dans notre étude. Ils sont basés sur une normalisation dans le *Feature Space* faisant intervenir la matrice des seconds moments empiriques des *expansions* $\boldsymbol{\phi}(\cdot)$, estimée sur les vecteurs du Monde. Nous noterons cette matrice définie positive (taille $D\times D$) :

$$\mathbf{S}_{\mathcal{B}} = \tfrac{1}{N}\sum_{i=1}^{N} \boldsymbol{\phi}(\mathrm{b}_i)\boldsymbol{\phi}(\mathrm{b}_i)^{\mathrm{T}} \tag{4.23}$$

Définition 5 (Noyaux FSN / "*Feature Space Normalized*" *kernels*).
Avec les notations introduites ci-dessus, le noyau FSN entre deux vecteurs $\{\mathrm{x},\mathrm{y}\} \in \mathbb{X}^2$ est donné par :

$$\widehat{k}(\mathrm{x},\mathrm{y}) = \boldsymbol{\phi}(\mathrm{x})^{\mathrm{T}}\left[\mathbf{S}_{\mathcal{B}} + \varepsilon\mathbf{I}_D\right]^{-1}\boldsymbol{\phi}(\mathrm{y}) \tag{4.24}$$

où $\varepsilon\mathbf{I}_D$ est la matrice identité multipliée par un scalaire positif $0 \leq \varepsilon \ll \mathrm{tr}(\mathbf{S}_{\mathcal{B}})/D$.

Définition 6 (Noyaux FSNS / "*Feature Space Normalized Sequence*" *kernels*).
Le noyau FSNS entre deux séquences $\mathcal{X} = \{\mathrm{x}_t\}_{t=1\cdots T_{\mathrm{X}}}$ et $\mathcal{Y} = \{\mathrm{y}_s\}_{s=1\cdots T_{\mathrm{Y}}}$ est la moyenne des noyaux FSN entre les couples de vecteurs inter-séquences :

$$\begin{aligned}\widehat{\kappa}(\mathcal{X},\mathcal{Y}) &= \tfrac{1}{T_{\mathrm{X}}T_{\mathrm{Y}}}\textstyle\sum_{t=1}^{T_{\mathrm{X}}}\sum_{s=1}^{T_{\mathrm{Y}}}\widehat{k}(\mathrm{x}_t,\mathrm{y}_s)\\ &= \overline{\boldsymbol{\phi}}(\mathcal{X})^{\mathrm{T}}\left[\mathbf{S}_{\mathcal{B}} + \varepsilon\mathbf{I}_D\right]^{-1}\overline{\boldsymbol{\phi}}(\mathcal{Y})\end{aligned} \tag{4.25}$$

où nous utilisons la notation $\overline{\boldsymbol{\phi}}(\mathcal{X}) = \frac{1}{T_{\mathrm{X}}}\sum_{t=1}^{T_{\mathrm{X}}}\boldsymbol{\phi}(\mathrm{x}_t)$.

Le noyau GLDS est bien un noyau FSNS avec l'expansion polynomiale $\boldsymbol{\phi} = \boldsymbol{\phi}^p$ et sans régularisation ($\varepsilon = 0$). Le noyau de Fisher (§3.4.3) peut être aussi vu comme un noyau FSNS (sans régularisation) si l'on constate que la matrice d'Information de Fisher n'est autre que la matrice des seconds moments des *expansions* de Fisher.

Les noyaux FSNS présentent plusieurs caractéristiques importantes :

1. La projection dans un espace à hautes dimensions
2. Le moyennage
3. La normalisation
4. La régularisation

La projection dans un espace à hautes dimensions reflète simplement un des intérêts essentiels des méthodes à noyaux, à savoir : augmenter la complexité de la modélisation pour capturer le pouvoir discriminant des variables d'entrée à travers des phénomènes non-linéaires.

Le moyennage permet de construire explicitement un noyau de Mercer, même si ce n'est certainement pas la manière optimale de combiner l'information des séquences quand les observations ne sont pas indépendantes. Il est aussi un moyen simple de conférer une invariance aux permutations des vecteurs dans les séquences d'entrée, ce qui est un plus pour l'application de vérification du locuteur en mode *Text-Independent*. En effet, nous rappelons que les noyaux adéquats à cette application sont des noyaux entre "paquets de vecteurs", pour lesquels les séquences sont des ensembles non ordonnées, tout comme avec l'approche de référence UBM-GMM. Soulignons que traiter les séquences sans tenir compte de l'ordre ne signifie pas forcément que l'information sur la dynamique du signal d'entrée soit totalement laissée de coté. Dans nos expériences de vérification du locuteur, l'information dynamique à *court terme* est prise en compte dans la phase de pré-traitement, en amont de la modélisation : les vecteurs d'entrée sont constitués des vecteurs cepstraux et de leurs dérivées.

La normalisation mettant en jeu les seconds moments est classique en *Machine Learning* et *Data Mining*. Elle permet d'introduire une invariance aux transformations linéaires (dans le *Feature Space*), et de rendre les algorithmes d'apprentissage par optimisation plus stables (§1.3.4) en évitant les variations trop importantes de certaines composantes du *Feature Space*. Plusieurs matrices de normalisation (autres que l'inverse des seconds moments) sont bien sûr envisageables. Par exemple, [Hatch et al., 2006] préconise de normaliser au moyen de la moyenne de matrices covariances intraclasses estimées sur des locuteurs séparées, ou encore [Xie et al., 2006] préconise d'aller au-delà des seconds moments, et de prendre en compte la kurtosis[16].

La régularisation est nécessaire pour des raisons numériques et statistiques dans les cas où la dimension du *Feature Space* dépasse l'ordre de grandeur du nombre de vecteurs d'apprentissage [Schölkopf et al., 1999]. Dans ce cas en effet, $\mathbf{S}_{\mathcal{B}}$ est généralement non inversible, d'où le besoin de régulariser en ajoutant une constante positive aux termes diagonaux ("bon conditionnement" de la matrice). De plus, la régularisation confère au problème un bon comportement du point de vue statistique : les estimateurs de "*shrinkage*" tel que celui obtenu en ajoutant une pondération de la matrice identité conduisent empiriquement et théoriquement à des erreurs quadratiques moyennes moins élevées [Daniels et Kass, 2001]. Remarquons que l'ajout d'une constante positive aux valeurs diagonales a pour effet d'augmenter les valeurs propres de la matrice de cette constante. Cette assertion a un intérêt pratique pour le choix de la constante,

[16] *kurtosis* : rapport entre le $4^{\text{ème}}$ moment et le carré du 2^{nd} moment.

lorsque l'on connaît la précision numérique des calculs effectués par un algorithme d'inversion. Cette régularisation n'est certes pas la seule alternative [Choi et al., 2006], mais comme nous le verrons plus tard, sa simplicité permet de la manipuler facilement dans les astuces de calcul.

4.2.2 Noyaux FSMS

Afin de permettre une interprétation intuitive de la quantité calculée par les noyaux FSNS, nous nous intéressons maintenant aux noyaux FSNS correspondant à des *expansions* centrées. Dans le cas général, nous remplaçons une *expansion* $\boldsymbol{\phi}$ quelconque par son *expansion* centrée que nous noterons $\widetilde{\boldsymbol{\phi}}$ (le symbole $\widetilde{\ }$ faisant référence à la notion de centrage). L'opération de centrage sur les vecteurs du Monde dans le *Feature Space* est défini par :

$$\widetilde{\boldsymbol{\phi}}(\mathrm{x}) = \boldsymbol{\phi}(\mathrm{x}) - \boldsymbol{\mu}_{\boldsymbol{\Phi}} \tag{4.26}$$

où nous notons

$$\boldsymbol{\mu}_{\boldsymbol{\Phi}} = \tfrac{1}{N} \sum_{i=1}^{N} \boldsymbol{\phi}(\mathrm{b}_i) \tag{4.27}$$

la moyenne des $\boldsymbol{\phi}(\cdot)$ sur les vecteurs du Monde. À cause de la non linéarité de $\boldsymbol{\phi}$, cette *expansion* $\boldsymbol{\mu}_{\boldsymbol{\Phi}}$ n'admet pas nécessairement de pré-image dans l'espace d'entrée ($\mathrm{z} \in \mathbb{X}$ tel que $\boldsymbol{\phi}(\mathrm{z}) = \boldsymbol{\mu}_{\boldsymbol{\Phi}}$) [Mika et al., 1999].

Remplacer $\boldsymbol{\phi}$ par $\widetilde{\boldsymbol{\phi}}$ dans la définition des noyaux FSNS (4.25) revient à remplacer la matrice $\mathbf{S}_{\mathcal{B}}$ des seconds moments par la matrice de covariance ("seconds moments centrés") :

$$\boldsymbol{\Sigma}_{\mathcal{B}} = \tfrac{1}{N} \sum_{i=1}^{N} \widetilde{\boldsymbol{\phi}}(\mathrm{b}_i) \widetilde{\boldsymbol{\phi}}(\mathrm{b}_i)^{\mathrm{T}} = \tfrac{1}{N} \sum_{i=1}^{N} \boldsymbol{\phi}(\mathrm{b}_i) \boldsymbol{\phi}(\mathrm{b}_i)^{\mathrm{T}} - \boldsymbol{\mu}_{\boldsymbol{\Phi}} \boldsymbol{\mu}_{\boldsymbol{\Phi}}{}^{\mathrm{T}} \tag{4.28}$$

Finalement, nous sommes amenés à définir une sous-catégorie de noyaux FSNS qui met en jeu les quantités familières $\boldsymbol{\mu}_{\boldsymbol{\Phi}}$ et $\boldsymbol{\Sigma}_{\mathcal{B}}$. Nous les désignerons par le sigle FSM (*Feature Space Mahalanobis*).

Définition 7 (Noyaux FSMS / "*Feature Space Mahalanobis Sequence*" *kernels*).
Avec les notations introduites ci-dessus, le noyau FSM entre deux vecteurs est donné par :

$$\widetilde{k}(\mathrm{x}, \mathrm{y}) = \big(\boldsymbol{\phi}(\mathrm{x}) - \boldsymbol{\mu}_{\boldsymbol{\Phi}}\big)^{\mathrm{T}} \big[\boldsymbol{\Sigma}_{\mathcal{B}} + \varepsilon \mathbf{I}_D\big]^{-1} \big(\boldsymbol{\phi}(\mathrm{y}) - \boldsymbol{\mu}_{\boldsymbol{\Phi}}\big) \tag{4.29}$$

Le noyau FSMS entre deux séquences est la moyenne des noyaux FSM entre les couples de vecteurs inter-séquences :

$$\widetilde{\kappa}(\mathcal{X}, \mathcal{Y}) = \big(\overline{\boldsymbol{\phi}}(\mathcal{X}) - \boldsymbol{\mu}_{\boldsymbol{\Phi}}\big)^{\mathrm{T}} \big[\boldsymbol{\Sigma}_{\mathcal{B}} + \varepsilon \mathbf{I}_D\big]^{-1} \big(\overline{\boldsymbol{\phi}}(\mathcal{Y}) - \boldsymbol{\mu}_{\boldsymbol{\Phi}}\big) \tag{4.30}$$

4.2.3 Interprétations

Pour simplifier les interprétations qui vont suivre, nous faisons abstraction du terme de régularisation ajouté aux matrices de covariance pour des raisons numériques (cas $\varepsilon = 0$).

Il est facile de vérifier que les noyaux FSMS que nous avons défini induisent une distance de Mahalanobis dans le *Feature Space* :

$$\begin{aligned} \widetilde{d}(\mathcal{X},\mathcal{Y}) &= \sqrt{\widetilde{\kappa}(\mathcal{X},\mathcal{X})^2 - 2\,\widetilde{\kappa}(\mathcal{X},\mathcal{Y}) + \widetilde{\kappa}(\mathcal{Y},\mathcal{Y})^2} \\ &= \sqrt{\left(\overline{\boldsymbol{\Phi}}(\mathcal{X}) - \overline{\boldsymbol{\Phi}}(\mathcal{Y})\right)^{\mathrm{T}} {\boldsymbol{\Sigma}_{\mathcal{B}}}^{-1} \left(\overline{\boldsymbol{\Phi}}(\mathcal{X}) - \overline{\boldsymbol{\Phi}}(\mathcal{Y})\right)} \end{aligned} \tag{4.31}$$

Cette distance entre séquences n'est autre que la moyenne quadratique des distances de Mahalanobis entre *expansions* inter-séquences. Fig.4.2 illustre alors la démarche inhérente au calcul du noyau FSNS centré, qui revient à appliquer une normalisation de Mahalanobis dans le *Feature Space*.

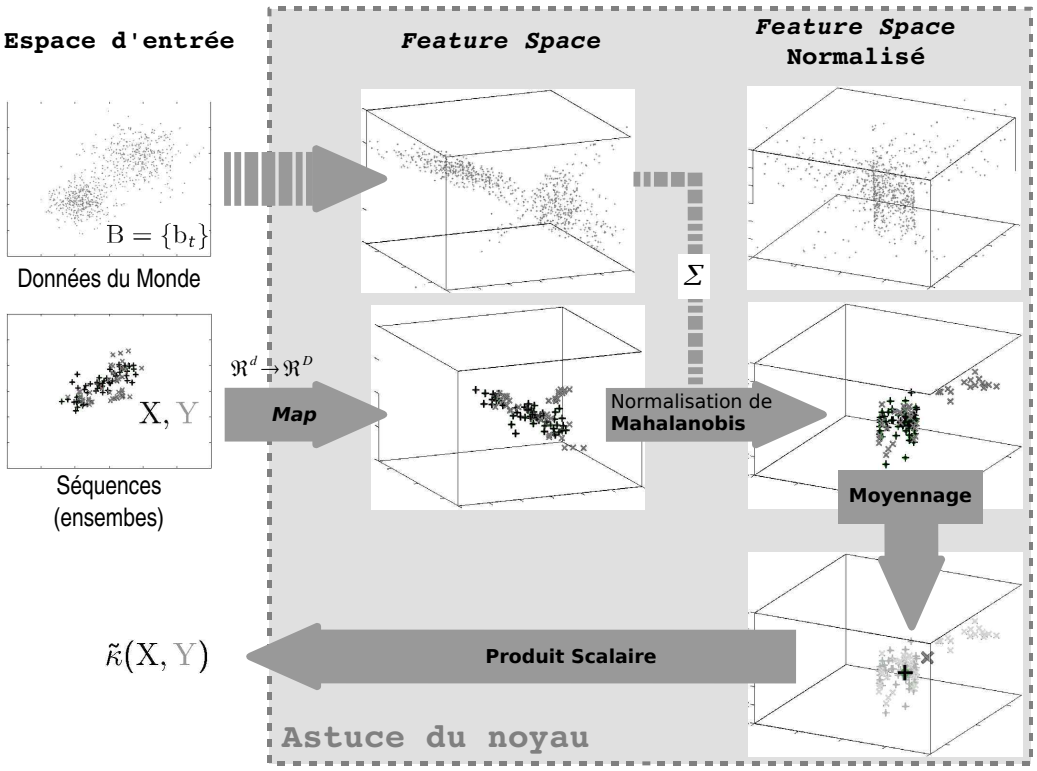


Fig. 4.2 - Noyau FSNS avec matrice de covariance : vue simplifiée de ce qui est calculé (implicitement si l'astuce du noyau est appliquée).

Interprétation probabiliste

Considérons les hypothèses suivantes :

1. Les *expansions* $\{\boldsymbol{\Phi}(\mathrm{x}_t)\}_{t=1\cdots T_{\mathrm{X}}}$ et $\{\boldsymbol{\Phi}(\mathrm{y}_s)\}_{s=1\cdots T_{\mathrm{Y}}}$ sont indépendantes et générées par deux vecteurs aléatoires (relatifs à chacune des séquences $\mathcal{X}, \mathcal{Y}$ respectives).

2. Les vecteurs aléatoires sont issus de distributions Gaussiennes ayant la même matrice de covariance fixée *a priori* (tout comme dans l'Analyse Discriminante de [Fisher, 1936]).

Dans ce contexte, les moyennes empiriques de ces distributions sont respectivement $\overline{\boldsymbol{\Phi}}(\mathcal{X})$ et $\overline{\boldsymbol{\Phi}}(\mathcal{Y})$, et une estimation de la covariance *a priori* est la covariance empirique $\boldsymbol{\Sigma}_{\mathcal{B}}$ sur les vecteurs du monde. Dans ce cas, la distance de séquences définie en (4.31) correspond à la (racine carrée de la) divergence KL symétrique entre densités estimées (§3.3, équation (3.26)). De ce point de vue, le noyau FSMS (4.25) peut être vu comme un noyau entre densités de probabilité dans le *Feature Space*.

L'hypothèse de distribution Gaussienne dans le *Feature Space* est en fait peu réaliste. Nous y faisons référence ici pour illustrer comment les noyaux FSMS mesurent une ressemblance entre les statistiques mesurées sur un grand nombre de caractéristiques au niveau des séquences. Même si l'hypothèse de Gaussianité dans le *Feature Space* a peu de chance d'être vérifiée, elle permet de capturer des structures complexes dans l'espace d'entrée. Les distributions Gaussiennes dans l'espace d'entrée utilisent les statistiques du premier et second ordre sur les variables d'entrée, ce qui est limité lorsque l'on est confronté à des données dont la structure n'est pas linéaire. Les distributions Gaussiennes dans le *Feature Space* utilisent quant à elles des statistiques du premier et second ordre sur des caractéristiques qui typiquement ne sont pas linéaires vis-à-vis des variables d'entrées. De cette manière, elles codent des statistiques d'ordres potentiellement élevés, estimées sur les vecteurs d'entrée. Parmi les travaux qui font l'hypothèse implicite d'une distribution Gaussienne dans le *Feature Space*, on peut citer les généralisations (par l'astuce du noyau) de l'Analyse en Composantes Principales [Schölkopf et al., 1998], de l'Analyse Factorielle Discriminante [Baudat et Anouar, 2000] et de l'Analyse en Composantes Indépendantes [Bach et Jordan, 2002].

L'idée d'utiliser un noyau probabiliste dans le *Feature Space* en supposant des distributions Gaussiennes a aussi été proposée par [Kondor et Jebara, 2003, Zhou et Chellappa, 2006]. Dans ces deux travaux, les séquences $\mathcal{X}$ (resp. $\mathcal{Y}$) sont représentées par des distributions Gaussiennes dans le *Feature Space*, de moyennes $\overline{\boldsymbol{\Phi}}(\mathcal{X})$ (resp. $\overline{\boldsymbol{\Phi}}(\mathcal{Y})$) et de covariance régularisée que nous notons $\boldsymbol{\Sigma}_{\mathrm{X}}$ (resp. $\boldsymbol{\Sigma}_{\mathrm{Y}}$). [Kondor et Jebara, 2003] montrent comment calculer de manière implicite un noyau de Bhattacharyya entre ces deux distributions Gaussiennes. [Zhou et Chellappa, 2006] donnent les expressions de différents noyaux construits à partir de plusieurs mesures de divergences entre les distributions Gaussiennes (§3.3.2, Tab.3.17) dans le *Feature Space*. En pratique, l'estimation de ces noyaux se fait à partir de la matrice de Gram $\mathbf{K}_{\{\mathrm{X,Y}\}}$ contenant les valeurs des noyaux vectoriels entre toutes les paires de vecteurs que l'on peut former à partir des séquences $\{\mathcal{X},\mathcal{Y}\}$. Cette matrice contient les deux matrices de Gram intraséquences $\mathbf{K}_{\mathrm{X}}$ et $\mathbf{K}_{\mathrm{Y}}$ (contenant les valeurs $k(\mathrm{x}_t,\mathrm{x}_s)$ et $k(\mathrm{y}_t,\mathrm{y}_s)$) et la matrice de Gram inter-séquences $\mathbf{K}_{\mathrm{X}\times\mathrm{Y}}$ (contenant les valeurs $k(\mathrm{x}_t,\mathrm{y}_s)$) :

$$\mathbf{K}_{\{\mathrm{X,Y}\}} = \left[\begin{array}{c|c} \mathbf{K}_{\mathrm{X}} & \mathbf{K}_{\mathrm{X}\times\mathrm{Y}} \\ \hline \mathbf{K}_{\mathrm{X}\times\mathrm{Y}}{}^{\mathrm{T}} & \mathbf{K}_{\mathrm{Y}} \end{array}\right]$$

Ces deux approches ne sont pas applicables à la vérification du locuteur à cause de la complexité calculatoire. Elles nécessitent en effet le calcul de $O(T^2)$ valeurs de noyaux vectoriels *à chaque fois* que l'on veut estimer le noyau entre deux séquences de longueur T. Le problème est du même ordre qu'avec les *Local Kernels* (§3.5.1) qui nécessitent le calcul de la matrice de Gram inter-séquences $\mathbf{K}_{\mathrm{X}\times\mathrm{Y}}$ (T^2 valeurs) pour chaque couple de séquences d'entrée.

4.3 Formulation duale

Le problème posé par le noyau GLDS et les noyaux FSNS tels qu'ils ont été formulés dans la section précédente est l'implémentation de l'*expansion* $\boldsymbol{\phi}$ lorsque le *Feature Space* a une dimension très grande ou infinie. Le but de cette section est de montrer comment pallier ce problème en exprimant autrement les noyaux FSNS.

4.3.1 Notions essentielles

Nous commençons par présenter quelques notions qui vont nous permettre par la suite d'écrire les noyaux FSNS sous une autre forme où l'*expansion* $\boldsymbol{\phi}$ n'apparaîtra pas de manière explicite.

Espace engendré par les *expansions* $\boldsymbol{\phi}$

Nous serons par moment amenés à faire appel à une hypothèse, formulée dans cet encadré.

Hypothèse 1 (Représentativité d'un ensemble).
Étant donné une *expansion* $\boldsymbol{\phi}$ sur $\mathbb{X}$ définissant un noyau de Mercer $k(\cdot,\cdot) = \boldsymbol{\phi}(\cdot)^{\mathrm{T}}\boldsymbol{\phi}(\cdot)$, on dit que l'ensemble $\mathcal{B} = \{\mathrm{b}_i \in \mathbb{X},\ i = 1 \cdots N\}$ vérifie "l'hypothèse de représentativité" pour $\mathrm{x} \in \mathbb{X}$ si l'*expansion* de x appartient à l'espace engendré par les *expansions* $\left\{\boldsymbol{\phi}(\mathrm{b}_i)\right\}_{i=1\cdots N}$.
Autrement dit, l'hypothèse est vérifiée pour un vecteur x tant que l'on peut supposer que :

$$\exists\ \omega_1, \cdots, \omega_D \in \mathbb{R}\ \mathrm{tq} \quad \boldsymbol{\phi}(\mathrm{x}) = \sum_{i=1}^{N} \omega_i\ \boldsymbol{\phi}(\mathrm{b}_i) \tag{4.32}$$

Plusieurs travaux [Hastie et Tibshirani, 1990, Mika, 1998, Tsuda, 1999] sont basés sur une telle hypothèse pour calculer une approximation des fonctions du RKHS (définition 1) générée par un noyau k. Typiquement, ces approximations sont faites à partir des fonctions $k(\mathrm{b}_i, \cdot)$. D'après le théorème 1 des représentants, ces fonctions suffisent à exprimer les solutions d'une grande famille de problèmes d'optimisation sur les données $\{\mathrm{b}_i\}$. En prenant l'exemple des SVMs, une fois qu'un modèle a été entraîné sur des données $\{\mathrm{b}_i\}$, alors l'application de ce modèle sur des données quelconques se fait en manipulant les fonctions scalaires $k(\mathrm{b}_i, \cdot)$. Autrement dit, utiliser un modèle SVM appris sur un $\mathcal{B}$ revient à manipuler les données de test comme si l'hypothèse 1 de représentativité était vérifiée sur $\mathcal{B}$.

De manière générale, l'hypothèse 1 peut être facilement respectée pour un ensemble $\mathcal{B}$ de taille $N \geq D$. Mais on rappelle que la dimension D du *Feature Space* peut être infinie, pour un noyau RBF Gaussien par exemple. Dans les cas où l'hypothèse 1 n'est pas respectée, les calculs restent valables en remplaçant l'*expansion* $\boldsymbol{\phi}(\mathrm{x})$ par la *projection orthogonale de l'expansion sur l'espace engendré par les expansions* $\{\boldsymbol{\phi}(\mathrm{b}_i)\}$. Nous noterons alors $\boldsymbol{\phi}_{|\mathcal{B}}(\mathrm{x})$ une telle projection :

$$\boldsymbol{\phi}_{|\mathcal{B}}(\mathrm{x}) = \arg\min_{\boldsymbol{\phi}_{\mathrm{B}}} \left\{ \left\| \boldsymbol{\phi}_{\mathrm{B}} - \boldsymbol{\phi}(\mathrm{x}) \right\|_2 ,\ \ \boldsymbol{\phi}_{\mathrm{B}} = \textstyle\sum_{i=1}^{N} \omega_i\ \boldsymbol{\phi}(\mathrm{b}_i) \right\}$$

Cette projection est aussi suggérée par [Kondor et Jebara, 2003]. Dans ce travail, la quantité estimée de manière implicite à travers le noyau est un noyau de Bhattacharyya entre Gaussiennes dans le *Feature Space* (détails dans [Kondor, 2005]) :

$$\kappa(\mathcal{X},\mathcal{Y}) = \int_{\mathbb{W}} \sqrt{\mathcal{N}_{\mathbb{W}}(\ \mathrm{z}\ |\overline{\boldsymbol{\Phi}}(\mathcal{X}), \boldsymbol{\Sigma}_{\mathrm{X}})}\ \sqrt{\mathcal{N}_{\mathbb{W}}(\ \mathrm{z}\ |\overline{\boldsymbol{\Phi}}(\mathcal{Y}), \boldsymbol{\Sigma}_{\mathrm{Y}})}\ \mathrm{dz}$$

où $\mathbb{W}$ est l'espace engendré par les *expansions* $\left\{\boldsymbol{\phi}(\mathrm{x}_1), \cdots, \boldsymbol{\phi}(\mathrm{x}_{T_\mathrm{X}}), \boldsymbol{\phi}(\mathrm{y}_1), \cdots, \boldsymbol{\phi}(\mathrm{y}_{T_\mathrm{Y}})\right\}$ (union des séquences $\mathcal{X}$ et $\mathcal{Y}$). La projection sur $\mathbb{W}$ permet de bien définir les Gaussiennes $\mathcal{N}_{\mathbb{W}}(\cdot|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, qui ne peuvent pas être des distributions sur espace de dimension infini. Pour les cas où la dimension du *Feature Space* est supérieure au nombre de données sur lesquelles sont estimées les statistiques $\boldsymbol{\mu}$ et $\boldsymbol{\Sigma}$, on parle techniquement de "processus Gaussien" [Rasmussen et Williams, 2006]. La fonction $\mathcal{N}_{\mathbb{W}}$ désigne alors la projection de la Gaussienne sur le sous-espace du RKHS engendré par les $\left\{k(\mathrm{x}_1,\cdot), \cdots, k(\mathrm{x}_{T_\mathrm{X}},\cdot), k(\mathrm{y}_1,\cdot), \cdots, k(\mathrm{y}_{T_\mathrm{Y}},\cdot)\right\}$.

Expansion empirique sur un ensemble

[Schölkopf et al., 1999] montrent que l'on peut approcher l'estimation d'un noyau à partir des fonctions $k(\mathrm{b}_i,\cdot)$. Ils introduisent pour cela une notion qui jouera un rôle central dans nos calculs : le "*empirical kernel map*".

Définition 8 (*expansion* empirique).
Étant donné un ensemble $\mathcal{B} = \{\mathrm{b}_i \in \mathbb{X},\ i = 1 \cdots N\}$ et un noyau de Mercer $k(\cdot,\cdot) = \boldsymbol{\phi}(\cdot)^{\mathrm{T}}\boldsymbol{\phi}(\cdot)$ défini sur $\mathbb{X}\times\mathbb{X}$, "l'*expansion* empirique" (sur $\mathcal{B}$) est l'application :

$$\boldsymbol{\psi}_{\mathcal{B}} \begin{cases} \mathbb{X} & \rightarrow & \mathbb{R}^N \\ \mathrm{x} & \mapsto & \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{x}) = \begin{bmatrix} \boldsymbol{\phi}(\mathrm{b}_1)^{\mathrm{T}}\boldsymbol{\phi}(\mathrm{x}) \\ \vdots \\ \boldsymbol{\phi}(\mathrm{b}_N)^{\mathrm{T}}\boldsymbol{\phi}(\mathrm{x}) \end{bmatrix} = \begin{bmatrix} k(\mathrm{b}_1,\mathrm{x}) \\ \vdots \\ k(\mathrm{b}_N,\mathrm{x}) \end{bmatrix} \end{cases} \tag{4.33}$$

Le noyau k entre les vecteurs d'un ensemble $\mathcal{B}$ peut s'exprimer comme un noyau linéaire généralisé entre *expansions* empiriques sur $\mathcal{B}$, d'après :

$$\forall\ \{\mathrm{b}_i, \mathrm{b}_j\} \in \mathcal{B}^2\ , \qquad k(\mathrm{b}_i,\mathrm{b}_j) = \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{b}_i)^{\mathrm{T}}\mathbf{K}^{\dagger}\boldsymbol{\psi}_{\mathcal{B}}(\mathrm{b}_j)$$

où l'on note par simplicité $\mathbf{K}$ (au lieu de $\mathbf{K}_{\mathrm{B}}$) la matrice de Gram sur $\mathcal{B}$. Dans cette équation, $\mathbf{K}^{\dagger}$ désigne la pseudo-inverse de $\mathbf{K}$ (*cf.* annexe A.1.2), qui est l'inverse $\mathbf{K}^{\dagger} = \mathbf{K}^{-1}$ si $\mathbf{K}$ est inversible.

On peut généraliser cette dernière égalité à un couple de vecteurs $\{\mathrm{x},\mathrm{y}\}$ quelconque si l'ensemble $\mathcal{B}$ vérifie l'hypothèse 1 de représentativité pour au moins un vecteur du couple. Le lemme suivant clarifie le contexte où la formule peut s'appliquer.

.../...

Lemme 1.

Si un ensemble $\mathcal{B} = \{\mathrm{b}_i\}_{i=1\cdots N}$ *vérifie l'hypothèse de représentativité pour au moins un vecteur du couple* $\{\mathrm{x}, \mathrm{y}\}$*, alors le noyau entre* x *et* y *peut s'exprimer comme un noyau linéaire généralisé entre les expansions empiriques :*

$$\forall\ \{\mathrm{x}, \mathrm{y}\} \in \mathbb{X}^2\ , \qquad k(\mathrm{x}, \mathrm{y}) = \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{x})^{\mathrm{T}} \mathbf{K}^{\dagger} \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{y}) \tag{4.34}$$

où $\mathbf{K}$ *est la matrice de Gram sur* $\mathcal{B}$*, et* $\mathbf{K}^{\dagger}$ *sa pseudo-inverse.*

Si l'hypothèse de représentativité n'est pas vérifiée, alors on peut toujours écrire :

$$\forall\ \{\mathrm{x}, \mathrm{y}\} \in \mathbb{X}^2\ , \qquad \boldsymbol{\phi}_{|\mathcal{B}}(\mathrm{x})^{\mathrm{T}} \boldsymbol{\phi}_{|\mathcal{B}}(\mathrm{y}) = \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{x})^{\mathrm{T}} \mathbf{K}^{\dagger} \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{y}) \tag{4.35}$$

où $\boldsymbol{\phi}_{|\mathcal{B}}$ *désigne le projeté orthogonal de l'expansion* $\boldsymbol{\phi}$ *sur l'espace engendré par les* $\{\boldsymbol{\phi}(\mathrm{b}_i)\}$*.*

Preuve :

Afin de simplifier les écritures matricielles dans toutes les démonstrations qui suivent, nous notons $\boldsymbol{\Phi}$ la matrice $D \times N$ des *expansions* des vecteurs du Monde. Cette matrice vérifie :

$$\boldsymbol{\Phi} = [\boldsymbol{\phi}(\mathrm{b}_1), \cdots, \boldsymbol{\phi}(\mathrm{b}_N)] \tag{4.36}$$

$$\boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi} = \mathbf{K} = [\boldsymbol{\psi}_{\mathcal{B}}(\mathrm{b}_1) \cdots \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{b}_N)] \tag{4.37}$$

Les projections orthogonales $\boldsymbol{\phi}_{|\mathcal{B}}(\mathrm{x})$ et $\boldsymbol{\phi}_{|\mathcal{B}}(\mathrm{y})$ appartiennent au sous-espace du *Feature Space* engendré par les $\{\boldsymbol{\phi}(\mathrm{b}_i)\}$. Il existe donc deux colonnes $\mathbf{w}_{\mathrm{x}}$ et $\mathbf{w}_{\mathrm{y}}$ de N scalaires telles que :

$$\begin{array}{cc} \boldsymbol{\phi}_{|\mathcal{B}}(\mathrm{x}) = \boldsymbol{\Phi}\mathbf{w}_{\mathrm{x}} & \boldsymbol{\phi}_{|\mathcal{B}}(\mathrm{y}) = \boldsymbol{\Phi}\mathbf{w}_{\mathrm{y}} \\ \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{x}) = \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\phi}(\mathrm{x}) = \mathbf{K}\mathbf{w}_{\mathrm{x}} & \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{y}) = \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\phi}(\mathrm{y}) = \mathbf{K}\mathbf{w}_{\mathrm{y}} \end{array}$$

En considérant que $\mathbf{K}$ est symétrique, on peut facilement prouver l'égalité :

$$\begin{aligned} \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{x})^{\mathrm{T}} \mathbf{K}^{\dagger} \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{y}) &= \mathbf{w}_{\mathrm{x}}^{\mathrm{T}} \mathbf{K}\ \mathbf{K}^{\dagger}\ \mathbf{K}\mathbf{w}_{\mathrm{y}} \\ &= \mathbf{w}_{\mathrm{x}}^{\mathrm{T}} \mathbf{K}\mathbf{w}_{\mathrm{y}} \\ &= \boldsymbol{\phi}_{|\mathcal{B}}(\mathrm{x})^{\mathrm{T}} \boldsymbol{\phi}_{|\mathcal{B}}(\mathrm{y}) \end{aligned}$$

Dans les cas où l'hypothèse de représentativité est respectée pour x par exemple, alors on peut écrire les égalités : $\boldsymbol{\phi}(\mathrm{x}) = \boldsymbol{\phi}_{|\mathcal{B}}(\mathrm{x})$ et $\boldsymbol{\phi}(\mathrm{x})^{\mathrm{T}} \boldsymbol{\phi}(\mathrm{y}) = \boldsymbol{\phi}(\mathrm{x})^{\mathrm{T}} \boldsymbol{\phi}_{|\mathrm{x}}(\mathrm{y}) = \boldsymbol{\phi}_{|\mathcal{B}}(\mathrm{x})^{\mathrm{T}} \boldsymbol{\phi}_{|\mathcal{B}}(\mathrm{y})$.

■

Le lemme 1 peut être généralisé pour les séquences par simple linéarité. Avec l'hypothèse de représentativité, il s'écrit :

$$\overline{\boldsymbol{\phi}}(\mathcal{X})^{\mathrm{T}} \overline{\boldsymbol{\phi}}(\mathcal{Y}) = \overline{\boldsymbol{\psi}_{\mathcal{B}}}(\mathcal{X})^{\mathrm{T}}\ \mathbf{K}^{\dagger}\ \overline{\boldsymbol{\psi}_{\mathcal{B}}}(\mathcal{Y}) \tag{4.38}$$

où $\overline{\boldsymbol{\psi}_{\mathcal{B}}}$ désigne l'*expansion* empirique moyenne :

$$\overline{\boldsymbol{\psi}_{\mathcal{B}}}(\mathcal{X}) = \tfrac{1}{T_{\mathrm{X}}} \sum_{t=1}^{T_{\mathrm{X}}} \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{x}_t) = \begin{bmatrix} \frac{1}{T_{\mathrm{X}}} \sum_t k(\mathrm{b}_1, \mathrm{x}_t) \\ \vdots \\ \frac{1}{T_{\mathrm{X}}} \sum_t k(\mathrm{b}_N, \mathrm{x}_t) \end{bmatrix} \tag{4.39}$$

Forme duale et complexité

Les termes de droite des équations (4.35) et (4.38) seront qualifiés dans ce qui suit de "forme duale" par analogie à la formulation du critère d'apprentissage des SVMs (§1.3.2). En effet le critère SVM à optimiser, qui met en jeu des vecteurs d'apprentissages étiquetées $\{\mathrm{a}_i, \boldsymbol{\ell}_i\}_{i=1\cdots N}$, peut s'exprimer dans une forme duale grâce au théorème des représentants :

$$\underbrace{\textit{Problème "primal"}}_{\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \tau\big(\boldsymbol{\theta}, \{\boldsymbol{\ell}_i \boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{\Phi}(\mathrm{a}_i)\}\big)} \quad \longleftrightarrow \quad \underbrace{\textit{Problème "dual"}}_{\boldsymbol{\alpha}^* = \arg\max_{\boldsymbol{\alpha}} \tau_D\big(\boldsymbol{\alpha}, \{\boldsymbol{\Lambda}_i \boldsymbol{\psi}_{\mathcal{A}}(\mathrm{a}_i)\}\big)}$$

(sous contraintes)

$$\text{avec} \begin{cases} \Lambda_{i,j} = \alpha_i \alpha_j \boldsymbol{\ell}_i \boldsymbol{\ell}_j \\ \boldsymbol{\psi}_{\mathcal{A}}(\mathrm{x}) = \big[\begin{array}{ccc} \boldsymbol{\Phi}(\mathrm{a}_1)^{\mathrm{T}} \boldsymbol{\Phi}(\mathrm{x}) & \cdots & \boldsymbol{\Phi}(\mathrm{a}_N)^{\mathrm{T}} \boldsymbol{\Phi}(\mathrm{x}) \end{array} \big]^{\mathrm{T}} \end{cases}$$

La forme duale introduite par le lemme 1 permet d'éviter le calcul explicite de l'*expansion* $\boldsymbol{\Phi}$ de taille D, mais requiert par contre le calcul de l'*expansion* empirique de taille N. Elle peut présenter un intérêt dans les cas où D est très grand ou infini, et convient dans des problèmes où l'on dispose de peu de vecteurs d'apprentissage (données coûteuses à collecter et/ou étiqueter). Ce n'est toutefois pas le cas du traitement de la parole, où d'importantes bases de données sont accessibles. La tâche de vérification du locuteur est un "*large-scale problem*" pour lequel le nombre N de vecteurs du Monde habituellement utilisé pour entraîner les systèmes performants est de l'ordre du million. Une telle quantité d'information est rédhibitoire pour le calcul du noyau, dont la complexité est de l'ordre de $O(N^2)$. Le but de la prochaine section sera de remédier à ce problème.

Notons que la forme duale du noyau revient tout comme la forme primale à un noyau linéaire généralisé entre *expansions* de séquences. Conserver une telle forme présente plusieurs intérêts pratiques cités en §3.5.1. Dans la suite de cette section, nous donnons les formes duales des noyaux FSNS. Autrement dit, nous dérivons le lemme 1 en rajoutant une matrice de normalisation impliquant l'inverse d'une matrice de seconds moments $\mathbf{S}_{\mathcal{B}} \neq \mathbf{I}_D$ et un terme de régularisation $\varepsilon \neq 0$.

4.3.2 Forme duale des noyaux FSNS

Proposition 1.

Supposons que le corpus (du Monde) $\mathcal{B} = \{\mathrm{b}_i\}$ vérifie l'hypothèse de représentativité pour au moins un vecteur du couple $\{\mathrm{x}, \mathrm{y}\}$. Alors *le noyau FSN (définition 5) peut s'écrire comme un noyau linéaire généralisé entre expansions empiriques (4.33) :*

$$\widehat{k}(\mathrm{x},\mathrm{y}) = \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{x})^{\mathrm{T}} \left[\tfrac{1}{N}\mathbf{K}^2 + \varepsilon\mathbf{K}\right]^{-1} \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{y}) \tag{4.40}$$

où $\mathbf{K}$ *est la matrice de Gram sur* $\mathcal{B}$*. Dans le cas où cette matrice n'est pas inversible, il suffit de prendre la pseudo-inverse au lieu de l'inverse du terme central.*

L'extension au noyau FSNS s'obtient par simple linéarité :

$$\widehat{\kappa}(\mathcal{X},\mathcal{Y}) = \overline{\boldsymbol{\psi}_{\mathcal{B}}}(\mathcal{X})^{\mathrm{T}} \left[\tfrac{1}{N}\mathbf{K}^2 + \varepsilon\mathbf{K}\right]^{-1} \overline{\boldsymbol{\psi}_{\mathcal{B}}}(\mathcal{Y}) \tag{4.41}$$

où $\overline{\boldsymbol{\psi}_{\mathcal{B}}}$ *est l'expansion empirique moyenne définie en (4.39), et où il suffit de prendre la pseudo-inverse au lieu de l'inverse du terme central si la matrice de Gram n'est pas inversible.*

Preuve :

La démonstration est ici faite dans le cas général où l'on n'a pas de garantie sur l'inversibilité de $\mathbf{K}$. Si $\mathbf{K}$ est inversible, les calculs qui suivent peuvent être simplifiés en remplaçant $\mathbf{K}^\dagger$ par $\mathbf{K}^{-1}$ (sachant que l'inverse vérifie la propriété $\mathbf{K}^{-1}\mathbf{K} = \mathbf{I}_N$, plus forte que $\mathbf{K}\mathbf{K}^\dagger\mathbf{K} = \mathbf{K}$).

En considérant la matrice $\boldsymbol{\Phi}$ (4.36) des *expansions* des vecteurs du Monde, le lemme 1 nous permet d'écrire les relations :

$$\boldsymbol{\phi}(\mathrm{x})^{\mathrm{T}}\boldsymbol{\phi}(\mathrm{y}) = \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{x})^{\mathrm{T}}\mathbf{K}^\dagger\boldsymbol{\psi}_{\mathcal{B}}(\mathrm{y}) \tag{4.42}$$

$$\boldsymbol{\phi}(\mathrm{x})^{\mathrm{T}}\boldsymbol{\Phi} = \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{x})^{\mathrm{T}}\mathbf{K}^\dagger\mathbf{K} \tag{4.43}$$

$$\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} = \mathbf{K} \text{ (trivial)} \tag{4.44}$$

Soulignons que seule la première équation (4.42) fait appel à l'hypothèse de représentativité des données du Monde. Dans (4.43) par exemple, les composantes de $\boldsymbol{\phi}(\mathrm{x})^{\mathrm{T}}\boldsymbol{\Phi}$ peuvent s'écrire $\boldsymbol{\phi}(\mathrm{x})^{\mathrm{T}}\boldsymbol{\phi}(\mathrm{b}_i) = \boldsymbol{\phi}_{|\mathcal{B}}(\mathrm{x})^{\mathrm{T}}\boldsymbol{\phi}_{|\mathcal{B}}(\mathrm{b}_i)$ (les valeurs restent inchangés par projection sur les $\boldsymbol{\phi}(\mathrm{b}_i)$).

La matrice des second moments régularisée peut quant à elle s'écrire :

$$\mathbf{S}_{\mathcal{B}} + \varepsilon\mathbf{I}_D = \tfrac{1}{N}\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}} + \varepsilon\mathbf{I}_D$$

Pour l'inversion de ce terme de normalisation intervenant dans le calcul du noyau FSN (4.47), nous considérons l'identité matricielle de [Woodbury, 1950]. D'après ce lemme, quelques soient deux matrices inversibles $\mathbf{P}$ et $\mathbf{Q}$ de tailles respectives $(D\times D)$ et $(N\times N)$, et une matrice $\mathbf{M}$ de taille $(D\times N)$:

$$[\mathbf{P} + \mathbf{M}\mathbf{Q}\mathbf{M}^{\mathrm{T}}]^{-1} = \mathbf{P}^{-1} - \mathbf{P}^{-1}\mathbf{M}\left(\mathbf{Q}^{-1} + \mathbf{M}^{\mathrm{T}}\mathbf{P}^{-1}\mathbf{M}\right)^{-1}\mathbf{M}^{\mathrm{T}}\mathbf{P}^{-1} \tag{4.45}$$

.../...

