Paramètres NPC pour la segmentation et le regroupement de locuteurs dans un flux audio

Bruno Gas, Christophe Charbuillet, Mohamed Chetouani, Jean-Luc Zarader

Laboratoire des Instruments et Systèmes d'Ile de France
Université Pierre et Marie Curie, Paris, France
Bruno.Gas@upmc.fr

Abstract

Nous présentons dans cet article une méthode d'extraction de caractéristiques du signal de parole : les paramètres NPC (Neural Predictive Coding). Nous détaillons les premiers résultats obtenus dans le cadre de la campagne d'évaluation ESTER sur une application de segmentation et regroupement de locuteurs (tâche SRL).

1. Introduction

Les paramètres NPC ont été proposés par Gas et Zarader [1], [2] pour l'extraction de caractéristiques appliquée à la reconnaissance de phonèmes [3]. Basés sur le principe d'une modélisation paramétrique non linéaire du spectre court-terme, ils consistent en l'estimation des poids d'un réseau de neurone utilisé en prédiction. En ce sens, il s'agit d'une extension des paramètres LPC à la modélisation de processus non linéaires. Un reproche que l'on peut faire aux paramètres LPC, et dont héritent naturellement les paramètres NPC, est leur caractère insuffisament discriminant. Ils sont de fait plus adaptés aux applications de codage [6] qu'aux applications de reconnaissance. Le renforcement du caractère discriminant est obtenu en adaptant l'étape d'extraction de caractéristiques à celle de classification. Le modèle NPC est donc le premier d'une série de codeurs faisant intervenir tour à tour différentes méthodes de collaboration classification-paramétrisation [4],[9]. Initialement validé sur des sous ensembles de classes phonétiques extraits des bases Darpa-TIMIT et NTIMIT, Chetouani [7] a proposé une architecture modulaire pour traiter l'ensemble des classes phonétiques de ces bases et obtenu 61.65% de reconnaissance contre 48,3% avec les paramètres LPC et 51,25% avec MFCC dans les mêmes conditions expérimentales. Il a également testé les paramètres NPC en reconnaissance du locuteur sur une base en langue espagnole [5] en collaboration avec Marcos Faundez-Zanuy [8] et obtenu sur une base de 49 locuteurs des taux allant de 61,63% à 100% de bonne reconnaissance. Dans les mêmes conditions expérimentales, LPC et MFCC permettaient d'obtenir respectivement 90,61% et 97,55%. L'objectif de notre participation à la campagne ESTER est de valider les paramètres NPC sur de grandes bases de données non segmentées, en l'occurence des flux audio radiophoniques. En contexte non supervisé (les locuteurs ne sont pas connus d'avance), il n'est plus possible d'envisager l'incorporation d'informations de classe d'appartenance des signaux dès l'étape d'extraction de caractéristiques. Nous proposons donc de tester les paramètres NPC-K qui sont une évolution récente des paramètres NPC permettant d'incorporer des contraintes discriminantes non supervisées. Le système s'inspire en cela fortement des cartes auto-organisantes de Kohonen. Nous préparons également actuellement une évaluation du modèle NPC-1, le modèle d'origine, qui ne nécessite pas de connaissance d'informations *a priori* sur les locuteurs traités, ces derniers étant supposés non connus.

2. Les paramètres NPC

Le Codeur Predictif Neuronal est une extension du codage LPC (Linear Predictive Coding) à la modélisation de signaux non linéaires. Etant donnée une séquence d'échantillons $\{y_{k-i,i=1,\dots\lambda}\}$ extraite d'une trame quelconque de parole, le réseau (de type MLP à une couche cachée et une cellule de sortie) effectue la prédiction de l'échantillon suivant y_k en fonction des λ précédents. Soit F de $\Re^\lambda \rightarrow \Re$ la fonction réalisée par le réseau. La prédiction $\hat{y}_k$ s'écrit :

$$\hat{y}_k = F([y_{k-1}, y_{k-2}, \dots, y_{k-\lambda}]^\top) \tag{1}$$

Nous désignerons par $\mathbf{x}_k$ le vecteur des échantillons précédents, soit :

$$\hat{y}_k = F(\mathbf{x}_k) \tag{2}$$

Soit $\Omega = [\omega_{ij}]$ le vecteur des poids du réseau. Ces poids sont calculés de sorte à minimiser l'erreur de prédiction $\epsilon_k = y_k - \hat{y}_k$ pour toutes les séquences $\mathbf{x}_k$ d'échantillons appartenant à la même trame. L'erreur quadratique de prédiction s'écrit :

