UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Traitement de l'incertitude pour la reconnaissance de la parole robuste au bruit

THÈSE

présentée et soutenue publiquement le 20 novembre 2015

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Dung Tien TRAN

Composition du jury

Président du jury:	**François CHARPILLET** Directeur de Recherche, Inria Nancy - Grand Est
Rapporteurs :	**Dorothea KOLOSSA** Associate professor, Ruhr-Universität Bochum **Yannick ESTÈVE** Professeur, Université du Mans
Examinateur :	**Shinji WATANABE** Senior Principal Member Research Staff, Mitsubishi Electric Research Laboratories
Directeurs de thèse :	**Emmanuel VINCENT** Chargé de Recherche, Inria Nancy - Grand Est **Denis JOUVET** Directeur de Recherche, Inria Nancy - Grand Est

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Mis en page avec la classe thesul.

Abstract

This thesis focuses on noise robust automatic speech recognition (ASR). It includes two parts. First, we focus on better handling of uncertainty to improve the performance of ASR in a noisy environment. Second, we present a method to accelerate the training process of a neural network using an auxiliary function technique.

In the first part, multichannel speech enhancement is applied to input noisy speech. The posterior distribution of the underlying clean speech is then estimated, as represented by its mean and its covariance matrix or uncertainty. We show how to propagate the diagonal uncertainty covariance matrix in the spectral domain through the feature computation stage to obtain the full uncertainty covariance matrix in the feature domain. Uncertainty decoding exploits this posterior distribution to dynamically modify the acoustic model parameters in the decoding rule. The uncertainty decoding rule simply consists of adding the uncertainty covariance matrix of the enhanced features to the variance of each Gaussian component.

We then propose two uncertainty estimators based on fusion to nonparametric estimation, respectively. To build a new estimator, we consider a linear combination of existing uncertainty estimators or kernel functions. The combination weights are generatively estimated by minimizing some divergence with respect to the oracle uncertainty. The divergence measures used are weighted versions of Kullback-Leibler (KL), Itakura-Saito (IS), and Euclidean (EU) divergences. Due to the inherent nonnegativity of uncertainty, this estimation problem can be seen as an instance of weighted nonnegative matrix factorization (NMF).

In addition, we propose two discriminative uncertainty estimators based on linear or nonlinear mapping of the generatively estimated uncertainty. This mapping is trained so as to maximize the boosted maximum mutual information (bMMI) criterion. We compute the derivative of this criterion using the chain rule and optimize it using stochastic gradient descent.

In the second part, we introduce a new learning rule for neural networks that is based on an auxiliary function technique without parameter tuning. Instead of minimizing the objective function, this technique consists of minimizing a quadratic auxiliary function which is recursively introduced layer by layer and which has a closed-form optimum. Based on the properties of this auxiliary function, the monotonic decrease of the new learning rule is guaranteed.

Résumé

Cette thèse se focalise sur la reconnaissance automatique de la parole (RAP) robuste au bruit. Elle comporte deux parties. Premièrement, nous nous focalisons sur une meilleure prise en compte des incertitudes pour améliorer la performance de RAP en environnement bruité. Deuxièmement, nous présentons une méthode pour accélérer l'apprentissage d'un réseau de neurones en utilisant une fonction auxiliaire.

Dans la première partie, une technique de rehaussement multicanal est appliquée à la parole bruitée en entrée. La distribution a posteriori de la parole propre sous-jacente est alors estimée et représentée par sa moyenne et sa matrice de covariance, ou incertitude. Nous montrons comment propager la matrice de covariance diagonale de l'incertitude dans le domaine spectral à travers le calcul des descripteurs pour obtenir la matrice de covariance pleine de l'incertitude sur les descripteurs. Le décodage incertain exploite cette distribution a posteriori pour modifier dynamiquement les paramètres du modèle acoustique au décodage. La règle de décodage consiste simplement à ajouter la matrice de covariance de l'incertitude à la variance de chaque gaussienne.

Nous proposons ensuite deux estimateurs d'incertitude basés respectivement sur la fusion et sur l'estimation non-paramétrique. Pour construire un nouvel estimateur, nous considérons la combinaison linéaire d'estimateurs existants ou de fonctions noyaux. Les poids de combinaison sont estimés de façon générative en minimisant une mesure de divergence par rapport à l'incertitude oracle. Les mesures de divergence utilisées sont des versions pondérées des divergences de Kullback-Leibler (KL), d'Itakura-Saito (IS) ou euclidienne (EU). En raison de la positivité inhérente de l'incertitude, ce problème d'estimation peut être vu comme une instance de factorisation matricielle positive (NMF) pondérée.

De plus, nous proposons deux estimateurs d'incertitude discriminants basés sur une transformation linéaire ou non-linéaire de l'incertitude estimée de façon générative. Cette transformation est entraînée de sorte à maximiser le critère de maximum d'information mutuelle boosté (bMMI). Nous calculons la dérivée de ce critère en utilisant la règle de dérivation en chaîne et nous l'optimisons par descente de gradient stochastique.