En prenant ici $\mathbf{P} = \varepsilon \mathbf{I}_D$, $\mathbf{Q} = \frac{1}{N}\mathbf{I}_N$ et $\mathbf{M} = \mathbf{\Phi}$, on peut écrire le noyau sous une nouvelle forme :

$$\widehat{k}(\mathrm{x},\mathrm{y}) = \boldsymbol{\phi}(\mathrm{x})^{\mathrm{T}} \Big[\varepsilon^{-1}\mathbf{I}_D - \varepsilon^{-1}\mathbf{\Phi} \left(N\varepsilon \mathbf{I}_N + \mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi} \right)^{-1} \mathbf{\Phi}^{\mathrm{T}} \Big] \boldsymbol{\phi}(\mathrm{y})$$

Si l'on applique maintenant les relations (4.42), (4.43) et (4.44) on peut faire apparaître l'*expansion* empirique :

$$\widehat{k}(\mathrm{x},\mathrm{y}) = \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{x})^{\mathrm{T}} \Big[\varepsilon^{-1}\mathbf{K}^{\dagger} - \varepsilon^{-1}\mathbf{K}^{\dagger}\mathbf{K} \left(N\varepsilon \mathbf{I}_N + \mathbf{K} \right)^{-1} \mathbf{K}\mathbf{K}^{\dagger} \Big] \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{y})$$

Pour simplifier cette expression, il faut utiliser la généralisation de l'identité de Woodbury au cas des pseudo-inverses, formulée par [Ogawa, 1988], qui s'écrit pour une matrice carrée $\mathbf{P}$ qui n'est pas nécessairement inversible :

$$[\mathbf{P} + \mathbf{M}\mathbf{Q}\mathbf{M}^{\mathrm{T}}]^{\dagger} = \mathbf{P}^{\dagger} - \mathbf{P}^{\dagger}\mathbf{M} \left(\mathbf{Q}^{-1} + \mathbf{M}^{\mathrm{T}}\mathbf{P}^{\dagger}\mathbf{M} \right)^{-1} \mathbf{M}^{\mathrm{T}}\mathbf{P}^{\dagger} \tag{4.46}$$

En identifiant $\mathbf{P} = \varepsilon\mathbf{K}$, $\mathbf{Q} = \frac{1}{N}\mathbf{I}_N$ et $\mathbf{M} = \mathbf{K}$, on obtient finalement :

$$\widehat{k}(\mathrm{x},\mathrm{y}) = \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{x})^{\mathrm{T}} \left[\tfrac{1}{N}\mathbf{K}^2 + \varepsilon\mathbf{K} \right]^{\dagger} \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{y})$$

■

Précisons que la matrice de Gram $\mathbf{K}$ n'est pas inversible dans les cas où le nombre N de vecteurs du Monde est inférieur à la dimension D du *Feature Space*, et dans les cas où il y a une redondance dans les vecteurs du Monde. Dans le premier cas ($D < N$), la forme duale ne présente pas d'intérêt étant donné qu'elle implique des complexités de calcul plus élevées. Nous rappelons alors que nous cherchons une forme duale pour les cas où le *Feature Space* est de dimension très grande. En particulier, avec un noyau RBF Gaussien, la dimension du *Feature Space* est infinie, et la matrice de Gram sur $\mathcal{B}$ est inversible du moment que les vecteurs de $\mathcal{B}$ sont distincts deux à deux (condition nécessaire et suffisante) [Micchelli, 1986a].

Rôle de l'hypothèse de représentativité

Avant de discuter sur l'hypothèse de représentativité, qui peut être peu ou prou contraignante selon le contexte applicatif, soulignons que cette hypothèse n'est utilisée dans la démonstration de la proposition 1 que pour la factorisation des termes provenant de la régularisation. Pour $\varepsilon = 0$, la forme duale donnée dans la proposition 1 reste valable sans faire appel à l'hypothèse 1.

Proposition 2.

Sans le terme de régularisation ($\varepsilon = 0$) les noyaux FSNS s'écrivent de manière exacte

$$\begin{aligned} \widehat{k}(\mathrm{x},\mathrm{y}) = \boldsymbol{\phi}(\mathrm{x})^{\mathrm{T}}\mathbf{S}_{\mathcal{B}}{}^{\dagger}\boldsymbol{\phi}(\mathrm{y}) &= \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{x})^{\mathrm{T}} \left(\tfrac{1}{N}\mathbf{K}^2 \right)^{\dagger} \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{y}) \\ \widehat{\kappa}(\mathcal{X},\mathcal{Y}) = \overline{\boldsymbol{\phi}}(\mathcal{X})^{\mathrm{T}}\mathbf{S}_{\mathcal{B}}{}^{\dagger}\overline{\boldsymbol{\phi}}(\mathcal{Y}) &= \overline{\boldsymbol{\psi}_{\mathcal{B}}}(\mathcal{X})^{\mathrm{T}} \left(\tfrac{1}{N}\mathbf{K}^2 \right)^{\dagger} \overline{\boldsymbol{\psi}_{\mathcal{B}}}(\mathcal{Y}) \end{aligned}$$

Preuve :

La démonstration est quelque peu différente de la preuve pour la proposition 1, qui était basée sur un ε non nul. Nous considérons ici la Décomposition en Valeur Singulière (SVD) *mince* (*cf.* annexe A.1.1) de la matrice $\mathbf{\Phi}$ des *expansions* des vecteurs du Monde :

$$\mathbf{\Phi} = \mathbf{U}_r \mathbf{D}_r \mathbf{V}_r{}^{\mathrm{T}}$$

Si l'on note $r \leq \min\{D, N\}$ désigne le rang de $\mathbf{\Phi}$, alors $\mathbf{U}_r$ et $\mathbf{V}_r$ sont des matrices orthogonales de tailles respectives $D \times r$ et $N \times r$ ($\mathbf{U}_r{}^{\mathrm{T}}\mathbf{U}_r = \mathbf{I}_r = \mathbf{V}_r{}^{\mathrm{T}}\mathbf{V}_r$), et $\mathbf{D}_r$ est une matrice diagonale contenant les valeurs singulières. Cette décomposition permet d'écrire la matrice des seconds moments $\mathbf{S}_{\mathcal{B}} = \frac{1}{N}\mathbf{U}_r \mathbf{D}_r{}^2 \mathbf{U}_r{}^{\mathrm{T}}$ et la matrice de Gram $\mathbf{K} = \mathbf{V}_r \mathbf{D}_r{}^2 \mathbf{V}_r{}^{\mathrm{T}}$. On peut ainsi réécrire la pseudo-inversion :

$$\mathbf{S}_{\mathcal{B}}{}^{\dagger} \;=\; {}_N\mathbf{U}_r \mathbf{D}_r{}^{-2} \mathbf{U}_r{}^{\mathrm{T}} \;=\; {}_N\mathbf{\Phi}\mathbf{V}_r \mathbf{D}_r{}^{-4} \mathbf{V}_r{}^{\mathrm{T}} \mathbf{\Phi}^{\mathrm{T}} \;=\; {}_N\mathbf{\Phi} \left(\mathbf{K}^{\dagger}\right)^2 \mathbf{\Phi}^{\mathrm{T}}$$

On retombe finalement sur le résultat à montrer en identifiant l'*expansion* empirique $\boldsymbol{\psi}_{\mathcal{B}}(\mathrm{x}) = \mathbf{\Phi}^{\mathrm{T}}\boldsymbol{\phi}(\mathrm{x})$.

■

Pour tenir compte du terme de régularisation et garder une forme factorisée (proposition 1), il faut faire appel à l'hypothèse de représentativité. Cette hypothèse n'est aucunement contraignante du moment que les données du Monde $\mathcal{B}$ incluent les données d'apprentissage $\mathcal{A}$ des modèles. En effet, selon le théorème 1 des représentants, les modèles à noyaux appris par optimisation d'un risque régularisé manipulent les vecteurs d'entrée dans le sous-espace du *Feature Space* engendré par les données d'apprentissage $\mathcal{A} = \{\mathrm{a}_i\}$: ils s'appliquent à un vecteur x quelconque en calculant les valeurs de noyau $k(\mathrm{a}_i, \mathrm{x})$ qui correspondent à des produits scalaires $\boldsymbol{\phi}(\mathrm{a}_i)^{\mathrm{T}}\boldsymbol{\phi}(\mathrm{x})$ dans le *Feature Space*. Si les données d'apprentissage sont inclues dans le corpus du monde ($\mathcal{A} \subset \mathcal{B}$), alors ce produit scalaire peut s'écrire $\boldsymbol{\phi}(\mathrm{a}_i)^{\mathrm{T}}\boldsymbol{\phi}_{|\mathcal{A}}(\mathrm{x}) = \boldsymbol{\phi}_{|\mathcal{B}}(\mathrm{a}_i)^{\mathrm{T}}\boldsymbol{\phi}_{|\mathcal{B}}(\mathrm{x})$ (notations du lemme 1) et l'hypothèse de représentativité ne pose aucune contrainte.

Toutefois, en vérification du locuteur, les données d'apprentissage (notamment celles observées pour un locuteur cible) peuvent provenir d'un corpus distinct du corpus du Monde (constitué *a priori*), d'où la nécessité de faire appel à l'hypothèse de représentativité. Cette hypothèse n'est alors pas garantie lorsque la dimension D du *Feature Space* est très grande ($D > N$) ou infinie comme c'est le cas avec un noyau RBF Gaussien. Dans ce cas, nous rappelons qu'utiliser l'hypothèse comme nous le faisons avec l'*expansion* empirique revient à projeter les données, dans le *Feature Space*, sur le sous-espace engendré par les $\{\boldsymbol{\phi}(\mathrm{b}_i)\}$. L'erreur commise par une telle approximation est minime du moment que les vecteurs du Monde sont suffisamment nombreux et représentatifs.

Finalement, si l'hypothèse de représentativité n'est pas respectée, la proposition 1 reste vraie si l'on considère l'*expansion* projetée $\boldsymbol{\phi}_{|\mathcal{B}}$ au lieu de l'*expansion* $\boldsymbol{\phi}$. La matrice des seconds moments $\mathbf{S}_{\mathcal{B}}$ reste la même après cette modification, étant donné que les *expansions* du Monde restent inchangées par la projection : $\boldsymbol{\phi}_{|\mathcal{B}}(\mathrm{b}_i) = \boldsymbol{\phi}(\mathrm{b}_i)$. les noyaux FSNS dont nous avons donné la forme duale dans la proposition 1 s'écrivent de manière exacte :

$$\widehat{k}_{|\mathcal{B}}(\mathrm{x}, \mathrm{y}) = \boldsymbol{\phi}_{|\mathcal{B}}(\mathrm{x})^{\mathrm{T}} \left[\mathbf{S}_{\mathcal{B}} + \varepsilon \mathbf{I}_D\right]^{-1} \boldsymbol{\phi}_{|\mathcal{B}}(\mathrm{y}) \tag{4.47}$$

$$\widehat{\kappa}_{|\mathcal{B}}(\mathcal{X}, \mathcal{Y}) = \tfrac{1}{T_{\mathrm{X}} T_{\mathrm{Y}}} \sum_{t=1}^{T_{\mathrm{X}}} \sum_{s=1}^{T_{\mathrm{Y}}} \widehat{k}_{|\mathcal{B}}(\mathrm{x}_t, \mathrm{y}_s) \tag{4.48}$$

4.3.3 Forme duale des noyaux FSMS

Proposition 3.

Si le corpus du Monde $\mathcal{B} = \{\mathrm{b}_i\}$ vérifie l'hypothèse de représentativité, alors le noyau FSM (définition 7) peut s'écrire comme un noyau linéaire généralisé entre expansions empiriques centrées (4.33) :

$$\widetilde{k}(\mathrm{x},\mathrm{y}) = \big(\boldsymbol{\psi}_{\mathcal{B}}(\mathrm{x}) - \boldsymbol{\mu}_{\boldsymbol{\Psi}}\big)^{\mathrm{T}} \ \big[\tfrac{1}{N}\mathbf{K}\Pi\mathbf{K} + \epsilon\mathbf{K}\big]^{-1} \ \big(\boldsymbol{\psi}_{\mathcal{B}}(\mathrm{y}) - \boldsymbol{\mu}_{\boldsymbol{\Psi}}\big) \tag{4.49}$$

où $\boldsymbol{\mu}_{\boldsymbol{\Psi}}$ est la moyenne des expansions empiriques sur les vecteurs du Monde, et Π est une matrice de centrage :

$$\boldsymbol{\mu}_{\boldsymbol{\Psi}} = \tfrac{1}{N}\textstyle\sum_{i=1}^{N}\boldsymbol{\psi}_{\mathcal{B}}(\mathrm{b}_i) \tag{4.50}$$

$$\Pi = \mathbf{I}_N - \tfrac{1}{N}\mathbf{1}_N\mathbf{1}_N^{\mathrm{T}} \tag{4.51}$$

Par simple linéarité, le noyau FSMS s'écrit :

$$\widetilde{\kappa}(\mathcal{X},\mathcal{Y}) = \big(\overline{\boldsymbol{\psi}_{\mathcal{B}}}(\mathcal{X}) - \boldsymbol{\mu}_{\boldsymbol{\Psi}}\big)^{\mathrm{T}} \ \big[\tfrac{1}{N}\mathbf{K}\Pi\mathbf{K} + \epsilon\mathbf{K}\big]^{-1} \ \big(\overline{\boldsymbol{\psi}_{\mathcal{B}}}(\mathcal{Y}) - \boldsymbol{\mu}_{\boldsymbol{\Psi}}\big) \tag{4.52}$$

Dans le cas où le terme central de (4.49) et (4.52) n'est pas inversible, il suffit de considérer la pseudo-inverse.

Si $\varepsilon = 0$ alors les égalités (4.49)et (4.52) sont vraies sans l'hypothèse de représentativité.

Preuve :

Notons que la matrice Π définie en (4.51) est une matrice de projection de rang $(N-1)$ ($\Pi\Pi\Pi = \Pi$). En identifiant la moyenne des *expansions* du monde $\boldsymbol{\mu}_{\boldsymbol{\Phi}} = \frac{1}{N}\boldsymbol{\Phi}\mathbf{1}$, elle permet d'écrire la matrice des *expansions* centrées du Monde $\big\{\widetilde{\boldsymbol{\phi}}(\mathrm{b}_i)\big\} = \big\{\boldsymbol{\phi}(\mathrm{b}_i) - \boldsymbol{\mu}_{\boldsymbol{\Phi}}\big\}$ sous la forme $\widetilde{\boldsymbol{\Phi}} = \boldsymbol{\Phi}\Pi$.

L'application du lemme 1 à l'*expansion* $\widetilde{\boldsymbol{\phi}}(\cdot)$ en utilisant l'hypothèse de représentativité donne, en introduisant l'*expansion* empirique moyenne $\boldsymbol{\mu}_{\boldsymbol{\Psi}}$ (4.50) :

$$\begin{aligned}
\widetilde{\boldsymbol{\phi}}(\mathrm{x})^{\mathrm{T}}\widetilde{\boldsymbol{\phi}}(\mathrm{y}) &= \boldsymbol{\phi}(\mathrm{x})^{\mathrm{T}}\boldsymbol{\phi}(\mathrm{y}) - \tfrac{1}{N}\textstyle\sum_{i=1}^{N}\big(\boldsymbol{\phi}(\mathrm{x}) + \boldsymbol{\phi}(\mathrm{y})\big)^{\mathrm{T}}\boldsymbol{\phi}(\mathrm{b}_i) \\
&\qquad\qquad + \tfrac{1}{N^2}\big(\textstyle\sum_{i=1}^{N}\boldsymbol{\phi}(\mathrm{b}_i)\big)^{\mathrm{T}}\big(\textstyle\sum_{i=1}^{N}\boldsymbol{\phi}(\mathrm{b}_i)\big) \\
&= \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{x})^{\mathrm{T}}\mathbf{K}^{\dagger}\boldsymbol{\psi}_{\mathcal{B}}(\mathrm{y}) - \big(\boldsymbol{\psi}_{\mathcal{B}}(\mathrm{x}) + \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{y})\big)^{\mathrm{T}}\mathbf{K}^{\dagger}\boldsymbol{\mu}_{\boldsymbol{\Psi}} + \boldsymbol{\mu}_{\boldsymbol{\Psi}}{}^{\mathrm{T}}\mathbf{K}^{\dagger}\boldsymbol{\mu}_{\boldsymbol{\Psi}} \\
&= \big(\boldsymbol{\psi}_{\mathcal{B}}(\mathrm{x}) - \boldsymbol{\mu}_{\boldsymbol{\Psi}}\big)^{\mathrm{T}}\mathbf{K}^{\dagger}\big(\boldsymbol{\psi}_{\mathcal{B}}(\mathrm{y}) - \boldsymbol{\mu}_{\boldsymbol{\Psi}}\big)
\end{aligned} \tag{4.53}$$

En remarquant que $\boldsymbol{\mu}_{\boldsymbol{\Psi}} = \frac{1}{N}\mathbf{K}^{\mathrm{T}}\mathbf{1}$, on peut en déduire :

$$\widetilde{\boldsymbol{\phi}}(\mathrm{x})^{\mathrm{T}}\widetilde{\boldsymbol{\Phi}} = \big(\boldsymbol{\psi}_{\mathcal{B}}(\mathrm{x}) - \boldsymbol{\mu}_{\boldsymbol{\Psi}}\big)^{\mathrm{T}}\mathbf{K}^{\dagger}\mathbf{K}\Pi \tag{4.54}$$

$$\widetilde{\boldsymbol{\Phi}}^{\mathrm{T}}\widetilde{\boldsymbol{\Phi}} = \Pi\mathbf{K}\mathbf{K}^{\dagger}\mathbf{K}\Pi = \Pi\mathbf{K}\Pi = \widetilde{\mathbf{K}} \text{ (matrice de Gram centrée)} \tag{4.55}$$

.../...

La matrice de covariance s'écrit quant à elle :

$$\mathbf{\Sigma}_{\mathcal{B}} = \tfrac{1}{N}\widetilde{\mathbf{\Phi}}\widetilde{\mathbf{\Phi}}^{\mathrm{T}} = \tfrac{1}{N}\mathbf{\Phi}\Pi\mathbf{\Phi}^{\mathrm{T}}$$

En considérant cette fois-ci $\mathbf{M} = \widetilde{\mathbf{\Phi}}$ (au lieu de $\mathbf{\Phi}$) dans l'identité de Woodbury (4.45), on peut inverser la covariance régularisée intervenant dans (4.57) pour obtenir :

$$\widetilde{k}(\mathrm{x},\mathrm{y}) = \widetilde{\boldsymbol{\phi}}(\mathrm{x})^{\mathrm{T}}\left[\varepsilon^{-1}\mathbf{I}_D - \varepsilon^{-1}\widetilde{\mathbf{\Phi}}\left(N\varepsilon\mathbf{I}_N + \widetilde{\mathbf{\Phi}}^{\mathrm{T}}\widetilde{\mathbf{\Phi}}\right)^{-1}\widetilde{\mathbf{\Phi}}^{\mathrm{T}}\right]\widetilde{\boldsymbol{\phi}}(\mathrm{y})$$

Si l'on applique maintenant les relations (4.53), (4.54) et (4.55) on peut faire apparaître l'*expansion* empirique centrée :

$$\widetilde{k}(\mathrm{x},\mathrm{y}) = \left(\boldsymbol{\psi}_{\mathcal{B}}(\mathrm{x}) - \boldsymbol{\mu}_{\mathbf{\Psi}}\right)^{\mathrm{T}}\left[\varepsilon^{-1}\mathbf{K}^{\dagger} - \varepsilon^{-1}\mathbf{K}^{\dagger}\mathbf{K}\Pi\left(N\varepsilon\mathbf{I}_N + \widetilde{\mathbf{K}}\right)^{-1}\Pi\mathbf{K}\mathbf{K}^{\dagger}\right]\left(\boldsymbol{\psi}_{\mathcal{B}}(\mathrm{y}) - \boldsymbol{\mu}_{\mathbf{\Psi}}\right)$$

Pour simplifier le terme central il suffit d'appliquer l'identité de Woodbury généralisée (4.46) avec $\mathbf{P} = \varepsilon\mathbf{K}$, $\mathbf{Q} = \frac{1}{N}\mathbf{I}_N$ et $\mathbf{M} = \mathbf{K}\Pi$. On obtient alors l'expression finale (4.49).

Pour $\varepsilon = 0$, la démonstration est similaire à celle de la proposition 2 en partant d'une décomposition SVD de $\mathbf{\Phi}\Pi$.

■

Si l'on veut utiliser ce noyau dans les SVMs, on peut faire abstraction du terme $\boldsymbol{\mu}_{\mathbf{\Psi}}$, étant donné que les SVMs sont invariants par translation dans le *Feature Space*. On obtiendra la même fonction discriminante en considérant le noyau :

$$\overline{\boldsymbol{\psi}_{\mathcal{B}}}(\mathcal{X})^{\mathrm{T}}\left[\tfrac{1}{N}\mathbf{K}\Pi\mathbf{K} + \varepsilon\mathbf{K}\right]^{\dagger}\overline{\boldsymbol{\psi}_{\mathcal{B}}}(\mathcal{Y}) \tag{4.56}$$

On peut facilement montrer que cela revient à calculer $\boldsymbol{\phi}(\mathrm{x})^{\mathrm{T}}\left[\mathbf{\Sigma}_{\mathcal{B}} + \varepsilon\mathbf{I}_D\right]^{-1}\boldsymbol{\phi}(\mathrm{y})$.

Tout comme pour le cas non centré présenté en §4.3.2, l'hypothèse de représentativité intervient à cause du terme de régularisation, et n'est pas gênante lorsque les données d'apprentissage sont inclues dans le corpus du Monde. Si $\varepsilon = 0$ la relation (4.49) reste tout le temps valable. Dans les cas où $\varepsilon \neq 0$ et où l'hypothèse de représentativité n'est pas respectée, il suffit de considérer les noyaux FSMS définis sur les *expansions* projetées $\boldsymbol{\phi}_{|\mathcal{B}}$ aux lieu des *expansions* $\boldsymbol{\phi}$. Ces noyaux sont de la forme :

$$\widetilde{k}(\mathrm{x},\mathrm{y}) = \widetilde{\boldsymbol{\phi}_{|\mathcal{B}}}(\mathrm{x})^{\mathrm{T}}\left[\mathbf{\Sigma}_{\mathcal{B}} + \varepsilon\mathbf{I}_D\right]^{-1}\widetilde{\boldsymbol{\phi}_{|\mathcal{B}}}(\mathrm{y}) \tag{4.57}$$

$$\text{où } \widetilde{\boldsymbol{\phi}_{|\mathcal{B}}}(\mathrm{x}) = \boldsymbol{\phi}_{|\mathcal{B}}(\mathrm{x}) - \boldsymbol{\mu}_{\mathbf{\Phi}} \tag{4.58}$$

$$\widetilde{\kappa}(\mathcal{X},\mathcal{Y}) = \tfrac{1}{T_{\mathrm{X}}T_{\mathrm{Y}}}\sum_{t=1}^{T_{\mathrm{X}}}\sum_{s=1}^{T_{\mathrm{Y}}}\widetilde{k}(\mathrm{x}_t,\mathrm{y}_s) \tag{4.59}$$

L'*expansion* que nous notons $\widetilde{\boldsymbol{\phi}_{|\mathcal{B}}}(\mathrm{x})$ représente l'*expansion* $\boldsymbol{\phi}(\mathrm{x})$ auquel il a été appliqué deux transformations linéaires successives :

1. Projection $\boldsymbol{\phi}(\mathrm{x})$ sur le sous-espace du *Feature Space* engendré par les *expansions* du Monde $\{\boldsymbol{\phi}(\mathrm{b}_i)\}$.

2. Soustraction de la moyenne empirique $\boldsymbol{\mu}_{\boldsymbol{\Phi}}$ (4.27) des projetés estimée sur les *expansions* du Monde, c'est-à-dire translation pour rendre les $\left\{\widetilde{\boldsymbol{\phi}_{|\mathcal{B}}}(\mathrm{b}_i)\right\}$ centrés.

Il est important de choisir d'appliquer la projection en amont du centrage, de manière à être en accord avec la définition des noyaux FSMS. Appliquée ainsi, la projection laisse inchangées les *expansions* centrées $\{\widetilde{\boldsymbol{\phi}}(\mathrm{b}_i)\}$ du Monde, et l'estimation de la matrice de covariance $\boldsymbol{\Sigma}_{\mathcal{B}}$.

4.4 Approximation par Décomposition de Cholesky Incomplète

Nous montrons maintenant une astuce pour approcher les expressions introduites par les propositions 1 et 3, de manière à réduire la complexité calculatoire $O(N^2)$ des noyaux FSNS et FSMS dans leurs formes duales. Un telle complexité est en effet rédhibitoire pour un problème de vérification du locuteur où le nombre N de données de Monde est élevé.

Notations

Comme dans la section précédente la matrice de Gram $\mathbf{K}$ va jouer un rôle essentiel dans notre raisonnement. Dans ce qui suit, cette matrice (taille $N\times N$) contient les valeurs de noyaux $k(\mathrm{b}_i, \mathrm{b}_j)$ relatives aux N vecteurs du Monde b_i. Elle est supposée symétrique et semi-définie positive, une condition suffisante étant que le noyau k vérifie les conditions de Mercer.

Pour faciliter la compréhension du lecteur, nous introduisons les notations suivantes.

- $\mathrm{I} = \left\{\ p_1\ ,\ \cdots\ ,\ p_m\ \right\}$ désigne une liste ordonnée d'entiers distincts deux à deux compris entre 1 et N, de taille $m < N$. Concrètement, I indexe certaines colonnes de $\mathbf{K}$.
- Si l'on considère la correspondance entre chaque vecteur du Monde b_j et chaque colonne j de $\mathbf{K}$ (valeurs $k(\cdot, \mathrm{b}_j)$), alors l'index I désigne un sous-ensemble des vecteurs du Monde (de taille m), auquel nous ferons référence par "*dictionnaire*". Nous le notons :

$$\begin{array}{rcl} \mathcal{C} & = & \left\{\ \mathrm{b}_{p_1}\ ,\ \cdots\ ,\ \mathrm{b}_{p_i}\ ,\ \cdots\ ,\ \mathrm{b}_{p_m}\ \right\} \\ & = & \left\{\ \mathrm{c}_1\ ,\ \cdots\ ,\ \mathrm{c}_i\ ,\ \cdots\ ,\ \mathrm{c}_m\ \right\} \end{array} \tag{4.60}$$

- $\mathbf{K}(:, \mathrm{I})$ désigne la matrice $N\times m$ des colonnes de $\mathbf{K}$ indexées par I (notation `matlab`). Elle contient les valeurs $k(\mathrm{b}_i, \mathrm{c}_j)$.
- $\mathbf{K}(\mathrm{I}, \mathrm{I})$ désigne la matrice $m\times m$ des valeurs de $\mathbf{K}$ aux indices $\mathrm{I}\times\mathrm{I}$, c'est-à-dire les lignes de $\mathbf{K}(:, \mathrm{I})$ indexées par I. C'est aussi la matrice de Gram sur le *dictionnaire* $\mathcal{C}$.
- $\boldsymbol{\psi}_{\mathcal{C}}(\mathrm{x})$ désigne l'*expansion* empirique sur le *dictionnaire* $\mathcal{C}$, qui correspond aux lignes de $\boldsymbol{\psi}_{\mathcal{B}}(\mathrm{x})$ indexées par I.

$$\boldsymbol{\psi}_{\mathcal{C}}(\mathrm{x}) = \begin{bmatrix} k(\mathrm{b}_{p_1}, \mathrm{x}) \\ \vdots \\ k(\mathrm{b}_{p_m}, \mathrm{x}) \end{bmatrix} = \begin{bmatrix} k(\mathrm{c}_1, \mathrm{x}) \\ \vdots \\ k(\mathrm{c}_m, \mathrm{x}) \end{bmatrix} \tag{4.61}$$

- $\overline{\boldsymbol{\psi}_{\mathcal{C}}}(\mathcal{X})$ désigne la moyenne $\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{\psi}_{\mathcal{C}}(\mathrm{x}_t)$ de l'*expansion* empirique sur une séquence.

4.4.1 Introduction à la réduction de complexité

L'idée pour réduire la complexité est de se baser sur une approximation de la matrice de Gram avec une matrice de rang plus faible (*low-rank decomposition*). Les méthodes de ce genre ont souvent été appliquées pour réduire la complexité des algorithmes d'apprentissage dans les méthodes à noyau tout en minimisant la perte d'information [Smola et Schölkopf, 2000, Williams et Seeger, 2001, Fine et Scheinberg, 2001]. En fait, même si la matrice de Gram est de rang plein, son spectre décroît exponeniellement dans la plupart des cas pratiques où le noyau est relativement adapté aux données (§3.1.3) [Williams et Seeger, 2000]. Les valeurs propres les plus faibles correspondent à des dimensions du *Feature Space* où la variance des (*expansions* des) données est relativement faible. Ces dimensions sont supposées comporter peu d'information et peuvent être supprimées. Le rang de la matrice de Gram est alors diminué d'autant.

L'idée sous-jacente est proche des motivations de la méthode "*kernel PCA*" [Schölkopf et al., 1998] qui consiste à faire une Analyse en Composantes Principales dans le *Feature Space*. Cette méthode n'est toutefois pas adéquate pour réduire la complexité car les directions intéressantes du *Feature Space* sont exprimées en fonction de tous les vecteurs de départ. En effet, les axes principaux déterminés par une *kernel PCA* correspondent à une famille d'éléments du *Feature Space*, que nous pouvons noter $\{\boldsymbol{\phi}_{\mathrm{c}_1}, \cdots, \boldsymbol{\phi}_{\mathrm{c}_m}\}$, et qui n'admettent pas forcément de pré-images[17] c_i dans l'espace d'entrée [Mika et al., 1999]. Chacun de ces éléments $\boldsymbol{\phi}_{\mathrm{c}_i}$ est manipulé implicitement au moyen d'un vecteur de poids $\boldsymbol{\alpha}_i = [\alpha_{i,1} \cdots \alpha_{i,N}]^{\mathrm{T}}$ qui sous-entend une combinaison linéaire des *expansions* du Monde de la forme $\boldsymbol{\phi}_{\mathrm{c}_i} = \sum_{j=1}^{N} \alpha_{i,j} \boldsymbol{\phi}(\mathrm{b}_i)$. La projection sur les axes du *Feature Space* ainsi définis se fait alors de manière implicite via la fonction noyau :

$$\underbrace{\boldsymbol{\phi}_{\mathrm{c}_i}{}^{\mathrm{T}} \boldsymbol{\phi}(\mathrm{x})}_{k(\text{“}\mathrm{c}_i\text{”},\mathrm{x})} = \sum_{j=1}^{N} \alpha_{i,j} k(\mathrm{b}_j, \mathrm{x}) = \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{x})^{\mathrm{T}} \boldsymbol{\alpha}_i$$

Si nous voulions intégrer à notre calcul de noyau la projection sur les m axes principaux[18] de la *kernel PCA*, on peut montrer qu'il suffit de remplacer, dans le résultat (4.40) de la proposition 1 :

- $\boldsymbol{\psi}(\mathrm{x})$ par $\left[\begin{array}{ccc} \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{x})^{\mathrm{T}} \boldsymbol{\alpha}_1 & \cdots & \boldsymbol{\psi}_{\mathcal{B}}(\mathrm{x})^{\mathrm{T}} \boldsymbol{\alpha}_m \end{array} \right]^{\mathrm{T}}$
- $\mathbf{K}^2$ par la matrice contenant les valeurs $\left(\boldsymbol{\alpha}_i{}^{\mathrm{T}} \mathbf{K}^2 \boldsymbol{\alpha}_j \right)_{i,j=1\cdots m}$
- $\mathbf{K}$ par la matrice contenant les valeurs $\left(\boldsymbol{\alpha}_i{}^{\mathrm{T}} \mathbf{K} \boldsymbol{\alpha}_j \right)_{i,j=1\cdots m}$

Même si la dimension m de ces matrices peut être nettement inférieure à la dimension N de départ, la complexité de calcul du noyau reste $O(Nm)$ linéaire en N. Cette complexité est rédhibitoire pour une tâche de vérification du locuteur où l'on cherche à classer les séquences en temps réel. Ce problème de complexité pourrait être atténué en utilisant un algorithme "*sparse kernel PCA*" [Tipping, 2001] qui fournit une solution parcimonieuse où une certaine proportion des $\alpha_{i,j}$ sont nuls. Mais malheureusement, cette technique n'est pas applicable dans notre contexte à cause de sa complexité $O(N^3)$ et plus encore de la mémoire requise $O(N^2)$.

En définitive, la réduction de complexité doit se faire par une méthode d'approximation de la matrice de Gram de rang m plus faible :

1. dont la capacité de mémoire requise et la complexité soient linéaires en N, et

[17] La pré-image du "supervecteur" $\boldsymbol{\phi}_{\mathrm{c}}$ par l'expansion $\boldsymbol{\phi}$ est un vecteur c de l'espace d'entrée tel que $\boldsymbol{\phi}(\mathrm{c}) = \boldsymbol{\phi}_{\mathrm{c}}$

[18] Nous supposons les "$\boldsymbol{\phi}_{\mathrm{c}_i}$" rangés par valeurs propres décroissantes.

2. qui débouche sur la sélection d'un ensemble de *vecteurs d'entrée*, dont le cardinal $m \ll N$ déterminera la complexité $O(m^2)$ du calcul des noyaux FSNS.

L'algorithme de Décomposition de Cholesky Incomplète (ICD) [Fine et Scheinberg, 2001] répond à ces deux exigences. Nous montrons maintenant comment une telle décomposition permet d'obtenir des expressions réduites des noyaux FSNS (§4.4.2) et FSMS (§4.4.3). Nous reviendrons ensuite plus en détail sur le critère à optimiser pour minimiser la perte d'information dans la réduction (§4.4.4), et sur l'algorithme ICD (§4.4.5).

4.4.2 Forme duale réduite des noyaux FSNS

Comme nous le verrons ensuite (§4.4.5), l'algorithme de Décomposition de Cholesky Incomplète permet de calculer une approximation de la matrice de Gram sur les données $\mathcal{B}$ de la forme :

$$\mathbf{K} \approx \mathbf{K}(:,\mathrm{I})\mathbf{K}(\mathrm{I},\mathrm{I})^{-1}\mathbf{K}(:,\mathrm{I})^{\mathrm{T}} \tag{4.62}$$

où I est un index d'un sous-ensemble des colonnes de $\mathbf{K}$, qui correspond à un sous-ensemble $\mathcal{C} = \{\mathrm{c}_i\}_i$ du corpus initial $\mathcal{B}$. La taille de I, notée m, correspond au rang des *expansions* $\{\boldsymbol{\phi}(\mathrm{c}_i)\}_i$. La proposition suivante montre qu'une telle décomposition permet de réduire la complexité de calcul des noyaux FSNS de $O(N^2)$ à $O(m^2)$.

Proposition 4.

Supposons que le corpus du Monde $\mathcal{B} = \{\mathrm{b}_i\}$ vérifie l'hypothèse de représentativité pour au moins un vecteur du couple $\{\mathrm{x},\mathrm{y}\}$ et que la matrice de Gram peut se décomposer sous la forme

$$\mathbf{K} = \mathbf{K}(:,\mathrm{I})\mathbf{K}(\mathrm{I},\mathrm{I})^{-1}\mathbf{K}(:,\mathrm{I})^{\mathrm{T}}$$

Alors *le noyau FSN peut s'écrire :*

$$\widehat{k}(\mathrm{x},\mathrm{y}) = \boldsymbol{\psi}_{\mathcal{C}}(\mathrm{x})^{\mathrm{T}} \left[\tfrac{1}{N}\mathbf{K}(:,\mathrm{I})^{\mathrm{T}}\mathbf{K}(:,\mathrm{I}) + \varepsilon\mathbf{K}(\mathrm{I},\mathrm{I})\right]^{-1} \boldsymbol{\psi}_{\mathcal{C}}(\mathrm{y}) \tag{4.63}$$

où $\boldsymbol{\psi}_{\mathcal{C}}$ *est l'expansion empirique réduite définie en (4.61), de la même taille que* I.

Aussi par linéarité, *le noyau FSNS s'écrit à son tour :*

$$\widehat{\kappa}(\mathcal{X},\mathcal{Y}) = \overline{\boldsymbol{\psi}_{\mathcal{C}}}(\mathcal{X})^{\mathrm{T}} \left[\tfrac{1}{N}\mathbf{K}(:,\mathrm{I})^{\mathrm{T}}\mathbf{K}(:,\mathrm{I}) + \varepsilon\mathbf{K}(\mathrm{I},\mathrm{I})\right]^{-1} \overline{\boldsymbol{\psi}_{\mathcal{C}}}(\mathcal{Y}) \tag{4.64}$$

Preuve :

Pour simplifier les expressions, nous notons $\boldsymbol{\Phi}$ la matrice des *expansions* du Monde et

$$\mathbf{G} = \mathbf{K}(:,\mathrm{I})\mathbf{K}(\mathrm{I},\mathrm{I})^{-1/2}$$

la matrice de taille $N \times m$ qui est une racine carré de la matrice de Gram $\mathbf{K} = \mathbf{G}\mathbf{G}^{\mathrm{T}}$. Le fait que ces deux matrices aient le même carré ($\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} = \mathbf{K} = \mathbf{G}\mathbf{G}^{\mathrm{T}}$) implique que $\boldsymbol{\Phi}$ peut se décomposer sous la forme [Golub et Van Loan, 1996] :

.../...

$$\mathbf{\Phi}^{\mathrm{T}} = \mathbf{G}{\mathbf{U}_m}^{\mathrm{T}} \tag{4.65}$$

où $\mathbf{U}_m$ est une matrice orthonormale de taille $N \times m$ (${\mathbf{U}_m}^{\mathrm{T}}\mathbf{U}_m = \mathbf{I}_m$).

Si l'on suppose vraie l'hypothèse de représentativité des données du Monde, alors on peut facilement généraliser cette relation pour tout vecteur x :

$$\boldsymbol{\phi}(\mathrm{x})^{\mathrm{T}} = \boldsymbol{\psi}_{\mathcal{C}}(\mathrm{x})^{\mathrm{T}}\mathbf{K}(\mathrm{I},\mathrm{I})^{-1/2}{\mathbf{U}_m}^{\mathrm{T}} \tag{4.66}$$

Cette relation permet de retrouver le résultat du lemme 1 sous la forme $\boldsymbol{\phi}(\mathrm{x})^{\mathrm{T}}\boldsymbol{\phi}(\mathrm{y}) = \boldsymbol{\psi}_{\mathcal{C}}(\mathrm{x})^{\mathrm{T}}\mathbf{K}(\mathrm{I},\mathrm{I})^{-1}\boldsymbol{\psi}_{\mathcal{C}}(\mathrm{y})$.

L'équation (4.65) permet aussi d'écrire la matrice des seconds moments régularisée sous la forme :

$$\mathbf{S}_{\mathcal{B}} + \varepsilon\mathbf{I}_D = \tfrac{1}{N}\mathbf{U}_m\mathbf{G}^{\mathrm{T}}\mathbf{G}\mathbf{U}_m + \varepsilon\mathbf{I}_D \tag{4.67}$$

On arrive ensuite à montrer le résultat en utilisant le même raisonnement que la preuve de la proposition 1, c'est-à-dire en appliquant l'identité de Woodbury (4.45) avec $\mathbf{P} = \varepsilon\mathbf{I}_D$, $\mathbf{Q} = \frac{1}{N}\mathbf{I}_N$ et $\mathbf{M} = \mathbf{U}_m\mathbf{G}^{\mathrm{T}}$.

■

Cette réduction de complexité repose sur l'approximation à bas rang de la matrice de Gram (4.62). On peut donc pressentir que le critère à minimiser pour le choix de l'approximation est une norme de la matrice résiduelle $\mathbf{K} - \mathbf{K}(:,\mathrm{I})\mathbf{K}(\mathrm{I},\mathrm{I})^{-1}\mathbf{K}(:,\mathrm{I})^{\mathrm{T}}$. Pour être conscient des écarts induits par les approximations faites et formuler clairement ce critère, Nous montrons maintenant que l'approximation que nous avons donné pour les noyaux FSNS revient à projeter les *expansions* des vecteurs d'entrée sur l'espace engendré par les *expansions* des vecteurs du *dictionnaire* $\mathcal{C}$ correspondant à l'index I (4.60). Notons $\boldsymbol{\phi}_{|\mathcal{C}}(\mathrm{x})$ le projeté orthogonal de l'*expansion* $\boldsymbol{\phi}(\mathrm{x})$ sur l'espace engendré par les *expansions* $\{\boldsymbol{\phi}(\mathrm{c}_i)\}_{i=1\cdots m}$. La matrice des seconds moments correspondant à ces "expansions projetées" est définie par :

$$\mathbf{S}_{\mathcal{B}|\mathcal{C}} = \tfrac{1}{N}\sum_{i=1}^{N}\boldsymbol{\phi}_{|\mathcal{C}}(\mathrm{b}_i)\boldsymbol{\phi}_{|\mathcal{C}}(\mathrm{b}_i)^{\mathrm{T}}$$

Avec ces notations, la proposition explicite la quantité calculée de manière exacte par la forme duale réduite introduite dans la proposition 4.

Proposition 5.

Pour tout sous-ensemble $\mathcal{C} \subset \mathcal{B}$ indexé par I*, on a l'égalité :*

$$\textit{Si } \varepsilon > 0, \quad \boldsymbol{\psi}_{\mathcal{C}}(\mathrm{x})^{\mathrm{T}}\left[\tfrac{1}{N}\mathbf{K}(:,\mathrm{I})^{\mathrm{T}}\mathbf{K}(:,\mathrm{I}) + \varepsilon\mathbf{K}(\mathrm{I},\mathrm{I})\right]^{-1}\boldsymbol{\psi}_{\mathcal{C}}(\mathrm{y}) = \boldsymbol{\phi}_{|\mathcal{C}}(\mathrm{x})^{\mathrm{T}}\left[\mathbf{S}_{\mathcal{B}|\mathcal{C}} + \varepsilon\mathbf{I}_D\right]^{-1}\boldsymbol{\phi}_{|\mathcal{C}}(\mathrm{y})$$

$$\textit{Si } \varepsilon = 0, \qquad \boldsymbol{\psi}_{\mathcal{C}}(\mathrm{x})^{\mathrm{T}}\left[\tfrac{1}{N}\mathbf{K}(:,\mathrm{I})^{\mathrm{T}}\mathbf{K}(:,\mathrm{I})\right]^{-1}\boldsymbol{\psi}_{\mathcal{C}}(\mathrm{y}) = \boldsymbol{\phi}_{|\mathcal{C}}(\mathrm{x})^{\mathrm{T}}\,\mathbf{S}_{\mathcal{B}|\mathcal{C}}{}^{\dagger}\,\boldsymbol{\phi}_{|\mathcal{C}}(\mathrm{y})$$

où $\mathbf{K}(:,\mathrm{I})$ et $\mathbf{K}(\mathrm{I},\mathrm{I})$ sont les matrices de taille respective $N \times m$ et $m \times m$ contenant les valeurs de noyaux $\{k(\mathrm{b}_i,\mathrm{c}_j)\}$ et $\{k(\mathrm{c}_i,\mathrm{c}_j)\}$ (cf. notations en début de §4.4).

Preuve :

• Cas $\varepsilon > 0$

Notons $\mathbf{\Phi}_{|\mathcal{C}}$ la matrice des *expansions* projetées $\boldsymbol{\phi}_{|\mathcal{C}}(\mathrm{b}_i)$ des vecteurs du Monde. Cette matrice permet d'identifier la matrice des seconds moments $\mathbf{S}_{\mathcal{B}|\mathcal{C}} = \mathbf{\Phi}_{|\mathcal{C}}\mathbf{\Phi}_{|\mathcal{C}}^{\mathrm{T}}$. L'application du lemme 1 en tenant compte de l'inversibilité de $\mathbf{K}(\mathrm{I},\mathrm{I})$ conduit aux égalités :

$$\boldsymbol{\phi}_{|\mathcal{C}}(\mathrm{x})^{\mathrm{T}}\boldsymbol{\phi}_{|\mathcal{C}}(\mathrm{y}) = \boldsymbol{\psi}_{\mathcal{C}}(\mathrm{x})^{\mathrm{T}}\mathbf{K}(\mathrm{I},\mathrm{I})^{-1}\boldsymbol{\psi}_{\mathcal{C}}(\mathrm{y}) \tag{4.68}$$

$$\boldsymbol{\phi}_{|\mathcal{C}}(\mathrm{x})^{\mathrm{T}}\mathbf{\Phi}_{|\mathcal{C}} = \boldsymbol{\psi}_{\mathcal{C}}(\mathrm{x})^{\mathrm{T}}\mathbf{K}(\mathrm{I},\mathrm{I})^{-1}\mathbf{K}(:,\mathrm{I})^{\mathrm{T}} \tag{4.69}$$

$$\mathbf{\Phi}_{|\mathcal{C}}^{\mathrm{T}}\mathbf{\Phi}_{|\mathcal{C}} = \mathbf{K}(:,\mathrm{I})\mathbf{K}(\mathrm{I},\mathrm{I})^{-1}\mathbf{K}(:,\mathrm{I})^{\mathrm{T}} \tag{4.70}$$

La suite de la démonstration est ensuite analogue à la preuve de la proposition 1, en introduisant $\mathbf{\Phi}_{|\mathcal{C}}$ au lieu de $\mathbf{\Phi}$.