$$\mathcal{L} = \sum_k (y_k - F_\Omega(\mathbf{x}_k))^2 \tag{3}$$

Après minimisation de $\mathcal{L}$ par l'algorithme de rétropropagation du gradient, F_Ω constitue une *modélisation* NLAR (Non Linear Auto-Regressive) de la trame considérée et Ω peut être considérée comme *caractéristique* de cette trame. Elle porte une information court-terme du signal vocal au même titre que les paramètres LPC sont considérés comme une représentation paramétrique du spectre court-terme. Bien entendu, Les mêmes conditions de stationnarité s'appliquent.

Un problème propre à cette approche est l'augmentation très rapide de la dimension du vecteur acoustique Ω lorsque l'on augmente la taille de la fenêtre de prédiction λ. La structure du codeur NPC est fondée sur l'hypotèse selon laquelle lorsque l'on écrit :

$$F(\mathbf{x}_k) = \sum_i a_i \sigma \left(\sum_j \omega_{ij} y_{k-j} \right) = G_\Omega \circ H_{\mathbf{a}}(\mathbf{x}_k) \tag{4}$$

où les $\Omega = [\omega_{ij}]$ sont maintenant les poids de la première couche et les $\mathbf{a}_i$ les poids de la deuxième couche, la fonction $H_{\mathbf{a}}$ dépend de la fonction $f : [-1,+1]^\lambda \longrightarrow [-1,+1]$ que l'on cherche à modéliser, tandis que G_Ω n'en dépend pas. Il ressort de cette hypothèse que seuls les coefficients de la deuxième

couche dépendent du processus ayant généré la trame de parole. Cette hypothèse a été démontrée à partir du théorème de Kolmogorov par plusieurs auteurs pour différents types de réseaux de neurones [10]. Par extension, un unique réseau prédicteur comportant autant de cellules de sortie qu'il existe de trames à représenter, modélise l'ensemble de ces fonctions par minimisation de la fonction de coût quadratique suivante :

$$\mathcal{L}(\Omega, \mathbf{a}_{1,\dots,L}) = \sum_{l}^{L} \sum_{k} (y_k - G_\Omega \circ H_{\mathbf{a}_l}(\mathbf{x}_k))^2 \delta_{\mathcal{T}(\mathbf{x}_k)-l} \quad (5)$$

où l désigne la trame, k les échantillons de la trame et $\mathcal{T}(\mathbf{x}_k)$ la trame d'appartenance des échantillons $\mathbf{x}_k$. On a $\delta_{\mathcal{T}(\mathbf{x}_k)-l} = 1$ si $\mathcal{T}(\mathbf{x}_k) = l$, et 0 sinon. L'idée principale du NPC est donc que les poids de la seconde couche constituent les paramètres représentatifs de la trame. Le vecteur acoustique ainsi généré voit sa dimension dépendre uniquement du nombre, arbitraire, de cellules cachées. La première couche ne dépend pas des trames considérées et peut donc être déterminée antérieurement à l'extraction de caractéristiques : c'est la phase de *paramétrisation*. Les modèles NPC-1 et NPC-K constituent deux méthodes différentes de *paramétrisation*, et donc deux méthodes de détermination des poids Ω. Lors de la phase d'extraction de caractéristiques d'une trame l de signal de parole, seuls les poids de la deuxième couche $\mathbf{a}_l$ sont estimés par minimisation de l'erreur quadratique. La première couche, issue de la phase de paramétrisation, réalise alors une transformation non linéaire du signal.

2.1. Le modèle NPC-1

2.1.1. Paramétrisation du codeur

La phase de paramétrisation est l'estimation de la fonction G_Ω. Elle s'obtient par minimisation du critère quadratique (5). Lors de cette phase, les paramètres $\mathbf{a}_l$ sont également estimés, mais ne sont plus utilisés par la suite.