Dans la seconde partie, nous introduisons une nouvelle méthode d'apprentissage pour les réseaux de neurones basée sur une fonction auxiliaire sans aucun réglage de paramètre. Au lieu de maximiser la fonction objectif, cette technique consiste à maximiser une fonction auxiliaire qui est introduite de façon récursive couche par couche et dont le minimum a une expression analytique. Grâce aux propriétés de cette fonction, la décroissance monotone de la fonction objectif est garantie.

Remerciements

I would like to acknowledge many people who have helped me along the way to this milestone. I will start by thanking my thesis supervisor, Emmanuel Vincent. I have learned a great deal of audio processing, machine learning from him, and have benefited from his skill and intuition at solving problems. I would also like to thank Denis Jouvet, who essentially co-supervised much of my PhD research. His enthusiasm for speech recognition is insatiable, and his support of this work has been greatly appreciated. Without their guidance, I would not have been able to complete this thesis.

I'm very grateful to all members of MULTISPEECH research team for sharing a great working atmosphere everyday with me. I have learned a lot from their enthusiasm, and immense knowledge. Many thanks to Antoine Liutkus, Yann Salaün and Nathan Souviraà-Labastie for very fruitful discussions about audio processing and to Imran Sheikh, Sunit Sivasankaran, Juan Andrés Morales Cordovilla, and Arie Nugraha for the numerous interesting discussions they provided in speech recognition and neural networks.

I would like to acknowledge Nobutaka Ono, Le Trung Kien, Daichi Kitamura, Keisuke Imoto, Eita Nakamura, Ta Duc Tuyen of the Ono Lab at the National Institute of Informatics for warmly welcoming me into their lab and providing me very good conditions for research. I am very grateful to Le Trung Kien for giving me wonderful advice about optimization and math when I was at the National Institute of Informatics as well as to Shoji Makino from the University of Tsukuba for his advice about career.

Many thanks also go to many Vietnamese friends in Nancy, who have been with me to share the good and bad times.

Last but not least, I would like to express my love to my parents and sisters who have always been there for me, encouraging me and helping me to be who I am. This thesis would not have been possible without the love and affection that they have provided.

Contents

Appendix A

Tables

List of Figures

1

Résumé étendu

Cette thèse porte sur la robustesse au bruit de la reconnaissance automatique de la parole (RAP). La RAP vise à obtenir la séquence de phones ou de mots correspondant à une séquence de descripteurs acoustiques observée [Rabiner and Juang, 1993; Gales and Young, 2008]. Sa performance se dégrade fortement en présence de réverbération et de bruit, qui engendrent un décalage entre les données d'apprentissage et de test. Les systèmes de RAP se divisent généralement au deux parties: le *front-end* et le *back-end*. Le décodage incertain [Kolossa and Haeb-Umbach, 2011; Droppo et al., 2002; Deng, 2011; Deng et al., 2005; Ion and Haeb-Umbach, 2006] est une méthode qui permet de connecter ces deux parties en tenant compte de l'incertitude sur le signal ¡¡débruité¿¿. Dans cette thèse, nous nous focaliserons uniquement sur la robustesse au bruit à partir d'enregistrements multicanaux avec des microphones distants impliquant des bruits complexes provenant de plusieurs sources fortement non-stationnaires. La thèse comporte deux parties. La première partie se focalise sur l'amélioration du décodage incertain et de la performance de RAP en environnement bruité. La deuxième partie présente une méthode pour accélérer l'apprentissage d'un réseau de neurones en utilisant une fonction auxiliaire.

1.1 Décodage incertain

1.1.1 État de l'art

Une approche classique pour augmenter la robustesse au bruit non-stationnaire consiste à appliquer une technique de rehaussement de la parole multicanal [Ozerov and Févotte, 2010; Ozerov et al., 2012] au signal de parole bruité en entrée. Le signal de parole rehaussé est alors considéré comme une estimée ponctuelle du signal de parole propre.

Le décodage incertain est une approche prometteuse qui, au-delà d'une telle estimée ponctuelle, vise à estimer la distribution *a posteriori* de la parole propre et à l'utiliser pour modifier dynamiquement les paramètres du modèle acoustique de la parole lors du décodage. Cette distribution *a posteriori* est approximée par une distribution gaussienne, dont la moyenne