• Cas $\varepsilon = 0$

Considérons ici la Décomposition an Valeurs Singulières mince (SVD, annexe A.1.1) de $\mathbf{\Phi}_{|\mathcal{C}}$:

$$\mathbf{\Phi}_{|\mathcal{C}} = \mathbf{U}_m\mathbf{D}_m{\mathbf{V}_m}^{\mathrm{T}}$$

où m correspond au rang de $\mathbf{\Phi}_{|\mathcal{C}}$, et où les tailles des matrices $\mathbf{U}_m$, $\mathbf{V}_m$ et $\mathbf{D}_m$ sont respectivement $D{\times}m$, $N{\times}m$ et $m{\times}m$. Cette décomposition permet d'écrire la SVD de la matrice des seconds moments $\mathbf{S}_{\mathcal{B}|\mathcal{C}} = \frac{1}{N}\mathbf{U}_m{\mathbf{D}_m}^2{\mathbf{U}_m}^{\mathrm{T}}$, ainsi que :

$$\mathbf{\Phi}_{|\mathcal{C}}^{\mathrm{T}}\mathbf{\Phi}_{|\mathcal{C}} \;=\; \mathbf{V}_m{\mathbf{D}_m}^2{\mathbf{V}_m}^{\mathrm{T}} \;=\; \mathbf{K}(:,\mathrm{I})\mathbf{K}(\mathrm{I},\mathrm{I})^{-1}\mathbf{K}(:,\mathrm{I})^{\mathrm{T}} \qquad \text{d'après (4.70)}$$

Avec le même raisonnement sur la décomposition SVD que dans la preuve de la proposition 2, on peut ainsi réécrire la pseudo-inversion

$$\mathbf{S}_{\mathcal{B}|\mathcal{C}}{}^{\dagger} = {}_N\mathbf{\Phi}_{|\mathcal{C}}\left(\left(\mathbf{K}(:,\mathrm{I})\mathbf{K}(\mathrm{I},\mathrm{I})^{-1}\mathbf{K}(:,\mathrm{I})^{\mathrm{T}}\right)^{\dagger}\right)^2{\mathbf{\Phi}_{|\mathcal{C}}}^{\mathrm{T}}$$

On arrive au résultat à montrer en se servant de (4.69), et en constatant que $rang(\mathbf{K}(:,\mathrm{I})^{\mathrm{T}}\mathbf{K}(:,\mathrm{I})) = rang(\mathbf{K}(:,\mathrm{I})^{\mathrm{T}}) = m$ (matrice de rang plein, donc inversible).

■

On peut remarquer qu'en l'absence de régularisation ($\varepsilon = 0$), la proposition 5 formule le noyau FSNS avec la pseudo-inverse de la matrice $\mathbf{S}_{\mathcal{B}|\mathcal{C}}$ qui contient les seconds moments des expansions projetées sur $\{\boldsymbol{\phi}(\mathrm{c}_i)\}$. En fait, cette matrice n'est pas inversible parce que le rang de $\{\boldsymbol{\phi}(\mathrm{b}_i)\}$ est supérieur au rang de $\{\boldsymbol{\phi}(\mathrm{c}_i)\}$. Normaliser par la pseudo-inverse de la matrice des seconds moments a une justification théorique et peut être interprété comme une régularisation adéquate [Kondor et Jebara, 2003]. En effet, cela revient à conserver uniquement les composantes principales (du *Feature Space*) qui correspondent à des valeurs propres non nulles de $\mathbf{S}_{\mathcal{B}|\mathcal{C}}$, comme dans une PCA sur les $\{\boldsymbol{\phi}_{|\mathcal{C}}(\mathrm{b}_i)\}$. Ces directions engendrent l'espace image de la projection sur les $\{\boldsymbol{\phi}(\mathrm{c}_i)\}$.

4.4.3 Forme duale réduite des noyaux FSMS

De la même manière que nous venons de le voir, la décomposition incomplète de la matrice de Gram (4.62) conduit à une expression réduite des noyaux FSMS. Avec une démonstration analogue à celles des propositions 3 et 4, on peut montrer la propriété suivante.

Proposition 6.
Supposons que les vecteurs du Monde $\mathcal{B} = \{\mathrm{b}_i\}$ vérifient l'hypothèse de représentativité pour zu moins un vecteur du couple $\{\mathrm{x}, \mathrm{y}\}$ et que la matrice de Gram peut se décomposer sous la forme

$$\mathbf{K} = \mathbf{K}(:,\mathrm{I})\mathbf{K}(\mathrm{I},\mathrm{I})^{-1}\mathbf{K}(:,\mathrm{I})^{\mathrm{T}}$$

Alors *le noyau FSMS peut s'écrire :*

$$\widetilde{\kappa}(\mathcal{X},\mathcal{Y}) \approx \left(\overline{\boldsymbol{\psi}_{\mathcal{C}}}(\mathcal{X}) - \boldsymbol{\mu}_{\boldsymbol{\Psi}_{\mathcal{C}}}\right)^{\mathrm{T}} \left[\tfrac{1}{N}\mathbf{K}(:,\mathrm{I})^{\mathrm{T}}\Pi\mathbf{K}(:,\mathrm{I}) + \varepsilon\mathbf{K}(\mathrm{I},\mathrm{I})\right]^{-1} \left(\overline{\boldsymbol{\psi}_{\mathcal{C}}}(\mathcal{Y}) - \boldsymbol{\mu}_{\boldsymbol{\Psi}_{\mathcal{C}}}\right) \qquad (4.71)$$

où $\Pi = \mathbf{I}_N - \frac{1}{N}\mathbf{1}\mathbf{1}^{\mathrm{T}}$ *est la matrice de centrage* $(N\times N)$ *et où* $\boldsymbol{\mu}_{\boldsymbol{\Psi}_{\mathcal{C}}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\psi}_{\mathcal{C}}(\mathrm{b}_i)$ *est la moyenne de l'expansion empirique* $\boldsymbol{\psi}_{\mathcal{C}}(\cdot)$ *estimée sur les données du Monde.*

Nous rappelons que lorsque l'on applique un algorithme invariant par translation dans le *Feature Space*, comme les SVMs, on peut faire abstraction du terme $\boldsymbol{\mu}_{\boldsymbol{\Psi}_{\mathcal{C}}}$. Cela permet de retrouver la même complexité de calcul que pour les noyaux FSNS (une fois le terme central de normalisation calculé).

Nous montrons maintenant que la forme réduite proposée revient à projeter les (*expansions* des) données sur les *expansions* du *dictionnaire* $\mathcal{C}$, puis à calculer le noyau FSMS exact. En reprenant la notation $\boldsymbol{\Phi}_{|\mathcal{C}}$ pour les expansions projetées, nous définissons les expansions projetées (puis) centrées $\widetilde{\boldsymbol{\Phi}}_{|\mathcal{C}}$ et la matrice de covariance $\boldsymbol{\Sigma}_{\mathcal{B}|\mathcal{C}}$ correspondante :

$$\begin{aligned}
\widetilde{\boldsymbol{\Phi}}_{|\mathcal{C}}(\mathrm{x}) &= \boldsymbol{\Phi}_{|\mathcal{C}}(\mathrm{x}) - \boldsymbol{\mu}_{\boldsymbol{\Phi}_{\mathcal{C}}} \\
\boldsymbol{\Sigma}_{\mathcal{B}|\mathcal{C}} &= \frac{1}{N}\sum_{i=1}^{N}\widetilde{\boldsymbol{\Phi}}_{|\mathcal{C}}(\mathrm{b}_i)\widetilde{\boldsymbol{\Phi}}_{|\mathcal{C}}(\mathrm{b}_i)^{\mathrm{T}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\Phi}_{|\mathcal{C}}(\mathrm{b}_i)\boldsymbol{\Phi}_{|\mathcal{C}}(\mathrm{b}_i)^{\mathrm{T}} - {\boldsymbol{\mu}_{\boldsymbol{\Phi}_{\mathcal{C}}}}^2 \\
\text{avec } \boldsymbol{\mu}_{\boldsymbol{\Phi}_{\mathcal{C}}} &= \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\Phi}_{|\mathcal{C}}(\mathrm{b}_i)
\end{aligned}$$

Proposition 7.
Avec les mêmes conventions que dans la proposition 5, on a les égalités :

$$\textit{Si } \varepsilon > 0, \quad \widetilde{\boldsymbol{\psi}_{\mathcal{C}}}(\mathrm{x})^{\mathrm{T}}\left[\tfrac{1}{N}\mathbf{K}(:,\mathrm{I})^{\mathrm{T}}\Pi\mathbf{K}(:,\mathrm{I}) + \varepsilon\mathbf{K}(\mathrm{I},\mathrm{I})\right]^{-1}\widetilde{\boldsymbol{\psi}_{\mathcal{C}}}(\mathrm{y}) = \widetilde{\boldsymbol{\Phi}}_{|\mathcal{C}}(\mathrm{x})^{\mathrm{T}}\left[\boldsymbol{\Sigma}_{\mathcal{B}|\mathcal{C}} + \varepsilon\mathbf{I}_D\right]^{-1}\widetilde{\boldsymbol{\Phi}}_{|\mathcal{C}}(\mathrm{y})$$

$$\textit{Si } \varepsilon = 0, \qquad \widetilde{\boldsymbol{\psi}_{\mathcal{C}}}(\mathrm{x})^{\mathrm{T}}\left[\tfrac{1}{N}\mathbf{K}(:,\mathrm{I})^{\mathrm{T}}\Pi\mathbf{K}(:,\mathrm{I})\right]^{\dagger}\widetilde{\boldsymbol{\psi}_{\mathcal{C}}}(\mathrm{y}) = \widetilde{\boldsymbol{\Phi}}_{|\mathcal{C}}(\mathrm{x})^{\mathrm{T}}\,\boldsymbol{\Sigma}_{\mathcal{B}|\mathcal{C}}{}^{\dagger}\,\widetilde{\boldsymbol{\Phi}}_{|\mathcal{C}}(\mathrm{y})$$

où $\widetilde{\boldsymbol{\psi}_{\mathcal{C}}}(\mathrm{x}) = \boldsymbol{\psi}_{\mathcal{C}}(\mathrm{x}) - \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\psi}_{\mathcal{C}}(\mathrm{b}_i)$ *est l'expansion empirique centrée sur les données du Monde, et où* Π *est la matrice de centrage définie dans la proposition 1.*

Preuve :

• Cas $\varepsilon > 0$

Notons $\widetilde{\mathbf{\Phi}_{|\mathcal{C}}} = \mathbf{\Phi}_{|\mathcal{C}}\Pi$ la matrice des *expansions* projetées $\widetilde{\boldsymbol{\phi}}_{|\mathcal{C}}(\mathrm{b}_i)$ des vecteurs du Monde. Elle permet d'identifier la covariance empirique $\mathbf{\Sigma}_{\mathcal{B}|\mathcal{C}} = \widetilde{\mathbf{\Phi}_{|\mathcal{C}}}\widetilde{\mathbf{\Phi}_{|\mathcal{C}}}^{\mathrm{T}}$. L'application du lemme 1 en tenant compte de l'inversibilité de $\mathbf{K}(\mathrm{I},\mathrm{I})$ conduit aux égalités :

$$\widetilde{\boldsymbol{\phi}}_{|\mathcal{C}}(\mathrm{x})^{\mathrm{T}}\widetilde{\boldsymbol{\phi}}_{|\mathcal{C}}(\mathrm{y}) = \widetilde{\boldsymbol{\psi}_{\mathcal{C}}}(\mathrm{x})^{\mathrm{T}}\mathbf{K}(\mathrm{I},\mathrm{I})^{-1}\widetilde{\boldsymbol{\psi}_{\mathcal{C}}}(\mathrm{y}) \tag{4.72}$$

$$\widetilde{\boldsymbol{\phi}}_{|\mathcal{C}}(\mathrm{x})^{\mathrm{T}}\widetilde{\mathbf{\Phi}_{|\mathcal{C}}} = \widetilde{\boldsymbol{\psi}_{\mathcal{C}}}(\mathrm{x})^{\mathrm{T}}\mathbf{K}(\mathrm{I},\mathrm{I})^{-1}\mathbf{K}(:,\mathrm{I})^{\mathrm{T}}\Pi \tag{4.73}$$

$$\widetilde{\mathbf{\Phi}_{|\mathcal{C}}}^{\mathrm{T}}\widetilde{\mathbf{\Phi}_{|\mathcal{C}}} = \Pi\mathbf{K}(:,\mathrm{I})\mathbf{K}(\mathrm{I},\mathrm{I})^{-1}\mathbf{K}(:,\mathrm{I})^{\mathrm{T}}\Pi \tag{4.74}$$

La suite de la démonstration est ensuite analogue à la preuve de la proposition 2, en introduisant dans le raisonnement $\widetilde{\mathbf{\Phi}_{|\mathcal{C}}}$ au lieu de $\mathbf{\Phi}$.

• Cas $\varepsilon = 0$

Considérons ici la SVD mince de la matrice $\widetilde{\mathbf{\Phi}_{|\mathcal{C}}}$:

$$\widetilde{\mathbf{\Phi}_{|\mathcal{C}}} = \mathbf{U}_m\mathbf{D}_m{\mathbf{V}_m}^{\mathrm{T}}$$

Cette décomposition permet d'écrire la SVD de la matrice des seconds moments $\mathbf{S}_{\mathcal{B}|\mathcal{C}} = \frac{1}{N}\mathbf{U}_m{\mathbf{D}_m}^2{\mathbf{U}_m}^{\mathrm{T}}$, ainsi que :

$$\begin{aligned}\widetilde{\mathbf{\Phi}_{|\mathcal{C}}}^{\mathrm{T}}\widetilde{\mathbf{\Phi}_{|\mathcal{C}}} &= \mathbf{V}_m{\mathbf{D}_m}^2{\mathbf{V}_m}^{\mathrm{T}} \\ &= \Pi\mathbf{K}(:,\mathrm{I})\mathbf{K}(\mathrm{I},\mathrm{I})^{-1}\mathbf{K}(:,\mathrm{I})^{\mathrm{T}}\Pi \qquad \text{d'après (4.74)}\end{aligned}$$

Avec le même raisonnement sur la décomposition SVD que dans la preuve de la proposition 2, on peut ainsi réécrire la pseudo-inversion

$$\mathbf{\Sigma}_{\mathcal{B}|\mathcal{C}}{}^{\dagger} = {}_N\widetilde{\mathbf{\Phi}_{|\mathcal{C}}}\left(\left(\Pi\mathbf{K}(:,\mathrm{I})\mathbf{K}(\mathrm{I},\mathrm{I})^{-1}\mathbf{K}(:,\mathrm{I})^{\mathrm{T}}\Pi\right)^{\dagger}\right)^2\widetilde{\mathbf{\Phi}_{|\mathcal{C}}}^{\mathrm{T}}$$

On arrive ensuite au résultat à montrer en se servant de (4.73).

■

4.4.4 Critère d'approximation

Supposons que l'on dispose d'un corpus de vecteurs du Monde $\mathcal{B}$ de taille N, et que l'on se fixe une taille $m < N$ pour le *dictionnaire* $\mathcal{C}$ sur lequel opérer la projection dans le *Feature Space*. Un but légitime, pour choisir une projection qui perde le moins d'information possible, est de minimiser la distance euclidienne entre les *expansions* des vecteurs du Monde et leurs projetés sur l'*expansion* du *dictionnaire*. Le critère des moindres carrés s'écrit :

$$\mathcal{C} = \arg\min_{\mathcal{C}\in\mathbb{X}^m}\sum_{i=1}^{N}\left\|\boldsymbol{\phi}(\mathrm{b}_i) - \boldsymbol{\phi}_{|\mathcal{C}}(\mathrm{b}_i)\right\|_2^2 \tag{4.75}$$

où $\Big\|\boldsymbol{\phi}(\mathrm{b}_i) - \boldsymbol{\phi}_{|\mathcal{C}}(\mathrm{b}_i)\Big\|_2^2 = \big(\boldsymbol{\phi}(\mathrm{b}_i) - \boldsymbol{\phi}_{|\mathcal{C}}(\mathrm{b}_i)\big)^{\mathrm{T}}\big(\boldsymbol{\phi}(\mathrm{b}_i) - \boldsymbol{\phi}_{|\mathcal{C}}(\mathrm{b}_i)\big)$ est la norme euclidienne au carré. Ce critère n'est pas le seul envisageable mais comme nous allons le montrer maintenant, il permet de justifier l'utilisation de la Décomposition de Cholesky Incomplète.

Le lemme 1 s'écrit dans le contexte d'une projection sur l'*expansion* du *dictionnaire* : $\boldsymbol{\phi}_{|\mathcal{C}}(\mathrm{b}_i)^{\mathrm{T}}\boldsymbol{\phi}_{|\mathcal{C}}(\mathrm{b}_i) = \boldsymbol{\psi}_{\mathcal{C}}(\mathrm{b}_i)^{\mathrm{T}}\mathbf{K}(\mathrm{I},\mathrm{I})^{\dagger}\boldsymbol{\psi}_{\mathcal{C}}(\mathrm{b}_i) = \boldsymbol{\phi}(\mathrm{b}_i)^{\mathrm{T}}\boldsymbol{\phi}_{|\mathcal{C}}(\mathrm{b}_i)$. Ces relations nous permettent de reformuler le critère :

$$\begin{aligned} \mathcal{C} &= \arg\min_{\mathcal{C}\in\mathbb{X}^m} \sum_{i=1}^{N} \Big(k(\mathrm{b}_i,\mathrm{b}_i) - \boldsymbol{\psi}_{\mathcal{C}}(\mathrm{b}_i)^{\mathrm{T}}\mathbf{K}(\mathrm{I},\mathrm{I})^{\dagger}\boldsymbol{\psi}_{\mathcal{C}}(\mathrm{b}_i)\Big) \\ &= \arg\min_{\mathcal{C}\in\mathbb{X}^m} \mathrm{tr}\Big(\mathbf{K} - \mathbf{K}(:,\mathrm{I})\mathbf{K}(\mathrm{I},\mathrm{I})^{\dagger}\mathbf{K}(:,\mathrm{I})^{\mathrm{T}}\Big) \end{aligned} \tag{4.76}$$

Le fait que le critère des moindres carrés s'exprime ainsi induit que le résidu $(\mathbf{K} - \mathbf{K}(:,\mathrm{I})\mathbf{K}(\mathrm{I},\mathrm{I})^{\dagger}\mathbf{K}(:,\mathrm{I})^{\mathrm{T}})$ est toujours semi-défini positif. La trace $\mathrm{tr}(\mathbf{M})$ est quant à elle une norme pour les matrices *semi-définies positives*. Nous rappelons (§3.1.3, théorème 4) qu'elle est aussi égale à la somme des valeurs propres, et il a été montré qu'elle est une borne inférieure de la norme de Frobenius $\|\mathbf{M}\|_F = \sqrt{\mathrm{tr}(\mathbf{M}\mathbf{M}^{\mathrm{T}})}$ [Smola et Schölkopf, 2000].

Le problème de minimisation (4.76) fait intervenir les fonctions $k(\mathrm{b}_i,\mathrm{c}_j)$ qui sont généralement non linéaires vis-à-vis des m inconnues c_j. Trouver la solution d'un tel problème se résout avec des algorithmes sous-optimaux qui sont instables et qui impliquent des complexités trop élevées dans notre cas où le nombre N d'équations à minimiser est de l'ordre de 10^6. Si l'on cherche le *dictionnaire* $\mathcal{C}$ parmi les vecteurs du Monde $\mathcal{B}$, alors on peut réécrire le problème d'optimisation en faisant intervenir un index I défini en (4.60) :

$$\mathrm{I} = \arg \min_{\substack{I\subset\{1\cdots N\},\\ \mathrm{card}(I)=m}} \mathrm{tr}\Big(\mathbf{K} - \mathbf{K}(:,\mathrm{I})\mathbf{K}(\mathrm{I},\mathrm{I})^{\dagger}\mathbf{K}(:,\mathrm{I})^{\mathrm{T}}\Big)$$

Le problème se ramène à trouver une bonne approximation de la matrice de Gram qui peut se décomposer sous la forme $\mathbf{K}(:,\mathrm{I})\mathbf{K}(\mathrm{I},\mathrm{I})^{\dagger}\mathbf{K}(:,\mathrm{I})$. Soulignons qu'il s'agit d'un problème d'optimisation *sous contraintes*, étant donné que l'on impose la taille du *dictionnaire*. Sans cette contrainte, une solution triviale qui fournit un résidu nul est $\mathrm{I} = \{1\cdots N\}$ ($\mathcal{C} = \mathcal{B}$). En fait, le résidu peut s'annuler à partir du moment où la taille du *dictionnaire* a atteint $r = rang\,(\mathbf{K}) = rang\,(\{\boldsymbol{\phi}(\mathrm{b}_i)\}_{i=1\cdots N})$. Tant que cette taille r n'a pas été atteinte, on peut montrer que le *dictionnaire* de taille $m < r$ qui minimisent le critère correspondent nécessairement à une matrice $\mathbf{K}(\mathrm{I},\mathrm{I})$ inversible. En effet, lorsque cette sous-matrice de Gram n'est pas de rang plein, les *expansions* $\{\boldsymbol{\phi}(\mathrm{c}_i)\}_{i=1\cdots m}$ du *dictionnaire* $\mathcal{C}_m$ correspondant ne sont pas indépendantes. Quitte à réordonner l'index I, on peut donc supposer que $\boldsymbol{\phi}(\mathrm{c}_m)$ appartient au sous-espace engendré par $\{\ \boldsymbol{\phi}(\mathrm{c}_i)\ \cdots\ \boldsymbol{\phi}(\mathrm{c}_{m-1})\ \}$. Or si le rang r des $\{\boldsymbol{\phi}(\mathrm{b}_i)\}$ n'est pas atteint, alors on peut améliorer le critère (4.75) en remplaçant c_m par n'importe quel vecteur b_{p_m} du Monde tel que $\boldsymbol{\phi}(\mathrm{b}_{p_m})$ n'appartient pas à ce sous-espace. Donc le *dictionnaire* $\{\mathrm{c}_i\}_{i=1\cdots m}$ n'est pas optimal parmi ceux de la même taille. Ces considérations nous permettent de simplifier le critère à optimiser en enlevant la pseudo-inversion superflue :

$$\mathrm{I} = \arg \min_{\substack{I\subset\{1\cdots N\},\\ \mathrm{card}(I)\leq m}} \underbrace{\mathrm{tr}\Big(\mathbf{K} - \mathbf{K}(:,\mathrm{I})\mathbf{K}(\mathrm{I},\mathrm{I})^{-1}\mathbf{K}(:,\mathrm{I})^{\mathrm{T}}\Big)}_{mse(\mathrm{I})} \tag{4.77}$$

Nous notons *mse* le critère à minimiser (abréviation de "*mean squared error*") vu que c'est une erreur quadratique moyenne d'après (4.76).

4.4.5 Justification de la décomposition de Cholesky incomplète

Trouver l'index I optimal pour (4.77) (ou autrement dit le *dictionnaire* $\mathcal{C}$ optimal pour (4.76)) est un problème NP-complet qui requiert d'essayer $\frac{(m+N)!}{m!\ N!}$ combinaisons possibles tout en mémorisant à chaque itérations $O(N^2)$ valeurs. C'est trop élevé pour notre application où N est de l'ordre de 10^6. Pour surmonter ce problème de complexité, nous décidons d'utiliser un algorithme glouton (*greedy*) comme il est classique de le faire dans ce genre de situations [Mallat et Zhang, 1993, Natarajan, 1995]. Dans notre cas, le principe d'un tel algorithme est de sélectionner de manière itérative les vecteurs du *dictionnaire* en optimisant à chaque étape la réduction du critère *mse*. À chaque itération, un vecteur du Monde est ainsi sélectionné pour être rajouté au *dictionnaire*. L'algorithme s'arrête dès que la taille du *dictionnaire* atteint une valeur m fixée, ou que le critère à minimiser passe en-dessous d'un seuil $\eta \geq 0$. Un tel algorithme peut se formuler comme suit.

1. Initialiser $\mathrm{I} \coloneqq \{p_1\}$ avec $p_1 = \arg\max_{j=1\cdots N} \mathbf{K}(j,j)$, et $\mathrm{J} \coloneqq \{1\cdots N\} \setminus \{p_1\}$ (complémentaire).
2. Itérer tant que la taille de l'index I est inférieure à m :
   - Pour chaque $j \in \mathrm{J}$, calculer (notation `matlab`)

$$se(\mathrm{I}, j) \coloneqq \mathbf{K}(j,j) - \mathbf{K}(j,\mathrm{I})\mathbf{K}(\mathrm{I},\mathrm{I})^{-1}\mathbf{K}(j,\mathrm{I})^{\mathrm{T}} \tag{4.78}$$

   - Si $\sum_{j\in\mathrm{J}} se(\mathrm{I}, j) < \eta$, arrêter.
   - Sélectionner le meilleur $p_i = \arg\max_{j\in\mathrm{J}} se(\mathrm{I}, j)$
   - Actualiser les index $\mathrm{I} \coloneqq \mathrm{I} \cup \{p_i\}$, $\mathrm{J} \coloneqq \mathrm{J} \setminus \{p_i\}$.

L'initialisation peut être vu comme une sélection du vecteur du Monde avec la plus mauvaise approximation quand le *dictionnaire* est vide, c'est-à-dire quand tous les vecteurs sont projetés sur l'origine. Le critère $se(\mathrm{I}, j)$ à maximiser lors de chaque itération est une borne inférieure du gain apporté en rajoutant b_j au *dictionnaire* :

$$se(\mathrm{I}, j) \leq mse(\mathrm{I}) - mse(\mathrm{I} \cup \{j\})$$

Cette valeur n'est pas calculée pour les vecteurs déjà incorporés au *dictionnaire* parce que le gain est nul pour ces vecteurs. Aussi, les $se(\mathrm{I}, j)$ permettent d'avoir accès après chaque itération à l'erreur commise par l'approximation :

$$mse(\mathrm{I}) = \sum_{j\in\mathrm{J}} se(\mathrm{I}, j) \leq (N - m) \max_{j\in\mathrm{J}} se(\mathrm{I}, j)$$

Appliquer l'algorithme glouton tel qu'il a été formulé ci-dessus et comme il a été fait par [Smola et Schölkopf, 2000, Franc et Hlavac, 2003] conduit à une complexité algorithmique $O(Nm^3)$. Mais cette complexité algorithmique peut être réduite à $O(Nm^2)$ en utilisant des astuces de calcul qui aboutissent sur l'algorithme connu sous le nom de "Décomposition de Cholesky Incomplète" (ICD) introduit par [Fine et Scheinberg, 2001]. L'équivalence entre l'ICD et l'algorithme glouton présenté ci-dessus est clarifiée par [Bach et Jordan, 2005]. La capacité de

mémoire requise pour les deux algorithmes gloutons est $O(Nm)$. À aucun moment l'algorithme ne nécessite de connaître les N^2 valeurs de $\mathbf{K}$; ce serait rédhibitoire pour notre application.

L'algorithme ICD est comme son nom l'indique inspiré de la Décomposition de Cholesky. Ces deux algorithmes sont détaillés dans l'annexe A.2. Nous en donnons ici une vue d'ensemble qui permet de montrer l'adéquation avec notre problématique.

Le but de la décomposition de Cholesky est de calculer la racine carrée $\mathbf{G}$ triangulaire inférieure d'une matrice *symétrique définie positive*

$$\mathbf{K} = \mathbf{G}\mathbf{G}^{\mathrm{T}} \tag{4.79}$$

À chaque itération de la méthode de Cholesky, la $i^{\text{ième}}$ colonne de $\mathbf{G}$ est calculée à partir des valeurs de la $i^{\text{ième}}$ colonne de $\mathbf{K}$. Ce procédé est répété jusqu'à $i = N$ la taille de $\mathbf{K}$, qui est aussi son rang (la méthode s'applique uniquement si $\mathbf{K}$ est inversible). Le but de l'ICD est quant à lui de trouver une approximation d'une racine carrée de $\mathbf{K}$ avec un rang $m < N$ plus faible. Comme dans la décomposition "complète", cette approximation est construite de manière itérative. La différence est qu'à chaque étape i, un pivot p_i est choisi (au lieu de i) de manière à maximiser un critère que l'on note $\xi(\mathrm{I}, j)$:

$$p_i = \arg\max_{j \in \mathrm{J}} \underbrace{\mathbf{K}(j,j) - \mathbf{H}(j,:)\ \mathbf{H}(j,:)^{\mathrm{T}}}_{\xi(\mathrm{I},j)}$$

où
- $\mathrm{I} = \{p_1\ \cdots\ p_{i-1}\}$ est l'index des pivots sélectionnés au cours des étapes précédentes,
- J est son complémentaire ordonné dans $\{1 \cdots N\}$, et
- $\mathbf{H}$ est une matrice triangulaire inférieure de taille $N \times (i-1)$ qui est construite de manière analogue à la méthode de Cholesky.

Ces critères sont en fait choisis de manière à minimiser :

$$\sum_{j \in \mathrm{J}} \xi(\mathrm{I}, j) = \mathrm{tr}\left(\mathbf{L} - \mathbf{H}\mathbf{H}^{\mathrm{T}}\right) \tag{4.80}$$

où $\mathbf{L} = \mathbf{K}([\mathrm{I}\ \mathrm{J}], [\mathrm{I}\ \mathrm{J}])$ est une matrice obtenue en permutant les entrées de $\mathbf{K}$ selon I et J. La matrice triangulaire $\mathbf{H}$ est une approximation de la racine carrée de $\mathbf{L}$. On fait apparaître l'équivalence $\xi(\mathrm{I}, j) = se(\mathrm{I}, j)$ en constatant que l'approximation correspondante est $\mathbf{K} \approx \mathbf{G}\mathbf{G}^{\mathrm{T}}$ avec :

$$\mathbf{G} = \mathbf{K}(:, \mathrm{I})\ \mathbf{K}(\mathrm{I}, \mathrm{I})^{-1/2} \tag{4.81}$$

En effet, on peut montrer [Bach et Jordan, 2005] qu'à chaque itération :

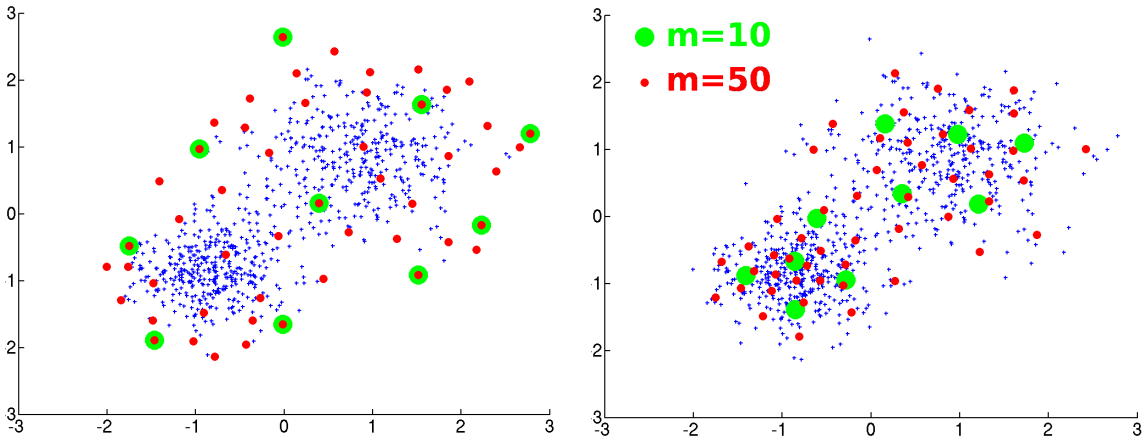
$$\mathbf{H}\mathbf{H}^{\mathrm{T}} = \left[\begin{array}{c|c} \mathbf{K}(\mathrm{I},\mathrm{I}) & \mathbf{K}(\mathrm{J},\mathrm{I})^{\mathrm{T}} \\ \hline \mathbf{K}(\mathrm{J},\mathrm{I}) & \mathbf{K}(\mathrm{J},\mathrm{I})\mathbf{K}(\mathrm{I},\mathrm{I})^{-1}\mathbf{K}(\mathrm{J},\mathrm{I})^{\mathrm{T}} \end{array}\right]$$

L'algorithme ICD s'arrête dès que :
– le nombre de colonnes de $\mathbf{H}$ a atteint une valeur m fixée,
ou
– le critère à minimiser (4.80) atteint d'un seuil $\eta \geq 0$ fixé.

Ce critère peut être identifié d'après (4.81) comme étant la norme $\mathrm{tr}(\mathbf{K} - \mathbf{K}(:,\mathrm{I})\mathbf{K}(\mathrm{I},\mathrm{I})^{-1}\mathbf{K}(:,\mathrm{I}))$. Ce résidu devient nul dès lors que i atteint le rang de $\mathbf{K}$. Dans ce cas l'algorithme s'arrête et permet de calculer une racine carrée de $\mathbf{K}$ même si cette matrice n'est pas de rang plein (contrairement à la méthode de Cholesky classique qui ne peut pas traiter ce cas, à cause de l'absence

de permutations par l'astuce du pivot). Le résidu devient négligeable lorsque l'écart entre le spectre de $\mathbf{L}$ (resp. $\mathbf{K}$) et celui de son approximation $\mathbf{H}\mathbf{H}^{\mathrm{T}}$ (resp. $\mathbf{G}\mathbf{G}^{\mathrm{T}}$) est négligeable. Dans notre application, N est de l'ordre de 10^6, et pour avoir des complexités de calcul raisonnables avec la forme duale réduite des noyaux FSNS/FSMS, il faut que m ne dépasse pas 10^4. La réduction de rang opérée est donc importante, et on peut difficilement espérer que le résidu soit négligeable. Autrement dit, le second critère d'arrêt listé ci-dessus est superflu. Les expériences montreront que nos performances sont améliorées lorsque l'on augmente le rang maximal m. Ce paramètre représente la taille du *Feature Space* et donc la complexité de la modélisation, mais en contrepartie il règle la complexité calculatoire des noyaux FSNS/FSMS.

En définitive, l'ICD est une méthode de technique de réduction d'information adaptée au noyau. Elle permet de faire du *clustering* dans le *Feature Space*. Fig.4.3 met en avant les similarités et les différences entre un *dictionnaire* obtenu en sortie d'une ICD avec un noyau RBF Gaussien, et un *dictionnaire* obtenu par une méthode classique de quantification vectorielle dans l'espace d'entrée. Les données bidimensionnelles ont été générées artificiellement à partir d'un GMM à deux Gaussiennes, et la taille du *dictionnaire* a été fixée à 20 puis 50.



(a) Codebook de taille m obtenu avec une ICD en prenant un noyau RBF Gaussien. Il correspond à un quadrillage de l'espace d'entrée

(b) Codebook de taille m obtenu avec un algorithme classique de *clustering K-means*.

Fig. 4.3 - Allure des *dictionnaires* résultant d'une ICD et d'un algorithme de *K-means*.

Chapitre 5

Mise en œuvre et évaluation expérimentale des noyaux de séquences

Sommaire

Ce chapitre présente une série d'expériences qui montrent que les SVMs à noyaux de séquences peuvent donner des performances aussi bonnes voire meilleures que les systèmes génératifs UBM-GMM, avec une complexité calculatoire comparable. Insistons sur le fait que toutes les approches qui ont été testées dans ce chapitre impliquent des temps de calcul raisonnables. Avec des processeurs de type `P4 2.6 GHz multithread`, la plupart des systèmes testés mettent entre une heure et une journée pour traiter environ 1200 séquences chacune testées sur 15 locuteurs cibles[19] (1200 séquences correspondant à 100 heures d'enregistrement et 40 heures de parole pure).

Les protocoles de développement et d'évaluation des différentes méthodes de modélisation pour la vérification du locuteur sont décrits dans la section 5.1. Ils mettent en jeu de grandes bases de données NIST et des modules d'extraction de paramètres classiques dans le traitement de la parole. La section 5.2 décrit les détails techniques des systèmes de référence qui serviront de repères pour juger des performances des modèles à noyaux FSNS, et autres noyaux de séquences. Ces systèmes sont :

1. Un système UBM-GMM (modélisation générative classique),
2. Un système SVM à noyau vectoriel, dont les faibles performances mettent en valeur la nécessité d'utiliser les noyaux de séquences (d'ensembles de vecteurs),
3. Un système SVM à noyau GLDS (cas particulier des noyaux FSNS).

La section 5.3 montre une série d'expériences relatives au développement des noyaux FSNS introduits dans le chapitre précédent. Elles montrent comment régler ce type de noyau pour un problème de vérification du locuteur. Ensuite, la section 5.4.1 montre l'application d'autres noyaux de séquence, basés sur la modélisation probabiliste. Ces noyaux ont été introduits de façon théorique dans le chapitre 3 (§3.3 à §3.5). Dans ce chapitre, nous donnons leur expression dans le cas de les GMMs adaptés (à partir d'un UBM), et nous fournissons quelques éléments permettant de régler ces noyaux pour une tâche de vérification du locuteur par SVM. Enfin la section 5.5 présente les résultats d'évaluation obtenus par les systèmes les plus performants avant de conclure.

[19] $1200 \times 15 = 18\,000$ tests, sans compter les délais de développement du système et d'apprentissage des modèles.

5.1 Protocole expérimental

5.1.1 Description des données

Corpus de développement

Pour développer les différents systèmes de vérification du locuteur, nous avons utilisé le protocole [BioSecure, 2005]. Dans ce protocole, les données sont issues des évaluations [NIST SRE, 2003] et [NIST SRE, 2004] dans les conditions standards (1side-1side "*core test*" pour NIST'04). Dans ces conditions, tous les enregistrements (d'apprentissage et de test) incluent environ 2 minutes de parole prononcée par un seul et même locuteur, provenant d'une conversation téléphonique de 5 minutes. Le genre homme/femme du locuteur est connu pour chaque enregistrement (étiquetage fourni par NIST) et aucun test inter-genres n'est réalisé[20]. Pour exploiter au mieux cette information *a priori*, la démarche classique consiste à différencier l'apprentissage des modèles selon le genre. Afin de réduire les délais de développement et d'évaluation, nous avons arbitrairement décidé de nous limiter dans toutes nos expériences aux locuteurs de genre féminin présents dans les bases NIST.

Les données NIST'03 sont issues de la base SWITCHBOARD et les données de NIST'04 de la base MIXER. Ces deux bases de données audio offrent une large variété de conditions d'acquisition du signal. Notamment, la transmission téléphonique peut être filaire ou mobile, avec différents formats de transmission. Les conditions rencontrées dans NIST'03 sont assez différentes de celles rencontrées dans NIST'04. En particulier, contrairement à NIST'03,

- NIST'04 fait intervenir des locuteurs bilingues. Même si la langue majoritairement utilisée est l'anglais américain, certains enregistrements contiennent des phrases prononcées dans d'autres langues (arabe, espagnol).
- Certains enregistrements de test de NIST'04 ne proviennent pas de conversations téléphoniques (microphones pour ordinateur, etc.), et font ainsi intervenir des conditions d'acquisition bien différentes du reste.
- Les données fournies par NIST'04 n'ont pas subis le même pré-traitement pour l'atténuation de l'écho[21] et aucun retrait automatique des silences n'a été appliqué.

Dans tous les cas, la qualité des enregistrements téléphoniques est dégradée par plusieurs sources de bruits, dont les interférences d'ondes et l'écho qui persiste malgré les diverses précautions lors du pré-traitement. Aussi le contenu phonétique est très variable : avant l'enregistrement d'une conversation téléphonique pour la collecte de données NIST, des sujets de conversations sont imposés aux locuteurs avec un soucis de diversité dans les thèmes abordés.

Dans le protocole de développement BioSecure que nous utilisons, les données sont organisées de la même manière que présenté en §2.1.2. À titre indicatif, les caractéristiques sur la population des femmes sont les suivantes :

- **(A) Le corpus du monde** est constitué de 283 enregistrements, dont 207 proviennent des locuteurs de NIST'03, et 76 d'une sous-population de locuteurs de NIST'04. Cela équivaut approximativement à 9 heures de parole. Après application du retrait automatique des

[20] sauf quelques erreurs de protocole.

[21] L'écho se traduit par la perception de la voix d'un des deux intervenants de la conversation, sur le canal correspondant à l'autre intervenant. Le même phénomène se produit lorsqu'on entend sa propre voix dans le téléphone portable. À cause de cet effet d'écho, la séparation de deux locuteurs en deux canaux droite/gauche n'est pas parfaite.

zones de faibles énergies (silence et bruit), environ deux millions de trames correspondant à 6 heures de paroles sont sélectionnées.
- **(B) Le corpus d'imposteurs** est constitué de 113 enregistrements de locuteurs de NIST'04.
- **(C) Le corpus de validation** fait intervenir 181 locuteurs cibles de NIST'04 (avec un enregistrement par locuteur pour l'apprentissage), et 368 autres enregistrements de NIST'04 pour les tests. L'évaluation de validation est composée de 7062 essais, dont 758 positifs (*i.e.* faisant intervenir un même locuteur).

Soulignons que les populations de locuteurs intervenant respectivement dans les corpus (A), (B) et (C) sont distinctes.

À l'origine, le protocole BioSecure a été conçu en s'inspirant des phases de développement d'un système UBM-GMM. Aussi les rôles des corpus (A) et (B) sont limpides pour ce système : les données du Monde (A) servent à l'apprentissage de l'UBM et les données imposteurs (B) servent à normaliser les scores (T-Norm dans nos expériences). Pour les systèmes discriminatifs (SVMs) par contre, les rôles respectifs de (A) et (B) ne sont pas clairs. Bien entendu, l'estimation de statistiques *a priori* sur les données d'entrée (pour les normalisations) se fait sur le corpus du Monde (A). Le problème réside plutôt dans le choix des séquences imposteurs qui vont servir d'exemples négatifs (étiquette -1) pour l'apprentissage des modèles locuteurs. Pour lever l'ambiguïté, nous sommes parti du constat suivant. Les données du Monde (A) jouent en fait un double rôle dans un système UBM-GMM :

1. En phase d'apprentissage, l'UBM sert en effet à initialiser l'apprentissage MAP des modèles locuteurs. Ainsi les données du Monde sont d'abord considérées comme des données non étiquetées servant à estimer des paramètres indépendants du locuteur (*a priori*).

2. En phase de test, l'UBM représente la distribution de l'hypothèse de refus dans le rapport de vraisemblance (densité *a posteriori* des imposteurs). Les données du Monde sont donc aussi considérées comme des données imposteurs en phase de test.

Partant de là, nous avons choisi de faire jouer des rôles analogues aux séquences du Monde (A) pour les systèmes discriminatifs SVMs, en les présentant comme exemples négatifs aux algorithmes d'apprentissage. Concernant le corpus (B), nous verrons plus tard qu'il est préférable de l'utiliser pour fournir des exemples négatifs supplémentaires, plutôt que pour une normalisation des scores du type T-Norm. Jusqu'à §5.3.4, les systèmes SVMs présentés utiliseront simplement les données du Monde (A) comme corpus d'imposteurs pour l'apprentissage, et ne procéderont à aucune normalisation des scores. À partir de §5.3.5, les systèmes SVMs présentés utiliseront l'ensemble des données (A) et (B) pour l'apprentissage.

Corpus d'évaluation

Les systèmes sont évalués sur [`NIST SRE`, 2005] dans les conditions standard ("`1conv-1conv`"). Les données sont issues de la base `MIXER` tout comme les données NIST'04.

L'évaluation met en jeu pour les locuteurs femmes :
- 372 locuteurs cibles,
- 1 287 séquences tests,
- 17 794 essais dont 1 551 positifs.

5.1.2 Critères d'évaluations

Même si les EER seront donné à titre indicatif dans nos descriptions de résultats, le véritable critère d'évaluation des systèmes considéré sera la DCF utilisés dans toutes les évaluations [NIST SRE, 1997], ..., [NIST SRE, 2006]. Dans toutes ces évaluations, les paramètres de la DCF sont :

- coût de faut rejet $\tau_{\text{FR}} = 10$
  et probabilité *a priori* d'apparition d'un locuteur cible $P_{\text{loc}} = 0.01$
- coût de fausse alarme $\tau_{\text{FA}} = 1$
  et probabilité *a priori* d'apparition d'un imposteur $P_{\text{imp}} = 1 - P_{\text{loc}} = 0.99$

La DCF considérée s'exprime alors en fonction des taux de faux rejets FR% et de fausses alarmes FA% selon :

$$\text{DCF} = \underbrace{\tau_{\text{FR}} P_{\text{loc}}}_{0.1} \text{FR}\% + \underbrace{\tau_{\text{FA}} P_{\text{imp}}}_{0.99} \text{FA}\%$$

Une telle fonction correspond à des point de fonctionnement optimaux en haut à gauche de la courbe DET (faibles taux de fausses alarmes). Elle est adaptée pour des applications comme le filtrage d'appels téléphoniques pour rechercher un criminel. Dans un tel contexte en effet, il est préférable d'affecter un coût élevé aux faux rejets ($\tau_{\text{FR}} > \tau_{\text{FA}}$), tout en tenant compte du fait que la probabilité d'apparition du criminel est faible ($P_{\text{loc}} \ll P_{\text{imp}}$).