2.1.2. Extraction de caractéristiques

La phase d'extraction de caractéristiques est la phase d'utilisation normale du NPC, c'est à dire la phase d'estimation des paramètres NPC. Pour une trame quelconque l et l'ensemble des séquences $\mathbf{x}_k$ qui la composent, la première couche du réseau est utilisée comme un opérateur de changement de représentation : $\mathbf{z} = G_\Omega(\mathbf{x}_k)$. Le vecteur des poids Ω étant celui obtenu lors de la phase de paramétrisation. L'estimation du vecteur caractéristique $\mathbf{a}_l$ s'obtient par minimisation de l'erreur de prédiction sur l'ensemble des vecteurs $\mathbf{z}_k$ calculés sur les séquences $\mathbf{x}_k$:

$$\mathcal{L}(\mathbf{a}_l) = \sum_{k} (y_k - H_{\mathbf{a}_l}(\mathbf{z}_k))^2 \quad (6)$$

2.2. Le modèle NPC-K

Optimiser les paramètres NPC pour une tâche de classification s'obtient, dans l'hypothèse simplificatrice d'une distribution gaussienne des paramètres, en minimisant la covariance intra-classe des paramètres générés tout en maximisant leur covariance inter-classes. Un tel procédé peut être introduit dans le codeur NPC sous la forme de contraintes sur les poids de la deuxième couche durant l'apprentissage [9] : lors de l'étape de paramétrisation, on limite le nombre de cellules de sorties pour n'en conserver que quelque unes plutôt qu'une par trame. Les vecteurs obtenus ne modélisent donc plus des trames de parole particulières mais des classes de trames et deviennent des vecteurs *représentants*. Typiquement, le modèle NPC-2 introduit par Chavy [4] ne conserve qu'un représentant par classe phonétique et est exploité en reconnaissance de phonèmes. Dans le cas qui nous préoccuppe nous n'avons pas connaissance des classes (les locuteurs par exemple). Le modèle NPC-K est une méthode d'apprentissage de la première couche qui permet d'incorporer des contraintes discriminantes non supervisées. Les cellules de la deuxième couche sont organisées selon une une carte 2D de type *carte de Kohonen* et on attend de l'apprentissage une auto-organisation de la carte permettant d'obtenir la propriété de tonotopie : à deux trames obéissant à des modèles prédictifs proches correspond deux neurones proches au sens de la distance définie sur la carte.

2.2.1. Distance NPC

La définition d'une carte de Kohonen (le lecteur se reportera sur l'abondante litérature concernant ce domaine pour une définition plus complète des cartes de Kohonen) nécessite de disposer d'une distance sur l'espace des formes en entrée permettant de comparer une forme quelconque avec les poids d'un neurone quelconque de la carte, représentant d'une classe de forme. Ici, les formes sont des signaux (trames de parole) et les représentants sur la carte, des vecteurs caractéristiques (les poids d'un neurone de la carte prédictive). La *distance NPC* entre deux trames de parole l et m est définie selon le principe de la distance d'Itakura étendue aux modèles NLAR de type NPC [9] :

$$d_\Omega(l, m) = \log \frac{\mathcal{L}(\Omega, \mathbf{a}_l, m)}{\mathcal{L}(\Omega, \mathbf{a}_m, m)} \quad (7)$$

où $\mathcal{L}(\Omega, \mathbf{a}_l, m)$ est l'erreur de prédiction calculée sur les échantillons de la trame m en utilisant les paramètres $\mathbf{a}_l$ estimés sur la trame l par le modèle NPC. (7) est le rapport entre l'erreur de prédiction de la trame m en utilisant les coefficients NPC estimés à partir de la trame l et l'erreur de prédiction obtenue en utilisant cette fois les coefficients NPC estimés à partir de la trame m. Ce rapport tend vers 1, donc la distance vers 0, lorsque les deux trames sont identiques. d_Ω est plus exactement une mesure de distorsion car elle n'est pas symétrique.

2.2.2. Paramétrisation NPC-K

L'algorithme d'apprentissage est celui d'une carte de Kohonen traditionnelle (algorithme KONLINE) dont on aurait remplacé la distance euclidienne par la distance NPC :

Pour toutes les trames m de l'ensemble d'apprentissage :

- 1) Calculer l'ensemble des distances $d_\Omega(l, m)$ entre la trame m et les $l = 1, \dots, L$ trames représentées sur la carte prédictive ;
- 2) choisir le neurone représentant la trame la plus proche au sens de cette distance ;
- 3) modifier les poids du neurone gagnant de sorte à diminuer la distance $d_\Omega(l^*, m)$ le séparant de la trame m (minimisation de l'erreur de prédiction) ;
- 4) déterminer le voisinage du neurone gagnant sur la carte en utilisant une fonction de voisinage $V(l, l^*)$ au sens de la distance définie sur la carte ;
- 5) modifier les poids des neurones du voisinage de sorte à les rapprocher également de la trame m (minimisation de l'erreur de prédiction).