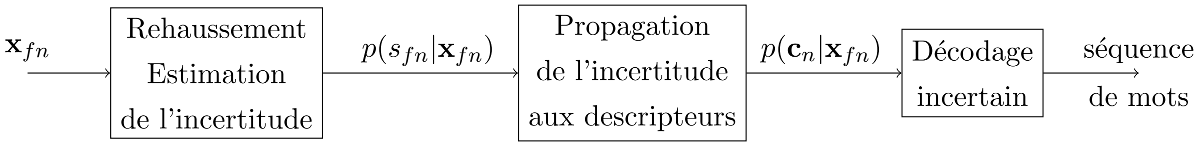

$\mathbf{x}_{fn}$
Rehaussement Estimation de l'incertitude
$p(s_{fn}|\mathbf{x}_{fn})$
Propagation de l'incertitude aux descripteurs
$p(\mathbf{c}_n|\mathbf{x}_{fn})$
Décodage incertain
séquence de mots

Figure 1.1: Schéma du décodage incertain.

représente la parole rehaussée et la variance représente l'incertitude, c'est-à-dire le degré de distorsion résiduelle de la parole après rehaussement [Astudillo, 2010] estimé à partir d'un modèle paramétrique de distorsion prenant en compte la réverbération ou le bruit. L'incertitude peut être soit calculée directement sur les descripteurs utilisés par la RAP [Deng, 2011; Krueger and Haeb-Umbach, 2013; Delcroix et al., 2013b; Liao, 2007; Delcroix et al., 2009] soit estimée dans le domaine spectral et propagée ensuite aux descripteurs [Kolossa et al., 2010; Astudillo and Orglmeister, 2013; Ozerov et al., 2013; Astudillo, 2010; Srinivasan and Wang, 2007; Kallasjoki et al., 2011; Nesta et al., 2013]. Nous nous focaliserons sur cette deuxième approche dans cette thèse. Lorsque le modèle acoustique utilisé pour la RAP est un modèle de Markov avec probabilités d'observation par mélanges de gaussiennes (GMM-HMM), le décodage incertain se résume alors à ajouter cette variance à celle de chaque gaussienne dans le calcul de la vraisemblance [Deng et al., 2005]. L'approche complète est illustrée dans la figure 1.1.

1.1.2 Propagation de l'incertitude comme une matrice de covariance pleine

Lorsque les descripteurs utilisés pour la RAP sont les coefficients cepstraux en échelle Mel (MFCC), l'incertitude présente une corrélation entre les descripteurs. Nous avons proposé une méthode pour calculer cette corrélation sous forme d'une matrice de covariance pleine.

Au départ, l'incertitude est estimée dans le domaine spectral complexe comme la variance a posteriori du filtre de Wiener utilisé pour le rehaussement en chaque point temps-fréquence [Astudillo, 2010]. Les moments croisés du spectre d'amplitude et du spectre de puissance sont calculés en utilisant les statistiques de la distribution de Rice. Ces moments sont ensuite propagés au vecteur constitué des MFCCs et de la log-énergie à travers une suite d'étapes (pré-emphase, banc de filtres Mel, logarithme, transformée en cosinus discrète, et liftrage) en utilisant une expansion en série de Taylor vectorielle (VTS) d'ordre 1 [Ozerov et al., 2013] fournissant la matrice de covariance pleine de l'incertitude sur ce vecteur. Cette matrice est finalement propagée à travers une matrice de dérivation temporelle afin d'obtenir la matrice de covariance de l'incertitude sur le vecteur de descripteurs complet constitué des MFCCs, de la log-énergie, et de leurs dérivées temporelles d'ordre 1 et 2. Un exemple de matrice de covariance pleine est montré dans la figure 1.2 et comparé à la vérité terrain (oracle).

La performance de RAP associée est évaluée dans le tableau 1.1 sur la tâche 1 du 2e défi CHiME [Vincent et al., 2013b]. Les résultats montrent que le décodage incertain avec une matrice de covariance pleine réduit le taux d'erreur de 5%, en relatif, par rapport à une matrice

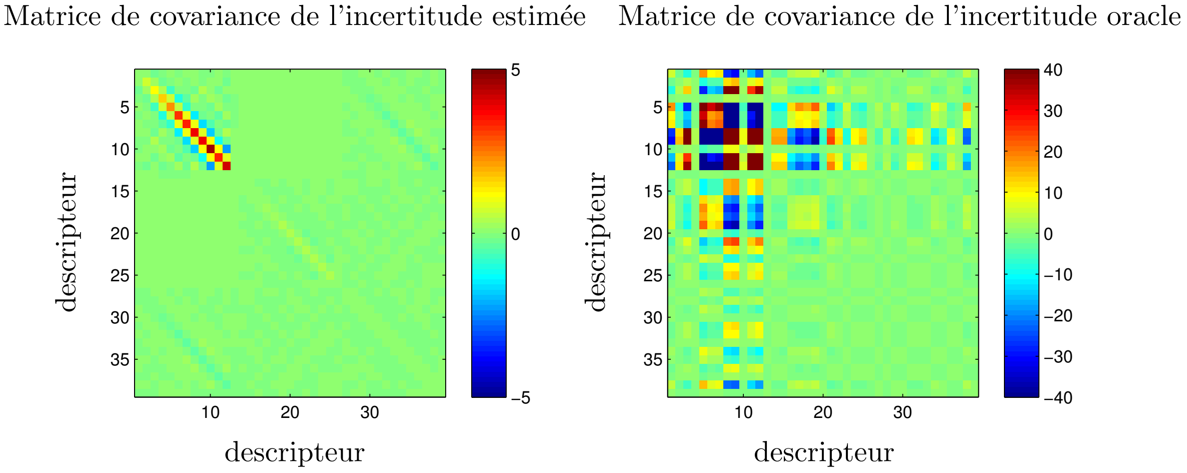


Figure 1.2: Exemple de matrice de covariance pleine de l'incertitude sur une trame. Gauche: Wiener + VTS. Droite: oracle.