En phase de développement, le critère à optimiser est le minimum de la DCF sur le corpus de validation. Pour une famille de systèmes donnée, les réglages en phase de développement sont faits de manière à minimiser ce critère. Les réglages optimaux, qui seront choisis pour l'évaluation, seront marqués par un ✔ dans les tables de performances. Le critère de minimum de DCF sert aussi à fixer le seuil de décision (point de fonctionnement optimum). En phase d'évaluation, le critère de performance mesuré est la valeur de la DCF au seuil de décision fixé (point de fonctionnement réel).

5.1.3 Pré-traitement

Étant donné que le but de ces recherches est d'évaluer les stratégies de modélisation, nous nous sommes limité à utiliser une pré-traitement classique. Comme dans la plupart des systèmes classiques de vérification du locuteur, nous avons opté pour une extraction de coefficients cepstraux sur des fenêtres de taille fixe avec un pas d'échantillonnage de 10ms. Les enregistrements sont ainsi convertis en séquences vectorielles de longueurs variables. Pour les enregistrements NIST dans les conditions standard, les longueurs des séquences sont de l'ordre de 9000 vecteurs après suppression automatique des zones de faible amplitude. Cela correspond à 1min30 de parole pure sélectionnée par locuteur.

Plusieurs paramétrisations ont été testées pour chaque système durant nos études. Au vu des performances sur le corpus de validation, il en ressort que le choix de la meilleure paramétrisation dépend de la stratégie de modélisation utilisée ensuite. La même ambiguïté a été soulignée par [Campbell et al., 2006a] dans la comparaison entre un système UBM-GMM et un système SVM avec noyau GLDS. En fait, le réglage automatique pour une paramétrisation optimale reste un problème ouvert, et les divers choix techniques se font à l'heure actuelle par validation.

Pour le système UBM-GMM et pour les systèmes SVMs où les séquences sont représentées par des GMMs, la meilleure paramétrisation est celle que nous noterons "LFCC". Pour les autres systèmes SVMs, elle est celle que nous désignerons par "MFCC". Nous décrivons maintenant ces deux types de paramétrisation.

Module de paramétrisation LFCC

Nous avons utilisé pour l'extraction de paramètres LFCC le logiciel [`SPRO`, 2004](version `4.0`). Pour le retrait des silences et la normalisation des paramètres issus de ce premier sous-module, nous avons utilisé l'outil de développement [`ALIZE`, 2005, Bonastre et al., 2005](version `1.2`). Les paramètres obtenus sont des vecteurs de dimension 33.

[`SPRO`] Le signal numérique de départ est filtré de manière à supprimer les fréquences au-dessous de 300 Hz ou au-dessus de 3400 Hz, sachant que la bande passante des appels téléphoniques est [300-3400 Hz]. Un filtre numérique de pré-accentuation des aigus est appliqué avec un coefficient de 0.97[22]. L'énergie E et 16 coefficients cepstraux $LFCC$ (du 2^{nd} au $17^{\text{ème}}$) et leurs dérivées ΔE et $\Delta LFCC$ sont alors extraits sur des fenêtres de Hamming de 20ms, à pas d'échantillonage régulier de 10ms.

[`ALIZE`] Le processus de sélection des trames utilisé (exécutable `EnergyDetector.exe` de [`LIA SpkDet`, 2005]) est basé sur une classification non supervisée à partir d'une modélisation tri-Gaussienne de l'énergie sur chaque enregistrement. Le paramètre d'énergie E ne sert que pour cette étape de retrait des silences (trames à basse énergie). Les autres paramètres sont ensuite normalisés globalement par centrage/réduction (exécutable `NormFeat.exe` de [`LIA SpkDet`, 2005]), de manière à avoir une moyenne nulle et une variance unitaire sur sur la totalité de chaque enregistrement.

Module de paramétrisation MFCC

Nous avons utilisé pour l'extraction de paramètres MFCC le logiciel [`HTK`, 2002](version `3.0`). Les techniques d'amélioration du signal d'entrée sont identiques à celles des paramètres LFCC. L'énergie E, les 12 premiers coefficients cepstraux $MFCC$ (à partir de 22 filtres Mel) et leurs dérivées ΔE et $\Delta MFCC$ sont alors extraits sur des fenêtres de Hamming de 16ms (toutes les 10ms). La suppression des silences se fait de la manière manière que pour le module LFCC, et l'énergie est ensuite supprimée pour avoir des vecteurs de dimension 35.

La normalisation utilisée sur ces paramètres $MFCCs$ est le *Feature Warping* [Pelecanos et Sridharan, 2001], implémenté par nos soins. Cette technique consiste à d'abord ranger chaque type de paramètre x_u par ordre décroissant sur des fenêtres de $N = 301$ trames (3 secondes) centrées en chaque trame t. Ensuite la valeur normalisée $\tilde{x}_{u,t}$ affectée à la trame centrale (de paramètre initial $x_{u,t}$) est déterminée en fonction de son rang R_t (égal à 1 s'il s'agit

[22] Après application du filtre à un signal d'amplitudes y_t, on récupère les valeurs $y'_t = y_t + 0.97y_{t-1}$

de la valeur la plus grande). Elle est obtenue en résolvant numériquement

$$\text{erf}(\tilde{x}_{u,t}) = \frac{N+\frac{1}{2}-R_t(x_{u,t})}{N} \quad \text{(valeurs entre } \tfrac{1}{2N} \text{ et } 1-\tfrac{1}{2N}\text{)}$$
$$\text{où} \quad \text{erf}(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{z^2}{2}} \, \mathrm{d}z \quad \text{est une fonction croissante.}$$

Étant donné que le rang R_t est quantifié dans $[1, \cdots, N]$, les paramètres le sont aussi. Ils peuvent prendre $N = 301$ valeurs qui peuvent être pré-calculées. Le but d'une telle normalisation est de faire épouser localement à chaque paramètre d'entrée une distribution normale de moyenne nulle et d'écart-type unitaire. Elle a aussi pour effet de borner les valeurs d'entrée, ce qui est un plus pour la stabilité des SVMs.

5.2 Développement des systèmes de référence

5.2.1 Système UBM-GMM

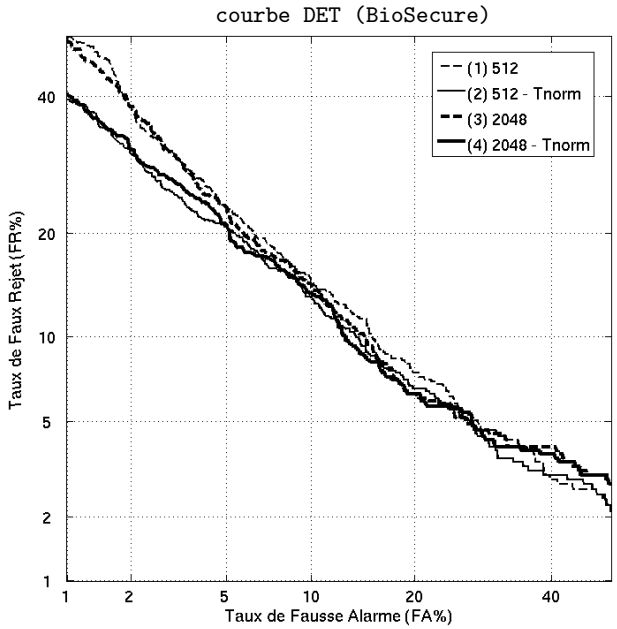
Choix techniques

Le système de référence UBM-GMM a été implémenté en utilisant les exécutables du package [`LIA SpkDet`, 2005] basé sur les sources [`ALIZE`, 2005]. Le module de paramétrisation considéré est le module LFCC décrit plus haut. Les différents paramètres intervenant dans l'apprentissage des modèles et dans les calculs de scores ont été réglés d'après la présentation de `LIA_SpkDet` au workshop BioSecure par [Scheffer, 2005]. Ils ont été choisis par validation pour le module de pré-traitement LFCC. Les détails sur les choix des différents coefficients sont listés ci-dessous avec les noms des exécutables utilisés (package `LIA_SpkDet`).

- *Développement* (`TrainWorld.exe`)
  L'UBM est appris à partir du corpus du Monde (A). Différents nombres de composantes Gaussiennes ont été testées parmi les puissances de deux (126, ..., 2048). À chaque itération de l'algorithme EM, un *variance flooring/ceiling* (§2.2.2) est appliqué aux matrices de covariance diagonales de manière à ce que les variances soient comprises entre 0.5 et 10 fois la variance des entrées. Le choix de ces paramètres est évidemment lié à la normalisation (centrage/réduction) des vecteurs d'entrée.
- *Apprentissage des modèles* (`TrainTarget.exe`)
  Pour l'apprentissage des modèles clients, un "*relevance factor*" $r = 14$ a été utilisé dans (2.3) pour l'adaptation des moyennes de l'UBM.
- *Attribution de scores* (`ComputeTest.exe`)
  En phase de test, les rapports de vraisemblance sont calculés à partir des 10 meilleures Gaussiennes de l'UBM ("10-*best scoring*", §2.2.3)
- *Normalisation des scores* (`ComputeNorm.exe`)
  Enfin, on applique une T-Norm aux scores (§2.1.6, en utilisant les statistiques obtenues par chaque séquence de test sur les modèles imposteurs à partir du corpus de développement (B). L'entraînement de ces modèles imposteurs utilise les mêmes réglages que pour les modèles clients (`TrainTarget.exe`).

Résultats de développement

Fig.5.1 montre les performances atteintes en phase de développement par le système UBM-GMM pour 512 et 2048 composantes Gaussiennes, avec et sans normalisation de scores T-Norm. Comme il a été observé par [Barras et Gauvain, 2003], la T-Norm a pour effet d'améliorer les performances dans les régions DET de fausses alarmes (où se situe le minimum de la DCF). Comme le montre ces expériences, la différence de performances en fonction du nombre de composantes des modèles n'est pas nette, du moment que ce nombre est d'ordre élevé. En fait, même si sans normalisation de scores, les GMMs à 512 composantes donnent de moins bonnes performances, ce n'est pas le cas lorsque l'on utilise la T-Norm. Soulignons qu'un tel résultat ne peut pas faire l'objet de généralisation et que des expériences menées par d'autres montrent que l'augmentation du nombre de Gaussiennes améliore les performances du moment que certaines précautions sont prises. Pour nos expériences, nous nous bornerons à prendre comme système génératif de référence l'UBM-GMM à 512 composantes avec T-Norm.



Systèmes UBM-GMM
Tailles de modèles / Normalisation de scores :
(1) 512 Gaussiennes
(2) 512 Gaussiennes + T-Norm
(3) 2048 Gaussiennes
(4) 2048 Gaussiennes + T-Norm

		EER(%)	minDCF(x10^{-3})
	(1)	12.53	56.4
✔	(2)	11.48	49.1
	(3)	11.89	57.5
	(4)	11.76	49.5

Fig. 5.1 - Performances du système UBM-GMM en phase de développement.

5.2.2 Système SVM avec noyaux de vecteurs

Problématique

Avant de présenter les niveaux de performance des SVMs avec noyaux de séquences, nous essayons l'approche "vectorielle" (§2.3.1). Elle consiste à apprendre des modèles SVMs au niveau des trames de parole, et à attribuer à une séquence de test un score qui soit une combinaison des scores obtenus sur ses trames. Même si cette démarche paraît triviale, sa mise en pratique pour la vérification du locuteur sur une grande base de donnée ne l'est pas.

Le principal problème est la complexité calculatoire. Si N est le nombre d'observations (vec-

teurs) disponibles pour le développement, alors nous rappelons que :

- La complexité de l'algorithme d'apprentissage SVM est au mieux $O(N^2)$, avec une capacité de mémoire nécessaire en $O(N^2)$.
- La taille des modèles est aussi linéaire en N. Il en est donc de même pour la complexité d'attribution d'un score à une trame.

Dans notre application N est de l'ordre de 10^6, et il faudrait des puissances de calculs et des capacités de mémoire énormes pour faire une évaluation NIST dans des temps raisonnables.

Choix et résultats de développement

De manière à limiter les problèmes de complexité, nous avons opté pour l'approche proposée par [Lei et al., 2005] dont une vue d'ensemble a été donnée en §2.3.1. C'est un des rares travaux sur les SVMs vectoriels pour la reconnaissance du locuteur dans ces dernières années. Les expérimentations de [Lei et al., 2005] ont été faites sur la base de données YOHO pour une tâche d'identification du locuteur. Ce corpus fait intervenir moins de données que les bases NIST, avec des enregistrements de meilleure qualité et des contenus phonétiques moins variés. Les détails algorithmiques sont listés ci-dessous.

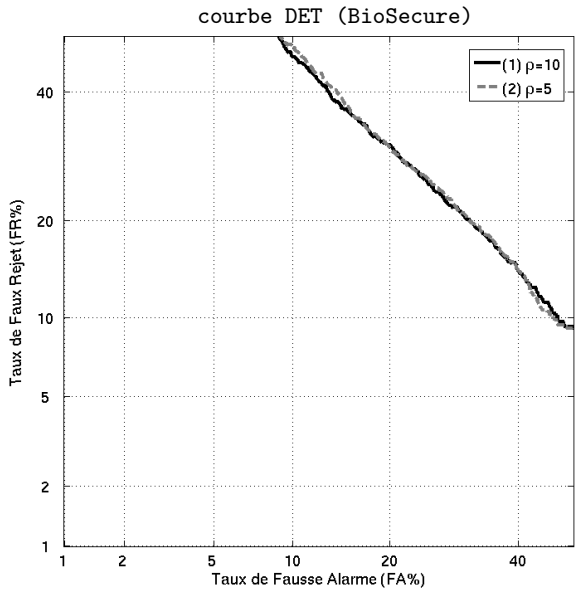
- *Développement*
  Un algorithme de Quantification Vectorielle est appliqué aux données du Monde, de manière à obtenir un dictionnaire de taille m fixée arbitrairement. Nous avons pour cela utilisé l'algorithme de quantification détaillé dans [Linde et al., 1980].
  D'après [Lei et al., 2005], les performances finales s'améliorent avec l'augmentation de m. Bien que les auteurs aient testé jusqu'à $m = 32$ sur la base YOHO, nous nous sommes limité à $m = 16$ sur nos bases de données NIST à cause des temps de calcul.

- *Apprentissage des modèles*
  Les séquences d'apprentissage sont découpées en régions par classification non supervisée à partir du dictionnaire. La région d'appartenance de chaque trame correspond à l'élément du dictionnaire le plus proche, selon une distance euclidienne sur les vecteurs de paramètres. Chaque trame est donc étiquetée par un entier entre 1 et m, qui détermine l'acheminement vers un "expert" SVM donné. Ainsi m experts sont entraînés pour constituer un mélange de SVMs vectoriels.
  Dans nos expériences, nous avons choisi d'utiliser un noyau RBF Gaussien. Le paramètre d'étalement de la Gaussienne qui a fournit les meilleures performances dans nos expériences est $\rho = 5$ qui correspond à la valeur recommandé ρ_0 dans l'équation (3.15) de §3.2.2.
  Précisons que le nombre d'exemples en entrée de l'algorithme SVM reste très élevé, même après une réduction des effectifs par $m = 16$. La solution préconisée par [Lei et al., 2005] est de rejeter, lors de la classification non supervisée en régions, les trames les plus distantes des éléments du dictionnaire. Dans nos expériences, nous avons appliqué une telle réduction de taille sur les séquences imposteurs. Pour chacune de ces séquences (longueur d'origine de l'ordre de 10^6), nous n'avons retenu pour chaque région que les 50 trames les plus proches de l'élément de référence du dictionnaire.

- *Attribution de scores*
  Pour attribuer un score à une séquence test, les trames sont d'abord rangées en m régions avec le même classifieur que lors de l'apprentissage. Chaque trame reçoit alors un score en sortie de l'expert SVM correspondant. Dans nos expériences, nous avons comparé deux

stratégies classiques pour combiner ces scores, qui sont les suivantes.

1. *Stratégie simple :* Le score d'une séquence est la moyenne arithmétique des scores sur les trames.
2. *Stratégie probabiliste :* Pour chaque expert SVM de chaque locuteur cible, la "probabilité d'apparition du locuteur sachant le score $f(\mathbf{x}_t)$ affecté à une trame t" est modélisée par une régression logistique : $p(\text{loc}|f(\mathbf{x}_t)) = \frac{1}{1+e^{(Af(\mathbf{x}_t)+B)}}$. Les coefficients $\{A, B\}$ de cette régression sont appris d'après l'algorithme fourni par [Platt, 2000], à partir des mêmes données de développement qui ont été utilisées pour entraîner les SVMs. Finalement, le score d'une séquence est la moyenne arithmétique des log-probabilités *a posteriori*, de manière analogue à un système UBM-GMM.



Mélanges de SVMs avec noyau RBF Gaussien
Stratégie d'attribution de score :
(1) Moyenne des scores sur les trames
(2) Moyenne des log-probabilités empiriques (logit)

	EER(%)	minDCF($\times 10^{-3}$)
(1)	25.30	95.1
(2)	25.46	96.1

Fig. 5.2 - Performances d'un mélange de SVM à noyaux vectoriels en phase de développement.

Fig.5.2 montre les résultats pour un mélange de $m = 16$ experts SVMs à noyau RBF Gaussien ($\rho = 1.5$), selon ces deux stratégies de combinaison des scores vectoriels. D'après les résultats montrés ici, la stratégie probabiliste n'améliore pas les performances bien qu'elle requiert un alourdissement des modules d'apprentissage et d'attribution de scores. Soulignons toutefois que l'approche simple (moyenne des scores SVM) n'est pas raisonable d'un point de vue théorique : les scores de 2 SVMs différents ne peuvent normalement pas être comparés car ils représentent la distance à la marge dans des espaces différents.

Aussi, les meilleures résultats que nous avons pu atteindre avec l'approche vectorielle sur les données de validation n'atteignent pas les niveaux de performance des SVMs à noyaux de séquences (présentés ci-après). De ce fait, nous abandonnons cette approche pour les comparaisons sur le corpus d'évaluation qui suivront.

5.2.3 Système SVM avec noyau GLDS

Généralités sur les noyaux de séquences pour la vérification du locuteur

Lors du développement d'un SVM pour un problème classique de classification binaire, il est coutume de régler le compromis biais-variance C d'apprentissage et les paramètres du noyau par validation croisée. Utiliser un noyau de séquence dans notre protocole de vérification du locuteur est particulier sur ce point-là. Le fait de n'avoir qu'une séquence par locuteur cible (exemple positif) rend peu légitime une validation croisée pour choisir les paramètres de manière spécifique à chaque locuteur. Au lieu de cela, la phase de développement d'un système SVM à noyau de séquences consiste à optimiser les paramètres globalement, au vu des performances sur le corpus de validation. Les paramètres sont alors fixés pour l'apprentissage de tous les modèles locuteurs cibles.

Considérer des séquences en entrée des algorithmes amène à se questionner sur la définition de la "séquence". Dans notre contexte de vérification du locuteur, nous avons été amené à définir l'unité de séquence comme suit :

> Une "séquence" désigne un ensemble d'observations émises par un seul et même locuteur, dans des conditions d'enregistrement identiques.

Ainsi, un lot de séquences différentes émises par un même locuteur renseigne sur les variations indépendantes du locuteurs (conditions d'enregistrement, variabilités intralocuteurs dans le moyen terme, etc.). A contrario une seule séquence *même découpée en plusieurs* ne comporte que peu d'information de ce type. Pour appuyer ce dernier point, nous avons comparé un système GLDS classique avec un système où les séquences d'apprentissage sont découpées, de manière à :

- augmenter le nombre d'exemples (séquences) en entrée de l'algorithme d'apprentissage.
- diminuer le déséquilibre entre le nombre d'exemples négatifs et le nombre d'exemples positifs, en découpant seulement les extraits correspondant aux locuteurs cibles (et non aux imposteurs).

Comme le montre le tableau de performances suivant, aucun de ces deux artifices ne semble rendre l'algorithme d'apprentissage SVM plus robuste, vu le faible écart relatif entre les performances.

Types de séquences découpées	Nombre de morceaux par séquence	EER(%)	minDCF(x10^{-3})
Aucune	(1)	13.29	55.6
`Loc`, `Imp`	5	13.32	52.9
`Loc`, `Imp`	10	12.86	53.0
`Loc` *seulement*	5	13.54	54.2
`Loc` *seulement*	10	13.56	56.4

`Loc` : Séquence du locuteur cible
`Imp` : Séquences imposteurs

Afin de combler le déséquilibre entre exemples négatifs / positifs inhérent à notre protocole de vérification du locuteur, nous avons aussi tenté de réajuster en conséquence le coût C des

erreurs pour l'algorithme d'apprentissage comme préconisé par [Lin et al., 2002]. En affectant un coût plus important aux erreurs de type "faux rejets" qu'aux erreurs de type "fausses acceptations" ($C^{+1} > C^{-1}$ au lieu d'un C unique), nous n'avons pas noté de changement notable dans les performances. La même chose a été observée empiriquement par [Mariéthoz, 2006] qui l'explique par le fait que le problème d'apprentissage soit séparable. En effet, le nombre de séquences d'apprentissage (de l'ordre de 300) étant de manière générale nettement inférieure à la dimension du *Feature Space*, il est facile de ne commettre aucune erreur sur les données d'apprentissage. Dans notre protocole, il s'agit de distinguer *la* (seule) séquence du locuteur cible des séquences imposteurs. Dans ce cas, étant donné que les données d'apprentissage n'offrent aucune information sur la variabilité des entrées clients :

- les meilleurs classifieurs sont ceux qui séparent parfaitement les données d'apprentissage (biais nul).
- le critère de marge des SVMs permet alors de choisir le meilleur classifieur parmi ceux qui remplissent cette condition (minimisation de la variance).

Au lieu du "compromis biais-variance" habituellement visé dans les situations standards d'apprentissage, on recherche ici à minimiser la variance tout en *annulant* le biais. Ceci correspond en pratique à un paramètre de coût C élevé. Les expériences confirment que les performances des systèmes SVMs à noyaux de séquences sur le corpus de validation sont améliorées lorsque l'on augmente ce paramètre C. Une fois une certaine valeur de $C_\text{o} < C$ atteinte, aucune erreur d'apprentissage n'est commise et les modèles appris restent les mêmes (ceux qui maximisent la marge "dure"). En reconsidérant une distinction entre le coût de faux rejets C^{+1} et le coût de fausses alarmes C^{-1}, le réglage optimal pour l'apprentissage correspond alors à $C^{+1}, C^{-1} > C_\text{o}$, et le rapport C^{+1}/C^{-1} n'influe pas sur le choix du modèle lorsque le biais est nul.

Choix et résultats de développement

Le noyau GLDS, tel qu'il est présenté en §4.1, présente assez peu de paramètres libres à régler lors du développement. Nous rappelons que cette rigidité limite fondamentalement les perspectives d'amélioration d'un système basé sur ce noyau, bien qu'elle ait l'avantage d'alléger la phase de développement. Les paramètres libres sont :

- le degré p de l'expansion polynomiale,
- le paramètre C de compromis biais-variance dans l'apprentissage SVM.

Concernant le premier point, les expériences montrent que les performances atteintes avec un degré $p = 3$ sont de loin meilleures à celles atteintes avec un degré $p = 1$ ou 2. Les expériences n'ont pas pu être menées au-delà d'un degré $p = 3$ à cause de la complexité calculatoire du noyau GLDS. Concernant le second point, les expériences montrent que les meilleures performances correspondent à paramètre de coût C élevé en apprentissage. Cette tendance a été discuté dans le paragraphe précédent.

Pour aller au-delà dans la complexité de la modélisation, nous avons essayé d'appliquer des noyaux non linéaires aux *expansions* polynomiales normalisées. En particulier, nous avons utilisé un noyau RBF Gaussien $\exp\left(\frac{-1}{2\rho^2}\left(\kappa(\mathbf{X},\mathbf{X})^2 - 2\kappa(\mathbf{X},\mathbf{Y}) + \kappa(\mathbf{Y},\mathbf{Y})^2\right)\right)$ au lieu du produit scalaire utilisé dans le noyau GLDS classique $\kappa(\mathbf{X},\mathbf{Y})$. Aucune tentative d'amélioration dans cette direction n'a aboutit.

Enfin, nous avons évalué les variations de performance selon différentes stratégies pour normaliser, de façon à pouvoir apprécier l'effet de la normalisation puisqu'elle a été notre point

de départ pour définir les noyaux FSNS. Pour cette étude, nous nous sommes focalisé sur une approximation diagonale de la matrice $\mathbf{S}_\mathcal{B}$ des seconds moments, comme il est commun de faire dans les systèmes GLDS classiques [Campbell, 2004, Mariéthoz et Bengio, 2005]. Quatre façons de régler cette matrice ont été comparées. Elles sont listée ci-dessous (on note s_u les valeurs diagonales de $\mathbf{S}_\mathcal{B}$).

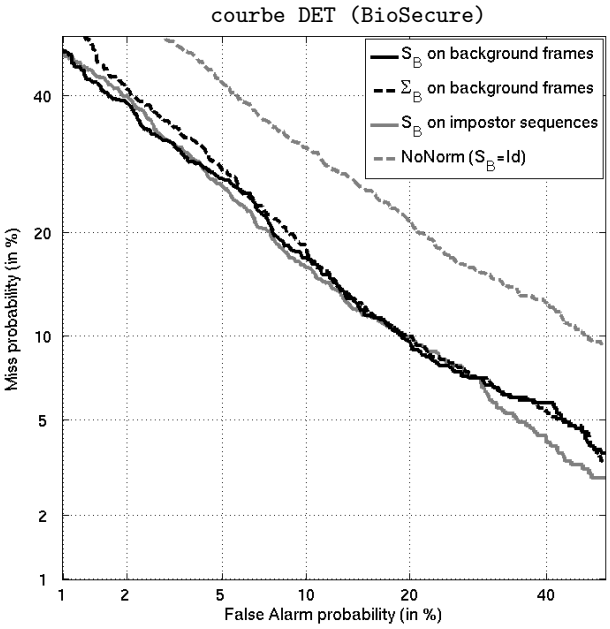
1. *Approche classique*
   Les seconds moments sont estimés sur les expansions du Monde.
   $$s_u = \frac{1}{N}\sum_{i=1}^{N} \phi_u^3(\mathbf{b}_i)^2$$
   où ϕ_u^3 désigne la $u^{\text{ième}}$ composante de l'expansion polynomiale (de degré 3), et où $\{\mathbf{b}_i\}_{i=1\cdots N}$ sont les vecteurs du Monde ($N = 2{\times}10^6$ trames).
2. *Approche centrée*
   Les variances sont estimées sur les données du Monde, au niveau des trames.
   $$s_u = \sigma_u^2 = \frac{1}{N}\sum_{i=1}^{N} \phi_u^3(\mathbf{b}_i)^2 - \left(\frac{1}{N}\sum_{i=1}^{N} \phi_u^3(\mathbf{b}_i)\right)^2$$
3. *Approche grossière*
   Les seconds moments sont estimés sur les expansions moyennes des séquences imposteurs.
   $$s_u = \frac{1}{S}\sum_{i=1}^{S} \overline{\phi}_u^3(\mathbf{A}_i^{(-)})^2$$
   où $\overline{\phi}_u^3$ désigne la $u^{\text{ième}}$ composante de l'expansion polynomiale moyenne, et où $\{\mathbf{A}_i^{(-)}\}_{i=1\cdots S}$ sont les séquences imposteurs ($S = 283$ séquences).
4. *Approche biaisée*
   Aucune normalisation n'est appliquée ($s_u = 1$).

En pratique, la normalisation est intégrée à l'expansion de séquences en multipliant par $(s_u)^{-1/2}$ la $u^{\text{ième}}$ composante de l'expansion moyenne. En effet, normaliser les entrées par ${\mathbf{S}_\mathcal{B}}^{-1/2}$ puis calculer un produit scalaire est plus avantageux en terme de complexité calculatoire que d'appliquer ${\mathbf{S}_\mathcal{B}}^{-1}$ dans noyau linéaire généralisé[23]. L'expansion polynomiale est dans nos expériences de taille $D = \frac{(25+3)!}{25!\ 3!} = 3276$.

Fig.5.3 présente les résultats des systèmes SVMs à noyau GLDS, avec les quatre types de normalisation citées ci-dessus. On peut dresser plusieurs conclusions intéressantes :

- *Comparaison* `(4)` *et autres :* L'absence de normalisation aboutit à un classifieur peu robuste.
- *Comparaison* `(1)/(2)` *:* Normaliser par la matrice de covariance, même si elle est plus familière que les seconds moments pour la normalisation, n'améliore pas la robustesse. Dans nos expériences, le centrage dégrade même légèrement les performances.
- *Comparaison* `(1)/(3)` *:* Calculer les valeurs diagonales de la matrice de normalisation à l'échelle des séquences d'apprentissage plutôt qu'à l'échelle des trames ne dégrade pas les performances malgré la plus grande incertitude sur l'estimation statistique. Cette conclusion nous servira pour alléger la complexité de calcul du noyau de Fisher.

[23] Supposons avoir déjà calculé les expansions (non normalisées) des S séquences d'apprentissage et les coefficients multiplicatifs (diagonaux) de normalisation. Pour calculer les valeurs des noyaux (symétriques) pour toutes les paires de séquences d'apprentissage, la première stratégie de normalisation nécessite $D(S+3)/2$ opérations supplémentaires, alors que la seconde en nécessite $D(S+1)$.



Systèmes SVM avec noyau GLDS
Type de normalisation :
(1) Classique ($\mathbf{S_B}$ estimé sur les trames du Monde)
(2) Centré ($\mathbf{\Sigma_B}$ estimé sur les trames)
(3) avec $\mathbf{S_B}$ estimé sur les séquences imposteurs
(4) sans normalisation ($\mathbf{S_B} = \mathbf{I}_D$)

	EER(%)	minDCF(x10^{-3})
(1) ✔	13.29	55.6
(2)	13.47	59.8
(3)	13.05	56.6
(4)	20.60	71.9

Fig. 5.3 - Performances du noyau GLDS selon le type de Normalisation.

5.3 Développement et évaluation des noyaux FSNS

Dans cette section, nous montrons comment nous avons réglé les divers paramètres mis en jeu par les noyaux FSNS et FSMS dans leur forme duale réduite (§4.4.2 et §4.4.3) avant de comparer les SVMs basés sur ces noyaux avec les systèmes de référence. L'expression générale correspondante des noyaux FSNS et FSMS est :

$$\kappa(\mathcal{X}, \mathcal{Y}) = \overline{\boldsymbol{\psi}_{\mathcal{C}}}(\mathcal{X})^{\mathrm{T}} \, \mathbf{R} \, \overline{\boldsymbol{\psi}_{\mathcal{C}}}(\mathcal{Y}) \tag{5.1}$$

Nous rappelons que $\overline{\boldsymbol{\psi}_{\mathcal{C}}}$ est une expansion empirique moyenne sur les séquences (§4.3.1) de taille m. Cette expansion est calculée à partir d'un ensemble de m vecteurs sélectionnés pendant le développement : le *dictionnaire*. $\mathbf{R}$ est une matrice de normalisation fixée après le choix du *dictionnaire* indépendamment des locuteurs cibles. Pour simplifier la complexité calculatoire pour le protocole d'évaluation NIST, le calcul des valeurs de noyaux entre les séquences d'apprentissage se fait en deux temps. Tout d'abord les expansions "normalisées" sont estimées et gardées en mémoire ; il s'agit des expansions $\mathbf{R}^{1/2}\overline{\boldsymbol{\psi}_{\mathcal{C}}}(\mathcal{X})$ où $\mathbf{R}^{1/2}$ est une matrice triangulaire issue d'une décomposition de Cholesky de $\mathbf{R}$ (annexe A.2). Cela permet ensuite d'utiliser un simple produit scalaire, dont la complexité est $O(m)$ au lieu de $O(m^2)$ pour le noyau linéaire généralisé (5.1). La complexité de test d'une séquence est en $O(m)$ lorsque les modèles SVM sont compactés comme suggéré en §3.5.1 dans l'équation (3.60).

5.3.1 Stratégie de normalisation

Un point-clé des noyaux FSNS est la normalisation dans le *Feature Space*. Nous rappelons que cette normalisation est opérée implicitement à travers une matrice de la forme :

$$\mathbf{R} = \left[\tfrac{1}{N}\mathbf{K}(:,\mathrm{I})^{\mathrm{T}}\mathbf{P}\mathbf{K}(:,\mathrm{I}) + \varepsilon\mathbf{K}(\mathrm{I},\mathrm{I})\right]^{-1} \tag{5.2}$$

où ε est un scalaire positif et où

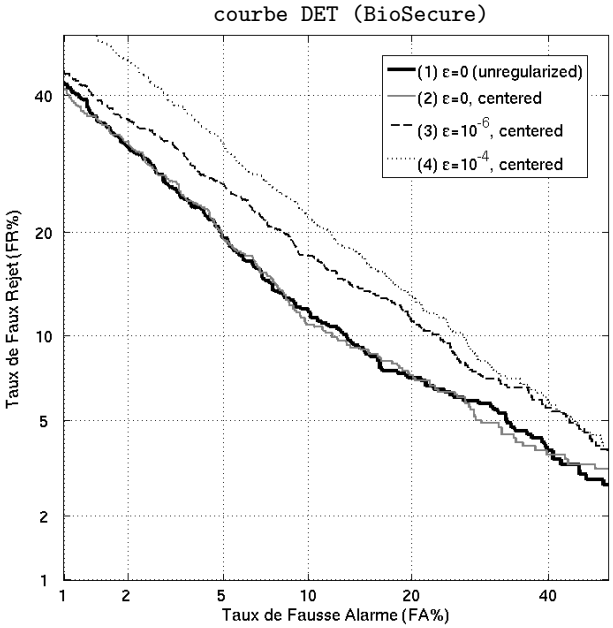
$$\mathbf{P} = \begin{cases} \mathbf{I}_N & \text{pour les noyaux FSNS} \\ \mathbf{I}_N - \frac{1}{N}\mathbf{1}\mathbf{1}^{\mathrm{T}} & \text{pour les noyaux FSMS} \end{cases}$$

Les noyaux FSMS sont équivalents aux noyaux FSNS avec une *expansion* centrée sur du Monde $\mathcal{B}$.

Pour un corpus du Monde $\mathcal{B}$ et un *dictionnaire* $\mathcal{C}$ donnés, plusieurs choix se présentent pour appliquer la normalisation :

– Appliquer ou non un centrage des *expansions*.
– Régler la valeur du paramètre de régularisation ε.

Nous avons réalisé plusieurs expériences pour comparer les performances relatives aux divers choix possibles. Fig.5.4 montre les résultats pour un noyau RBF Gaussien ($\rho = 5$) et un *dictionnaire* avec $m = 5000$ éléments issus de l'ICD.



Systèmes SVM avec noyaux FSNS/FSMS
Type de normalisation :
(1) FSNS avec $\varepsilon = 0$ (sans régularisation)
(2) FSMS (centré) avec $\varepsilon = 0$ (sans régularisation)
(3) FSMS (centré) avec $\varepsilon = 10^{-6}$
(4) FSMS (centré) avec $\varepsilon = 10^{-4}$

	EER(%)	minDCF(x10^{-3})
(1)	10.95	49.9
(2)	10.62	49.3
(3)	14.11	53.1
(4)	15.83	63.5

Fig. 5.4 - Performances de développement des noyaux FSNS selon le type de normalisation.

Les conclusions que l'on peut tirer de ces expériences ont été validées pour plusieurs autres réglages. Elles sont les suivantes :

– Le centrage (dans le *Feature Space*) ne semble pas vraiment améliorer les performances, tout comme il avait été observé avec le noyau GLDS.
– Les meilleures performances sont obtenues sans régularisation ($\varepsilon = 0$).

Nous n'avons pas d'interprétation théorique pour la première assertion. Mais dans la suite, nous ne présenterons que des expériences faisant intervenir les noyaux FSNS (sans centrage), afin d'alléger le calcul du terme de normalisation (5.2).

Concernant le second point, on peut avancer qu'approcher les noyaux FSNS avec une ICD revient à appliquer une première régularisation. En effet, nous avons vu que cela équivalait à projeter les *expansions* du *Feature Space* sur un sous-espace de dimension réduite m avant de

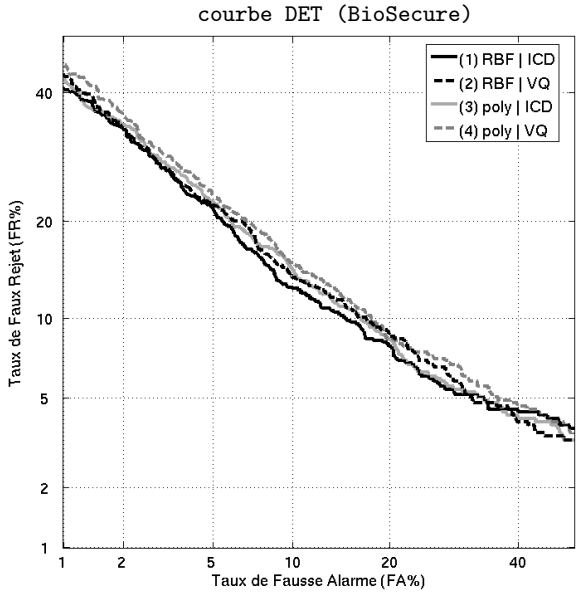
calculer le noyau normalisé. L'estimation des seconds moments sur ce sous-espace ne pose pas de problème statistique tant qu'ils sont estimés sur un nombre de données $N > m$ suffisamment élevé. Autrement dit, la matrice des seconds moments est inversible et ne nécessite donc pas de régularisation supplémentaire. La présence d'une régularisation mettant en jeu $\varepsilon > 0$ est donc superflue. Notons qu'augmenter le paramètre ε équivaut à limiter l'effet de la normalisation : à l'extrême $\varepsilon \to +\infty$ le premier terme de (5.2) devient désuet, et le noyau calculé équivaut à un noyau sans normalisation d'après le lemme 1 ($\mathbf{S}_{\mathcal{B}} := \mathbf{I}_D$).

5.3.2 Choix du *dictionnaire*

Calcul des éléments du *dictionnaire*

À cause de la capacité de mémoire $O(Nm)$ requise par l'ICD, nous n'avons pas pu réaliser cette décomposition sur tous les vecteurs du Monde disponibles ($N \sim 10^6$). Dans les expériences présentées, nous avons arbitrairement choisi d'exécuter l'ICD sur une sous-population de $5{\times}10^4$ vecteurs sélectionnés aléatoirement parmi les vecteurs du Monde. D'autres alternatives ont été essayé pour pallier le problème de mémoire mais elles n'ont pas abouti sur des améliorations de performance. Par exemple, nous avons essayé plusieurs variantes d'algorithmes gloutons, dans lesquels plusieurs ICDs sont appliquées successivement de manière à finalement tenir compte de tous les vecteurs du Monde.

À cause des complexités calculatoires, nous n'avons pas pu comparer l'ICD à d'autres algorithmes d'approximation conçus pour les méthodes à noyau. Cependant, afin de juger de la pertinence de l'ICD dans notre calcul de noyau, nous avons comparé l'ICD à une méthode d'approximation indépendante du choix du noyau : la Quantification Vectorielle (VQ) [Linde et al., 1980]. Fig.5.5 montre les différences de performances obtenues selon que le *dictionnaire* est choisi par



Systèmes SVM avec noyaux FSNS
Choix du *dictionnaire* (taille $m = 4096$) :
(1) Noyau RBF ($\rho = 5$), *dictionnaire* choisi par ICD
(2) Noyau RBF ($\rho = 5$), *dictionnaire* choisi par VQ
(3) Noyau polynomial (degré 5), *dictionnaire* choisi par ICD
(4) Noyau polynomial (degré 5), même *dictionnaire* que (1) (VQ)

	EER(%)	minDCF(x10^{-3})
(1)	11.61	50.3
(2)	12.42	51.9
(3)	12.13	51.4
(4)	12.92	53.8

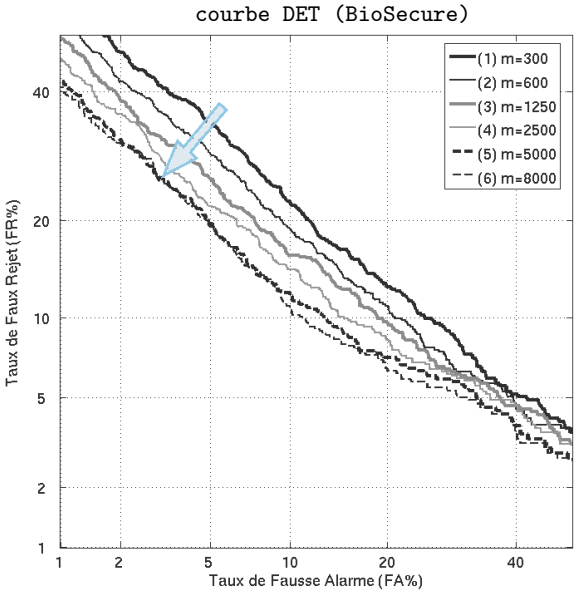
Fig. 5.5 - Performances de développement des noyaux FSNS selon le choix du *dictionnaire*.

ICD ou par VQ, pour une taille de *dictionnaire* fixée à $m = 4096$. Nous reviendrons dans la sous-section suivante sur les deux types de noyaux utilisés (RBF Gaussien et polynomial). Nous pouvons remarquer que les performances avec l'ICD sont meilleures, mais elles ne le sont que légèrement. Choisir le *dictionnaire* par VQ présente l'avantage pratique d'alléger la phase de développement : il n'y a pas la nécessité de ré-estimer un *dictionnaire* à chaque fois que l'on change le réglage du noyau vectoriel $k(\mathbf{x}, \mathbf{y})$. C'est pourquoi nous préconisons la démarche suivante pour le développement des noyaux FSNS : d'abord choisir le noyau vectoriel et le paramétrer en regardant les performances fournis avec un *dictionnaire* issu d'une VQ, et ensuite exécuter l'ICD une fois le choix du noyau arrêté.

Précisons, enfin, que les performances restent les mêmes selon que le *dictionnaire* est choisi par VQ, ou selon qu'il est choisi comme l'ensemble des vecteurs moyennes des composantes d'un GMM. Il semble donc empiriquement inutile d'obtenir une partition plus précise que celle obtenu par VQ en rajoutant des étapes d'optimisation EM[24].

Taille du *dictionnaire*

La taille du *dictionnaire* désigne le nombre m de vecteurs qu'il contient, et correspond en théorie à la dimension du *Feature Space* dans lequel les données sont réellement projetées. Elle règle ainsi la complexité de la modélisation, et détermine la complexité calculatoire du noyau de séquences. Comme la montre Fig.5.6 pour un noyau RBF Gaussien ($\rho = 5$), les performances



Systèmes SVM avec noyaux FSNS Gaussiens
Taille du *dictionnaire* :
(1) $m = 300$ éléments
(2) $m = 600$
(3) $m = 1250$
(4) $m = 2500$
(5) $m = 5000$
(6) $m = 8000$

	EER(%)	minDCF(x10^{-3})
(1)	15.70	65.0
(2)	14.38	61.2
(3)	13.59	57.4
(4)	12.52	53.5
(5)	10.95	49.9
(6)	10.17	48.9

Fig. 5.6 - Performances de développement des noyaux FSNS selon la taille du *dictionnaire*.

ont tendance à s'améliorer lorsque la taille du *dictionnaire* augmente. Choisir m revient donc à choisir un compromis entre niveau de performance et efficacité. Pour la suite des expériences, nous nous sommes limité à $m = 5000$ vecteurs dans le *dictionnaire*, étant donné que le niveau de performance stagne au-delà.

[24] Nous rappelons que le résultat d'une VQ peut servir d'initialisation à l'apprentissage d'un GMM (§1.2.1).