Plusieur fonctions de voisinage peuvent être utilisées. Nous avons utilisé la fonction gaussienne permettant de pondérer les modifications des neurones du voisinage en fonction de leur distance au gagnant selon une loi gaussienne :

$$V(l, l^*) = e^{-\frac{d(l,l^*)}{2\sigma}} \quad (8)$$

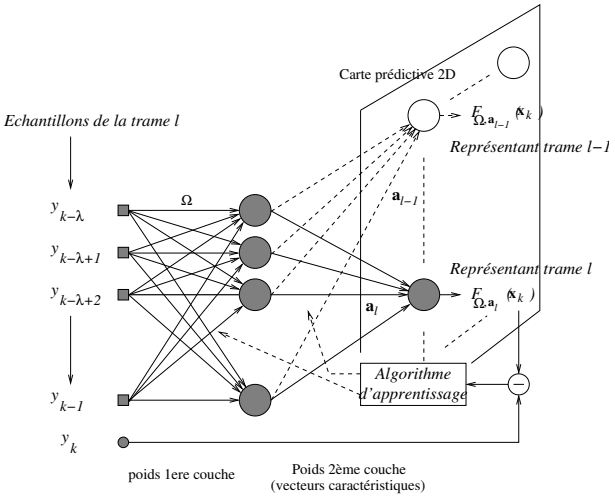

FIG. 1 – *Schéma du codeur NPC-K utilisé.*

où $d(l, l^*)$ désigne la distance sur la carte séparant les deux neurones prédictifs l et l^*. La variance σ représente l'étendue du voisinage. C'est une fonction décroissante du temps en cours d'apprentissage. La détermination du neurone gagnant lorsque l'on présente une trame m inconnue nécessite d'estimer l'ensemble des distances NPC $d_\Omega(l, m)$ et donc d'estimer les paramètres NPC $\mathbf{a}_m$ de la trame m. Le détail de la règle de décision montre qu'un tel calcul n'est pas nécessaire :

$$l^* = \arg \min_{l=1,\ldots,L} \{d_\Omega(l, m)\} \tag{9}$$

entraîne :

$$l^* = \arg \min_{l=1,\ldots,L} \{\log \frac{\mathcal{L}(\Omega, \mathbf{a}_l, m)}{\mathcal{L}(\Omega, \mathbf{a}_m, m)}\} \tag{10}$$

soit,

$$l^* = \arg \min_{l=1,\ldots,L} \{\mathcal{L}(\Omega, \mathbf{a}_l, m)\} \tag{11}$$

Ainsi, seule l'estimation de l'erreur de prédiction des échantillons de la trame m sur l'ensemble des cellules $\mathbf{a}_l$ est nécessaire pour déterminer le neurone gagnant.

2.2.3. Extraction de caractéristiques

L'extraction de caractéristique d'une trame quelconque, une fois les poids Ω de la première couche estimés, s'effectue de la même façon que précédemment avec le modèle NPC-1.

3. Premiers résultats expérimentaux

Nous présentons les premiers résultats obtenus à partir d'une paramétrisation NPC-K effectuée sur le signal `19981207_0700_0800_inter_fm_dga.wav` du corpus 50h phase 1. Ces résultats sont préliminaires dans la mesure où nous n'avons pas encore constitué une base de signaux représentative de l'ensemble des média traités dans la campagne ESTER. Des simulations sur une paramétrisation NPC-1 sont actuellement en cours dont les résultats seront présentés ultérieurement.

3.1. Contexte expérimental de la tâche SRL

Notre campagne de tests sur la tâche SRL comporte les principales étapes suivantes :

- 1) Estimation des poids de la première couche du codeur (*paramétrisation* du codeur). Cette étape s'effectue à l'aide d'une base de signaux de parole n'appartenant pas aux signaux utilisés pour l'évaluation.
- 2) Extraction à l'aide du codeur précédemment défini des paramètres acoustiques des signaux de parole utilisés pour l'évaluation
- 3) Segmentation et regroupement des locuteurs à partir des *features* précédemment extraits à l'aide de l'outil *audioseg*[11]

3.1.1. Paramétrisation du codeur

50 itérations d'apprentissage ont été effectuées sur le signal mentionné plus haut (une heure de parole), représentant 23 heures de simulation sur un PC pentium cadencé à 3.2 GHz. Les trames étaient de 20ms, extraites toutes les 10ms. Le modèle comprenait une première couche de 16 neurones (fenêtre de prédiction sur 20 échantillons) et une carte prédictive 2D comprenant 10x10 = 100 neurones prédictifs. Cette structure permet d'estimer ensuite des vecteurs caractéristiques de dimension 16. Les poids initiaux étaient choisis selon une distribution aléatoire. En revanche, les poids de la carte prédictive (2ème couche) étaient initialisés à 0.