Covariance de l'incertitude	Descripteurs incertains	Ensemble de test						
		-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Moyenne
sans incertitude		73.75	78.42	84.33	89.50	91.83	92.25	85.01
diagonale	statiques	75.00	79.00	84.75	90.13	91.92	93.67	85.74
	dynamiques	75.00	79.00	84.92	90.33	91.92	92.33	85.58
	tous	76.93	79.17	85.92	90.00	92.00	93.75	86.29
pleine	statiques	76.75	79.33	85.50	90.33	92.33	93.67	86.31
	dynamiques	76.75	79.17	85.75	90.33	92.00	93.83	86.30
	tous	77.92	80.75	86.75	90.50	92.92	93.75	**87.00**

Table 1.1: Précision d'estimation des mots-clés (%) sur la tâche 1 du 2e défi CHiME obtenue par décodage incertain des descripteurs statiques et dynamiques. Les précisions moyennes ont un intervalle de confiance à 95% de ±0.8%.

diagonale et de 13% par rapport au système de base (rehaussement sans incertitude). Une autre expérience (non décrite dans le tableau) a montré que cette amélioration est essentiellement due à la modélisation des corrélations de l'incertitude entre les MFCCs et leurs dérivées. La corrélation de l'incertitude entre les MFCCs et la log-énergie est très faible et sa modélisation n'améliore pas la performance.

1.1.3 Estimation de l'incertitude basée sur l'apprentissage génératif

Malgré l'amélioration de performance constatée en RAP, il est visible que l'incertitude estimée reste inférieure à l'incertitude réelle. Cette constatation est valable non seulement pour le filtre de Wiener mais aussi pour les autres estimateurs de l'incertitude dans le domaine spectral.

La visualisation de ces estimateurs dans la figure 1.3 montre qu'ils ont un comportement

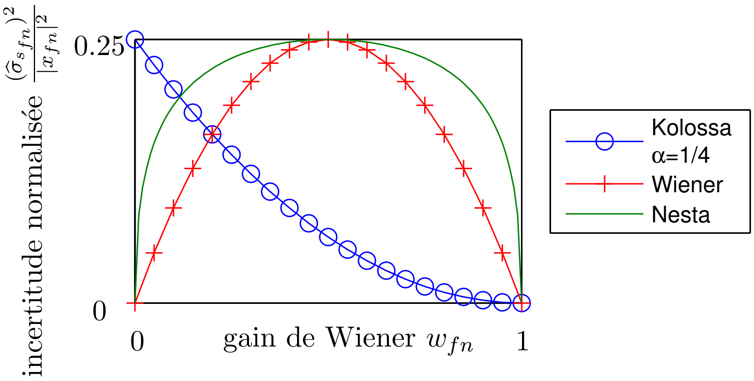


Figure 1.3: Comportement des estimateurs d'incertitude. L'axe horizontal représente la proportion de parole dans le spectre de puissance de la parole bruitée observé. L'axe vertical est proportionnel à l'incertitude. L'incertitude est normalisée par le spectre de puissance de la parole bruitée pour illustrer le fait que la forme des courbes n'en dépend pas.

différent. L'estimateur de Kolossa [Kolossa et al., 2010] décroît lorsque le spectre de puissance estimé de la parole croît. Les deux autres estimateurs atteignent un maximum lorsque les spectres de puissance estimés de la parole et du bruit sont égaux. L'estimateur de Nesta [Nesta et al., 2013] croît plus vite que celui de Wiener [Astudillo, 2010].

Alors que les approches existantes reposent sur un seul de ces estimateurs, nous proposons de les fusionner afin d'obtenir un meilleur estimateur. Cette fusion est réalisée indépendamment à chaque fréquence. En notant par E le nombre d'estimateurs originels, l'estimateur fusionné $(\widehat{\sigma}^{\text{fus}}_{s_{fn}})^2$ sur la trame n à la fréquence f peut s'exprimer comme

$$(\widehat{\sigma}^{\text{fus}}_{s_{fn}})^2 = \sum_{e=1}^{E} \theta^e_{s_f} \, (\widehat{\sigma}^e_{s_{fn}})^2 \tag{1.1}$$