5.3.3 Paramètres du noyau vectoriel

Noyaux polynomiaux

Dans un premier temps, nous présentons les résultats des noyaux FSNS, prenant comme noyau vectoriel le noyau polynomial :

$$k(\mathbf{x}, \mathbf{y}) = (\delta + \nu\ \mathbf{x}^{\mathrm{T}}\mathbf{y})^p \qquad \text{avec} \quad \delta \geq 0\ ,\ \nu > 0 \tag{5.3}$$

D'après l'équation (3.14) de §3.2.1, ce noyau correspond à une expansion polynomiale de degré maximal p, dont les composantes sont multipliées par des coefficients qui dépendent de p, ν et δ. Dans le cas particulier où $\delta = 0$, les composantes correspondant aux monomes de degrés strictement inférieurs à p ont un poids nul et la dimension du *Feature Space* est réduite en conséquence. Les performances sont généralement moins bonnes dans ce cas particulier, comme le montre Fig.5.7 pour $p = 5$, $\nu = 1$, et $\delta = 1$ pour le système `(2)` (resp. $\delta = 0$ pour `(3)`). Cette baisse de performance est due à la complexité réduite de la modélisation. Dans le cas général, le noyau polynomial code toutes les corrélations entre composantes vectorielles d'entrée jusqu'à l'ordre p. Dans le cas où $\delta = 0$, seulement les corrélations d'ordre strictement égal à p sont prises en considération.

Sans approximation par ICD et sans régularisation ($\varepsilon = 0$), les coefficients théoriques qui multiplient les composantes de l'expansion polynomiale (3.14) sont désuets pour les noyaux FSNS "exacts" : ils s'annulent si l'on tient compte de l'effet de la normalisation par la matrice des seconds moments. Du moment que $\delta > 0$, le choix de ν et δ ne change donc pas la valeur des noyaux FSNS polynomiaux. Aussi, dans le cas d'un degré $p = 3$ et d'un δ non nul, nous avons vérifié que le noyau FSNS polynomial donnait les mêmes résultats que le noyau GLDS $p = 3$ avec une matrice de normalisation pleine, du moment que la taille du *dictionnaire* est au moins égale à la taille $D = 3276$ du *Feature Space*. Les performances correspondantes sont représentées par la courbe `(1)` de Fig.5.7. Dans le cas où une ICD est utilisée pour approcher le noyau FSNS, le choix de ν et δ a par contre un impact sur les valeurs du noyau de séquence. Étant donné que les vecteurs d'entrée sont normalisés, nous avons arbitrairement fixé $\nu = 1$ et $\delta = 1$. Nous nous sommes par contre intéressés au paramètre p. Par rapport au noyau FSNS de degré $p = 3$ (noyau GLDS à matrice de normalisation pleine `(1)`), les performances peuvent être améliorées en augmentant le degré comme le montre Fig.5.7. Mais elles ne le sont finalement que modérément pour une taille de *dictionnaire* m fixée.

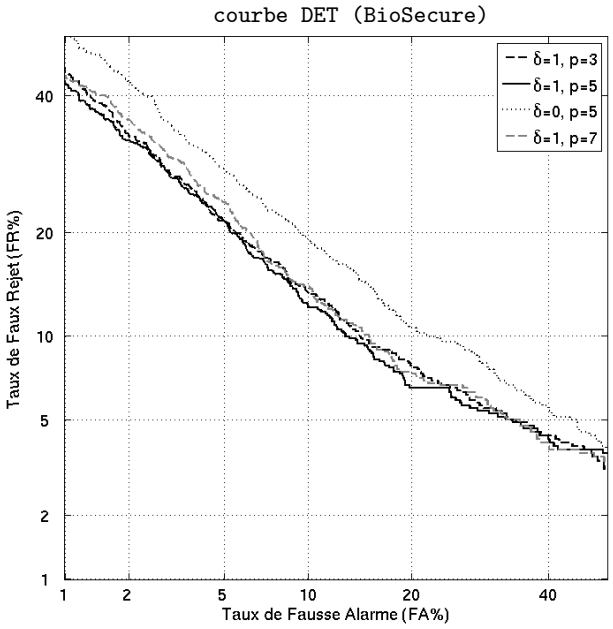
- *Comparaison* `(1)/(3)` *:* Les performances sont améliorées en passant à un degré p supérieur à 3 avec un *dictionnaire* de taille $m > 3276$,
- *Comparaison* `(3)/(4)` *:* Les performances ne sont pas forcément améliorées lorsqu'augmente le degré p, pour une taille m fixe. Dans les expériences que nous avons menées, le degré optimal parmi les degrés de 1 à 10 semble être $p = 5$.

Notre interprétation de ces phénomènes est la suivante.

- D'un coté, passer à un degré p supérieur rend la modélisation *a priori* plus complexe, puisque la dimension du *Feature Space* $D = \frac{(d+p)!}{d!\ p!}$ est augmentée. Étant donné que le critère de marge des SVMs permet d'éviter le "fléau de la dimensionalité" dans le *Feature Space* (§1.3.1), on peut s'attendre à ce que les SVMs bénéficient de cette complexification du noyau pour donner de meilleures performances.

Mais

- d'un autre coté, la dimension du *Feature Space* où sont finalement projetées les données reste la même pour une taille de *dictionnaire* m fixée (proposition 5).



Systèmes SVM avec noyaux FSNS polynomiaux

Paramètres du polynôme :

(1) $k(\mathbf{x},\mathbf{y}) = (1+\mathbf{x}^{\mathrm{T}}\mathbf{y})^3$ $(m = 3276)$
(2) $k(\mathbf{x},\mathbf{y}) = (1+\mathbf{x}^{\mathrm{T}}\mathbf{y})^5$ $(m = 5000)$
(3) $k(\mathbf{x},\mathbf{y}) = (\mathbf{x}^{\mathrm{T}}\mathbf{y})^5$ $(m = 5000)$
(4) $k(\mathbf{x},\mathbf{y}) = (1+\mathbf{x}^{\mathrm{T}}\mathbf{y})^7$ $(m = 5000)$

	EER(%)	minDCF(x10^{-3})
(1)	12.14	52.6
(2)	11.61	50.8
(3)	14.66	59.8
(4)	12.00	52.8

Fig. 5.7 - Performances de développement des noyaux FSNS avec noyau polynomial selon le degré.

Concernant ce dernier point, rappelons tout de même que l'ICD est sensée fournir des axes du *Feature Space* de départ (dimension D) qui contiennent le plus d'information (au sens du critère PCA).

Dans le but d'améliorer la robustesse, nous avons tenté d'utiliser des noyaux polynomiaux normalisés sphériquement (§3.1.4, équation 3.8) comme préconisé par [Wan, 2003]. Les performances n'étaient pas améliorées.

Zone d'influence du noyau RBF Gaussien

Le noyau avec lequel nous avons obtenu les meilleures performances est le noyau RBF Gaussien, dont l'efficacité a été montrée pour la plupart des méthodes d'apprentissage à noyaux [Shawe-Taylor et Cristianini, 2004].

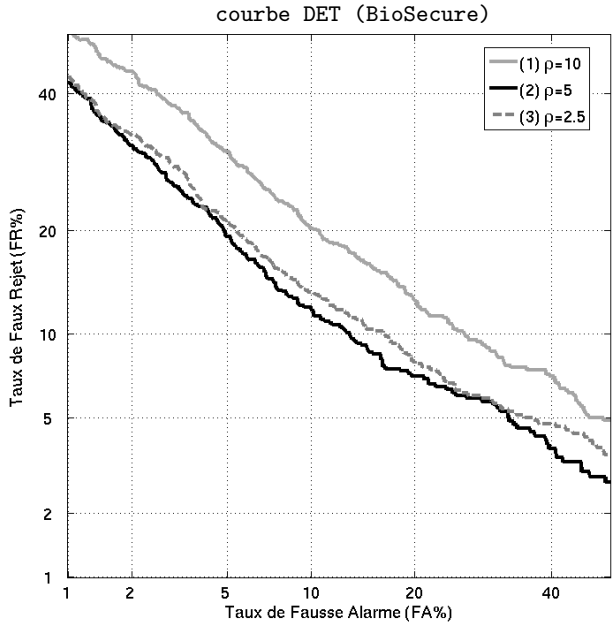
$$k(\mathbf{x},\mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\rho^2}} \tag{5.4}$$

Nous rappelons que le paramètre d'échelle ρ règle la notion de localité (§3.2.2). Plus ρ est élevé, et plus la "zone d'influence" des vecteurs d'apprentissage est étendue. Ce paramètre est donc relié à la variance des composantes d'entrée. [Schölkopf et al., 1999] recommande de prendre ρ de l'ordre de

$$\rho_0 = \sqrt{d}\overline{\sigma} \tag{5.5}$$

où d est la dimension des vecteurs d'entrées, et $\overline{\sigma} = \frac{1}{d}\sum_{u=1}^{d}\left(\mathrm{E}[x_u^2] - \mathrm{E}[x_u]^2\right)$ est la moyenne quadratique des écarts-types de chaque composante vectorielle. Dans notre cas, $d = 25$ et $\overline{\sigma}$ est proche de 1 après la normalisation *Feature Warping*, ce qui donne $\rho_0 = 5$. Fig.5.8 montre

que cette valeur semble bien être optimale pour la conception de notre noyau de séquence. Autour de cette valeur, les performances restent à peu près les mêmes. Lorsque l'on s'en éloigne (multiplication/division par 2) les performances se dégradent, et ce particulièrement pour des valeurs élevées de ρ. Aussi, les performances que l'on peut atteindre avec un noyau RBF Gaussien sont meilleures que celles atteintes avec les noyaux polynomiaux.



Systèmes SVM avec noyaux FSNS Gaussien
Zone d'influence du noyau vectoriel RBF :
(1) $\rho = 10$
(2) $\rho = 5 = \rho_0$
(3) $\rho = 2.5$

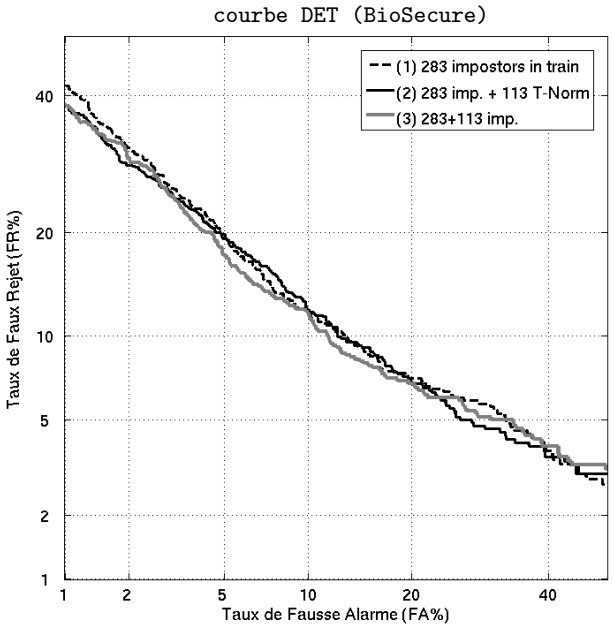
	EER(%)	minDCF(x10^{-3})
(1)	15.7	60.2
(2)	10.95	49.9
(3)	11.90	50.5

Fig. 5.8 - Performances de développement des noyaux FSNS avec noyau RBF Gaussien selon la zone d'influence.

5.3.4 Normalisation des scores

Dans les systèmes SVMs à noyaux de séquences dont nous avons montré les performances jusqu'à présent, les 283 séquences du corpus du Monde (A) servent de séquences imposteurs lors de l'apprentissage (classe -1), et aucune normalisation des scores n'a encore été appliquée. Pour le système SVM à noyaux FSNS comme pour le système UBM-GMM, la T-Norm améliore les performances en terme de DCF, comme le montre Fig.5.9 (comparaison (1)/(2)). Pour le calcul des statistiques pour la T-Norm (2), les 113 séquences du corpus d'imposteurs (B) ont été utilisées.

Cependant l'amélioration relative est faible par rapport à l'amélioration observée pour le système UBM-GMM. D'autre part, la même amélioration de performance peut être atteinte en rajoutant les 113 séquences normalement utilisées pour la T-Norm aux 283 séquences servant d'exemples imposteurs lors de l'apprentissage (comparaison (2)/(3)). Le même phénomène a été observé sur des systèmes SVMs avec d'autres noyaux de séquences dans des expériences dont nous ne montrons pas les résultats ici. Étant donné que la dernière stratégie (3) implique des complexités calculatoires bien moins lourdes qu'une T-Norm en phase de test, nous l'avons finalement adoptée pour tous les systèmes SVMs à noyau de séquences. Tous les résultats suivant relatifs aux noyaux de séquences correspondront à un apprentissage avec 396 séquences imposteurs (A+B).



Systèmes SVM avec noyaux FSNS Gaussien
Séquences d'apprentissage / Normalisation de scores :
(1) 283 séquences (A)
(2) 283 séquences (A) + T-Norm avec 113 séquences (B)
(3) 396 séquences (A+B)

		EER(%)	minDCF($\times 10^{-3}$)
	(1)	10.95	49.9
	(2)	11.35	46.9
✔	(3)	10.55	47.3

Fig. 5.9 - Noyaux FSNS et normalisation des scores.

La faible amélioration procurée par la T-Norm pour les systèmes SVMs pourrait s'expliquer par le fait que l'hypothèse de Gaussianité ne soit pas respectée sur les distributions de scores clients/imposteurs. La robustesse de la T-Norm dépend en effet de la véracité de cette hypothèse. Mais d'après les tests de Gaussianité que nous avons réalisé à partir des histogrammes cumulés des scores, il s'avère que les scores en sortie des systèmes SVMs ne vérifit pas moins l'hypothèse que les log-rapports de vraisemblance des GMMs.

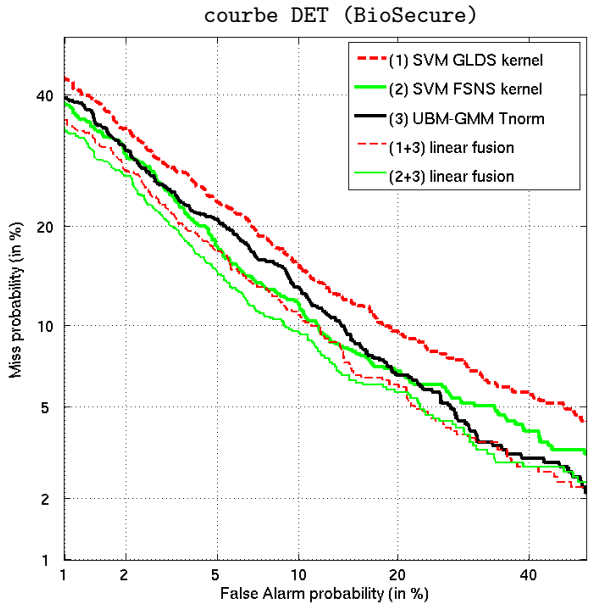
5.3.5 Résultats de l'évaluation

Fig.5.10 montre les performances du système SVM à noyau FSNS Gaussien sélectionné lors de la phase de développement (✔), ainsi que celles du système SVM à noyau GLDS analogue (396 séquences imposteurs en apprentissage) et du système de référence UBM-GMM. La comparaison de ces trois systèmes nous amène à conclure :

- *Comparaison* `(1)/(2)` *:* La généralisation du noyau GLDS offre la possibilité d'améliorer les performances du système SVM.
- *Comparaison* `(2)/(3)` *:* Les systèmes SVMs à noyaux FSNS (discriminatifs) offrent des résultats compétitifs par rapport aux systèmes UBM-GMM (génératifs).

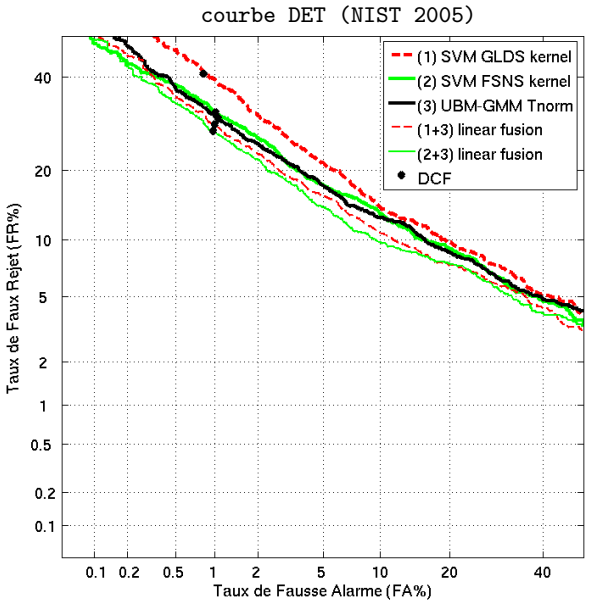
Notons que les performances sur les données d'évaluation (NIST SRE 2005) sont meilleures que celles obtenues sur le corpus de validation (NIST SRE 2004). Cette tendance a été observée pour la plupart des systèmes de vérification du locuteur qui ont été testés sur les évaluations NIST 2004 et NIST 2005. La plus grande "facilité" des tests de NIST SRE 2005 peut expliquer le faible écart entre le minimum de la DCF et la DCF réelle.

Fig.5.10 montre aussi les gains de performance apportés par une combinaison linéaire des scores de sortie des systèmes SVMs et du système UBM-GMM. Les coefficients de pondération de cette combinaison sont choisis de manière à optimiser les performances (minDCF) sur le corpus



	EER(%)	minDCF(x10^{-3})
(1)	12.80	51.9
(2)	10.55	47.5
(3)	11.48	49.1
(1+3)	10.32	45.0
(2+3)	9.50	43.5

(a) Performances sur le corpus de validation



	EER(%)	DCF(x10^{-3}) min	DCF(x10^{-3}) réel
(1)	12.54	48.5	48.8
(2)	11.91	41.4	41.6
(3)	12.06	40.5	40.6
(1+3)	10.54	38.3	38.8
(2+3)	9.71	36.7	37.0

(b) Performances sur le corpus d'évaluation

Systèmes et Fusion de systèmes :
(1) SVM, noyau GLDS
(2) SVM, noyau FSNS (Gaussien)
(3) UBM-GMM (T-Norm)
(1+3) Fusion linéaire de (1) et (3)
(2+3) Fusion linéaire de (2) et (3)

Fig. 5.10 - Évaluation des noyaux FSNS.

de validation (Fig.5.10(a)). Précisions que la fusion linéaire des systèmes SVMs (noyau GLDS / noyau FSNS), qui n'est pas montrée ici, n'améliore pas les performances par rapport au meilleur système (noyau FSNS Gaussien dans notre cas). Par contre fusionner les scores obtenus par la modélisation générative UBM-GMM d'une part, et la modélisation discriminante SVM d'autre part, améliore les performances (systèmes (1+3) et (2+3)). Cela suggère que les erreurs commises par les deux types de classifieurs sont complémentaires, ou encore que les scores de sortie des systèmes sont assez peu corrélés linéairement pour permettre d'améliorer la robustesse en étant combinés. Par contre, même si le gain en performance est important lorsque l'on généralise le noyau GLDS aux noyaux FSNS ((1)→(2)) le gain en performance est moins flagrant pour la fusion linéaire ((1+3)→(2+3)).

5.4 Développement des autres noyaux de séquences

5.4.1 Noyaux de produit de probabilités

Comme expliqué en §3.5.2, on peut concevoir comme noyau entre séquences un noyau de produit de probabilités (§3.3.1), entre les distributions empiriques estimées sur les séquences. L'application d'un tel noyau pour la vérification du locuteur n'a fait l'objet d'aucun article publié, à notre connaissance.

Nous choisissons comme famille de distributions les GMMs pour représenter les séquences parce qu'ils ont montré leur succès en vérification du locuteur. L'expérience montre que la technique d'apprentissage la plus robuste des GMMs est l'adaptation MAP à partir d'un UBM, en gardant fixes les poids et les matrices de covariance. Nous nous focalisons donc à ce cas particulier qui a l'avantage de simplifier les calculs (en plus d'augmenter la robustesse). Dans ce contexte, nous notons :

- G le nombre de composantes Gaussiennes de l'UBM (et des GMMs adaptés),
- $\boldsymbol{\omega} = \left[\ \omega_1\ \cdots\ \omega_G\ \right]^{\mathrm{T}}$ la colonne formée des poids de l'UBM (et des GMMs adaptés),
- $\left\{\boldsymbol{\Sigma}_g\right\}_{g=1\cdots G}$ les matrices de covariance de l'UBM (et des GMMs adaptés).
- $\left\{\boldsymbol{\mu}_{X,g}\right\}_{g=1\cdots G}$ les vecteurs moyenne du GMM adapté sur une séquence $\mathbf{X}$
- $\boldsymbol{\theta}_{\mathrm{X}}$ (resp. $\boldsymbol{\theta}_{\mathrm{Y}}$) les paramètres du GMM appris sur une séquence $\mathbf{X}$ (resp. $\mathbf{Y}$) :

$$p(\mathbf{z}|\boldsymbol{\theta}_{\mathrm{X}}) = \textstyle\sum_{g=1}^{G} \omega_g \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_{X,g}, \boldsymbol{\Sigma}_g) \qquad p(\mathbf{z}|\boldsymbol{\theta}_{\mathrm{Y}}) = \textstyle\sum_{h=1}^{G} \omega_h \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_{Y,h}, \boldsymbol{\Sigma}_h)$$

Si l'on suppose que les matrices de covariances $\boldsymbol{\Sigma}_g$ sont diagonales, l'expression du "noyau de corrélation" (§3.3) dans le cas des GMMs se simplifie comme suit [Lyu, 2005] :

$$\kappa_1^{\mathrm{pp}}(\mathbf{X}, \mathbf{Y}) = \int_{\mathbb{R}^d} p(\mathbf{z}|\boldsymbol{\theta}_{\mathrm{X}}) p(\mathbf{z}|\boldsymbol{\theta}_{\mathrm{Y}})\, \mathrm{d}\mathbf{z} \ = \ (2\pi)^{-\frac{d}{2}}\ \boldsymbol{\omega}^{\mathrm{T}} \boldsymbol{\Gamma} \boldsymbol{\omega} \tag{5.6}$$

où $\boldsymbol{\Gamma}$ est la matrice symétrique (taille $G \times G$) composé des valeurs :

$$\Gamma_{g,h} = \tfrac{1}{\sqrt{\det(\boldsymbol{\Sigma}_g + \boldsymbol{\Sigma}_h)}} e^{-\frac{1}{2}(\boldsymbol{\mu}_{X,g} - \boldsymbol{\mu}_{Y,h})^{\mathrm{T}} (\boldsymbol{\Sigma}_g + \boldsymbol{\Sigma}_h)^{-1} (\boldsymbol{\mu}_{X,g} - \boldsymbol{\mu}_{Y,h})} \tag{5.7}$$

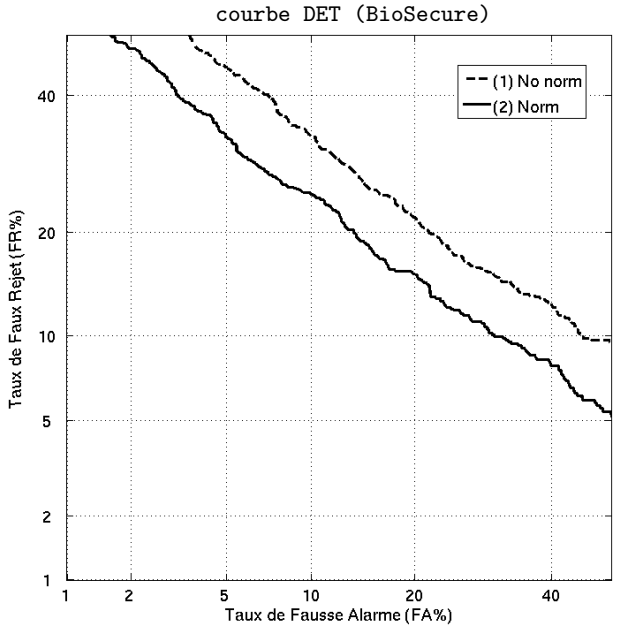
Pour améliorer la robustesse de l'apprentissage SVM, nous normalisons le noyau selon la projection gnomonique (§3.1.4, Tab.3.8) :

$$\mathring{\kappa}_1^{\mathrm{pp}}(\mathbf{X}, \mathbf{Y}) = \frac{\kappa_1^{\mathrm{pp}}(\mathbf{X}, \mathbf{Y})}{\sqrt{\kappa_1^{\mathrm{pp}}(\mathbf{X}, \mathbf{X})\ \kappa_1^{\mathrm{pp}}(\mathbf{Y}, \mathbf{Y})}} \tag{5.8}$$

Ce noyau permet de bien conditionner la matrice de Gram, dont les valeurs diagonales sont unitaires ($\mathring{\kappa}_1^{\mathrm{pp}}(\mathbf{X}, \mathbf{X}) = 1$), tout comme avec un noyau de Bhattacharyya $\kappa_{\mathrm{B}}^{\mathrm{pp}}(\mathbf{X}, \mathbf{Y}) = \int \sqrt{p(\mathbf{z}|\boldsymbol{\theta}_{\mathrm{X}}) p(\mathbf{z}|\boldsymbol{\theta}_{\mathrm{Y}})}\, \mathrm{d}\mathbf{z}$. Ce dernier noyau n'admet malheureusement pas d'expression analytique simple pour les GMMs. Son estimation nécessiterait un calcul d'intégrale sur $\mathbb{R}^d$, qui pénaliserait l'efficacité et la robustesse du noyau.

Comme le montre Fig.5.11, la normalisation du noyau permet de gagner nettement en performance. Toutefois, le noyau de corrélation normalisé fournit des performances moyennes par rapport aux autres noyaux de séquence.

Les expériences dont les performances sont ici montrées ont été réalisées en prenant $G = 512$ composantes Gaussiennes dans les GMMs, qui sont appris avec les mêmes réglages que pour le système UBM-GMM de référence (§5.2.1). Nous avons aussi essayé avec $G = 2048$ Gaussiennes, mais en plus d'induire des complexités calculatoires très élevées, les performances s'en trouvaient dégradées.



Systèmes SVM :
(1) Noyau de corrélation κ_1^{pp}
(2) Noyau de corrélation normalisé $\mathring{\kappa}_1^{\mathrm{pp}}$

	EER(%)	minDCF(x10^{-3})
(1)	20.83	76.1
✔ (2)	16.75	64.5

Fig. 5.11 - Performances du noyau de corrélation en phase de développement.

5.4.2 Supervecteurs GMM

Parmi les autres noyaux de séquences basés sur les densités estimées des séquences, on compte les noyaux dérivées de divergences probabilistes, présentés en §3.3.2 et §3.5.2. Nous nous sommes focalisés sur les noyaux basés sur la divergence de Kullback parce qu'ils ont été appliqués avec succcès à la vérification du locuteur en utilisant la modélisation GMM [Moreno et Ho, 2003b, Campbell et al., 2006b, Dehak et Chollet, 2006].

L'approche de [Moreno et Ho, 2003b] présente plusieurs inconvénients du fait que la divergence entre deux séquences soit calculée par une approximation d'intégrale sur des distributions GMMs (par des méthodes de Monte Carlo). Étant donnée la multi-dimensionalité de l'espace vectoriel d'entrée, cette estimation d'intégrale est coûteuse et peu robuste. À cause de l'instabilité des valeurs de noyaux basées sur cette approximation numérique, nous ne sommes pas arrivé à reproduire les niveaux de performance compétitifs annoncés par [Moreno et Ho, 2003b, Ho et Moreno, 2004], dont les résultats sont donnés pour un sous-échantillon des corpus `HUB4` et `KING`. Nous avons par contre appliqué avec succès une autre

méthode plus récente que nous expliquons dans la suite. Cette approche est basée sur une majoration analytique de la divergence de Kullback et débouche sur un noyau pour lequel les conditions de Mercer sont garanties (contrairement à l'approche de [Moreno et Ho, 2003b]).

[Do, 2003] montre que la divergence de Kullback (§3.3.2, Tab.3.16) entre deux Mélanges de modèles est majorée par une expression analytique simple :

$$\mathcal{D}^{\mathrm{KL}}(\sum_{g=1}^{G}\omega_g\mathcal{N}_g\|\sum_{h=1}^{G}\omega'_h\mathcal{N}'_h) \leq \sum_{g=1}^{G}\omega_g\left(\log\frac{\omega_g}{\omega'_g}+\mathcal{D}^{\mathrm{KL}}(\mathcal{N}_g\|\mathcal{N}'_g)\right) \tag{5.9}$$

Dans le cas des GMMs dérivées d'un UBM commun ($\omega_g = \omega'_g$), on peut simplifier l'expression de la divergence de Kullback symétrique en utilisant les expressions analytiques de la divergence de Kullback entre Gaussiennes (§3.3.2, Tab.3.17). En utilisant les mêmes notations que dans la sous-section précédente §5.4.1 :

$$\begin{aligned}\bar{\mathcal{D}}^{\mathrm{KL}}(p(\cdot|\boldsymbol{\theta}_{\mathrm{X}}),p(\cdot|\boldsymbol{\theta}_{\mathrm{Y}})) &= \mathcal{D}^{\mathrm{KL}}(p(\cdot|\boldsymbol{\theta}_{\mathrm{X}})\|p(\cdot|\boldsymbol{\theta}_{\mathrm{Y}}))+\mathcal{D}^{\mathrm{KL}}(p(\cdot|\boldsymbol{\theta}_{\mathrm{Y}})\|p(\cdot|\boldsymbol{\theta}_{\mathrm{X}})) \\ &\leq \underbrace{\sum_{g=1}^{G}\omega_g\left(\boldsymbol{\mu}_{X,g}-\boldsymbol{\mu}_{Y,g}\right)^{\mathrm{T}}\boldsymbol{\Sigma}_g{}^{-1}\left(\boldsymbol{\mu}_{X,g}-\boldsymbol{\mu}_{Y,g}\right)}_{\mathcal{D}^{\mathrm{gm}}\left(\boldsymbol{\theta}_{\mathrm{X}},\boldsymbol{\theta}_{\mathrm{Y}}\right)^2}\end{aligned} \tag{5.10}$$

La quantité $\mathcal{D}^{\mathrm{gm}}$ qui apparaît ici est une somme quadratique des distances de Mahalanobis entre moyennes des GMMs. C'est une mesure de distance définie pour les GMMs qui partagent les mêmes poids et matrices de covariances. De tels modèles peuvent être caractérisés par la concaténation des vecteurs moyennes : on parle de "supervecteur GMM" [Campbell et al., 2006b]. La mesure de distance introduite en (5.10) vérifie en particulier l'inégalité triangulaire, et confère les propriétés de Mercer au noyau Gaussien :

$$\kappa^{\mathrm{gm}}(\mathbf{X},\mathbf{Y}) = e^{-\frac{1}{2\rho^2}\mathcal{D}^{\mathrm{gm}}\left(\boldsymbol{\theta}_{\mathrm{X}},\boldsymbol{\theta}_{\mathrm{Y}}\right)^2} \tag{5.11}$$

Ce noyau a été exploité par [Dehak et Chollet, 2006] dans une tâche de vérification sur les données NIST. Dans ce travail, l'auteur préconise une normalisation des supervecteurs GMMs $\boldsymbol{\mu}_{X,g}$ de manière à ce que les supervecteurs normalisés $\bar{\boldsymbol{\mu}}_{X,g}$ soient à égale distance de l'UBM. En notant $\boldsymbol{\theta}_{\mathrm{o}} = \left\{\omega_g,\boldsymbol{\mu}^{\mathrm{o}}_g,\boldsymbol{\Sigma}_g\right\}_{g=1\cdots G}$ les paramètres de l'UBM et $\bar{\boldsymbol{\theta}}_{\mathrm{X}}$ les paramètres normalisés, le but de la normalisation est d'obtenir :

$$\forall\ \bar{\boldsymbol{\theta}}_{\mathrm{X}} \neq \boldsymbol{\theta}_{\mathrm{o}}\ ,\qquad \mathcal{D}^{\mathrm{gm}}\left(\bar{\boldsymbol{\theta}}_{\mathrm{X}},\boldsymbol{\theta}_{\mathrm{o}}\right) = \mathcal{D}_{\mathrm{o}} \tag{5.12}$$

où $\mathcal{D}_{\mathrm{o}}$ est une distance de référence fixée arbitrairement. Un moyen simple et efficace pour arriver à cela est la technique "D-MAP" conçue par [Ben et Bimbot, 2003], qui consiste à appliquer une même transformation linéaire à tous les vecteurs moyennes, selon :

$$\begin{aligned}&\bar{\boldsymbol{\mu}}_{X,g} = \lambda_{\mathrm{X}}\ \boldsymbol{\mu}_{X,g} + (1-\lambda_{\mathrm{X}})\,\boldsymbol{\mu}^{\mathrm{o}}_g \\ \text{avec}\quad &\lambda_{\mathrm{X}} = \frac{\mathcal{D}_{\mathrm{o}}}{\mathcal{D}^{\mathrm{gm}}\left(\bar{\boldsymbol{\theta}}_{\mathrm{X}},\boldsymbol{\theta}_{\mathrm{o}}\right)}\end{aligned} \tag{5.13}$$

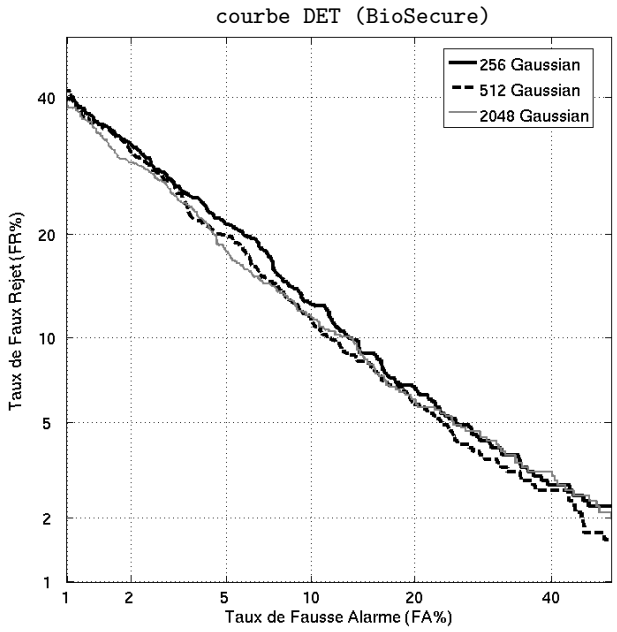
D'après les inégalités triangulaires, la normalisation (5.12) a pour effet de garantir pour tout couple de paramètres normalisés $(\bar{\boldsymbol{\theta}}_{\mathrm{X}},\bar{\boldsymbol{\theta}}_{\mathrm{Y}})$:

$$0 \leq \mathcal{D}^{\mathrm{gm}}\left(\bar{\boldsymbol{\theta}}_{\mathrm{X}},\bar{\boldsymbol{\theta}}_{\mathrm{Y}}\right) \leq 2\mathcal{D}_{\mathrm{o}} \tag{5.14}$$

Cela permet de borner dans $[e^{-2\left(\frac{\mathcal{D}_o}{\rho}\right)^2}, 1]$ le noyau normalisé défini par :

$$\bar{\kappa}^{\mathrm{gm}}(\mathbf{X}, \mathbf{Y}) = e^{-\frac{1}{2\rho^2}\mathcal{D}^{\mathrm{gm}}\left(\bar{\boldsymbol{\theta}}_{\mathrm{X}}, \bar{\boldsymbol{\theta}}_{\mathrm{Y}}\right)^2} \tag{5.15}$$

Les performances de ce noyau sont montrés dans Fig.5.12 en prenant pour paramètres $\mathcal{D}_{\mathrm{o}} = \rho = 1$. Ce jeu de paramètres a fourni les meilleures performances quelque soit le nombre de composantes Gaussiennes dans les GMMs. Sans la normalisation des modèles, il faut réajuster le paramètre ρ selon le nombre de composantes et la divergence moyenne entre GMMs, pour que la matrice de Gram reste bien conditionnée[25]. D'après les résultats de Fig.5.12, les performances du noyau Gaussien entre supervecteurs GMM (5.15) ont tendance à s'améliorer lorsque l'on augmente le nombre G de composantes Gaussiennes. Mais à partir de $G = 512$, les performances stagnent.



Systèmes SVM :
(1) Noyau Gaussien entre supervecteurs GMM $\bar{\kappa}^{\mathrm{gm}}$ avec des GMMs à $G = 256$ composantes Gaussiennes
(2) *idem* avec $G = 512$ composantes
(3) *idem* avec $G = 2048$ composantes

		EER(%)	minDCF(x10^{-3})
	(1)	11.48	49.1
	(2)	10.66	48.4
✔	(3)	10.82	47.6

Fig. 5.12 - Performances de développement du noyau Gaussien entre supervecteurs GMMs.

[25] Le paramètre ρ peut en général être réglé "à vue de nez" pour que la matrice de Gram ne soit ni diagonalement dominante, ni quasi-constante

5.4.3 Noyau de Fisher & Noyau TOP

Nous appliquons maintenant à la vérification du locuteur le noyau de Fisher et le noyau TOP présentés de façon théorique en §3.4.3 et §3.4.4. Tout comme pour les noyaux précédents basés sur des modèles génératifs, nous exploitons la modélisation UBM-GMM. Nous notons :

- G le nombre de composantes Gaussiennes de l'UBM,
- $\boldsymbol{\theta}_{\text{o}} = \left\{\omega_g, \boldsymbol{\mu}_g^{\text{o}}, \boldsymbol{\Sigma}_g\right\}_{g=1\cdots G}$ les paramètres (poids, vecteurs moyennes, matrices de covariance diagonales) de l'UBM,
- x_u la $u^{\text{ième}}$ coordonnée d'un vecteur d'entrée $\mathbf{x} = \left[\ x_1\ \cdots\ x_d\ \right]^{\text{T}}$ de dimension d,
- $\mu_{g,u}^{\text{o}}$ la $u^{\text{ième}}$ coordonnée du vecteur moyenne de la $g^{\text{ième}}$ composante de l'UBM : $\boldsymbol{\mu}_g^{\text{o}} = \left[\ \mu_{g,1}\ \cdots\ \mu_{g,d}\ \right]^{\text{T}}$,
- ${\sigma_{g,u}}^2$ la $u^{\text{ième}}$ valeur diagonale de la covariance de la $g^{\text{ième}}$ composante de l'UBM : $\boldsymbol{\Sigma}_g = \text{diag}\left(\ {\sigma_{g,1}}^2,\ \cdots,\ {\sigma_{g,d}}^2\ \right)$.

Noyau de Fisher

Commençons par donner l'expression de l'*expansion* de Fisher en prenant l'UBM comme modèle de probabilité *a priori*. L'UBM à covariances diagonales fait intervenir $D = G(2d+1)$ paramètres libres. Ce nombre D est la dimension de l'*expansion* de Fisher telle que l'introduit [Jaakkola et Haussler, 1998], et qui s'écrit pour un vecteur d'entrée $\mathbf{x}$:

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}} \log p(\mathbf{x}|\boldsymbol{\theta})\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{o}}} = \left[\begin{array}{l} \frac{\partial}{\partial\omega_1} \log p(\mathbf{x}|\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{o}}} \\ \boldsymbol{\nabla}_{\boldsymbol{\mu}_1^{\text{o}}} \log p(\mathbf{x}|\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{o}}} \\ \boldsymbol{\nabla}_{\boldsymbol{\Sigma}_1} \log p(\mathbf{x}|\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{o}}} \\ \vdots \\ \frac{\partial}{\partial\omega_G} \log p(\mathbf{x}|\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{o}}} \\ \boldsymbol{\nabla}_{\boldsymbol{\mu}_G^{\text{o}}} \log p(\mathbf{x}|\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{o}}} \\ \boldsymbol{\nabla}_{\boldsymbol{\Sigma}_G} \log p(\mathbf{x}|\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{o}}} \end{array}\right] \begin{array}{l} \\ \left.\begin{array}{l}\\\end{array}\right\} \text{ vecteur } \mathbb{R}^d \\ \left.\begin{array}{l}\\\end{array}\right\} \text{ vecteur } \mathbb{R}^d \\ \\ \\ \left.\begin{array}{l}\\\end{array}\right\} \text{ vecteur } \mathbb{R}^d \\ \left.\begin{array}{l}\\\end{array}\right\} \text{ vecteur } \mathbb{R}^d \end{array} \tag{5.16}$$

Les composantes de cette *expansion* s'écrivent [Layton et Gales, 2004] :

$$\frac{\partial}{\partial\omega_g} \log p(\mathbf{x}|\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{o}}} = \gamma_g^{\text{o}}(\mathbf{x}) = \frac{\mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}_g^{\text{o}}, \boldsymbol{\Sigma}_g\right)}{\sum_{h=1}^{G} \omega_h\ \mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}_h^{\text{o}}, \boldsymbol{\Sigma}_h\right)} \tag{5.17}$$

$$\frac{\partial}{\partial\mu_{g,u}^{\text{o}}} \log p(\mathbf{x}|\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{o}}} = \omega_g \gamma_g^{\text{o}}(\mathbf{x}) \left(\frac{x_u - \mu_{g,u}^{\text{o}}}{{\sigma_{g,u}}^2}\right) \tag{5.18}$$

$$\frac{\partial}{\partial{\sigma_{g,u}}^2} \log p(\mathbf{x}|\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{o}}} = \frac{\omega_g \gamma_g^{\text{o}}(\mathbf{x})}{2\ {\sigma_{g,u}}^2} \left(\frac{\left(x_u - \mu_{g,u}^{\text{o}}\right)^2}{{\sigma_{g,u}}^2} - 1\right) \tag{5.19}$$

On peut aussi considérer au lieu de (5.19) la dérivée partielle par rapport aux écart-types $\sigma_{g,u}$, comme il est fait dans [Wan, 2003], ce qui reviendrait à multiplier par $2\sigma_{g,u}$ la dérivée par rapport à ${\sigma_{g,u}}^2$. Quoiqu'il en soit, les facteurs multiplicatifs indépendants de $\mathbf{x}$ dans les

équations (5.17), (5.18) et (5.19) sont obsolètes si l'on utilise une normalisation par la matrice d'Information de Fisher (3.43), ou par tout autre matrice qui encode les seconds moments empiriques des *expansions* de Fisher. En supposant que ce soit le cas, on peut prendre comme *expansion* :

$$\boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \mathbf{x}) = \begin{bmatrix} \boldsymbol{\delta}^{(1)}(\boldsymbol{\theta}_{\mathrm{o}}, \mathbf{x}) \\ \vdots \\ \boldsymbol{\delta}^{(G)}(\boldsymbol{\theta}_{\mathrm{o}}, \mathbf{x}) \end{bmatrix} \tag{5.20}$$

$$\text{où} \quad \boldsymbol{\delta}^{(g)}(\boldsymbol{\theta}_{\mathrm{o}}, \mathbf{x}) = \gamma_g^{\mathrm{o}}(\mathbf{x}) \begin{bmatrix} 1 \\ \mathbf{x} - \boldsymbol{\mu}_g \\ \mathrm{diag}\Big((x-\boldsymbol{\mu}_g)(x-\boldsymbol{\mu}_g)^{\mathrm{T}} - \boldsymbol{\Sigma}_g\Big) \end{bmatrix} \tag{5.21}$$

Cette *expansion* s'étend aux séquences de vecteurs en considérant les vecteurs comme indépendants (hypothèse sous-jacente de la modélisation GMM), et en incorporant une normalisation par la longueur de la séquence comme suggéré en §3.4.3. Pour une séquence $\mathbf{X} = \{\mathbf{x}_t\}_{t=1\cdots T_{\mathrm{X}}}$, nous considérons l'*expansion* :

$$\boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \mathbf{X}) = \tfrac{1}{T_{\mathrm{X}}} \sum_{t=1}^{T_{\mathrm{X}}} \boldsymbol{\delta}(\boldsymbol{\theta}_{\mathrm{o}}, \mathbf{x}_t) \tag{5.22}$$

Notons que les auteurs de [Smith et Gales, 2002, Wan et Renals, 2004] préconisent de concaténer la vraisemblance $\log p(\mathbf{X}|\boldsymbol{\theta})$ à l'*expansion* de Fisher (constituée des dérivées de cette vraisemblance). D'après des expériences non montrées ici, les performances restent les mêmes à quelques variations non significatives près. Nous avons donc décidé de montrer les résultats sans cette composante superflue.