3.1.2. Extraction des paramètres acoustiques

Nous présentons dans ce papier une première évaluation des paramètres acoustiques sur l'ensemble des signaux de la base *DEV-2* d'évaluation de la phase 2 (environ 8h de parole). Pour toutes les trames à traiter, l'extraction des vecteurs caractéristiques NPC s'est effectuée en 10 itérations d'apprentissage (descente de gradient stochastique) avec une initialisation "linéaire" de ces vecteurs. Cette initialisation a été faite dans le soucis de comparer l'apport du traitement non linéaire relativement à une méthode linéaire LPC de même type (méthode prédictive). Les temps de calcul sont de l'ordre du temps réel et peuvent être sensiblement améliorés par une optimisation du code.

Afin de permettre une comparaison des paramètres NPC avec d'autres méthodes d'extraction de caractéristiques, nous avons également estimé les paramètres LPC et LFCC sur les mêmes signaux. Dans tous les cas, les trames étaient de 20ms, espacées de 10 ms et les *features* de dimension 16. Les coefficients log-énergie étaient également estimés. En revanche nous n'avons pas considéré dans cette étape les coefficients dynamiques Δ et $\Delta\Delta$.

3.1.3. Segmentation et regroupement de locuteurs

Le système de segmentation repose sur l'utilisation du critère BIC (Bayesian Information Criterion) appliqué à une modélisation mono-Gaussienne à matrice de covariance diagonale. Pour deux fenêtres adjacentes, on cherche à déterminer quelle modélisation est la plus vraisemblable compte tenue de sa complexité. Un modèle à une Gaussienne par fenêtre est comparé à une modélisation mono-Gaussienne de l'ensemble des données. Si la modélisation bi-Gaussienne s'avère être la plus vraisemblable alors un changement de locuteur est détecté à l'interface des fenêtres. Dans notre étude, la mise en oeuvre de l'algorithme de segmentation à été effectuée par le système *audioseg* [11] dans lequel une segmentation BIC en trois passes est implémentée. Un algorithme de regroupement hiearchique est ensuite appliqué sur les segments, basé sur la distance BIC, et toujours réalisé par l'outil *audioseg*.

3.2. Résultats intermédiaires

Les trois tableaux suivants donnent le détail des scores obtenus pour les trois paramètres utilisés : LPC, LFCC et NPC-K

Fichiers	LPC	LFCC
20030418_0700_0800_INTER_DGA	43.70	15.42
20030418_0800_0900_INTER_DGA	51.29	46.19
20030418_1200_1300_FINFO_DGA	35.17	27.58
20030418_1700_1800_FINFO_DGA	41.01	27.71
20030508_1400_1500_RFI_ELDA	27.85	14.67
20030509_1400_1500_RFI_ELDA	26.59	23.25
20030717_0700_0715_RTM_ELDA	33.31	4.8
20030717_1300_1320_RTM_ELDA	21.15	1.82
20030717_2000_2020_RTM_ELDA	8.28	6.42
20030717_2300_2315_RTM_ELDA	25.28	23.62
20030719_0700_0715_RTM_ELDA	19.11	42.98
20030719_1300_1320_RTM_ELDA	12.76	1.41
20030719_2000_2015_RTM_ELDA	21.08	14.71
20030719_2300_2310_RTM_ELDA	9.92	0.63
Moyenne Pondérée :	33.08%	22.50%

Au vu de ces premiers résulats, les scores obtenus par les paramètres NPC se situent entre ceux obtenus par les paramètres LPC et ceux obtenus par les paramètres LFCC. Ces résultats sont encourageants en ce qu'ils montrent que le modèle non linéaire NPC, apporte des informations supplémentaires utiles à la segmentation locuteur, relativement au modèle linéaire de la même famille (modèle prédictif). Des travaux sont en cours pour étudier l'apport des contraintes discriminantes par auto-organisation relativement à un modèle plus simple non contraint tel que le NPC-1. Enfin, l'apprentissage de la première couche du codeur nécessite d'être complété sur une base de signaux plus importante et plus représentative de l'application considérée.