où $(\widehat{\sigma}^e_{s_{fn}})^2$ sont les estimateurs originels et $\theta^e_{s_f}$ les coefficients de fusion. Ces coefficients sont appris sur un ensemble d'apprentissage de sorte à minimiser la divergence de Kullback-Leibler (KL), la divergence d'Itakura-Saito (IS), ou la divergence euclidienne (EU) entre l'incertitude estimée et l'incertitude oracle, calculée comme la différence au carré entre le spectre complexe de la parole rehaussée et celui de la parole propre [Deng et al., 2005]. La divergence est pondérée entre les points temps-fréquence de sorte à prendre compte les différences d'échelle. Il s'agit d'un problème de factorisation matricielle positive pondérée [Lee and Seung, 1999], qui est résolu à l'aide de mises à jour multiplicatives. Trois estimateurs fusionnés correspondant aux trois mesures de divergence ci-dessus sont alors obtenus et propagés aux descripteurs par VTS. Ces trois estimateurs propagés sont fusionnés à nouveau de sorte à obtenir un meilleur estimateur sur les descripteurs. La fusion est effectuée de la même façon, en calculant l'incertitude oracle comme la différence au carré entre les descripteurs de la parole rehaussée et ceux de la parole propre. La performance de RAP associée est présentée plus loin dans le tableau 1.2.

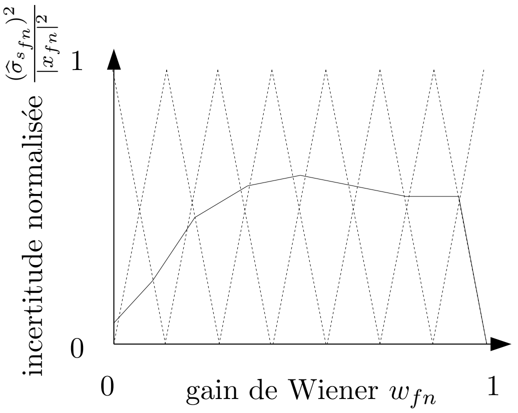


Figure 1.4: $E = 8$ fonctions noyaux triangulaires $\mathrm{b}^e(w_{fn})$ (traits pointillés) et estimateur non-paramétrique de l'incertitude $(\widehat{\sigma}^{\text{fus}}_{s_{fn}})^2/|x_{fn}|^2$ (trait plein). L'axe horizontal représente la proportion de parole dans le spectre de puissance de la parole bruitée observé. L'axe vertical est proportionnel à l'incertitude. L'incertitude est normalisée par le spectre de puissance de la parole bruitée pour illustrer le fait que la forme de l'estimateur (le trait plein) n'en dépend pas. Le trait plein est obtenu en sommant les traits pointillés avec différents poids positifs.

Bien que l'estimateur fusionné soit potentiellement meilleur que les estimateurs originels, sa forme reste contrainte par ces estimateurs originels. Afin de s'en affranchir, nous proposons une méthode d'estimation non-paramétrique. L'expression mathématique des divers estimateurs illustrés dans la figure 1.3 montre qu'ils partagent deux propriétés. Premièrement, l'incertitude estimée est proportionnelle au spectre de puissance de la parole bruitée. Deuxièmement, elle s'exprime comme une fonction du gain de Wiener, c'est-à-dire du rapport entre le spectre de puissance de la parole seule et celui de la parole bruitée. Compte tenu de ces propriétés, nous définissons un ensemble de fonctions noyaux

$$(\widehat{\sigma}^e_{s_{fn}})^2 = |x_{fn}|^2 \mathrm{b}^e(w_{fn}) \tag{1.2}$$

où $\mathrm{b}^e(.)$ sont des fonctions normalisées du gain de Wiener réparties de façon régulière entre 0 et 1 et indexées par $e \in \{1, \ldots, E\}$ et $|x_{fn}|^2$ est le spectre de puissance de la parole bruitée. Par la suite, nous choisissons des fonctions noyaux triangulaires:

$$\mathrm{b}^e(w_{fn}) = (E-1)\max(0, 1 - |(E-1)w_{fn} - (e-1)|). \tag{1.3}$$

L'incertitude est maintenant exprimée de la même façon que dans l'équation (1.1), mais les estimateurs originels sont remplacés par les fonctions noyaux. L'estimateur résultant est une fonction linéaire par morceaux du gain de Wiener, tel qu'illustré dans la figure 1.4. Cet estimateur est propagé aux descripteurs par VTS et passé à travers une deuxième fonction non-paramétrique afin d'obtenir l'incertitude finale. Les poids des deux fonctions non-paramétriques (dans le domaine spectral et dans celui des descripteurs) sont appris par factorisation matricielle