Étant donné que les dimensions mises en jeu par l'*expansion* $\boldsymbol{\delta}$ sont très élevées, il n'est en pratique pas envisageable d'estimer la matrice de normalisation de Fisher. Au lieu de cela, on normalise les *expansions* de manière à ce que chacune de leurs composantes soit de second moment unitaire sur les séquences imposteurs d'apprentissage. En notant $\{\mathbf{A}_1, \cdots, \mathbf{A}_N\}$ les séquences d'apprentissage, les composantes de l'*expansion* normalisée $\widetilde{\boldsymbol{\delta}}$ se calculent à partir des composantes δ_u de $\boldsymbol{\delta}$ selon :

$$\widetilde{\delta}_u(\boldsymbol{\theta}_{\mathrm{o}}, \mathbf{X}) = \frac{\delta_u(\boldsymbol{\theta}_{\mathrm{o}}, \mathbf{X})}{\frac{1}{N}\sqrt{\sum_{i=1}^{N} \delta_u(\boldsymbol{\theta}_{\mathrm{o}}, \mathbf{A}_i)^2}} \tag{5.23}$$

Cela équivaut à faire une approximation diagonale de la matrice d'information de Fisher. Nous estimons les seconds moments sur les séquences imposteurs simplement pour alléger le système. D'une part, nous avons vu qu'une estimation grossière des seconds moments au niveau des séquences produisaient les mêmes performances qu'une estimation sur un nombre bien plus élevé de vecteurs (§5.2.3). D'autre part, estimer les seconds moments indépendamment du locuteur cible permet d'utiliser un seul et même noyau pour tous les locuteurs.

Finalement, nous considérons dans nos expériences le noyau de Fisher :

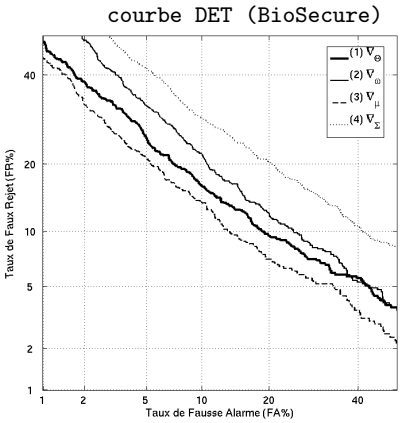
$$\kappa^{\mathrm{Fsh}}(\mathbf{X}, \mathbf{Y}) = \widetilde{\boldsymbol{\delta}}(\boldsymbol{\theta}_{\mathrm{o}}, \mathbf{X})^{\mathrm{T}}\ \widetilde{\boldsymbol{\delta}}(\boldsymbol{\theta}_{\mathrm{o}}, \mathbf{Y}) \tag{5.24}$$

Lors du développement du noyau de Fisher, l'influence de chaque catégorie de caractéristiques encodées par $\boldsymbol{\delta}^{(g)}$ (5.21) a été étudié empiriquement. Fig.5.13 montre les performances obtenus avec une *expansion* de Fisher :

1. classique (5.16),
2. restreinte aux dérivées partielles par rapport aux poids $\{\frac{\partial}{\partial \omega_g}\}$,
3. restreinte aux gradients par rapport aux moyennes $\{\boldsymbol{\nabla}_{\boldsymbol{\mu}_g}\}$,
4. restreinte aux gradients par rapport aux covariances $\{\boldsymbol{\nabla}_{\boldsymbol{\Sigma}_g}\}$.

À cause de la complexité calculatoire impliquée par la dimension de l'*expansion* de Fisher, nous nous sommes exceptionnellement limité à 512 composantes Gaussiennes pour ces expériences.

Les meilleures performances sont atteintes avec les dérivées par rapport à la moyenne. Même si le noyau correspondant fait intervenir une *expansion* de dimension Gd, il est plus robuste qu'un noyau standard qui met en jeu des *expansions* de dimension $G(2d+1)$. Cela peut s'expliquer par le fait que l'information de l'*expansion* de Fisher classique est redondante. En effet, pour une composante Gaussienne g et une coordonnée u données, les dérivées par rapport aux poids / moyennes / covariances font toutes intervenir les scalaires $\gamma_g^{\mathrm{o}}(\mathbf{x})$ et $(x_u - \mu_{g,u}^{\mathrm{o}})$. Si une caractéristique non linéaire $\phi\left(\gamma_g^{\mathrm{o}}(\mathbf{x}), x_u - \mu_{g,u}^{\mathrm{o}}\right)$ a un pouvoir discriminant plus fort qu'une autre caractéristique $\phi'\left(\gamma_g^{\mathrm{o}}(\mathbf{x}), x_u - \mu_{g,u}^{\mathrm{o}}\right)$, alors concaténer ϕ et ϕ' pénalise le pouvoir discriminant de ϕ.



Systèmes SVM :

(1) Noyau de Fisher standard κ^{Fsh} (UBM : 512 Gaussiennes)
(2) Restriction aux dérivées/$^{\mathrm{t}}$ poids $\boldsymbol{\delta} = \boldsymbol{\nabla}_{\{\omega_g\}}$
(3) " aux gradients/$^{\mathrm{t}}$ moyennes $\boldsymbol{\delta} = \boldsymbol{\nabla}_{\{\boldsymbol{\mu}_g\}}$
(4) " aux gradients/$^{\mathrm{t}}$ covariances $\boldsymbol{\delta} = \boldsymbol{\nabla}_{\{\boldsymbol{\Sigma}_g\}}$

	EER(%)	minDCF(x10^{-3})
(1)	13.45	56.4
(2)	15.31	68.2
(3)	11.61	52.2
(4)	20.18	75.0

Fig. 5.13 - Performances de développement du noyau de Fisher.

Noyau TOP

Le noyau TOP est similaire au noyau de Fisher. Il consiste à appliquer l'opérateur $\boldsymbol{\nabla}_{\boldsymbol{\theta}}$ (gradient par rapport aux paramètres de la modélisation) au log-rapport de vraisemblance.

Pour un locuteur cible donnée, supposons avoir entraîné un GMM en adaptant un UBM (paramètres $\boldsymbol{\theta}_{\mathrm{o}}$) sur une ou plusieurs séquences prononcées par le locuteur. En considérant la même adaptation MAP que dans §5.4.1, on peut noter $\boldsymbol{\theta}_{\mathrm{loc}} = \{\omega_g, \boldsymbol{\mu}_{\mathrm{loc},g}, \boldsymbol{\Sigma}_g\}$ les paramètres de ce GMM. L'*expansion* permettant de calculer le noyau TOP s'écrit alors :

$$\boldsymbol{\nabla}_{[\boldsymbol{\theta}^{+1}\ \boldsymbol{\theta}^{-1}]} \left(\log p(\mathbf{X}|\boldsymbol{\theta}^{+1}) - \log p(\mathbf{X}|\boldsymbol{\theta}^{-1})\right)\Big|_{[\boldsymbol{\theta}^{+1}\ \boldsymbol{\theta}^{-1}]=[\boldsymbol{\theta}_{\mathrm{loc}}\ \boldsymbol{\theta}_{\mathrm{o}}]} \tag{5.25}$$

En considérant la même normalisation que pour le noyau de Fisher, et avec les mêmes

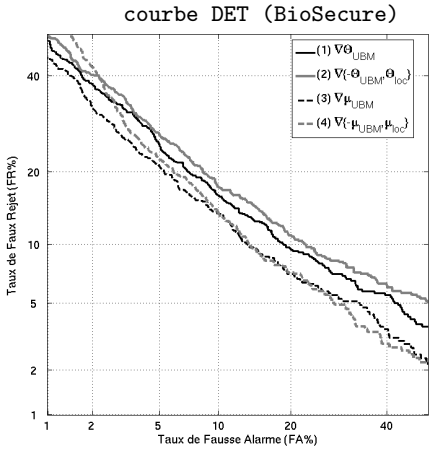
notations que précédemment, on peut formuler le noyau TOP comme suit :

$$\kappa^{\mathrm{TOP}}(\mathbf{X},\mathbf{Y}) = \widetilde{\boldsymbol{\delta}}(\boldsymbol{\theta}^{\pm},\mathbf{X})^{\mathrm{T}}\,\widetilde{\boldsymbol{\delta}}(\boldsymbol{\theta}^{\pm},\mathbf{Y}) \tag{5.26}$$

$$\text{avec}\quad \widetilde{\boldsymbol{\delta}}(\boldsymbol{\theta}^{\pm},\mathbf{X}) = \begin{bmatrix} \widetilde{\boldsymbol{\delta}}(\boldsymbol{\theta}_{\mathrm{loc}},\mathbf{X}) \\ -\widetilde{\boldsymbol{\delta}}(\boldsymbol{\theta}_{\mathrm{o}},\mathbf{X}) \end{bmatrix} \tag{5.27}$$

Un tel noyau présente un inconvénient pratique flagrant pour le problème de vérification du locuteur : il dépend du locuteur cible. Cette dépendance pénalise l'approche en terme de complexité calculatoire par rapport au noyau de Fisher standard. Admettons que l'on veuille classer une séquence selon plusieurs locuteurs cibles (dans un scénario de recherche de criminels par exemple). Avec l'approche Fisher classique, on peut garder en mémoire l'*expansion* normalisée pour ensuite calculer de manière efficace les scores fournis par différents modèles. Pour l'approche TOP, il est nécessaire de recalculer la moitié des composantes de l'*expansion* (5.27) à chaque test. La phase d'apprentissage des modèles est alourdie de façon analogue : à chaque nouveau locuteur cible, il faut recalculer les dérivées des vraisemblances des séquences imposteurs par rapport au modèle du locuteur.

De plus, les performances du noyau TOP sont moins bonnes que celles du noyau de Fisher, comme le montre Fig.5.14. Toutefois [Wan, 2003] rapporte une amélioration des performances sur la base de donnée PolyVar [Chollet et al., 1996] pour la vérification du locuteur (l'auteur ne donne pas la comparaison analogue sur la base de donnée YOHO). En comparaison des bases de données NIST ici considérées, la base PolyVar implique moins de locuteurs cibles avec plus de données d'apprentissage par locuteur[26], et met en jeu des enregistrements de meilleure qualité, avec une variation moindre du contenu phonétique. La dégradation que nous observons pour les noyaux TOP sur la base de donnée NIST peut venir du fait que la distribution des scores obtenus en phase de test est trop dépendante du locuteur cible, à cause de la spécificité des paramètres du noyau. Si c'est le cas, les performances pourraient être améliorées en appliquant une normalisation des scores du type Z-Norm (§2.1.6). Nous n'avons malheureusement pas testé une telle normalisation pour garder le protocole expérimental cohérent et équitable.



Systèmes SVM :
(1) Noyau de Fisher κ^{Fsh}
(UBM : 512 Gaussiennes)
(2) Noyau TOP κ^{TOP}
(3) Fisher restreint aux gradients/$^{\mathrm{t}}$ moyennes
(4) TOP restreint aux gradients/$^{\mathrm{t}}$ moyennes

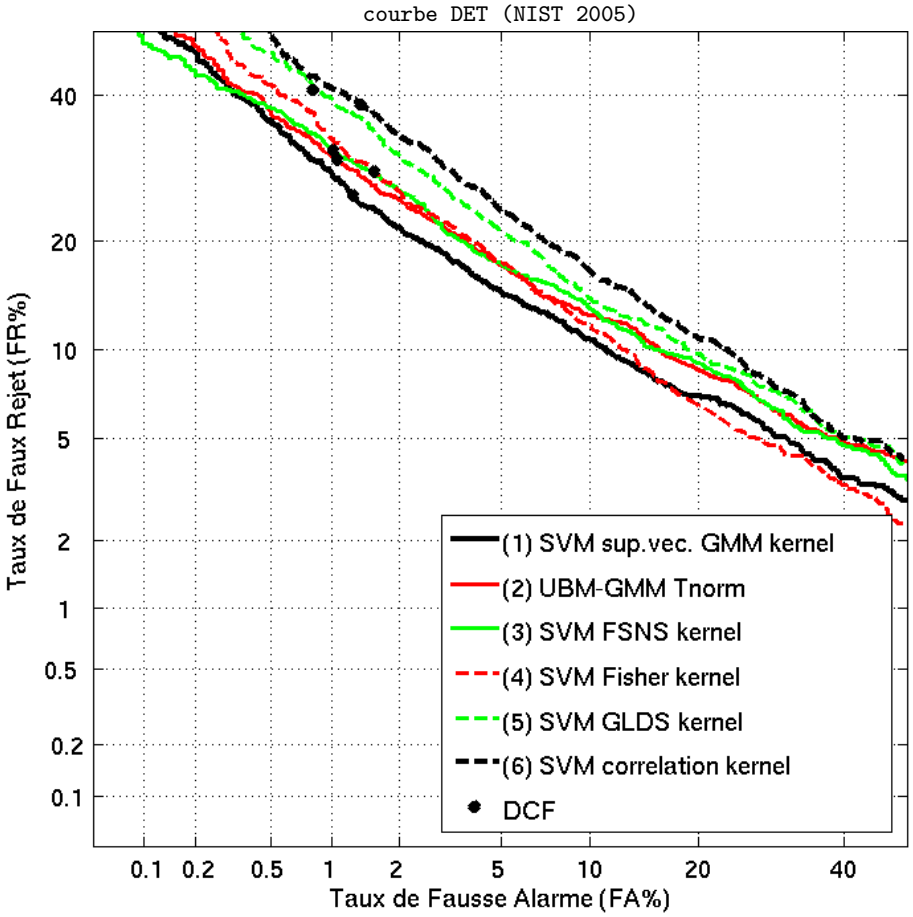
		EER(%)	minDCF(x10^{-3})
	(1)	13.45	56.4
	(2)	14.61	57.9
✔	(3)	11.61	52.2
	(4)	12.00	60.0

Fig. 5.14 - Performances de développement du noyau TOP comparé au noyau de Fisher.

[26] 38 locuteurs cibles - 85 phrases par locuteur pour l'apprentissage

5.5 Synthèse de l'évaluation

Les performances sur le corpus d'évaluation des différents systèmes à noyaux de séquences présentés dans ce chapitre (et de l'UBM-GMM de référence) sont montrées dans Fig.5.15, où les systèmes sont indexés par ordre décroissant de performance. Cette évaluation montre que la famille de noyaux la plus robuste pour la vérification du locuteur semblent être les noyaux dérivés des supervecteurs GMM (inspirés d'une distance de Kullback-Leibler). Hormis le noyau de produits de probabilité, les autres noyaux donnent globalement de bonnes performances, comparables à celles que l'on obtient avec la modélisation générative UBM-GMM.



Classement des systèmes par DCF

(1) SVM, Noyau "supervecteurs GMM"
(2) UBM-GMM (T-Norm)
(3) SVM, Noyau FSNS
(4) SVM, Noyau de Fisher
(5) SVM, Noyau GLDS
(6) SVM, Noyau de produit de probabilité

	EER(%)	DCF($\times 10^{-3}$) min	DCF($\times 10^{-3}$) réel
(1)	10.40	37.4	37.7
(2)	12.06	40.5	40.6
(3)	11.91	41.4	41.6
(4)	11.90	42.5	44.0
(5)	12.54	48.5	48.8
(6)	13.92	50.5	52.1

Fig. 5.15 - Évaluation des SVMs à noyaux de séquence.

Tous les systèmes n'impliquent pas les mêmes niveaux de complexité calculatoire. Dans la

liste suivante, les systèmes sont classés par ordre décroissant d'efficacité en phase de test sur le protocole d'évaluation NIST :

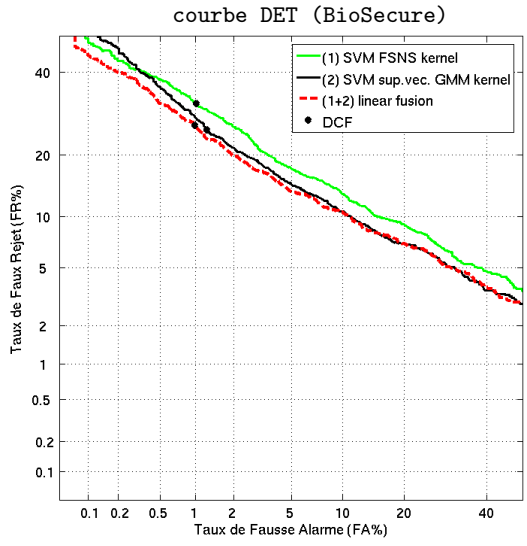
1. SVM à noyau GLDS
2. SVM à noyau FSNS
3. SVM à noyau de Fisher
   et UBM-GMM
4. SVM à noyau "supervecteurs GMM"
5. SVM à noyau de produit de probabilité

— *Non sélectionnés pour l'évaluation (complexité élevée et résultats moyens)* —

6. SVM à noyau TOP
7. SVM à noyaux vectoriels

La complexité relativement élevée du noyau "supervecteurs GMM" et du noyau de produit de probabilité viennent en particulier du fait que les modèles SVMs ne puissent pas être compactés comme c'est le cas lorsque la modélisation passe par une *expansion* des séquences dans un espace de dimension fini, suivie d'un noyau linéaire. Une telle réduction des modèles est possible pour les noyaux FSNS et le noyau de Fisher. La complexité élevée du noyau TOP vient quant à elle de la spécificité du noyau à chaque locuteur.

En ce qui concerne la fusion des systèmes, nous nous sommes limités à estimer les gains de performances que peuvent apporter les combinaisons linéaires des scores des systèmes. Nous avons combiné les systèmes deux à deux, et les poids de la combinaison linéaires ont été choisis de manière à minimiser la DCF sur le corpus de validation. Il ressort de cette étude que la fusion entre le système UBM-GMM et les autres noyaux de séquences basés sur les GMMs ne permet pas de gagner en robustesse, hormis pour le noyau de Fisher. Par contre, les performances sont généralement améliorées par la fusion d'un système SVM à noyau FSNS et d'un système SVM dont le noyau utilise une modélisation probabiliste. Fig.5.16 montre une telle amélioration avec le noyau "supervecteurs GMM" qui fournit individuellement les meilleurs résultats.



Systèmes SVM :
(1) SVM, Noyau FSNS
(UBM : 512 Gaussiennes)
(2) SVM, Noyau "supervecteurs GMM"
(1+2) Fusion linéaire de (1) et (2)

	EER(%)	DCF($\times 10^{-3}$) min	DCF($\times 10^{-3}$) réel
(1)	11.91	41.4	41.6
(2)	10.40	37.4	37.7
(1+2)	10.28	35.3	36.1

Fig. 5.16 - Gain de performance apporté par une fusion linéaire de deux systèmes SVMs à noyaux de séquence.

Chapitre 6

Noyau entre paires de séquences pour la vérification du locuteur

Sommaire

DANS ce chapitre, nous présentons un de nos travaux très récent [Louradour et Daoudi, 2007, Daoudi et Louradour, 2007]. Il s'agit d'une étude préliminaire sur une nouvelle façon d'envisager la vérification du locuteur, où la prise de décision est indépendante du locuteur cible. Le principe de base est de construire un système qui détermine *si deux séquences ont été prononcées ou non par un même locuteur*. Même si ce principe est général, nous utilisons les SVMs pour traiter le problème de reconnaissance du locuteur ainsi formulé. Nous appliquons le nouveau système à l'évaluation NIST considérée dans le chapitre précédent. Même si les performances n'égalent pas celles des systèmes classiques, elles restent acceptables et offrent de belles perspectives pour de futures améliorations.

6.1 Une nouvelle approche pour la vérification du locuteur

6.1.1 Principe général

Les systèmes classiques de vérification du locuteur, tels qu'ils ont été présentés dans le chapitre 2, sont basés sur un principe commun :

1. Apprendre un modèle représentant le locuteur cible, à partir d'un ensemble de séquences prononcées par ce locuteur (et d'un ensemble de séquences prononcées par d'autres locuteurs).
2. Attribuer un score à une séquence test étant donné le modèle cible.

Cette démarche est schématisée de façon sommaire dans Fig.6.1. L'illustration correspond au cas où une seule séquence par locuteur cible n'est disponible pour régler les modèles de locuteur, comme c'est le cas dans le protocole d'évaluation NIST qui reste la référence dans cette étude.

Les systèmes de vérification classiques donnent ainsi un rôle asymétrique à la (aux) séquence(s) "*train*" prononcée(s) par le locuteur cible et à la séquence "*test*" prononcée par un locuteur inconnu. Dans le protocole NIST de référence, on peut envisager de permuter les rôles de ces séquences, c'est-à-dire d'entraîner un modèle à partir de la séquence de test, et d'attribuer un score à la séquence prononcée par le locuteur cible. Cette idée a été proposée sous le nom de "*input swapping*" lors des évaluations NIST, entre autres par [Brümmer, 2005b].

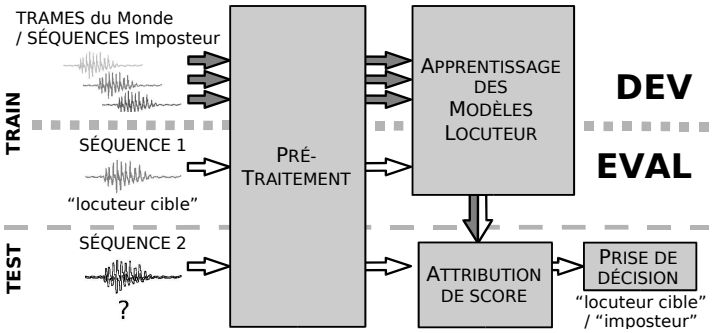
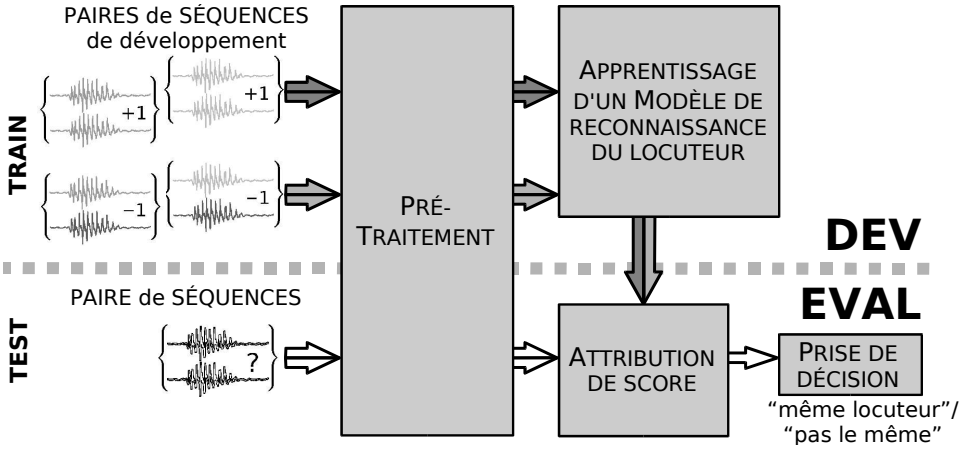


Illustration dans le cas où une seule séquence n'est disponible pour caractériser le "locuteur cible".

Fig. 6.1 - Approche classique pour la vérification du locuteur.

Motivé par ce constat, nous envisageons de concevoir un système de vérification du locuteur où les entrées sont des *paires de séquences* jouant un rôle symétrique, au lieu de séquences individuelles redirigées arbitrairement vers un module d'apprentissage ou de test. L'architecture générale d'un tel système est représentée dans Fig.6.2. L'objectif est de déterminer si les deux séquences de la paire *ont été prononcées ou non par le même locuteur*. Cette formulation du problème de reconnaissance du locuteur est celle rencontrée dans les approches classiques de segmentation du locuteur [Moraru et al., 2003]. Pour cette application où un enregistrement sonore doit être indexé selon les intervenants, l'enregistrement est d'abord découpé arbitrairement en segments de courte durée, qui sont ensuite regroupés de façon hiérarchique selon que le système détecte un même locuteur ou non.



Les entrées sont des paires de séquence. L'étiquette +1 (resp. -1) est attribuée aux paires de séquences prononcées par un même locuteur (resp. par des locuteurs différents).

Fig. 6.2 - Nouvelle approche pour la vérification du locuteur.

Données d'apprentissage

Les données d'apprentissage du nouveau système de vérification du locuteur (Fig.6.2) doivent être organisées différemment des données servant à l'apprentissage des modèles spécifiques aux locuteurs cibles utilisés dans les systèmes classiques (Fig.6.1). D'abord, les séquences de développement doivent être rangées par locuteur intervenant, alors que dans les approches classiques les séquences *train* sont rangées selon qu'elles ont été produites ou non par le locuteur cible. Ensuite, le corpus d'apprentissage doit être structuré de la même manière que le corpus de *validation* dans les approches classiques. Les données d'apprentissage du nouveau système sont des paires de séquences, étiquetées ± 1 selon qu'elles ont ou non été produites par le même locuteur. Pour simplifier la lecture dans ce qui suit, nous parlerons de "*trials*" ("positifs / négatifs") $\{\mathbf{X}, \mathbf{Y}\}$ pour les paires de séquences (étiquetées ± 1). Ce terme est emprunté au vocabulaire utilisé dans les évaluations NIST, où un *trial* désigne un couple {*locuteur cible – séquence test*}.

Notons que la nouvelle approche, qui prend en entrée des *trials*, peut être un remède à un facteur qui limite la capacité des approches discriminantes comme les SVMs : le déséquilibre entre le nombre d'entrées positives et d'entrées négatives en apprentissage [Mariéthoz, 2006]. En effet, dans le protocole d'évaluation NIST considéré dans cette étude, une seule séquence par locuteur cible est disponible pour apprendre les modèles d'un système SVM classique (contre plusieurs centaines pour caractériser les locuteurs imposteurs). Ce déséquilibre peut être facilement évité avec la nouvelle approche. Supposons disposer de n_l locuteurs de développement avec $S > 2$ séquences par locuteur. Un tel corpus de développement permet de simuler jusqu'à $n_\mathrm{l} S(S-1)/2$ *trials* positifs et $S^2 n_\mathrm{l}(n_\mathrm{l}-1)/2$ *trials* négatifs. En pratique, Il est possible de se limiter de manière à atteindre un rapport fixé entre *trials* positifs et *trials* négatifs.

Dans cette étude, nous ne traitons pas le cas où plus d'une séquence est disponible pour caractériser un locuteur cible. Plusieurs pistes sont envisageables pour appliquer la nouvelle approche à un tel protocole, comme combiner les sorties du système ou généraliser la méthode de manière à prendre en entrées non plus de paires mais des ensembles de séquences.

6.1.2 Travaux antérieurs

Un système à noyau entre paires de séquences

Un système similaire à celui suggéré dans la figure 6.2 a été proposé par [Brümmer, 2005b] lors du workshop NIST SRE 2005, mais n'a fait l'objet d'aucune publication. Même si les performances individuelles du système sont mauvaises (EER de l'ordre de 30% selon un communiqué personnel de Brümmer), la fusion avec d'autres systèmes classiques améliore sensiblement les performances. Nous présentons ici l'approche sous-jacente. Précisons avant de rentrer dans les détails qu'une des principales différences avec l'approche que nous proposons ensuite est que les séquences impliquées dans un *trial* ne jouent pas le même rôle : comme dans un système UBM-GMM, la séquence du locuteur cible est utilisée pour entraîner un GMM, et la séquence de test est utilisée par le biais de calculs de vraisemblances GMM.

Dans son raisonnement, Brümmer part d'un développement limité de Taylor au premier ordre du log-rapport de vraisemblance, identique à celui considéré dans le noyau de Fisher (§3.4.3). Ce développement s'écrit :

$$\tfrac{1}{T_{\mathbf{X}}}\big(\log p(\mathbf{X}|\boldsymbol{\theta}_{\text{loc}}) - \log p(\mathbf{X}|\boldsymbol{\theta}_{\text{o}})\big) \approx \tfrac{1}{T_{\mathbf{X}}} \underbrace{\boldsymbol{\nabla}_{\boldsymbol{\theta}} \log p(\mathbf{X}|\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{o}}}}_{\boldsymbol{\delta}(\mathbf{X})}^{\text{T}} \underbrace{\big(\boldsymbol{\theta}_{\text{loc}} - \boldsymbol{\theta}_{\text{o}}\big)}_{\boldsymbol{\Delta}(\boldsymbol{\theta}_{\text{loc}})}$$

où $\boldsymbol{\theta}_{\text{o}}$ réunit les paramètres de l'UBM, $\boldsymbol{\theta}_{\text{loc}}$ représente les paramètres d'un GMM adapté de l'UBM pour représenter un locuteur cible, et $\mathbf{X}$ est une séquence de test. Le vecteur $\boldsymbol{\delta}(\mathbf{X})$ est l'expansion de Fisher de la séquence de test, avec l'UBM comme distribution de probabilité *a priori*, et $\boldsymbol{\Delta}(\boldsymbol{\theta}_{\text{loc}})$ contient les différences entre les paramètres des GMMs locuteur cible et de l'UBM. Selon Brümmer, calculer le produit scalaire $\boldsymbol{\delta}(\mathbf{X})^{\text{T}}\boldsymbol{\Delta}(\boldsymbol{\theta}_{\text{loc}})$ fournit des résultats similaires à l'approche GMM classique. Son idée est d'introduire dans le calcul de ce produit scalaire une matrice de normalisation $\mathbf{Z}$ apprise de manière discriminante avec une méthode à noyau. Le score de sortie du système est de la forme :

$$f(\{\mathbf{X}, \boldsymbol{\theta}_{\text{loc}}\}) = \boldsymbol{\delta}(\mathbf{X})^{\text{T}}\mathbf{Z}\,\boldsymbol{\Delta}(\boldsymbol{\theta}_{\text{loc}}) + \beta_0 = \sum_i \alpha_i \ell_i \kappa\big(\{\mathbf{X}, \boldsymbol{\theta}_{\text{loc}}\}; \{\mathbf{X}_i, \boldsymbol{\theta}_i\}\big) + \beta_0 \qquad (6.1)$$

où κ est un noyau entre paires {*séquence test - modèle locuteur*}, où les paires d'apprentissage $\{\mathbf{X}_i, \boldsymbol{\theta}_i\}$ sont étiquetées $\ell_i = \pm 1$ selon la correspondance ou non à un même locuteur, et où les α_i sont des poids positifs appris de manière discriminante. Le noyau qui permet de déterminer $\mathbf{Z}$ est le noyau bilinéaire :

$$\kappa\big(\{\mathbf{X}_1, \boldsymbol{\theta}_1\}; \{\mathbf{X}_2, \boldsymbol{\theta}_2\}\big) = \boldsymbol{\delta}(\mathbf{X}_1)^{\text{T}}\boldsymbol{\delta}(\mathbf{X}_2)\;\boldsymbol{\Delta}(\boldsymbol{\theta}_1)^{\text{T}}\boldsymbol{\Delta}(\boldsymbol{\theta}_2)$$

[Brümmer, 2005b] fait le réglage des poids ℓ_i de (6.1) par Régression Logistique à noyau (*Regularized Kernel Logistic Regression*) [Zhu et Hastie, 2001]. Cette technique d'apprentissage est similaire aux SVMs (§1.3.2), à la différence près que la fonction de coût des erreurs dans le critère d'apprentissage a une forme différente : $\mathcal{L}(\ell, f(\mathbf{x})) = \log\left(1 + e^{-\ell f(\mathbf{x})}\right)$ au lieu du *hinge loss* (1.12). Avec une telle fonction de coût, les solutions apprises ne sont pas parcimonieuses, contrairement aux SVMs. L'avantage de la Régression Linéaire mis en avant Brümmer est que les scores renvoyés par les modèles appris se comportent comme des log-rapports de vraisemblance.

Étude des modèles de distribution sur les séquences

[Ben, 2004] suggère de traiter le problème de vérification du locuteur dans un espace des modèles. La démarche consiste en pratique à estimer les distributions GMM sur chacune des séquences d'un *trial*, et à mesurer une distance entre les GMMs. Cette distance est le score à partir duquel le classifieur prend la décision. Un tel système fait jouer un rôle symétrique aux deux séquences d'un *trial*. Le principal avantage mis en avant par [Ben, 2004] est l'économie des ressources mémoire que permettent de faire les modèles probabilistes en "compactant" l'information.

La méthode SVM proposée dans la section suivante a des liens étroits avec la méthode proposée par [Ben, 2004]. La grande différence est, qu'au lieu de fixer une mesure de distances entre modèles de façon arbitraire, nous apprenons cette mesure à partir d'exemples joués. Nos motivations rejoignent celles évoquées par [Brümmer, 2005b] : le but est de faire apprendre à la machine à distinguer les variations intralocuteurs des variations inter-locuteurs (dont le fameux "*channel mismatch*"). Nous verrons que notre approche, comme celle de Brümmer, revient à rendre plus complexe une modélisation de base en rajoutant des paramètres libres, et à apprendre ces paramètres de manière discriminante.

Critères de segmentation du locuteur

Comme mentionné précédemment, l'approche du problème de vérification que nous envisageons est analogue aux approches classiques pour la segmentation du locuteur. Il s'agit dans les deux cas de déterminer si deux séquences ont été prononcées ou non par un même locuteur. Ainsi, une première idée que nous avons eu était de se baser sur les critères classiques pour la segmentation du locuteur : le critère BIC [Chen et Gopalakrishnan, 1998] et le rapport de vraisemblance généralisé GLLR [Solomonoff et al., 1998]. Ces critères sont basés sur une modélisation paramétrique de la distribution des vecteurs sur les séquences. En segmentation du locuteur, ils sont habituellement appliqués à des modèles Gaussiens avec matrices de covariance pleines. Nous avons essayé de dériver plusieurs noyaux à partir d'une décomposition de ces critères formulés pour les GMMs, mais les systèmes SVM basés sur ces noyaux donnaient des performances nettement moins bonnes que l'approche que nous exposons maintenant [Daoudi et Louradour, 2007].

6.2 Nouveau système SVM à noyaux entre paires de séquences

Nous présentons maintenant un nouveau noyau entre paires de séquences pour la vérification du locuteur.

6.2.1 Conception du noyau

La problématique pour mettre au point un nouveau système SVM comme suggéré par Fig.6.2 est de concevoir un noyau entre *trials*. Au lieu de réfléchir directement en terme de noyau, il est plus naturel de concevoir un espace de représentation adéquat pour les paires de séquences. Ainsi, nous cherchons une expansion $\boldsymbol{\phi}(\{\mathbf{X}, \mathbf{Y}\})$ qui projette les *trials* dans un espace de Hilbert

de dimension fixe. À partir de cette expansion $\boldsymbol{\phi}$ nous pourrons définir un noyau de Mercer

$$\kappa\big(\ \{\mathbf{X}_1, \mathbf{Y}_1\}\ ,\ \{\mathbf{X}_2, \mathbf{Y}_2\}\ \big) = k\big(\ \boldsymbol{\phi}(\{\mathbf{X}_1, \mathbf{Y}_1\})\ ,\ \boldsymbol{\phi}(\{\mathbf{X}_2, \mathbf{Y}_2\})\ \big) \tag{6.2}$$

où k est un noyau vectoriel vérifiant les conditions de Mercer (§3.2). Un bon espace de représentation doit permettre une bonne séparation des *trials* positifs et des *trials* négatifs. Autrement dit, un vecteur caractéristique $\boldsymbol{\phi}$ correspondant à des paires de séquences similaires doit être relativement distant du vecteur $\boldsymbol{\phi}$ correspondant à des paires de séquences dissemblables.

Un bon moyen de mesurer la similitude entre deux séquences est d'analyser l'écart entre les distributions des vecteurs sur chacune des séquences. Pour faire cela, nous nous inspirons du noyau de séquence qui a fourni les meilleures performances dans nos expériences du chapitre 5 : le noyau entre supervecteurs GMM (§5.4.2). Nous rappelons que ce noyau se base sur une distance entre GMMs, dans le cas particulier où les GMMs sont adaptés d'un UBM en gardant fixes les poids et les matrices de covariance. Cette distance s'écrit :

$$\mathcal{D}^{\text{gm}}\big(\boldsymbol{\theta}_{\text{X}}, \boldsymbol{\theta}_{\text{Y}}\big)^2 = \sum_{g=1}^{G} \omega_g \left(\boldsymbol{\mu}_{X,g} - \boldsymbol{\mu}_{Y,g}\right)^{\text{T}} \boldsymbol{\Sigma}_g{}^{-1} \left(\boldsymbol{\mu}_{X,g} - \boldsymbol{\mu}_{Y,g}\right) \tag{6.3}$$

où l'on reprend les mêmes notations que dans le chapitre 5, à savoir :

- G est le nombre de composantes Gaussiennes de l'UBM (et des GMMs adaptés),
- ω_g et $\boldsymbol{\Sigma}_g$ sont respectivement le poids et la matrice de covariance de la $g^{\text{ième}}$ composante de l'UBM (et des GMMs adaptés),
- $\boldsymbol{\mu}_{X,g}$ désigne le vecteur moyenne de la $g^{\text{ième}}$ composante d'un GMM adapté sur une séquence $\mathbf{X}$
- $\boldsymbol{\theta}_{\text{X}}$ (resp. $\boldsymbol{\theta}_{\text{Y}}$) réunit les paramètres du GMM appris sur une séquence $\mathbf{X}$ (resp. $\mathbf{Y}$).

La distance (6.3) est une somme quadratique pondérée de distances de Mahalanobis entre les composantes Gaussiennes, que l'on note :

$$d_g(\boldsymbol{\mu}_{\text{X},g}, \boldsymbol{\mu}_{\text{Y},g}) = \sqrt{\left(\boldsymbol{\mu}_{X,g} - \boldsymbol{\mu}_{Y,g}\right)^{\text{T}} \boldsymbol{\Sigma}_g{}^{-1} \left(\boldsymbol{\mu}_{X,g} - \boldsymbol{\mu}_{Y,g}\right)} \tag{6.4}$$

Nous rappelons que ces distances au carré correspondent à la divergence de Kullback symétrique entre Gaussiennes, ou encore à la distance de Bhattacharyya modulo un facteur multiplicatif (§5.4.2). Plus deux séquences sont similaires et plus les distances d_g sont faibles. Ainsi ces distances sont des mesures adéquates pour construire l'espace de représentation recherché. Nous avons essayé plusieurs variantes pour construire l'expansion de *trials* $\boldsymbol{\phi}$, selon l'utilisation des poids ω_g et l'exposant de la distance. L'expansion qui a donné les meilleures performances empiriques est tout simplement :

$$\begin{aligned} \boldsymbol{\phi}(\{\mathbf{X}, \mathbf{Y}\}) &= \boldsymbol{\phi}(\{\boldsymbol{\theta}_{\text{X}}, \boldsymbol{\theta}_{\text{Y}}\}) \\ &= \left[\ d_1(\boldsymbol{\mu}_{\text{X},g}, \boldsymbol{\mu}_{\text{Y},g}) \quad \cdots \quad d_G(\boldsymbol{\mu}_{\text{X},g}, \boldsymbol{\mu}_{\text{Y},g})\ \right]^{\text{T}} \end{aligned} \tag{6.5}$$

Plus les distributions GMM correspondant aux séquences d'une paire sont proches et plus leur expansion $\boldsymbol{\phi}$ est proche de l'origine. Si un noyau linéaire est par exemple appliqué aux expansions $\boldsymbol{\phi}$, alors le modèle discriminant sera de la forme :

$$f(\{\mathbf{X}, \mathbf{Y}\}) = \sum_g \beta_g d_g(\boldsymbol{\mu}_{\text{X},g}, \boldsymbol{\mu}_{\text{Y},g}) + \beta_0 \tag{6.6}$$

Cette fonction score correspond à une mesure de distance entre distributions GMMs. Cette mesure peut être apprise de manière discriminante pour répondre au problème de vérification du locuteur. Dans nos expériences, nous utilisons un apprentissage SVM, et le noyau vectoriel qui a donné les meilleurs résultats est le noyau RBF Gaussien :

$$\kappa\big(\ \{\mathbf{X}_1, \mathbf{Y}_1\}\ ,\ \{\mathbf{X}_2, \mathbf{Y}_2\}\ \big) = e^{-\frac{\|\boldsymbol{\phi}(\{\mathbf{X}_1,\mathbf{Y}_1\}) - \boldsymbol{\phi}(\{\mathbf{X}_2,\mathbf{Y}_2\})\|^2}{2\rho^2}} \tag{6.7}$$

6.2.2 Normalisation

Nous rappelons que la normalisation du noyau et/ou des entrées est une étape cruciale pour la robustesse des méthodes à noyau qui ne sont pas invariantes aux transformations linéaires dans l'espace de représentation, comme c'est le cas des SVMs (§1.3.4). En particulier, nous avons vu que les performances empiriques du noyau entre supervecteurs GMMs étaient améliorées par une normalisation "D-MAP" des GMMs [Ben et Bimbot, 2003] (§5.4.2). Pour notre noyau entre paires de séquences, nous utilisons une normalisation similaire pour borner les expansions des *trials*.

Les vecteurs moyennes d'un GMM appris sur une séquence $\mathbf{X}$ (et adapté de l'UBM) sont réajustés selon :

$$\begin{aligned} \bar{\boldsymbol{\mu}}_{X,g} &= \lambda_{\mathrm{X}}\ \boldsymbol{\mu}_{X,g} + (1 - \lambda_{\mathrm{X}})\, \boldsymbol{\mu}_g^{\mathrm{o}} \\ \text{où}\ \ \lambda_{\mathrm{X}} &= \frac{1}{2\|\boldsymbol{\phi}\left(\{\boldsymbol{\theta}_{\mathrm{o}}, \boldsymbol{\theta}_{\mathrm{X}}\}\right)\|_2} \end{aligned} \tag{6.8}$$

où $\|\cdot\|_2$ est la norme euclidienne et où $\boldsymbol{\theta}_{\mathrm{o}}$ (resp. $\boldsymbol{\mu}_g^{\mathrm{o}}$) représente les paramètres (resp. les vecteurs moyennes) de l'UBM. Il est facile de vérifier que cette normalisation des modèles garantit que $\|\boldsymbol{\phi}\left(\{\boldsymbol{\theta}_{\mathrm{o}}, \bar{\boldsymbol{\theta}}_{\mathrm{X}}\}\right)\|_2 = 1/2$. Cela implique que les normes des expansions (6.5) des paires de GMMs adaptés par D-MAP sont majorés par 1, d'après l'inégalité triangulaire :

$$0 \leq\ \|\boldsymbol{\phi}\left(\{\bar{\boldsymbol{\theta}}_{\mathrm{X}}, \bar{\boldsymbol{\theta}}_{\mathrm{Y}}\}\right)\|_2\ \leq\ \|\boldsymbol{\phi}\left(\{\boldsymbol{\theta}_{\mathrm{o}}, \bar{\boldsymbol{\theta}}_{\mathrm{Y}}\}\right)\|_2 + \|\boldsymbol{\phi}\left(\{\boldsymbol{\theta}_{\mathrm{o}}, \bar{\boldsymbol{\theta}}_{\mathrm{Y}}\}\right)\|_2\ \leq 1$$

Nous considérons finalement l'expansion *normalisée* :

$$\bar{\boldsymbol{\phi}}(\{\mathbf{X}, \mathbf{Y}\}) = \boldsymbol{\phi}\left(\{\bar{\boldsymbol{\theta}}_{\mathrm{X}}, \bar{\boldsymbol{\theta}}_{\mathrm{Y}}\}\right) \tag{6.9}$$

Les composantes de cette expansion sont toutes comprises entre 0 et 1, comme c'est le cas après les techniques de normalisation classiques pour les méthodes à noyau [Sarle, 1997, Imbiriba et al., 2004]. Nous verrons que cette normalisation améliore les performances empiriques du système SVM.

6.3 Évaluation expérimentale

Dans cette section, nous appliquons le nouveau noyau entre paires de séquences en utilisant un apprentissage SVM. Les résultats finaux sont montrés sur l'évaluation NIST 2005 (limitée aux locuteurs femmes) tout comme dans le chapitre précédent. Pour simplifier la lecture, nous désignons par "système PoS" (*Pair-of-Sequences*) le système SVM avec le noyau entre paires de séquences défini dans la section précédente.

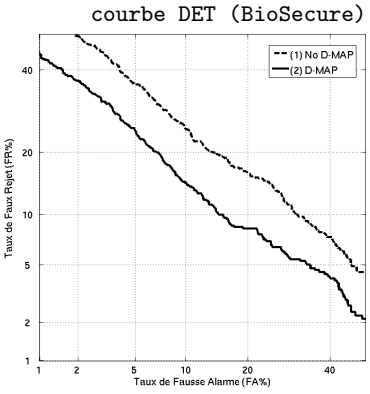
6.3.1 Protocole expérimental

Le protocole de développement BioSecure utilisé dans les expériences du chapitre précédent (§5.1.1) ne peut pas être appliqué tel quel avec la nouvelle approche, étant donné que les données d'apprentissage doivent être structurées sous forme de *trials* positifs/négatifs. Pour l'apprentissage du système PoS, le corpus de validation des systèmes classiques est augmenté d'autres *trials* formés à partir des séquences disponibles pour les locuteurs de NIST 2004 (aucune autre information n'est utilisée pour régler le système). Nous nous sommes arbitrairement limité à 16 166 *trials* faisant intervenir 277 locuteurs différents. Aussi, afin d'assurer l'adéquation du critère d'apprentissage du classifieur SVM avec la DCF = 0.1 FR% + 0.99 FA% à minimiser, nous avons fait en sorte qu'il y ait environ dix fois moins de *trials* positifs que de *trials* négatifs. Soulignons que les données de développement utilisées en apprentissage n'incluent aucun des locuteurs intervenant dans le corpus d'évaluation.