4. Conclusions

Nous avons présenté une première évaluation des paramètres NPC sur la tâche SRL de segmentation et regroupement de locuteurs. Les résultats présentés sont des résultats intermédiaires, le système d'évaluation étant en cours de réalisation. Ils montrent déjà une amélioration des performances des paramètres NPC relativement aux paramètres LPC mais restent en deça des scores obtenus avec les paramètres LFCC. Les travaux en cours visent à améliorer les conditions d'utilisation du codeur, en portant notament l'effort sur l'estimation des poids de la première couche.

Fichiers	Scores NPC-K
20030418_0700_0800_INTER_DGA	22.54
20030418_0800_0900_INTER_DGA	23.51
20030418_1200_1300_FINFO_DGA	38.52
20030418_1700_1800_FINFO_DGA	44.28
20030508_1400_1500_RFI_ELDA	22.59
20030509_1400_1500_RFI_ELDA	21.89
20030717_0700_0715_RTM_ELDA	22.47
20030717_1300_1320_RTM_ELDA	11.07
20030717_2000_2020_RTM_ELDA	8.33
20030717_2300_2315_RTM_ELDA	24.02
20030719_0700_0715_RTM_ELDA	15.17
20030719_1300_1320_RTM_ELDA	4.92
20030719_2000_2015_RTM_ELDA	23.09
20030719_2300_2310_RTM_ELDA	10.64
Moyenne Pondérée :	25.22%

5. Remerciements

Nous voudrions remercier tout particulièrement Jean-François Bonastre et son équipe (LIA, Avignon) et Guillaume Gravier (IRISA/RENNES) pour leur aide précieuse dans l'utilisation des logicels ALIZE, SPRO et AUDIOSEG qui nous permettent de valider les paramètres NPC dans le contexte de la campagne ESTER.

6. References

[1] Gas, B. and Zarader, J.L. and Sellem, P. and Didiot, J.C., "Speech coding by limited weights neural network", IEEE Inter. Conf. on Systems Man and Cybernetics, 1997.

[2] Zarader, J.L. and Gas, B. and Didiot, J.C. and Sellem, P., "Neural Predictive Coding : application to phoneme recognition", Inter. Conf. on Neural Information Processing, 1997.

[3] Gas, B. and Zarader, J.L. and Chavy, C., "A New Approach to Speech Coding : The Neural Predictive Coding", J. of Advanced Computational Intelligence, Vol. 4(1) :120–127, 2000. IEEE Inter. Conf. on Systems Man and Cybernetics, 1997.

[4] Chavy, C. and Gas, B. and Zarader, J.L., "Discriminative coding with predictive neural networks", Inter. Conf. on Artificial Neural Network, Vol. 4(1) :219–222, 1999.

[5] Ortega-Garcia, J. and all, "A large speech corpus in spanish for speaker identification and verification", Proc. of Inter. Conf. on Acoustic, Speech and Signal Processing, Vol. 2 :773–776, 1998.

[6] Zarader, J.L. and Gas, B. and Charlelie-Nelson, D. and Chavy, C., "New compression and decompression of speech signals by NPC", Inter. Conf. on Signal, Speech and Image Processing, Vol. 119–125, 2001.

[7] Chetouani, M. and Gas, B. and Zarader, J.L., "Modular neural predictive coding for discriminative feature extraction", IEEE Inter. Conf. on Acoustic Speech and Signal Processing, Vol. 2 :33–36, 2003.

[8] Chetouani, M. and Faundez-Zanuy, M. and Gas, B. and Zarader, J.L., "A New Nonlinear speaker parameterization algorithm for speaker identification", Proc. of ISCA Tutorial and Research Workshop on Speaker and Recognition Langage Workshop, 309–314, 2004.

[9] Gas, B. and Zarader, J.L. and Chavy, C., and Chetouani, M., "Discriminant neural predictive coding applied to phoneme recognition", Neurocomputing, Vol. 56 :141–166, 2004.

[10] Hornik, K., "Multilayer feedforward networks are universal approximators", Neural Networks, Vol. 2 :359–366, 1989.

[11] Gravier, G. and Betser, M., "Audioseg : Audio Segmentation Toolkit", 2005.