Estimation	Propagation	Covariance	Ensemble de test						
		de l'incertitude	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Moyenne
Wiener	VTS	diagonale	76.93	79.17	85.92	90.00	92.00	93.75	86.29
fusion	VTS		78.33	80.17	85.92	90.08	92.08	94.17	86.97
fusion	fusion		80.50	82.17	88.25	91.33	92.50	93.58	88.05
non-paramétrique	VTS		80.00	81.92	87.25	91.50	92.25	93.08	87.66
non-paramétrique	non-paramétrique		81.75	83.50	88.33	91.08	92.75	93.00	**88.40**
Wiener	VTS	pleine	77.92	80.75	86.75	90.50	92.92	93.75	87.00
fusion	VTS		81.00	81.50	87.33	91.00	93.50	94.92	88.20
fusion	fusion		83.17	84.33	89.75	91.17	93.33	93.33	89.18
non-paramétrique	VTS		82.33	82.58	88.00	92.00	93.33	93.92	88.69
non-paramétrique	non-paramétrique		83.78	84.92	88.42	91.25	93.75	94.42	**89.42**

Table 1.2: Précision d'estimation des mots-clés (%) sur la tâche 1 du 2e défi CHiME obtenue par différents estimateurs fusionnés ou non-paramétriques.

positive pondérée comme ci-dessus. Le nombre de noyaux et les pondérations optimales ont été déterminés expérimentalement.

Nous avons analysé les résultats à la fois pour la tâche 1 (petit vocabulaire) et la tâche 2 (vocabulaire moyen) du 2e défi CHiME. L'estimateur non-paramétrique a permis une réduction relative du taux d'erreur de 29% et 28% sur ces deux tâches respectivement par rapport au système de base (sans incertitude). Par rapport à la fusion tardive par ROVER, cette approche apporte une réduction relative de 9% du taux d'erreur sur la tâche 1. Les résultats pour la tâche 1 détaillés dans le tableau 1.2 montrent que la fusion et l'apprentissage non-paramétrique améliorent la performance à la fois dans le domaine spectral et dans le domaine des descripteurs par rapport à Wiener + VTS et que l'approche non-paramétrique fournit les meilleurs résultats dans ces deux domaines.

1.1.4 Estimation de l'incertitude basée sur l'apprentissage discriminant

Bien qu'elle soit meilleure que celle obtenue avec un estimateur classique, la performance de l'approche non-paramétrique reste inférieure à celle potentiellement atteignable avec l'incertitude oracle. Cela s'explique en partie par la nature fortement non-linéaire des descripteurs et par l'hypothèse supplémentaire d'indépendance de l'incertitude entre différents points temps-fréquence. Une autre explication est que l'estimateur non-paramétrique proposé a été entraîné sans tenir compte de l'état du modèle acoustique de parole. Dans un travail antérieur, Delcroix avait proposé d'entraîner une simple fonction affine au sens du maximum de vraisemblance [Delcroix et al., 2013a; Delcroix et al., 2009]. Ces deux approches peuvent être considérées comme sous-optimales car il s'agit d'approches d'apprentissage génératif, où le même poids est appliqué à chaque état du modèle acoustique (qu'il corresponde à un alignement correct ou à un alignement incorrect) durant l'apprentissage. Récemment, l'apprentissage discriminant au sens des critères de maximum d'information mutuelle (MMI) [McDermott et al., 2010] et de *boosted MMI* (bMMI) [Povey et al., 2008; Tachioka et al., 2013a] a été employé avec succès pour la transformation affine de la moyenne et de la matrice de covariance diagonale de l'incertitude dans [Delcroix et al., 2011].

Nous présentons une méthode pour la transformation discriminante de la matrice de covariance pleine de l'incertitude, dépendant ou non de l'état du modèle acoustique. Dans le cas indépendant de l'état, en partant de l'estimateur non-paramétrique obtenu ci-dessus, une transformation non-linéaire g représentée par un ensemble de paramètres $\boldsymbol{\theta}$ est appliquée aux coefficients diagonaux $(\widehat{\boldsymbol{\sigma}}_{\mathbf{c}_n})^2$ de la matrice de covariance pleine $\widehat{\boldsymbol{\Sigma}}_{\mathbf{c}_n}$:

$$(\widehat{\boldsymbol{\sigma}}^{\mathrm{t}}_{\mathbf{c}_n})^2 = g\left((\widehat{\boldsymbol{\sigma}}_{\mathbf{c}_n})^2, \boldsymbol{\theta}\right). \tag{1.4}$$

Par la suite, nous supposons que g est un réseau de neurones.