Le module de paramétrisation LFCC est appliqué en pré-traitement (§5.1.3), et la taille des modèles GMM a été arbitrairement fixée à 2048. Les paramètres du système PoS relatifs à l'apprentissage SVM sont réglés par validation croisée sur le corpus d'apprentissage. Il s'agit de la zone d'influence ρ pour un noyau RBF Gaussien (6.7) et du paramètre C de compromis biais-variance pour l'apprentissage SVM. Si l'on adoptait la même démarche qu'avec les systèmes classiques, le seuil de décision devrait aussi être choisi par validation croisée. Mais nous avons pu remarquer que les seuils de décision pour une DCF optimale étaient très proche de zéro, du fait de l'adéquation entre les proportions positifs/négatifs du corpus d'apprentissage et la DCF, comme mentionné ci-dessus. Les résultats sur le corpus d'évaluation NIST 2005 seront présentés avec un seuil de décision à zéro pour le système PoS. Notons que ce seuil est le seuil "normal" pour l'application des SVMs dans les situations standards.

6.3.2 Résultats

Les performances du système PoS sur le corpus de validation utilisé pour le développement des systèmes classiques (§5.1.2) sont montrées dans Fig.6.3. Les performances ont été obtenues par validation croisée et ne peuvent donc pas être comparées aux performances des systèmes classiques sur la même évaluation. Un phénomène intéressant illustré par Fig.6.3 est le gain en robustesse apporté par la normalisation par D-MAP.



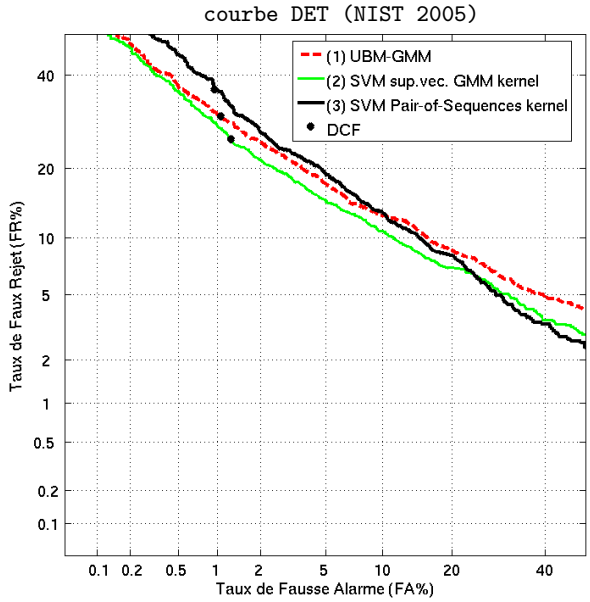
Systèmes SVM à noyau de paire de séquence (PoS) :
(1) sans D-MAP des GMMs
(2) avec D-MAP

		EER(%)	minDCF(x10^{-3})
	(1)	17.28	67.3
✔	(2)	12.29	53.4

Fig. 6.3 - Performances du système à noyau entre paires de séquences (validation croisée).

Les performances du système PoS sur le corpus d'évaluation sont montrés dans Fig.6.4. Même si elles n'atteignent pas les performances des systèmes conventionnels les plus robustes, elles sont encourageantes.

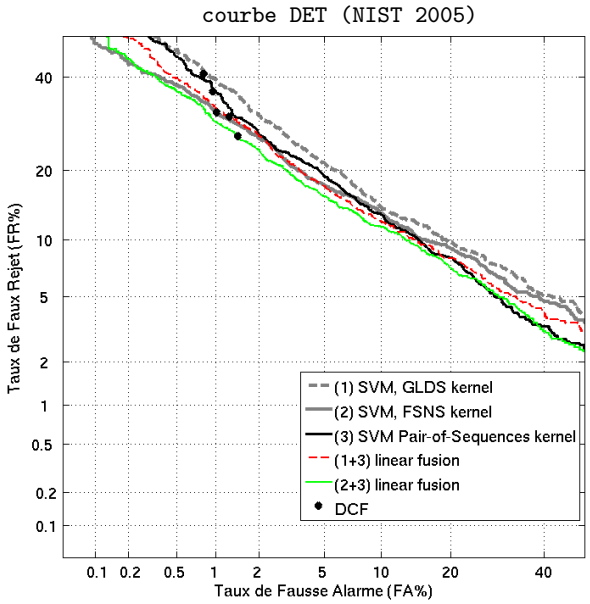
La fusion par combinaison linéaire des scores n'a pas permis d'améliorer les performances (DCF) des systèmes classiques basés sur la modélisation GMM (systèmes `(1)` et `(2)` de Fig.6.4(a)). Par contre, un gain en performance a été observé en combinant linéairement le système PoS avec les systèmes conventionnels "purement" discriminatifs, comme il est illustré dans Fig.6.4(b).



`(1)` UBM-GMM (T-Norm)
`(2)` SVM, Noyau "supervecteurs GMM"
`(3)` SVM, Noyau entre Paires de Séquences

	EER(%)	DCF($\times 10^{-3}$) min	DCF($\times 10^{-3}$) réel
`(1)`	12.06	40.5	40.6
`(2)`	10.40	37.4	37.7
`(3)`	11.58	44.1	46.0

(a) Performances individuelles.



`(1)` SVM, Noyau GLDS
`(2)` SVM, Noyau FSNS (Gaussien)
`(3)` SVM, Noyau entre Paires de Séquences
`(1+3)` Fusion linéaire de `(1)` et `(3)`
`(2+3)` Fusion linéaire de `(2)` et `(3)`

	EER(%)	DCF($\times 10^{-3}$) min	DCF($\times 10^{-3}$) réel
`(1)`	12.54	48.5	48.8
`(2)`	11.91	41.4	41.6
`(3)`	11.58	44.1	46.0
`(1+3)`	11.43	42.2	43.0
`(2+3)`	10.93	39.1	40.5

(b) Gains apportés par la fusion.

Fig. 6.4 - Évaluation du système SVM à noyaux entre paires de séquence.

À titre indicatif, le nombre de Vecteurs de Support avec la nouvelle approche SVM est de l'ordre de $n_o = 2000$ (sur 16 166 *trials* d'apprentissage) avec les paramètres optimaux déterminés par validation croisée sur le corpus d'apprentissage. En prenant $G = 2048$ composantes dans les GMMs, la taille du modèle "universel" de reconnaissance du locuteur appris par le système PoS

est $n_o G \approx 4{\times}10^6$. C'est très peu si l'on compare à son homologue dans les approches classiques : le système SVM à noyau entre supervecteurs GMM. Dans ce système, la taille *de chaque* modèle de locuteur cible est $n_o dG \approx 5{\times}10^6$, où n_o est le nombre de vecteurs de support (de l'ordre de 100 dans nos expériences) et d est la dimension de l'espace d'entrée. Précisons tout de même que la complexité calculatoire des deux systèmes sont comparables.

Soulignons que les résultats montrés dans ce chapitre correspondent à une première implémentation du système SVM à noyaux entre paires de séquences. Ils peuvent certainement être améliorés dans le futur par une plus grande expérience dans le développement de la nouvelle approche pour la reconnaissance du locuteur. Tout d'abord, il est tout à fait concevable de dériver une normalisation de type Z/T-Norm pour la nouvelle approche (§2.1.6). Il s'agirait par exemple de normaliser le score obtenu par un *trial* $\{\mathbf{X}, \mathbf{Y}\}$ en utilisant les statistiques des scores obtenus par les *trials* $\{\mathbf{X}, \mathbf{L}_i\}_i$ et $\{\mathbf{L}_i, \mathbf{Y}\}_i$ formés à partir de locuteurs de développement $\mathbf{L}_i$. Aussi, la fusion par combinaison linéaire n'est certes pas la manière optimale de fusionner l'information fourni par divers systèmes. Les résultats de fusion annoncés peuvent être améliorés en utilisant des techniques non linéaires plus élaborées.

Conclusion

Les approches probabilistes dominent l'état de l'art des méthodes utilisées en vérification automatique du locuteur (VAL). D'un autre coté, les Machines à Vecteurs de Support (SVM), basés sur la théorie d'apprentissage de Vapnik, sont réputés pour donner de bonnes performances dans de nombreux problèmes de classification. Bien que la VAL soit un problème de classification binaire, les SVMs ne sont pas encore une méthode classique pour la VAL. Même si quelques techniques pour appliquer les SVMs à la VAL commencent à émerger, elles ne sont que minoritairement utilisées lors des campagnes d'évaluation NIST en reconnaissance du locuteur. Plusieurs difficultés liées au problème de VAL en situation réelle sont à l'origine du retard de la mise en œuvre des SVMs pour cette application. Dans cette conclusion, nous commençons par récapituler les principales difficultés auxquelles ce travail de thèse apporte des réponses. Nous terminerons en soulevant des problématiques qui peuvent faire l'objet d'un développement supplémentaire de nos travaux.

Une première difficulté est la nature séquentielle de la parole. En fait, il ne s'agit pas là d'un véritable obstacle pour les SVMs, mais plutôt d'un axe de recherche récent. Même si les premières applications qui ont fait connaître le succès aux SVMs traitaient des objets non structurés de taille fixe, l'astuce du noyau permet aux SVMs de manipuler tout type de données. La problématique réside dans le choix du noyau. Les premières recherches portant sur les noyaux entre séquences de tailles variables datent de 1998. Dans cette étude, nous avons décrit les différentes manières possibles pour construire les noyaux de séquences. Nous nous sommes notamment intéressés aux noyaux entre ensembles vectoriels de tailles variables, c'est-à-dire aux noyaux qui ne modélisent pas les dépendances temporelles entre les vecteurs d'une séquence. Une telle focalisation est due au cadre applicatif : une majeure partie de l'information discriminante pour la VAL en mode "indépendant du texte" réside dans les statistiques des vecteurs acoustiques sur une séquence de parole. Les systèmes classiques, qui ne tiennent pas compte de l'ordre des vecteurs acoustiques dans les séquences, sont très robustes lorsque les conditions d'enregistrement sont bonnes et peu variables. La présente étude a mis en évidence trois principaux types de noyaux pouvant s'appliquer sur des ensembles vectoriels de tailles variables :

1. Les combinaisons de noyaux vectoriels ;
2. Les noyaux entre densités de probabilité ;
3. Les noyaux d'information mutuelle, qui sont des mesures de similarité construites à partir d'une distribution de probabilité.

Des systèmes SVMs basés sur chacun de ces trois types de noyaux ont été appliqués à la VAL sur des enregistrements téléphoniques mettant en jeu de fortes variabilités. Ils atteignent tous des performances acceptables par rapport à l'approche générative de référence basée sur les Mélanges

de Modèles Gaussiens (GMM). Dans notre étude, nous avons proposé une famille de noyaux appartenant à la première catégorie. Ces noyaux sont une généralisation d'un noyau connu dans la VAL pour son efficacité : le noyau GLDS. Le principe est de combiner des noyaux vectoriels normalisés afin d'exploiter des corrélations non linéaires entre les caractéristiques d'entrée. La généralisation proposée, bien que sujette à des approximations pour réduire la complexité, permet d'augmenter la robustesse du système SVM par rapport à un noyau GLDS conventionnel. Aussi nous avons établi un lien entre les noyaux proposés et les noyaux entre densités. Combiner les valeurs de noyaux vectoriels permet de manipuler implicitement des distributions dans un espace à très haute dimension : le "*Feature Space*", dont les coordonnées représentent autant de caractéristiques non linéaires extraites du signal d'entrée.

Une difficulté majeure pour la mise en application des SVMs à la VAL est liée au volume des bases de données nécessaires pour que les machines puissent apprendre à réaliser des tâches automatique sur le signal de parole de manière suffisamment robuste. La complexité calculatoire des algorithmes d'apprentissage et de prise de décision des SVMs peut être rédhibitoire dans le cas d'un corpus d'apprentissage trop volumineux. Ce mémoire montre deux solutions pour remédier à ce problème de façon élégante. Premièrement, adopter un noyau de séquence permet de synthétiser l'information de manière judicieuse. Deuxièmement, des approximations "de bas rang" des matrices contenant les valeurs de noyaux permettent de réduire les complexités calculatoire tout en gardant l'information essentielle. Pour la mise en application de nos travaux théoriques à la VAL, nous nous sommes appuyé sur une technique qui peut être appliquée à de grands volumes de données : la Décomposition de Cholesky Incomplète.

Une particularité spécifique à la VAL est le grand déséquilibre dans les données d'apprentissage. Dans beaucoup de contexte applicatifs, le nombre de séquences disponibles pour caractériser le "locuteur cible" est nettement moins élevé que le nombre de séquences nécessaires pour caractériser de manière fiable le "monde" des autres locuteur ("imposteurs" potentiels). Ce déséquilibre limite la capacité des SVMs à noyaux de séquences. Dans notre étude, nous introduisons une nouvelle approche qui permet de contourner ce problème dans le cas où une seule séquence d'apprentissage est accessible pour caractériser le locuteur cible. Nous appliquons à la VAL un classifieur SVM basé sur un *noyau entre paires de séquence*, dont l'objectif est de déterminer si deux séquences ont été prononcées ou non par le même locuteur. Même si les performances n'atteignent pas celles des meilleurs systèmes SVM à noyaux de séquences, elles sont acceptables et encouragent de futurs travaux dans la conception de la nouvelle catégorie de noyaux considérée. Le noyau entre paires de séquence que nous proposons est construit de manière intuitive. Il est inspiré d'un noyau de séquences qui permet d'atteindre des performances particulièrement bonnes en VAL. Ce noyau tire parti de la maîtrise des GMMs après une longue expérience relative à cette modélisation paramétrique dans la VAL.

Perspectives

Les techniques d'adaptation des modèles sont cruciales pour améliorer la robustesse des systèmes de traitement automatique de la parole. Elles permettent entre autres de limiter la dégradation des performances lorsque les modèles appris sous certaines conditions d'enregistrement sont appliqués dans d'autres conditions. Cette variabilité est connue sous le nom de "*channel mismatch*" dans la communauté de la VAL. Il s'agit d'un des principaux facteurs qui limitent la robustesse des systèmes lors des évaluations NIST. L'adaptation des modèles probabilistes est

maintenant bien maîtrisée et les méthodes génératives classiques pour la VAL peuvent bénéficier de ce savoir-faire pour atteindre de très bonnes performances pour des conditions d'utilisation variées. A contrario, les techniques d'adaptation sont à l'heure actuelle difficiles à mettre en œuvre avec les méthodes discriminantes à noyaux comme les SVMs. Quelques techniques commencent à émerger pour certains types de noyaux qui ont une forme particulière. Une étude importante reste à faire pour concevoir des techniques génériques pouvant s'appliquer à toute sorte de noyaux de séquences.

Un autre moyen de gagner en robustesse est de combiner plusieurs sources d'information ou plusieurs techniques de modélisation. En ce qui concerne la "fusion" de systèmes, nous nous sommes limités dans nos expériences à calculer des sommes pondérées de scores fournis par différents systèmes basés sur les paramètres acoustiques. Même si cela permet de se faire une idée des corrélations entre les erreurs commises par différents types de classifieurs, la combinaison linéaire des scores n'est certainement pas la manière optimale de gagner en performance. Une analyse plus fine des scores de sortie pourrait constituer un développement supplémentaire de ces travaux.

Signalons enfin que, d'un point de vue théorique, la nouvelle famille de noyaux de séquences que nous avons proposé (FSNS) ont un lien étroit avec la théorie émergente des processus Gaussiens [Rasmussen et Williams, 2006]. En effet, une distribution Gaussienne dans le *Feature Space* est équivalente à un processus Gaussien dans l'espace de Hilbert à noyau reproduisant (RKHS). Il serait donc intéressant d'analyser ces noyaux dans le cadre de ce formalisme pour voir les perspectives qu'il peut offrir.

A

Annexes

A.1 Quelques notions de calcul matriciel

A.1.1 Décomposition en Valeurs Singulières mince

La décomposition d'une matrice désigne sa réécriture en un produit de matrices vérifiant des propriétés particulières. Les décompositions matricielles sont utilisées soit pour réduire la complexité calculatoire, soit pour prouver des relations de manière synthétique. Dans le présent mémoire, nous avons fait référence à la Décomposition en Valeurs Singulières (SVD) pour faciliter certaines preuves (§4.3.2 et §4.4.2).

Le théorème central de la SVD est le suivant [Golub et Van Loan, 1996] :

Théorème 11 (SVD).
Toute matrice $\mathbf{M}$ *de taille* $D{\times}N$ *peut se décomposer sous la forme*

$$\mathbf{M} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}}$$

$$\text{où} \begin{cases} \mathbf{U} \text{ et } \mathbf{V} \text{ sont des matrices orthonormales de tailles } D{\times}D \text{ et } N{\times}N \\ \mathbf{D} \text{ est une matrice } D{\times}N. \text{ dont les valeurs non diagonales sont nulles.} \end{cases}$$

La matrice $\mathbf{D}$ est la généralisation d'une matrice diagonale qui indexe par $D_{i,i}$ les valeurs singulières de $\mathbf{M}$, habituellement rangées en ordre décroissant. Ces valeurs singulières sont en fait les valeurs propres de la matrice $\mathbf{M}^{\mathrm{T}}\mathbf{M}$, dont les vecteurs propres sont les vecteurs colonnes de V (base orthonormale). Aussi les vecteurs colonnes de $\mathbf{U}$ sont les vecteurs propres de $\mathbf{M}\mathbf{M}^{\mathrm{T}}$.

Les matrices $\mathbf{M}^{\mathrm{T}}\mathbf{M}$ et $\mathbf{M}\mathbf{M}^{\mathrm{T}}$ sont de même rang que la matrice $\mathbf{M}$. Ce rang r est inférieur ou égal à la plus petite dimension de $\mathbf{M}$ ($\min\{D, N\}$). Dans les cas où il est *strictement* inférieur à cette dimension, alors la décomposition peut prendre une autre forme car les valeurs singulières $D_{i,i}$ sont nulles au-delà de $i > r$. Une formulation alternative du théorème 11 introduit la "SVD mince" pour manipuler ces cas :

Théorème 12 (*thin* SVD).
Toute matrice $\mathbf{M}$ *non nulle de taille* $D{\times}N$ *peut se décomposer sous la forme :*

$$\mathbf{M} = \mathbf{U}_r\mathbf{D}_r{\mathbf{V}_r}^{\mathrm{T}} , \qquad 0 < r \leq \min\{D, N\}$$

$$\text{où} \begin{cases} \mathbf{U}_r \text{ et } \mathbf{V}_r \text{ sont des matrices orthonormales de tailles } D{\times}r \text{ et } N{\times}r. \\ \mathbf{D}_r \text{ est une matrice diagonale } r{\times}r \text{ à coefficients diagonaux non nuls.} \end{cases}$$

Les vecteurs colonnes de $\mathbf{U}_r$ (resp. $\mathbf{V}_r$) sont les r premières colonnes de $\mathbf{U}$ (resp. $\mathbf{V}$) et la diagonale de $\mathbf{D}_r$ contient les r valeurs singulières non nulles.La propriété orthogonale pour les matrices $\mathbf{U}_r$ et $\mathbf{V}_r$ (non carrées) s'écrit : ${\mathbf{U}_r}^{\mathrm{T}}\mathbf{U}_r = \mathbf{I}_r = {\mathbf{V}_r}^{\mathrm{T}}\mathbf{V}_r$.

A.1.2 Pseudo-inversion de matrice

La pseudo-inverse permet de généraliser la notion d'inverse d'une matrice pour une matrice quelconque, même rectangulaire ou non inversible. Parfois désignée comme "inverse de Moore-Penrose", elle est définie de manière unique [Golub et Van Loan, 1996].

Définition 9 (Pseudo-inverse d'une matrice).
La pseudo inverse d'une matrice $\mathbf{M}$ est la matrice $\mathbf{M}^\dagger$ qui vérifie les quatres conditions :

(i) $\mathbf{M}\mathbf{M}^\dagger\mathbf{M} = \mathbf{M}$
(ii) $\mathbf{M}^\dagger\mathbf{M}\mathbf{M}^\dagger = \mathbf{M}^\dagger$
(iii) $\mathbf{M}\mathbf{M}^\dagger$ est symétrique
(iv) $\mathbf{M}^\dagger\mathbf{M}$ est symétrique

La matrice $\mathbf{M}^\dagger$ est défini de manière unique pour tout $\mathbf{M}$. Elle est égale à l'inverse $\mathbf{M}^{-1}$ dans les cas où $\mathbf{M}$ est une matrice carrée inversible.

Il existe plusieurs manières de calculer $\mathbf{M}^\dagger$ selon la taille et le rang de $\mathbf{M}$. Dans notre étude, nous avons évoqué la pseudo-inversion pour des matrice carrés symétriques éventuellement non inversibles (§4.3et §4.4). Dans le cas, la pseudo-inverse peut être obtenue à partir d'une SVD de $\mathbf{M}$ (pour laquelle $\mathbf{U} = \mathbf{V}$ si $\mathbf{M}$ est inversible).

Théorème 13.
Si $\mathbf{M}$ *est une matrice carrée symétrique, alors sa pseudo-inverse est donnée par :*

$$\mathbf{M}^\dagger = \mathbf{U}_r\mathbf{D}_r{}^{-1}\mathbf{U}_r{}^{\mathrm{T}}$$

où $\mathbf{S}_r$ *et* $\mathbf{U}_r$ *sont obtenus par une décomposition SVD mince* $\mathbf{M} = \mathbf{U}_r\mathbf{D}_r\mathbf{U}_r{}^{\mathrm{T}}$.

Les valeurs propres de $\mathbf{M}^\dagger$ sont les inverses des valeurs propres non nulles de $\mathbf{M}$. Notons que dans les cas où $\mathbf{M}$ n'est pas inversible, $(\mathbf{M}^\dagger)^\dagger \neq \mathbf{M}$.

A.2 Algorithme de décomposition de Cholesky et ICD

L'algorithme de Cholesky permet de calculer la racine carrée d'une matrice *symétrique définie positive* qui soit triangulaire inférieure :

$$\mathbf{K} = \mathbf{G}\mathbf{G}^{\mathrm{T}}$$

La matrice triangulaire $\mathbf{G}$ vérifiant cette condition est définie de manière unique. Si $\mathbf{K}$ n'est pas inversible (simplement semi-définie positive), alors $\mathbf{G}$ existe mais n'est pas définie de manière unique. Dans ce cas l'algorithme de Cholesky, tel qu'il est formulé dans Tab.A.5 n'est pas applicable (problème d'inversion par zéro lors d'une itération).

Tab. A.5 - Algorithme de décomposition de Cholesky

- Entrées
  - Matrice définie positive $\mathbf{K}$ de taille $N \times N$.

ou

  - Données distinctes $\{\mathrm{b}_i\}_{i=1\cdots N}$ et noyau défini positif k : $\mathbf{K}[i,j] = k(\mathrm{b}_i, \mathrm{b}_j)$.
- Itération $i = 1 \cdots N$

  Calcul de la $i^{\text{ème}}$ colonne de $\mathbf{G}$:

  (Si $i = 1$) $\mathbf{G}[i,i] \coloneqq \sqrt{\mathbf{K}[i,i]}$

  (Si $i > 1$) $\mathbf{G}[i,i] \coloneqq \sqrt{\mathbf{K}[i,i] - \sum_{l=1}^{i-1} \mathbf{G}[i,l]^2}$

  Pour tout $j = (i+1) \cdots N$,

  (Si $i = 1$) $S \coloneqq 0$

  (Si $i > 1$) $S \coloneqq \sum_{l=1}^{i-1} \mathbf{G}[j,l]\mathbf{G}[i,l]$

  $\mathbf{G}[j,i] \coloneqq \dfrac{\mathbf{K}[i,j] - S}{\mathbf{G}[i,i]}$
- Sorties

  Matrice triangulaire inférieure $\mathbf{G}$ telle que $\mathbf{K} = \mathbf{G}\mathbf{G}^{\mathrm{T}}$.

Étant donnée une matrice symétrique semi-définie positive $\mathbf{K}$, l'algorithme de Décomposition de Cholesky Incomplète [Fine et Scheinberg, 2001] permet de trouver une approximation d'une racine carré $\mathbf{G}$ ayant un rang m inférieur ou égal à celui de $\mathbf{K}$:

$$\mathbf{K} \approx \mathbf{G}\mathbf{G}^{\mathrm{T}}$$

où $\mathbf{G}$ est de taille $N \times m$, si N est la taille de $\mathbf{K}$. Dans le cas où m atteind le rang de $\mathbf{K}$ (inférieur ou égal à N), alors l'algorithme ICD trouve une racine carrée exacte. Dans le cas où l'on fixe une valeur maximale pour le rang m de sortie, l'objectif de l'ICD est de minimiser une norme de l'écart entre $\mathbf{K}$ et son approximation $\mathbf{G}\mathbf{G}^{\mathrm{T}}$. Comme nous l'avons vu en §4.4.5, il s'agit de la somme des valeurs propres, ou encore la trace, de la matrice résiduelle : $\mathrm{tr}(\mathbf{K} - \mathbf{G}\mathbf{G}^{\mathrm{T}})$. Cette quantité est une norme pour les matrices symétriques semi-définies positives, et [Bach et Jordan, 2005] montre que la matrice résiduelle vérifie ces conditions.

Tab.A.6 décrit en détail les étapes de l'algorithme ICD. Il s'agit d'un algorithme glouton inspiré de la décomposition (complète) de Cholesky. La principale différence est qu'à chaque itération, un pivot est choisi de manière à réduire au mieux la trace du résidu. À la fin des itérations, l'approximation $\mathbf{H}$ *triangulaire inférieure* calculée par l'ICD est en fait une approximation

d'une matrice $\mathbf{L}$ obtenue par simple permutation des éléments de $\mathbf{K}$. En gardant en mémoire l'index des pivots à l'origine des permutations, on peut facilement retrouver la racine carrée (non triangulaire) $\mathbf{G}$ telle que $\mathrm{tr}(\mathbf{K} - \mathbf{G}\mathbf{G}^{\mathrm{T}}) = \mathrm{tr}(\mathbf{L} - \mathbf{H}\mathbf{H}^{\mathrm{T}})$. L'algorithme ICD permet de connaître une borne supérieure de ce résidu, qui peut être un critère d'arrêt tant que le rang maximal m fixé n'est pas atteind.

Tab. A.6 - Algorithme de décomposition de Cholesky incomplète (ICD)

• Entrées

- Matrice semi-définie positive $\mathbf{K}$ de taille $N \times N$.

ou

- Ensemble de données $\{\mathrm{b}_i\}_{i=1\cdots N}$ et noyau défini positif k : $\mathbf{K}[i,j] = k(\mathrm{b}_i, \mathrm{b}_j)$.

• Initialisation ($i = 0$)

- Initialisation des listes d'indices $I \coloneqq \{\}$ $\quad J \coloneqq \{1 \cdots N\}$
- Stockage des valeurs diagonales de $\mathbf{K}$ dans $\mathbf{D} \coloneqq \{\mathbf{K}[1,1] \cdots \mathbf{K}[N,N]\}$

• Itération (i)

- Choix du meilleur pivot $p_i \coloneqq \arg\max_j \mathbf{D}[j,j]$
- Codage de la permutation :
  (Actualisaction les index) $\quad I[i] \coloneqq p_i \quad J \coloneqq J \setminus J[p_i]$
  (Permutation lignes de $\mathbf{H}$) Pour tout $j = 1 \cdots i$, $\mathbf{H}[p_i, j]$ <-> $\mathbf{H}[i,j]$
- Calcul de la $i^{\text{ième}}$ colonne de $\mathbf{H}$:
  $\mathbf{H}[i,i] \coloneqq \sqrt{\mathbf{D}[p_i, p_i]}$
  Pour tout $j = (i+1) \cdots N$,
  (Si $i = 1$) $S \coloneqq 0$
  (Si $i > 1$) $S \coloneqq \sum_{l=1}^{i-1} \mathbf{H}[j,l]\mathbf{H}[p_i,l]$
  $\mathbf{H}[j,i] \coloneqq \dfrac{\mathbf{K}\big[p_i, J[j-i]\big] - S}{\mathbf{H}[i,i]}$
- Ré-estimation des résidus par ligne dans $\mathbf{D}$:
  Pour tout $j = (i+1) \cdots N$,
  $\mathbf{D}[j] \coloneqq \mathbf{K}\big[J[j-i], J[j-i]\big] - \sum_{l=1}^{i} \mathbf{H}[j,l]^2$
- Estimation du résidu global $\eta \coloneqq \sum_{j=i+1}^{N} \mathbf{D}[j]$

• Test d'arrêt

- Lorsque le résidu global η devient suffisamment faible.

ou

- Lorsque i a atteind un seuil maximal fixé (inférieur à N).

• Sorties

- Index ordonné $I = \{p_1 \cdots p_i\}$ des pivots, ou *dictionnaire* correspondant $\{\mathrm{b}_i\}_{i \in I}$.

et/ou

- Matrice triangulaire inférieure $\mathbf{H}$, approximation d'une racine carrée de :
  $\mathbf{L} = \mathbf{K}[\{I\ J\}, \{I\ J\}]$ (permutation de $\mathbf{K}$ selon I et son complémentaire ordonné).

et/ou

- Résidu η vérifiant $\|\mathbf{L} - \mathbf{H}\mathbf{H}^{\mathrm{T}}\| \leq \eta$.

A.3 Algorithme EM

L'algorithme EM (*Expectation Maximisation*) permet de régler les paramètres d'un modèle de distribution GMM pour atteindre un maximum (local) de vraissemblance d'un ensemble d'observations. Ces observations sont typiquement des vecteurs d'apprentissage non étiquetés. Les paramètres libres sont constitués des poids, des vecteurs moyennes et des matrices de covariance du GMM. Dans Tab.A.7, nous les notons respectivement $\omega[g]$, $\boldsymbol{\mu}[g]$ et $\boldsymbol{\Sigma}[g]$, où g désigne l'index des Gaussiennes. Le nombre de Gaussiennes composant le GMM doit être fixé avant de dérouler l'algorithme EM. Aussi plusieurs techniques d'initialisation sont envisageables avant de commencer les itérations d'EM. Nous citons deux exemples dans Tab.A.7.

Tab. A.7 - Algorithme EM

• Entrées :

- Vecteurs d'apprentissages non étiquetés $\{\mathbf{b}_i\}_{i=1\cdots N}$.
- nombre de Gaussiennes G.

• Initialisation (t=0)

- Extraction aléatoire parmi les données d'apprentissage pour initialiser les moyennes $\boldsymbol{\mu}^{(0)}[g]$.
- Initialiaton des matrices de covariance $\boldsymbol{\Sigma}^{(0)}[g]$ à la matrice unité.
- Initialisation équiprobable des poids $\omega^{(0)}[g] \coloneqq {}^1\!/_G$.

ou

- Utilisation d'un algorithme de quantification vectorielle non supervisé. Le dictionnaire est utilisé pour classer les vecteurs par *plus proche voisin*. Les poids, moyennes et covariances initiaux sont alors estimés sur chaque *cluster*.

• Itération (t+1)

Pour tout g={1...G},

-E/ *Phase d'estimation :* Calcul des probabilités que chaque vecteur $\mathbf{b}_i$ ait été généré par la Gaussienne g.

$$\mathrm{P}[g,i] \coloneqq p(g|\mathbf{b}_i) = \frac{\omega[g]\mathcal{N}(\mathbf{b}_i|\boldsymbol{\mu}[g],\boldsymbol{\Sigma}[g])}{\sum_{h=1}^{G}\omega[h]\mathcal{N}(\mathbf{b}_i|\boldsymbol{\mu}[h],\boldsymbol{\Sigma}(t)[h])}$$

-M/ *Phase de maximisation :* Réestimation des paramètres.

$$\omega[g] \coloneqq \frac{1}{N}\sum_{i=1}^{N}\mathrm{P}[g,i]$$

$$\boldsymbol{\mu}[g] \coloneqq \frac{\sum_{i=1}^{N}\left(\mathrm{P}[g,i]\,\mathbf{b}_i\right)}{\sum_{i=1}^{N}\mathrm{P}[g,i]}$$

$$\boldsymbol{\Sigma}[g] \coloneqq \frac{\sum_{i=1}^{N}\left(\mathrm{P}[g,i](\mathbf{b}_i-\boldsymbol{\mu}[g])(\mathbf{b}_i-\boldsymbol{\mu}[g])^{\mathrm{T}}\right)}{\sum_{i=1}^{N}\mathrm{P}[g,i]}$$

• Test d'arrêt

- Lorsque la variation de la vraisemblance normalisée du corpus d'apprentissage (estimée lors de E) devient suffisamment faible.

ou

- Lorsque t a atteind un seuil maximal fixé.

• Sorties : Paramètres du GMM $\{\omega[g],\ \boldsymbol{\mu}[g],\ \boldsymbol{\Sigma}[g]\}$.

Bibliographie

[Abe, 2003] Abe, S. (2003). On invariance of support vector machines. Dans *Proc. Int. Conf. on Intelligent Data Engineering and Automated Learning (IDEAL)*.

[Adami et al., 2003] Adami, A., Mihaescu, R., Reynolds, D., et Godfrey, J. (2003). Modeling prosodic dynamics for speaker recognition. Dans *Proc. of Int. Conf. on Acoustics Speech and Signal Processing (ICASSP)*.

[Adhikara et Joshi, 1956] Adhikara, B. et Joshi, D. (1956). Distance discrimination et resume exhaustif. *Publs. Inst. Stats*, 5 :57–74.

[Andrews et al., 2002] Andrews, W., Kohler, M., Campbell, J., Godfrey, J., et Hernandez-Cordero, J. (2002). Gender-dependent phonetic refraction for speaker recognition. Dans *Proc. ICASSP*.

[Arcienega et Drygajlo, 2002] Arcienega, M. et Drygajlo, A. (2002). A bayesian network approach for combining pitch and spectral envelope features for speaker verification. Dans *Proc. COST 275 Workshop on the Advent of Biometrics on the Internet*.

[Arias et al., 2005] Arias, J., Pinquier, J., et André-Obrecht, R. (2005). Evaluation of classification techniques for audio indexing. Dans *Proc. EUSIPCO*.

[Auckenthaler et al., 2000] Auckenthaler, R., Carey, M., et Lloyd-Thomas, H. (2000). Score normalization for text-independent speaker verification systems. *Digital Signal Processing*, 10(1-3) :42–54.

[Bach et Jordan, 2002] Bach, F. et Jordan, M. (2002). Kernel independent component analysis. *Journal of Machine Learning Research*, 3 :1–48.

[Bach et Jordan, 2005] Bach, F. et Jordan, M. (2005). Predictive low-rank decomposition for kernel methods. Dans *Proc. ICML*.

[Bahler et al., 1994] Bahler, L., Porter, J., et Higgins, A. (1994). Improved voice identification using a nearest-neighbor distance measure. Dans *Proc. (ICASSP)*.

[Bahlmann et al., 2002] Bahlmann, C., Haasdonk, B., et Burkhardt, H. (2002). On-line handwriting recognition with support vector machines : a kernel approach. Dans *Proc. IWFHR*.

[Baker, 1977] Baker, C. (1977). *The numerical treatment of integral equations*. Oxford : Clarendon Press.

[Barras et Gauvain, 2003] Barras, C. et Gauvain, J.-L. (2003). Feature and score normalization for speaker verification of cellular data. Dans *Proc. ICASSP*.

[Baudat et Anouar, 2000] Baudat, G. et Anouar, F. (2000). Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12(10) :2385–2404.

[Baum et al., 1970] Baum, L., Petrie, T., Soules, G., et Weiss, N. (1970). A maximization technique occuring in the statistical analysis of probabilistic functions of markov chains. *Annals of Mathematical Statistics*, 41(1) :164–171.

[Bellman, 1961] Bellman, R. (1961). *Adaptative Control Processes : a Guided Tour*. Princeton University Press.

[Ben, 2004] Ben, M. (2004). *Approches Robustes pour la Vérification Automatique du Locuteur par Normalisation et Adaptation Hiérarchique*. Thèse de doctorat, Université de Rennes 1.

[Ben et Bimbot, 2003] Ben, M. et Bimbot, F. (2003). D-MAP : a distance-normalized MAP estimation of speaker models for automatic speaker verification. Dans *Proc. ICASSP*.

[Ben et al., 2002] Ben, M., Blouet, R., et Bimbot, F. (2002). A Monte-Carlo Method for Score Normalization in Automatic Speaker Verification using Kullback-Leibler Distances. Dans *Proc. ICASSP*.

[Bengio et Mariéthoz, 2001] Bengio, S. et Mariéthoz, J. (2001). Learning the decision function for speaker verification. Dans *Proc. ICASSP*.

[Bengio et Mariéthoz, 2004] Bengio, S. et Mariéthoz, J. (2004). The expected performance curve : a new assessment measure for persion authentification. Dans *Proc. IEEE Odyssey : The Speaker and Language Recognition Workshop*.

[Bengio et LeCun, 2006] Bengio, Y. et LeCun, Y. (2006). *Large Scale Kernel Machines*, chapter Scaling Learning Algorithms towards AI (1.4 Fundamental Limitation of Local Learning). MIT Press.

[Berg et al., 1984] Berg, C., Reus Christensen, J., et Ressel, P. (1984). *Harmonic Analysis on Semigroups*. Springer Verlag.

[Besacier et Bonastre, 1998] Besacier, L. et Bonastre, J.-F. (1998). Frame pruning for speaker recognition. Dans *Proc. ICASSP*.

[Bhattacharyya, 1943] Bhattacharyya, A. (1943). On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math. Soc.*, 35 :99–109.

[Bilmes, 1997] Bilmes, J. (1997). A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. ICSI-TR-97-021, University of Berkeley.

[Blouet et al., 2004] Blouet, R., Mokbel, C., et Chollet, G. (2004). BECARS : un logiciel libre pour la vérification du locuteur. Dans *Proc. JEP*.

[Boë et al., 1999] Boë, L.-J., Bimbot, F., Bonastre, J.-F., et Dupont, P. (1999). De l'évaluation des systèmes de vérification du locuteur à la mise en cause des expertises vocales en identification juridique. *Langues*, 2(4) :270–288.

[Boë et al., 2001] Boë, L.-J., Bonastre, J.-F., et Bimbot, F. (2001). Pourquoi la justice doit arrêter les expertises vocales. *Justice*, 169 :5–11.

[Bonastre et al., 2003] Bonastre, J.-F., Bimbot, F., Boë, L.-J., Campbell, J., Reynolds, D., et Magrin-Chagnolleau, I. (2003). Person authentification by voice : A need for caution. Dans *Proc. Eurospeech*.

[Bonastre et al., 2004] Bonastre, J.-F., Scheffer, N., Fredouille, C., et Matrouf, D. (2004). NIST'04 speaker recognition evaluation campaign : New LIA speaker detection plateform based on ALIZE toolkit. Dans *NIST SRE'04 Workshop : Speaker Detection Evaluation Campaign*.

[Bonastre et al., 2005] Bonastre, J.-F., Wils, F., et Meignier, S. (2005). Alize, a free toolkit for speaker recognition. Dans *Proc. ICASSP*.

[Bouchard, 2005] Bouchard, G. (2005). *Les modèles génératifs en classification supervisée et applications à la catégorisation d'images et à la fiabilité industrielle*. Thèse de doctorat, Université Joseph Fourier (Grenoble I). sous-section 2.3.1,.

[Boughorbel et al., 2004] Boughorbel, S., Tarel, J.-P., et Fleuret, F. (2004). Non-mercer kernels for SVM object recognition. Dans *Proc. British Machine Vision Conference (BMVC)*.

[Bousquet et Hermann, 2003] Bousquet, O. et Hermann, D. (2003). On the complexity of learning the kernel matrix. Dans *Advances in Neural Information Processing System (NIPS'03)*, volume 15, pages 415–422.

[Bredin et al., 2006] Bredin, H., Dehak, N., et Chollet, G. (2006). GMM-based SVM for face recognition. Dans *Proc. ICPR*.

[Brugnara et De Mori, 1998] Brugnara, F. et De Mori, R. (1998). *Spoken Dialogue with Computers*, chapter Training of Acoustic Models, pages 171–196. Academic Press.

[Brümmer, 2005a] Brümmer, N. (2005a). Focal, tools for fusion and calibration of automatic speaker detection systems. `http ://www.dsp.sun.ac.za/~nbrummer/focal/index.htm`.

[Brümmer, 2005b] Brümmer, N. (2005b). Spescom datavoice and university of stellenbosch NIST2005 SRE system description. Description Technique de système, workshop NIST SRE 2005.

[Brümmer et du Preez, 2005] Brümmer, N. et du Preez, J. (2005). Application-independent evaluation of speaker detection. *Computer Speech and Language*.

[Burges, 1998] Burges, C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2) :955–974.

[Camastra, 2004] Camastra, F. (2004). *Kernel Methods for Unsupervised Learning*. Thèse de doctorat, Università degli Studi di Genova.

[Campbell, 1997] Campbell, J. (1997). Speaker recognition : A tutorial. *Proc. of the IEEE*, 85(9).

[Campbell et al., 2003a] Campbell, J., Reynolds, D., et Dunn, R. (2003a). Fusing high- and low-level features for speaker recognition. Dans *Proc. Eurospeech*.

[Campbell, 2001] Campbell, W. (2001). A sequence kernel and its application to speaker verification. Dans *Proc. NIPS*.

[Campbell, 2002] Campbell, W. (2002). Generalized linear discriminant sequence kernels for speaker recognition. Dans *Proc. ICASSP*.

[Campbell, 2003] Campbell, W. (2003). A SVM/HMM system for speaker recognition. Dans *Proc. ICASSP*.

[Campbell, 2004] Campbell, W. (2004). Language recognition with support vector machines. Dans *Proc. IEEE Odyssey*.

[Campbell et Assaleh, 1999] Campbell, W. et Assaleh, K. (1999). Polynomial classifier techniques for speaker verification. Dans *Proc. ICASSP*.

[Campbell et al., 2003b] Campbell, W., Campbell, J., Reynolds, D., Jones, D., et Leek, T. (2003b). Phonetic speaker recognition with support vector machines. Dans *Proc. NIPS*.

[Campbell et al., 2006a] Campbell, W., Campbell, J., Reynolds, D., Singer, E., et Torres-Carrasquillo, P. (2006a). Support vector machines for speaker and language recognition. *Computer Speech and Language*, 20 :210–229.

[Campbell et al., 2004] Campbell, W., Reynolds, D., et Campbell, J. (2004). Fusing discriminative and generative methods for speaker recognition : Experiments on switchboard and nfi/tno field data. Dans *Proc. IEEE Odyssey*.

[Campbell et al., 2003c] Campbell, W., Reynolds, D., et Dunn, R. (2003c). Fusing high- and low-level features for speaker recognition. Dans *Proc. Eurospeech*.

[Campbell et al., 2006b] Campbell, W., Sturim, D., Reynolds, D., et Solomonoff, A. (2006b). SVM based speaker verification using a GMM supervector kernel and NAP variability compensation. Dans *Proc. ICASSP*.

[Carey et al., 1996] Carey, M., Parris, E., Lloyd-Thomas, H., et Bennett, S. (1996). Robust prosodic features for speaker identification. Dans *Proc. ICSLP*.

[Chan et al., 2004] Chan, A., Vasconcelos, N., et Moreno, P. (2004). A family of probabilistic kernels based on information divergence. Technical report, SVCL.

[Chapelle et al., 1999] Chapelle, O., Haffner, H., et Vapnik, V. (1999). SVMs for histogram-based image classification. *IEEE Trans. on Neural Networks*, 10 :1055–1064.

[Chen, 2003] Chen, K. (2003). Towards better making a decision in speaker verification. *Pattern Recognition*, 36 :329–346.

[Chen et Gopalakrishnan, 1998] Chen, S. et Gopalakrishnan, P. (1998). Speaker, environment and channel change detection and cluestering via the bayesian information criterion. Dans *Proc. DARPA workshop*.

[Chernoff, 1952] Chernoff, H. (1952). A measure of asymptotic efficiency of tests for a hypothesis based on a sum of observations. *Annals of Mathematical Statistics*, 23 :493–507.

[Chetty et Wagner, 2005] Chetty, G. et Wagner, M. (2005). Audio-visual multimodal fusion for biometric person authentification and liveness verification. Dans *Proc. NICTA-HCSNet MultiModal User Interaction Workshop (MMUI)*.

[Cheung et al., 2005] Cheung, M.-C., Mak, M.-W., et Kung, S.-Y. (2005). *Intelligent Multimedia Processing with Soft Computing*, chapter Probabilistic Fusion of Sorted Score Sequences for Robust Speaker Verification, pages 369–381. Springer.

[Choi et al., 2006] Choi, Y.-S., Shin, H.-C., et Song, W.-J. (2006). Affine projection algorithms with adaptative regularization matrix. Dans *Proc. ICASSP*.

[Chollet et al., 1996] Chollet, G., Cochard, J., Constantinescu, A., Jaboulet, C., et Langlais, P. (1996). Swiss french polyphone and polyvar : Telephone speech databases to model inter- and intra-speaker variability. Technical report idiap research report 96-01, IDIAP.

[Cohen et Zigel, 2002] Cohen, A. et Zigel, Y. (2002). On feature selection for speaker verification. Dans *Proc. COST 275 Workshop on the Advent of Biometrics on the Internet*.

[Collobert et al., 2002] Collobert, R., Bengio, S., et Bengio, Y. (2002). A parallel mixture of SVMs for very large scale problems. *Neural Computation*, 14(5) :1105–1114.

[Conrad et Paliwal, 2001] Conrad, S. et Paliwal, K. (2001). Information fusion for robust speaker verification. Dans *Proc. Eurospeech*.

[Cooper et Herskovits, 1992] Cooper, G. et Herskovits, E. (1992). A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9 :309–347.

[Cortes et al., 2003] Cortes, C., Haffner, P., et Mohri, M. (2003). Weighted automata kernels – general framework and algorithms. Dans *Proc. Eurospeech*.

[Cortes et al., 2004] Cortes, C., Haffner, P., et Mohri, M. (2004). Rational kernels : Theory and algorithms. *Journal of Machine Learning Research*, 5 :1035–1062.

[Cortes et Vapnik, 1995] Cortes, C. et Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3) :273–297.

[Cover et Thomas, 1991] Cover, T. et Thomas, J. (1991). *Elements of Information Theory*. Wiley.

[Cox et O'Sullivan, 1990] Cox, D. et O'Sullivan, F. (1990). Asymptotic analysis of penalized likelihood and related estimators. *Annals of Statistics*, 18 :1676–1695.

[Crammer et al., 2003] Crammer, K., Keshet, J., et Singer, Y. (2003). *Advances in Neural Information Processing Systems*, volume 15, chapter Kernel design using boosting. MIT Press.

[Cristianini et al., 2002] Cristianini, N., Shawe-Taylor, J., Elisseeff, A., et Kandola, J. (2002). On kernel-target alignment. *Advances in Neural Information Processing Systems*, 14 :367–373.

[Daniels et Kass, 2001] Daniels, M. et Kass, R. (2001). Shrinkage estimators for covariance matrices. *Biometrics*, 57(4) :1173–1184.

[Daoudi et Louradour, 2007] Daoudi, K. et Louradour, J. (2007). A novel strategy for speaker verification based on SVM classification of pair of speech sequences. Dans *Proc. Int. Symposium on Signal Processing and its Applications (ISSPA'07)*.

[Dasgupta et al., 2002] Dasgupta, N., Carin, L., et Couchman, L. (2002). Relevance-vector-machine quantization and density-function estimation : Application to HMM-based multi-aspect target classification. Dans *Proc. IEEE Pattern Analysis and Machine Intelligence*.

[Davenport et al., 2006] Davenport, M., Baraniuk, R., et Scott, C. (2006). Controlling false alarms with support vector machines. Dans *Proc. ICASSP*.

[Davis et Mermelstein, 1980] Davis, S. et Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuous speech sentences. *IEEE Trans. on Acoustic, Speech, Signal Processing*, 28 :357–366.

[Dean et al., 2006] Dean, D., Wark, T., et Sridharan, S. (2006). An examination of audio-viual fused HMMs for speaker recognition. Dans *Proc. MMUA*.

[DeCoste et Schölkopf, 2002] DeCoste, D. et Schölkopf, B. (2002). Training invariant support vector machines. *Machine Learning*, 46.

[Dehak et Chollet, 2006] Dehak, N. et Chollet, G. (2006). Support vector GMMs for speaker verification. Dans *Proc. IEEE Odyssey*.

[Dempster et al., 1977] Dempster, A., Laird, N., et Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, series B*, 39(1) :1–38.

[Desobry et al., 2005a] Desobry, F., Davy, M., et Doncarli, C. (2005a). An on-line kernel change detection algorithm. *IEEE Trans. on Signal Processing*, 53(8) :2961–2974.

[Desobry et al., 2005b] Desobry, F., Davy, M., et Fitzgerald, W. (2005b). A class of kernels for sets of vectors. Dans *Proc. ESANN*.

[Devroye et al., 1996] Devroye, L., Györfi, L., et Lugosi, G. (1996). *A Probabilistic Theory of Pattern Recognition*. Springer.

[Do, 2003] Do, M. (2003). Fast approximation of kullback-leibler distance for dependence trees and hidden markov models. *Signal Processing Letters*, 10(4) :115–118.

[Doddington, 2001] Doddington, G. (2001). Speaker recognition based on idiolectal differences between speakers. Dans *Proc. Eurospeech*.

[Duda et al., 2000] Duda, R., Hart, P. E., et Stork, D. (2000). *Pattern Classification (2nd edition)*. John Wiley & Sons.

[El Hannani et Petrovska-Delacrétaz, 2005] El Hannani, A. et Petrovska-Delacrétaz, D. (2005). *Nonlinear Speech Modeling and Applications*, chapter Segmental Scores Fusion for ALISP-Based GMM Text-Independent Speaker Verification, pages 351–356. Springer Berlin.

[Ezzaidi et al., 2001] Ezzaidi, H., Rouat, J., et O'Shaughnessy, D. (2001). Towards combining pitch and MFCC for speaker identification systems. Dans *Proc. Eurospeech*.

[Fahlman et Lebiere, 1990] Fahlman, S. et Lebiere, C. (1990). The cascade-correlation learning architecture. Dans *Advances in Neural Information Processing Systems 2*.

[Faltlhauser et Ruske, 2001] Faltlhauser, R. et Ruske, G. (2001). Improving speaker recognition performance using phonetically structured gaussian mixture models. Dans *Proc. Eurospeech*.

[Fiérrez-Aguilar et al., 2006] Fiérrez-Aguilar, J., Garcia-Romero, D., Ortega-Garcia, J., et González-Rodríguez, J. (2006). Speaker verification using adapted user-dependent multilevel fusion. *Computer Speech and Language*, 20(2–3) :192–209.

[Fine et al., 2001] Fine, S., Navrátil, J., et Gopinath, R. (2001). Enhancing GMM scores using SVM «hints». Dans *Proc. Eurospeech*.

[Fine et Scheinberg, 2001] Fine, S. et Scheinberg, K. (2001). Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research*, 2 :243–264.

[Fisher, 1936] Fisher, R. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7 :179–188. Academic Press.

[Franc et Hlavac, 2003] Franc, V. et Hlavac, V. (2003). Greedy algorithm for a training set reduction in the kernel methods. Dans *Proc. Int. Conf. Computer Analysis of Images and Patterns*.

[Freund et Schapire, 1999] Freund, Y. et Schapire, R. (1999). A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5) :771–780.

[Fritz et al., 2005] Fritz, M., Leibe, B., Caputo, B., et Schiele, B. (2005). Integrating representative and discriminative models for object category detection. Dans *Proc. ICCV*.

[Fuentes et al., 2002] Fuentes, M., Mostefa, D., Kharroubi, J., Garcia-Salicetti, S., Dorizzi, B., et Chollet, G. (2002). Vérification de l'identité par fusion de données biométriques : Signatures en-ligne et parole. Dans *Proc. Coloque International Francophone sur l'Ecrit et le Document (CIFED)*.

[Ganapathiraju et Picone, 2000] Ganapathiraju, A. et Picone, J. (2000). Hybrid SVM/HMM architectures for speech recognition. *Neural Information Processing Systems*.

[Ganchev et al., 2003] Ganchev, T., Tasoulis, D., Vrahatis, M., et Fakotakis, N. (2003). Locally recurrent probabilistic neural network for text-independent speaker verification. Dans *Proc. Eurospeech*.

[Garcia-Romero et al., 2003] Garcia-Romero, D., Fiérrez-Aguilar, J., González-Rodríguez, J., et Ortega-Garcia, J. (2003). Support vector machine fusion for idiolectal and acoustic speaker information in spanish conversational speech. Dans *Proc. ICASSP*.

[Garcia-Romero et al., 2004] Garcia-Romero, D., Fierrez-Aguilar, J., Gonzalez-Rodriguez, J., et Ortega-Garcia, J. (2004). On the Use of Quality Measures for Text-Independent Speaker Recognition. Dans *Proc. IEEE Odyssey*.

[Gauvain et Lee, 1994] Gauvain, J.-L. et Lee, C.-H. (1994). Maximum *a Posteriori* estimation for multivariate gaussian mixture observations of markov chains. *IEEE Trans. on Speech and Audio Processing*, 2(2) :291–298.

[Goddard et al., 2003] Goddard, J., Martinez, A., Martinez, F., et Rufiner, H. (2003). A comparison of string kernels and discrete hidden markov models on a spanish digit recognition task. Dans *proc. Annual Int. Conference of the IEEE Engineering in Medicine and Biology Society.*

[Goddard et al., 2004] Goddard, J., Martinez, A., Martinez, F., et Rufiner, H. (2004). Noisy speech recognition using string kernels. Dans *proc. Conf. on SPEech and COMputer (SPECOM).*

[Golub et Van Loan, 1996] Golub, G. et Van Loan, C. (1996). *Matrix Computation.* The John Hopkins Univ. Press.

[Gretton et al., 2005] Gretton, A., Herbrich, R., Smola, A., Bousquet, O., et Schölkopf, B. (2005). Kernel methods for measuring independence. *Journal of Machine Learning Research*, 6 :2075–2129.

[Gunn, 1998] Gunn, S. (1998). Support vector machines for classification and regression. Technical report, University of Southampton.

[Gärtner, 2003] Gärtner, T. (2003). A survey of kernels for structured data. *ACM SIGKDD Explorations Newsletter*, 5(1).

[Haasdonk, 2005] Haasdonk, B. (2005). Feature space interpretation of SVMs with non positive definite kernels. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(4) :482–492.

[Hastie et Tibshirani, 1990] Hastie, T. et Tibshirani, R. (1990). *Monographs on Statistics and Applied Probability*, volume 48, chapter Generalized Additive Models. Chapman & Hall, London.

[Hatch et al., 2006] Hatch, A., Kajarekar, S., et Stolcke, A. (2006). Within-class covariance normalization for SVM speaker recognition. Dans *Proc. ICSLP.*

[Hatch et Stolcke, 2006] Hatch, A. et Stolcke, A. (2006). Generalized linear kernels for one-versus-all classification : Application to speaker recognition. Dans *Proc. ICASSP.*

[Hebb, 1949] Hebb, D. (1949). *The Organization if Behaviour.* John Wiley.

[Heck et al., 2000] Heck, L., Konig, Y., Sönmez, M., et Weintraub, M. (2000). Robustness to telephone handset distortion in speaker recognition by discriminative feature design. *Speech Communication*, 31 :181–192.

[Heck et Weintraub, 1997] Heck, L. et Weintraub (1997). Handset-dependent background models for robust text-independent speaker recognition. Dans *Proc. ICASSP.*

[Heckerman, 1995] Heckerman, D. (1995). A tutorial on learning with bayesian networks. Msr-tr-95-06, Microsoft Research.

[Hein et Bousquet, 2004] Hein, M. et Bousquet, O. (2004). Hilbertian metrics and positive kernels on probability measures. *AISTATS.*

[Hermansky, 1990] Hermansky, H. (1990). Perceptual linear predictive (plp) analysis of speech. *Journal of Acoustics Society America*, 87(4) :1738–1752.

[Hermansky et Morgan, 1994] Hermansky, H. et Morgan, N. (1994). RASTA processing of speech. *IEEE Trans. on Speech and Audio Processing*, 2(4) :587–589.

[Ho et Moreno, 2004] Ho, P. et Moreno, P. (2004). SVM kernel adaptation in speaker classification and verification. Dans *Proc. ICSLP.*

[Hou et Wang, 2003] Hou, F. et Wang, B. (2003). Text-independent speaker recognition using probabilistic SVM with GMM adjustment. Dans *Proc. ICASSP*.

[Hsu et Lin, 2002] Hsu, C.-W. et Lin, C.-J. (2002). A comparison of methods for multi-class support vector machines. *IEEE Trans. Neural Networks*, 13(5) :415–425.

[Imbiriba et al., 2004] Imbiriba, T., Klautau, A., Parihar, N., Raghavan, S., et Picone, J. (2004). GMM and kernel-based speaker recognition with the ISIP toolkit. Dans *Proc. Int. Workshop on Machine Learning for Signal Processing (IWMLSP)*.

[Jaakkola et Haussler, 1998] Jaakkola, T. et Haussler, D. (1998). Exploiting generative models in discriminative classifiers. *Advances in Neural Information Processing Systems*, 11.

[Jebara et Kondor, 2003] Jebara, T. et Kondor, R. (2003). Bhattacharyya and expected likelihood kernels. Dans *Proc. Annual Conference on Computational Learning Theory and Kernel Workshop*.

[Jebara et al., 2004] Jebara, T., Kondor, R., et Howard, A. (2004). Probability product kernels. *Journal of Machine Learning Research*, 5.

[Juneja et Espy-Wilson, 2002] Juneja, A. et Espy-Wilson, C. (2002). Segmentation of continuous speech using acoustic-phonetic parameters and statistical learning. Dans *Proc. Int. Conf. on Neural Information Processing*, volume 2.

[Justino et al., 2004] Justino, E., Bortolozzi, F., et Sabourin, R. (2004). A comparison of SVM and HMM classifiers in the off-line signature verification. *Pattern Recognition Letters*.

[Kaas et Raftery, 1995] Kaas, R. et Raftery, A. (1995). Bayes factors and model uncertainty. *Journal of the American Statistical Association*, 90 :773–795.

[Kandola et al., 2002] Kandola, J., Shawe-Taylor, J., et Cristianini, N. (2002). Optimizing kernel alignement over combinations of kernels. Technical report 121, University of London, Department of Computer Science.

[Kanokphara et al., 2006] Kanokphara, S., Macek, J., et Carson-Berndsen (2006). Comparative study : SVM & HMM for automatic articulatory feature extraction. Dans *Proc. Int. Conf. Industrial, Engeneering & Other Applications for Applied Intelligent Systems*.

[Kartik et al., 2005] Kartik, V., Satish, D., et Sekhar, C. (2005). Speaker change detection using support vector machines. Dans *Proc. NOLISP*.

[Kashima et al., 2004] Kashima, H., Tsuda, K., et Inokuchi, A. (2004). *Kernel Methods in Computational Biology*, volume 7, chapter Kernels for Graphs, pages 155–170. MIT Press.

[Kersting et Gärtner, 2004] Kersting, K. et Gärtner, T. (2004). Fisher kernels for logical sequences. Dans *Proc. ECML*.

[Kharroubi et al., 2001] Kharroubi, J., Petrovska-Delacretaz, D., et Chollet, G. (2001). Combining GMM's with suport vector machines for text-independent speaker verification. Dans *Proc. Eurospeech*.

[Kim et Lee, 1999] Kim, H. et Lee, H. (1999). Use of spectral autocorrelation in spectral envelope linear prediction for speech recognition. *IEEE Trans. on Speech and Audio Processing*, 7(5) :533–541.

[Kimeldorf et Wahba, 1971] Kimeldorf, G. et Wahba, G. (1971). Some results on tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33 :82–95.

[Klusácek et al., 2003] Klusácek, D., Navrátil, J., Reynolds, D., et Campbell, J. (2003). Conditional pronunciation modeling in speaker detection. Dans *Proc. ICASSP*.

[Kondor, 2005] Kondor, R. (2005). Computing the bhattacharyya kernel. note écrite à la suite de l'article "A Kernel between Sets of Vectors" du même auteur, `http ://www1.cs.columbia.edu/~risi/papers/computeBhatta.ps`.

[Kondor et Jebara, 2003] Kondor, R. et Jebara, T. (2003). A kernel between sets of vectors. Dans *Proc. ICML*.

[Kunn et al., 2000] Kunn, R., Junqua, J.-C., Nguyen, P., et Niedzielski, N. (2000). Rapid speaker adaptation in eigenvoice space. *IEEE Trans. on Speech and Audio Processing*, 8 :695–707.

[Lafferty et Lebanon, 2004] Lafferty, J. et Lebanon, G. (2004). Diffusion kernels on statistical manifolds. Technical report CMU-CS-04-101, School of Computer Science, Carnegie Mellon University, Pittsburg.

[Lanckriet et al., 2002] Lanckriet, G., Christianini, N., Bartlett, P., El Ghaoui, L., et I. Jordan, M. (2002). Learning the kernel matrix with semi-definite programming. Dans *Proc. ICML*.

[Lau et al., 2004] Lau, C., Ma, B., Meng, H., Moon, Y., et Yam, Y. (2004). Fuzzy logic decision fusion in a multimodal biometric system. Dans *Proc. ICSLP*.

[Lauritzen, 1995] Lauritzen, S. (1995). The EM algorithm for graphical models with missing data. *Computational Statistics and Data Analysis*, 19 :191–201.

[Layton et Gales, 2004] Layton, M. et Gales, M. (2004). Maximum margin training of generative kernels. Technical report CUED/F-INFENG/TR.484, Cambridge University Engineering Dept.

[Le et Bengio, 2003] Le, Q. et Bengio, S. (2003). Client dependent GMM-SVM models for speaker verification. Dans *Proc. International Conference on Artificial Neural Networks (ICANN)*.

[LeCun et al., 1990] LeCun, Y., Denker, J., Solla, S., Howard, R., et Jackel, L. (1990). Optimal brain damage. Dans *Advances in Neural Information Processing Systems 2*.

[Lei et al., 2005] Lei, Z., Yang, Y., et Wu, Z. (2005). Mixture of support vector machines for text-independent speaker recognition. Dans *Proc. Interspeech*.

[Leslie et al., 2002a] Leslie, C., Eskin, E., et Noble, W. (2002a). The spectrum kernel : A string kernel for SVM protein classification. Dans *proc. Pacific Symposium on Biocomputing (PSB)*.

[Leslie et al., 2002b] Leslie, C., Eskin, E., Weston, J., et Noble, W. (2002b). Mismatch string kernels for SVM protein classification. Dans *proc. NIPS*.

[Li et al., 2002] Li, Q., Zheng, J., Tsai, A., et Zhou, Q. (2002). Robust endpoint detection and energy normalization for real-time speech and speaker recognition. *IEEE Trans. on Speech and Audio Processing*, 10(3).

[Li et al., 2003] Li, S., Zhang, D., Ma, C., Shum, H.-Y., et Chang, E. (2003). Learning to boost GMM based speaker verification. Dans *Proc. Eurospeech*.

[Li et Bilmes, 2006] Li, X. et Bilmes, J. (2006). Regularized adaptation of discriminative classifiers. Dans *Proc. ICASSP*.

[Lin, 1991] Lin, J. (1991). Divergence measures based on shannon entropy. *IEEE Trans. Information Theory*, 37(14) :145–151.

[Lin et al., 2002] Lin, Y., Lee, Y., et Wahba, G. (2002). Support vector machines for classification in non-standard situations. *Machine Learning*, 46 :191–202.

[Linde et al., 1980] Linde, Y., Buzo, A., et Gray, R. (1980). An algorithm for vector quantization design. *IEEE Trans. on Communications*, 28(1) :84–95.

[Lissack et Fu, 1976] Lissack, T. et Fu, K. (1976). Error estimation in pattern recognition via l-distance between posterior density functions. *IEEE Trans. Information Theory*, 22 :34–45.

[Liu et al., 2002] Liu, M., Chang, E., et Dai, B.-Q. (2002). Hierarchical gaussian mixture models for speaker verification. Dans *Proc. ICSLP*.

[Liu et al., 2006] Liu, M., Dai, B., Xie, Y., et Yao, Z. (2006). Improved GMM-UBM/SVM for speaker verification. Dans *Proc. ICASSP*.

[Lodhi et al., 2002] Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., et Watkins, C. (2002). Text classification using string kernels. *Journal of Machine Learning Research*, 2 :419–444.

[Louradour et al., 2004] Louradour, J., Andre-Obrecht, R., et Daoudi, K. (2004). Segmentation and relevance measure for speaker verification. Dans *Proc. ICSLP*.

[Louradour et Daoudi, 2005a] Louradour, J. et Daoudi, K. (2005a). Conceiving a new sequence kernel and applying it to SVM speaker verification . Dans *Proc. 9th European Conf. on Speech Communication and Technology (INTERSPEECH)*.

[Louradour et Daoudi, 2005b] Louradour, J. et Daoudi, K. (2005b). SVM speaker verification using a new sequence kernel. Dans *Proc. 13th European Conf. on Signal Processing (EUSIPCO)*.

[Louradour et Daoudi, 2005c] Louradour, J. et Daoudi, K. (2005c). Nouveau noyau de séquence pour la vérification du locuteur. Dans *Proc. 20e colloque GRETSI sur le traitement du signal et des images (GRETSI)*.

[Louradour et Daoudi, 2006] Louradour, J. et Daoudi, K. (2006). SVM speaker verification using an incomplete cholesky decomposition sequence kernel. Dans *Proc. IEEE Odyssey*.

[Louradour et Daoudi, 2007] Louradour, J. et Daoudi, K. (2007). Pair-of-sequences SVM speaker verification. Dans *soumis à EUSIPCO 2007*.

[Louradour et al., 2005] Louradour, J., Daoudi, K., et Andre-Obrecht, R. (2005). Discriminative power of transient frames in speaker recognition. Dans *Proc. ICASSP*.

[Louradour et al., 2006a] Louradour, J., Daoudi, K., et Bach, F. (2006a). Feature space mahalanobis sequence kernels : Application to SVM speaker verification. *soumis à IEEE Trans. on Audio, Speech and Language Processing*.

[Louradour et al., 2006b] Louradour, J., Daoudi, K., et Bach, F. (2006b). SVM speaker verification using an incomplete cholesky decomposition sequence kernel. Dans *Proc. IEEE Odyssey : The Speaker and Language Recognition Workshop*.

[Lyngs et al., 1999] Lyngs, R. B., Pedersen, C. N. S., et Nielsen, H. (1999). Metrics and similarity measures for hidden markov models. Dans *Proc. ISMB*.

[Lyu, 2005] Lyu, S. (2005). A kernel between unordered sets of data : the gaussian mixture approach. Dans *Proc. ECML*.

[Magrin-Chagnolleau et al., 2001] Magrin-Chagnolleau, I., Gravier, G., et Blouet, R. (2001). Overview of the 2000-2001 ELISA consortium research activities. Dans *Proc. IEEE Odyssey*.

[Mahalanobis, 1936] Mahalanobis, P. (1936). On the generalized distance in statistics. *Proc. National Inst. Sci.*, 12 :49–55.

[Mak et al., 2003] Mak, M., Cheung, M., et Kung, S. (2003). Robust speaker verification from GSM-coded speech based on decision fusion and feature transformation. Dans *Proc. IEEE Int. Conf. on Acoustic, Speech and Signal Processing*.

[Mallat et Zhang, 1993] Mallat, C. et Zhang, Z. (1993). Matching pursuit in a time-frequency dictionary. *Trans. on Signal Processing*, 41 :3397–3415.

[Mariéthoz, 2006] Mariéthoz, J. (2006). *Discriminant Models for Text-independent Speaker Verification*. Thèse de doctorat, IDIAP.

[Mariéthoz et Bengio, 2002] Mariéthoz, J. et Bengio, S. (2002). A comparative study of adaptation methods for speaker verification. Dans *Proc. ICSLP*.

[Mariéthoz et Bengio, 2005] Mariéthoz, J. et Bengio, S. (2005). A kernel trick for sequences applied to text-independent speaker verification systems. *IDIAP Research Report*.

[Mariéthoz et Bengio, 2006] Mariéthoz, J. et Bengio, S. (2006). A max kernel for text-independent speaker verification systems. Dans *Proc. MMUA*.

[Martin et al., 1997] Martin, A., Doddington, G., Kamm, T., Ordowski, M., et Przybocki, M. (1997). The DET curve in assessment of detection task performance. Dans *Proc. Eurospeech*.

[Mashao, 2005] Mashao, D. (2005). Comparing SVM and GMM classifiers on the parametric feature-sets. *print South Africa Institute of Electrical Engineers (SAIEE) Trans.*

[Matusita, 1955] Matusita, K. (1955). Decision rules based on the distance for problems of fit, two samples and estimation. *Annals of Mathematical Statistics*, 26 :631–640.

[McCulloch et Pitts, 1943] McCulloch, W. et Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5 :115–133.

[Meila, 2003] Meila, M. (2003). Data centering in feature space. Dans *proc. Int. Workshop on Artificial Intelligence and Statistics*.

[Melin et al., 1998] Melin, H., Koolwaaij, J., Lindberg, J., et Bimbot, F. (1998). A comparative evaluation of variance flooring techniques in HMM-based speaker verification. Dans *Proc. ICSLP*.

[Mengusoglu, 2003] Mengusoglu, E. (2003). Confidence measure based model adaptation for speaker verification. Dans *Proc. Communication, Internet and Information Technology*.

[Micchelli, 1986a] Micchelli, C. (1986a). Interpolation of scattered data : Distance matrices and conditionnally positive definite functions. *Constructive Approximation*, 2 :11–22.

[Micchelli, 1986b] Micchelli, C. (1986b). Interpolation of scattered data : Distance metrics and conditionally positive definite functions. *Constructive Approximation*, 2 :11–22.

[Mika, 1998] Mika, S. (1998). Nichtlineare signalverarbeitung in feature-rä umen. Technical report, Technische Universitä t Berlin.

[Mika et al., 1999] Mika, S., Schölkopf, B., Smola, A., Müller, K., M., S., et G., R. (1999). *Advances in Neural Information Processing Systems*, volume 11, chapter Kernel PCA and de-noising in feature spaces, pages 536–542. MIT Press.

[Moraru et al., 2003] Moraru, D., Meignier, S., Besacier, L., Bonastre, J.-F., et Magrin-Chagnolleau, I. (2003). The ELISA consortium approaches in speaker segmentation during the NIST 2002 speaker recognition evaluation. Dans *Proc. ICASSP*.

[Moreno et Ho, 2003a] Moreno, P. et Ho, P. (2003a). A generative model based kernel for SVM classification in multimedia applications. Dans *Proc. NIPS*.

[Moreno et Ho, 2003b] Moreno, P. et Ho, P. (2003b). A new SVM approach to speaker identification and verification using probabilistic distance kernels. Dans *Proc. Eurospeech*.

[Naïm et al., 2004] Naïm, P., Wuillemin, P., Leray, P., Pourret, O., et Beckler, A. (2004). *Les Réseaux Bayésiens*. Eyrolles.

[Natarajan, 1995] Natarajan, B. (1995). Sparse approximate solutions to linear systems. *SIAM Journal of Computing*, 24(2) :227–234.

[Ng et Jordan, 2001] Ng, A. et Jordan, M. (2001). On discriminative vs. generative classifiers : a comparison of logistic regression and naive bayes. Dans *Proc. NIPS*.

[Nickel et al., 2004] Nickel, R., Oswal, S., et Iyer, A. (2004). Robust speaker verification with principal pitch components. Dans *Proc. EUSIPCO*.

[Noda et al., 1998] Noda, H., Harada, K., Kawaguchi, E., et Sawai, H. (1998). A context-dependent approach for speaker verification using sequential decision. Dans *Proc. ICSLP*.

[Ogawa, 1988] Ogawa, H. (1988). An operator pseudo-inverson lemma. *SIAM Journal of Applied Mathematics*, 48(6) :1527–1531.

[Ong et al., 2004] Ong, C., Mary, X., Canu, S., et Smola, A. (2004). Learning with non-positive kernels. Dans *Proc. Int. Conf. on Machine Learning*.

[Patrick et Fisher, 1969] Patrick, E. et Fisher, F. (1969). Non-parametric feature selection. *IEEE Trans. Information Theory*, 15 :577–584.

[Pedersen et Lyngsø, 2001] Pedersen, C. et Lyngsø, R. (2001). Complexity of comparing hidden markov models. Dans *Proc. ISAAC Conference on "Analysis, Applications, and Computation"*.

[Pelecanos et Sridharan, 2001] Pelecanos, J. et Sridharan, S. (2001). Feature warping for robust speaker verification. Dans *Proc. IEEE Odyssey*.

[Peltonen et al., 2004] Peltonen, J., Klami, A., et Kaski, S. (2004). Improved learning of riemannian metrics for exploratory analysis. *Neural Networks*, 17 :1087–1100.

[Peskin et al., 2003] Peskin, B., Navrátil, J., Abramson, J., Jones, D., Klusácek, D., Reynolds, D., et Xiang, B. (2003). Using prosodic and conversational features for high-performance speaker recognition : Report from JHU WS'02. Dans *Proc. ICASSP*.

[Platt, 1999] Platt, J. (1999). *Fast training of Support Vector Machines using Sequential Minimal Optimization*. MIT Press.

[Platt, 2000] Platt, J. (2000). *Advanced in Large Margin Classifiers*, chapter Probabilities for SV Machines, pages 61–74. MIT Press.

[Pothin et Richard, 2005] Pothin, J.-B. et Richard, C. (2005). Kernel machines : une nouvelle méthode pour l'optimisation de l'alignement des noyaux et l'amélioration des performances. Dans *proc. GRETSI*.

[Qing et Chen, 2006] Qing, X.-K. et Chen, K. (2006). On use of GMM for multilingual speaker verification : An empirical study. Dans *Proc. ICSLP*.

[Rabiner, 1989] Rabiner, L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2) :257–286.

[Rabiner et Juang, 1993] Rabiner, L. et Juang, B. (1993). *Fundamentals of Speech Recognition*, volume 1. Prentice-Hall.

[Raina et al., 2003] Raina, R., Shen, Y., Y.Ng, A., et McCallum, A. (2003). Classification with hybrid generative/discriminative models. Dans *Proc. NIPS*.

[Rasmussen et Williams, 2006] Rasmussen, C. et Williams, C. (2006). *Gaussian Processes for Machine Learning*. MIT Press.

[Rényi, 1960] Rényi, A. (1960). On measures of entropy and information. Dans *Proc. Berkeley Symposium on Mathematical Statistics and Probability*.

[Reynolds, 1995] Reynolds, D. (1995). Speaker identification and verification using gaussian mixture speaker models. *Speech Communication*, 17(1–2) :91–108.

[Reynolds, 1997] Reynolds, D. (1997). Comparison of Background Normalization methods for Text-Independent Speaker Recognition. Dans *Proc. Eurospeech*.

[Reynolds, 2003] Reynolds, D. (2003). Channel robust speaker verification via feature mapping. Dans *Proc. ICASSP*.

[Reynolds et al., 2000] Reynolds, D., Quatieri, T., et Dunn, R. (2000). Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10 :19–41.

[Richiardi et al., 2006] Richiardi, J., Drygajlo, A., et Prodanov, P. (2006). Confidence and reliability measures for speaker verification. *Journal of the Franklin Institute*. à paraître.

[Rosenberg et al., 1992] Rosenberg, A. E., DeLong, J., Lee, C.-H., Juang, B.-H., et Soong, F. K. (1992). The Use of Cohort Normalized Scores for Speaker Verification. Dans *Proc. Int. Conf. on Spoken Language Processing (ICSLP)*.

[Rouas et al., 2006] Rouas, J.-L., Louradour, J., et Ambellouis, S. (2006). Audio events detection in public transport vehicle. Dans *Proc. Intelligent Transportation Systems Conf. (ITSC)*.

[Saenko et al., 2004] Saenko, K., Darrell, T., et Glass, J. (2004). Articulatory features for robust visual speech recognition. Dans *Proc. ICMI*.

[Sanchez-Soto, 2005] Sanchez-Soto, E. (2005). *Réseaux Bayésiens Dynamiques pour la Vérification du Locuteur*. Thèse de doctorat, Télécom Paris (ENST).

[Sarle, 1997] Sarle, W. (1997). Neural network FAQ. Periodic posting to the Usenet newsgroup, http ://comp.ai.neural-nets.

[Saunders, 2002] Saunders, C. (2002). String kernels, fisher kernels and finite state automata. Dans *proc. NIPS*.

[Schafföner et al., 2006] Schafföner, M., Krüger, S., Andelic, E., Katz, M., et Wendemuth, A. (2006). Limited training data robust speech recognition using kernel-based acoustic models. Dans *Proc. ICASSP*.

[Scheffer, 2005] Scheffer, N. (2005). LIA speaker detection package. `http ://www.lia.univ-avignon.fr/heberges/ALIZE/Doc/NIST_LIARAL_Tutorial.tgz`. tutorial at the BIOSECURE Workshop.

[Scheffer et Bonastre, 2006] Scheffer, N. et Bonastre, J.-F. (2006). Fusing generative and discriminative UBM-based systems for speaker verification. Dans *Proc. of the 2nd international workshop on MMUA (MultiModal User Authentification)*.

[Schmidt et Gish, 1996] Schmidt, M. et Gish, H. (1996). Speaker identification via support vector machines. Dans *Proc. ICASSP*.

[Schoenberg, 1938] Schoenberg, I. (1938). Metric spaces and positive definite functions. *Trans. Amer. Math. Soc.*, 44 :522–536.

[Schölkopf et al., 1999] Schölkopf, B., Mika, S., J.C.Burges, C., Knirsch, P., Müller, K.-R., Rätsch, G., et J. Smola, A. (1999). Input space versus feature space. *IEEE Trans. Neural Networks*, 100(5) :1000–1017.

[Schölkopf et al., 2001] Schölkopf, B., Platt, J., Shawe-Taylor, J., Smola, A., et Williamson, R. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation*, 13 :1443–1471.

[Schölkopf et Smola, 2002] Schölkopf, B. et Smola, A. (2002). *Learning with kernels*. MIT Press.

[Schölkopf et al., 1998] Schölkopf, B., Smola, A., et Müller, K. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5) :1299–1319.

[Schölkopf et al., 2000] Schölkopf, B., Smola, A., Williamson, R., et L. Bartlett, P. (2000). New support vector algorithms. *Neural Computation*, 12 :1207–1245.

[Seeger, 2002a] Seeger, M. (2002a). Covariance kernels from bayesian generative models. *Advances in Neural Information Processing Systems*, 14.

[Seeger, 2002b] Seeger, M. (2002b). Covariance kernels from bayesian generative models. Technical report, Institute for Adaptative and Neural Computation.

[Shakhnarovish et al., 2005] Shakhnarovish, Darrell, et Indyk, editors (2005). *Nearest-Neighbor Methods in Learning and Vision*. MIT Press.

[Shawe-Taylor et Cristianini, 2004] Shawe-Taylor, J. et Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press.

[Shawe-Taylor et al., 2005] Shawe-Taylor, J., Williams, C., Cristianini, N., et Kandola, J. (2005). On the eigenspectrum of the gram matrix and the generalisation error of kernel PCA. *IEEE Trans. on Information Theory*, 51(7).

[Shimodaira et al., 2001] Shimodaira, H., Noma, K., Nakai, M., et Sagayama, S. (2001). Support vector machine with dynamic time-alignment kernel for speech recognition. Dans *Proc. Interspeech*.

[Shriberg et al., 2005] Shriberg, E., Ferrer, L., Kajarekar, S., Venkataraman, A., et Stolcke, A. (2005). Modeling prosodic feature sequences for speaker recognition. *Speech Communication*.

[Siohan et al., 1999] Siohan, O., Chesta, C., et Lee, C.-H. (1999). Hidden Markov Model Adaptation using Maximum A Posteriori Linear Regression. Dans *Proc. Workshop for Robust Methods for Speech Recognition in Adverse Conditions*.

[Smith et Gales, 2002] Smith, N. et Gales, M. (2002). Using SVMs to classify variable length speech pattern. Technical report CUED/F-INFENG/TR.412, Cambridge University Engineering Dept.

[Smith et al., 2001] Smith, N., Gales, M., et Niranjan, M. (2001). Data-dependent kernels in SVM classification of speech patterns. Technical report, Cambridge University Engineering Dept.

[Smola et Schölkopf, 2000] Smola, A. et Schölkopf, B. (2000). Sparse greedy matrix approximation for machine learning. Dans *Proc. ICML*.

[Solewicz, 2005] Solewicz, Y. (2005). *Optimized Fusion Methods for Speaker Verification*. Thèse de doctorat, Bar-Ilan University (Israël).

[Solomonoff et al., 2004] Solomonoff, A., Campbell, W., et Boardman, I. (2004). Advances in channel compensation for SVM speaker recognition. Dans *Proc. ICASSP*.

[Solomonoff et al., 1998] Solomonoff, A., Mielke, A., Schmidt, M., et Gish, H. (1998). Clustering speakers by their voices. Dans *Proc. ICASSP*.

[Staroniewicz et Majewski, 2004] Staroniewicz, P. et Majewski, W. (2004). SVM based text-dependent speaker identification for large sets of voices. Dans *Proc. EUSIPCO*.

[Steinwart, 2003] Steinwart, I. (2003). Sparseness of support vector machines. *Journal of Machine Learning Research*, 4 :1071–1105.

[Steinwart, 2005] Steinwart, I. (2005). Consistency of support vector machines and other regularized kernel classifiers. *IEEE Trans. Information Theory*, 51 :128–142.

[Sturim et Reynolds, 2005] Sturim, D. et Reynolds, D. (2005). Speaker Adaptative Cohort Selection for TNorm in Text-Independent Speaker Verification. Dans *Proc. ICASSP*.

[Sturim et al., 2002] Sturim, D., Reynolds, D., Dunn, R., et Quatieri, T. (2002). Speaker verification using text-constrained gaussian mixture models. Dans *Proc. ICASSP*.

[Tan et Wang, 2004] Tan, Y. et Wang, J. (2004). A support vector machine with a hybrid kernel and minimal vapnik-chervonenkis dimension. *IEEE Trans. on Knowledge and Data Engineering*, 16(4) :385–395.

[ALIZE, 2005] ALIZE (2005). Alize page. http ://www.lia.univ-avignon.fr/heberges/ALIZE/.

[BioSecure, 2005] BioSecure (2005). Biosecure network of excellence : Biometrics for secure authentification. http ://www.biosecure.info.

[HTK, 2002] HTK (2002). Htk homepage and htkbook. http ://htk.eng.cam.ac.uk.

[LIA SpkDet, 2005] LIA SpkDet (2005). Lia ral page. http ://www.lia.univ-avignon.fr/heberges/ALIZE/LIA_RAL/index.html.

[NIST SRE, 1997] NIST SRE (1997). The 1997 speaker recognition plan. http ://www.nist.gov/speech/tests/spk/1997/sp_v1p1.htm.

[NIST SRE, 2003] NIST SRE (2003). The NIST year 2003 speaker recognition evaluation plan. http ://www.nist.gov/speech/tests/spk/2003/doc/2003-spkrec-evalplan-v2.2.pdf.

[NIST SRE, 2004] NIST SRE (2004). The NIST year 2004 speaker recognition evaluation plan. http ://www.nist.gov/speech/tests/spk/2004/SRE-04_evalplan-v1a.pdf.

[NIST SRE, 2005] NIST SRE (2005). The NIST year 2005 speaker recognition evaluation plan. http ://www.nist.gov/speech/tests/spk/2005/sre-05_evalplan-v5.pdf.

[NIST SRE, 2006] NIST SRE (2006). The NIST year 2006 speaker recognition evaluation plan. http ://www.nist.gov/speech/tests/spk/2006/sre-06_evalplan-v9.pdf.

[SPRO, 2004] SPRO (2004). Spro release 4.0. http ://www.irisa.fr/metiss/guig/spro/. by Gravier, G.

[Tipping, 2000] Tipping, M. (2000). The relevance vector machine. *Advances in Neural Information Processing Systems*.

[Tipping, 2001] Tipping, M. (2001). Sparse kernel principal component analysis. *Advances in Neural Information Processing Systems*, 13.

[Titov et Henderson, 2005] Titov, I. et Henderson, J. (2005). Deriving kernels from MLP probability estimators for large categorization problems. Dans *Proc. Int. Joint Conf. on Neural Networks (IJCNN)*.

[Tomasi, 2005] Tomasi, C. (2005). Estimating gaussian mixture densities with EM - a tutorial. Technical report, Duke University.

[Tsang et al., 2005] Tsang, I., Kwok, J., et Cheung, P.-M. (2005). Core vector machines : Fast SVM training on very large data sets. *Journal of Machine Learning*, 6 :363–392.

[Tsuda, 1999] Tsuda, K. (1999). Support vector classifier with asymmetric kernel function. Dans *proc. ESANN*.

[Tsuda et al., 2004] Tsuda, K., Akaho, S., Kawanabe, M., et Müller, K.-R. (2004). Asymptotic properties of the fisher kernel. *Neural Computation*, 16(1) :115–137.

[Tsuda et al., 2002] Tsuda, K., Kawanabe, M., Rätsch, G., Sonnenburg, S., et Müller, K.-R. (2002). A new discriminative kernel from probabilistic models. *Neural Computation*, 14(10) :2397–2414.

[Van Trees, 1968] Van Trees, H. (1968). *Detection, Estimation and Modulation Theory*, volume 1. Wiley, New York.

[Vapnik, 1998] Vapnik, V. (1998). *Statistical Learning Theory*. John Wiley & Sons.

[Vasconcelos et al., 2004] Vasconcelos, N., Ho, P., et Moreno, P. (2004). The kullback-leibler kernel as a framework for discriminant and localized representations for visual recognition. Dans *Proc. European Conf. on Computer Vision*.

[Viterbi, 1967] Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE trans. on Information Theory*, 13(2) :260–269.

[Wallraven et al., 2003] Wallraven, C., Caputo, B., et Graf, A. (2003). Recognition with local features : the kernel recipe. Dans *Proc. ICCV*.

[Wan, 2003] Wan, V. (2003). *Speaker Verification using Support Vector Machines*. Thèse de doctorat, University of Sheffield.

[Wan et Carmichael, 2005] Wan, V. et Carmichael, J. (2005). Polynomial dynamic time warping kernel support vector machines for dysarthric speech recognition with sparse training data. Dans *Proc. Interspeech*.

[Wan et Renals, 2002] Wan, V. et Renals, S. (2002). Evaluation of kernel methods for speaker verification and identification. Dans *Proc. ICASSP*.

[Wan et Renals, 2004] Wan, V. et Renals, S. (2004). Speaker verification using sequence discriminant support vector machines. *IEEE Trans. on Speech and Audio Processing*.

[Wang et Luk Chan, 2002] Wang, L. et Luk Chan, K. (2002). Learning kernel parameters by using class separability measure. Dans *proc. NIPS*.

[Williams et Seeger, 2000] Williams, C. et Seeger, M. (2000). Effect of the input density distribution on kernel-based classifiers. Dans *Proc. ICML*.

[Williams et Seeger, 2001] Williams, C. et Seeger, M. (2001). *Advances in Neural Information Processing Systems*, volume 13, chapter Using the Nyström Method to Speed Up Kernel Machines. MIT Press.

[Woodbury, 1950] Woodbury, M. (1950). Inverting modified matrices. Memorandum rept. 42, statistical research group, Princeton University.

[Woodland, 1999] Woodland, P. (1999). Speaker adaptation : Techniques and challenges. Dans *Proc. IEEE ASRU*.

[Xiang et al., 2002] Xiang, B., Chaudhari, U., Navr'atil, J., Ramaswamy, G., et Gopinath, R. (2002). Short-time gaussianization for robust speaker verification. Dans *Proc. ICASSP*.

[Xie et al., 2006] Xie, Y., Dai, B., Yao, Z., et Liu, M. (2006). Kurtosis normalization in feature space for robust speaker verification. Dans *Proc. ICASSP*.

[Xu et al., 2006] Xu, J.-W., Pokharel, P., Jeong, K.-H., et C.Principe, J. (2006). An explicit construction of a reproducing gaussian kernel hilbert space. Dans *Proc. ICASSP*.

[Zhou et Hansen, 2003] Zhou, B. et Hansen, J. (2003). Discriminative acoustic model using eigenspace mapping for rapid speaker adaptation. Dans *Proc. ICASSP*.

[Zhou et Chellappa, 2006] Zhou, S. et Chellappa, R. (2006). From sample similarity to ensemble similarity : Probabilistic distance measures in reproducing kernel hilbert space. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(6) :917–929.

[Zhu et Hastie, 2001] Zhu, J. et Hastie, T. (2001). Kernel logistic regression and the import vector machine. *Advances in Neural Information Processing Systems*, 14.

[Zilca et al., 2004] Zilca, R., Pelecanos, J., Chaudhari, U., et Ramaswamy, G. (2004). Real Time Robust Speech Detection for Text Independent Speaker Recognition. Dans *Proc. IEEE Odyssey*.