Le but est de trouver l'ensemble de paramètres $\boldsymbol{\theta}$ qui maximise la probabilité *a posteriori* de la vérité terrain par rapport aux autres séquences de mots possibles. Pour cela, nous maximisons le critère bMMI donné par [Povey et al., 2008]

$$F_{bMMI} = \log\left(\frac{p(\widehat{\boldsymbol{\mu}}_{\mathbf{c}}, \widehat{\boldsymbol{\Lambda}}^{\mathrm{t}}_{\mathbf{c}}|\boldsymbol{q}^*, \boldsymbol{\theta})p(\boldsymbol{q}^*)}{\sum_{\boldsymbol{q}} p(\widehat{\boldsymbol{\mu}}_{\mathbf{c}}, \widehat{\boldsymbol{\Lambda}}^{\mathrm{t}}_{\mathbf{c}}|\boldsymbol{q}, \boldsymbol{\theta})p(\boldsymbol{q})e^{\epsilon A(\boldsymbol{q},\boldsymbol{q}^*)}}\right) \tag{1.5}$$

où $\widehat{\boldsymbol{\mu}}_{\mathbf{c}}$ et $\widehat{\boldsymbol{\Lambda}}^{\mathrm{t}}_{\mathbf{c}}$ sont la moyenne et l'incertitude diagonale transformée, $\boldsymbol{q} = \{q_n\}_{n=1}^N$ est une séquence d'états associée à une séquence de mots possible et $\boldsymbol{q}^*$ est la séquence d'états associée à la bonne séquence de mots. En utilisant la règle de dérivation en chaîne, la dérivée du critère bMMI par rapport à chaque paramètre de la transformation est donnée par

$$\frac{\partial F_{bMMI}}{\partial \theta_k} = \sum_{i,n} \frac{\partial F_{bMMI}}{\partial (\widehat{\sigma}^{\mathrm{t}}_{\mathbf{c}_n})^2_i} \frac{\partial (\widehat{\sigma}^{\mathrm{t}}_{\mathbf{c}_n})^2_i}{\partial \theta_k} \tag{1.6}$$

L'optimisation est effectuée par descente de gradient stochastique [Rumelhart et al., 1986] où le premier terme de l'équation (1.6) fait appel à la dérivée usuelle du critère bMMI et le second terme est calculé par *back-propagation*. Une fois la convergence obtenue, le facteur de correction de l'incertitude avant/après transformation est appliqué de façon heuristique à la matrice de covariance pleine.

La performance de RAP est évaluée dans le tableau 1.3. L'usage d'une transformation non-linéaire entraînée par bMMI réduit le taux d'erreur relatif de 5% supplémentaires par rapport à l'estimateur non-paramétrique entraîné de façon générative.

1.2 Apprentissage de réseaux de neurones

1.2.1 État de l'art

L'émergence des réseaux de neurones profonds comme modèles de parole pour le rehaussement et la RAP nous a amené à étudier le problème de leur apprentissage. Les réseaux de neurones

Méthode	Dépendant	Ensemble de test						
	de l'état	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Moyenne
Wiener + VTS		77.92	80.75	86.75	90.50	92.92	93.75	87.00
Estimation & propagation non-paramétrique		83.78	84.92	88.42	91.25	93.75	94.42	89.42
non-paramétrique + bMMI	non	84.17	85.00	88.75	91.33	93.75	94.42	89.57
[Delcroix et al., 2011]	oui	79.92	82.00	87.17	90.67	92.92	93.42	87.68
non-paramétrique + bMMI	oui	84.75	85.50	89.00	91.75	93.75	95.00	**89.95**

Table 1.3: Précision d'estimation des mots-clés (%) sur la tâche 1 du 2e défi CHiME obtenue par différentes méthodes d'apprentissage discriminant.

profonds sont constitués d'une couche d'entrée, d'une couche de sortie et de plusieurs couches cachées. La première couche prend un vecteur en entrée. Chaque couche est totalement connectée à la suivante et effectue une transformation affine représentée par une matrice de poids et un vecteur de biais, suivie d'une fonction d'activation non-linéaire de type sigmoïde, tangente hyperbolique ou rectification [Zeiler et al., 2013]. Le vecteur de sortie de la dernière couche vise à prédire un vecteur cible. Les paramètres optimaux sont estimés de sorte à minimiser un critère tel que l'erreur quadratique moyenne ou l'entropie croisée entre la sortie et la cible. L'apprentissage est typiquement effectué par descente de gradient stochastique [Rumelhart et al., 1986], par une méthode de gradient adaptatif (ADAGRAD) [Duchi et al., 2011] ou une méthode du second ordre telle que la méthode de Newton [Bordes et al., 2009]. Ces méthodes requièrent la fixation d'un pas d'apprentissage. Lorsque ce pas est trop petit, la convergence est lente. Lorsqu'il est trop grand, l'algorithme devient instable et peut diverger.

1.2.2 Approche proposée basée sur une fonction auxiliaire

Nous introduisons une nouvelle méthode d'apprentissage pour les réseaux de neurones basée sur une fonction auxiliaire qui ne nécessite pas la fixation d'un pas d'apprentissage. L'optimisation basée sur les fonctions auxiliaires [de Leeuw, 1994; Heiser, 1995; Becker et al., 1997; Lange et al., 2000; Hunter and Lange, 2004] est récemment devenue populaire dans d'autres domaines comme illustré par les méthodes de séparation de sources audio HPSS [Ono et al., 2008] et AuxIVA [Ono, 2011], par exemple. Cette technique consiste à introduire une fonction auxiliaire et à optimiser cette fonction plutôt que la fonction de coût originelle. La fonction auxiliaire est plus facile à optimiser, mais sa construction est souvent difficile.

Dans le cas d'un réseau de neurones à N couches, nous introduisons récursivement couche par couche une fonction auxiliaire quadratique dont le minimum est calculable de façon analytique. Par exemple, en ce qui concerne la couche de sortie, nous considérons le cas d'une fonction d'activation tangente hyperbolique et d'une fonction de coût euclidienne. La fonction de coût s'exprime comme

$$\mathbf{E} = \frac{1}{2}\sum_{p=1}^{P}\sum_{i=1}^{k_N}(z_{i,p}^{(N)} - y_{i,p})^2 + \frac{\lambda}{2}\sum_{n=1}^{N}\sum_{i}\sum_{j}(w_{i,j}^{(n)})^2 \tag{1.7}$$

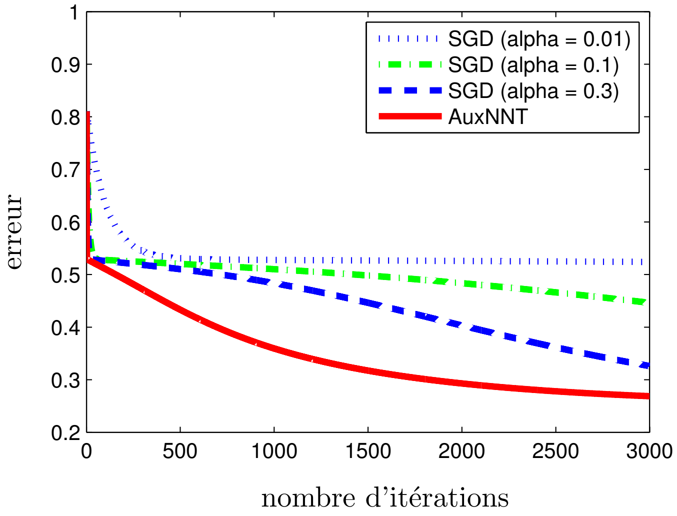
où le premier terme est la distance euclidienne au carré entre les éléments de la sortie $z_{i,p}^{(N)}$ sur la trame p et la cible correspondante $y_{i,p}$ et le deuxième terme est un terme de régularisation des poids $w_{i,j}^{(n)}$ de toutes les couches n qui évite le sur-apprentissage. Nous avons montré que, pour tous x et x_0 positifs et pour tout nombre réel y, l'inégalité suivante est satisfaite (voir les formules pour a, b et c dans le chapitre 9):

$$(\tanh(x) - y)^2 \leq ax^2 - 2bx + c = \mathbf{Q} \tag{1.8}$$

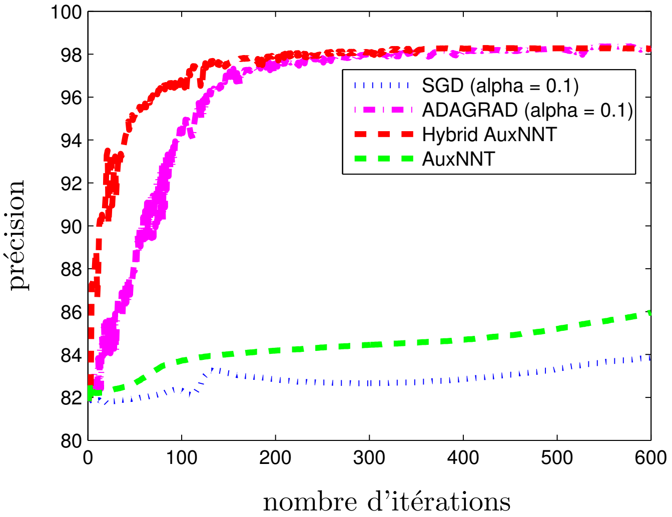
Nous avons aussi montré comment propager cette fonction auxiliaire aux couches inférieures. Compte tenu de la relation d'inégalité entre la fonction de coût et la fonction auxiliaire, la convergence monotone de la fonction de coût au fil des itérations est garantie.

De plus, nous avons proposé une approche hybride qui exploite à la fois les avantages de la fonction auxiliaire et d'ADAGRAD. Plus précisément, lorsque le changement de valeur des paramètres est faible, plusieurs itérations d'ADAGRAD sont effectuées. Nous sélectionnons alors l'itération pour laquelle le gradient est le plus grand et nous continuons l'optimisation avec la méthode basée sur la fonction auxiliaire, jusqu'à ce que le changement devienne à nouveau petit.

Les résultats expérimentaux sur la base d'images MNIST [LeCun et al., 1998] montrent que l'algorithme proposé converge plus rapidement et vers une meilleure solution que la descente de gradient stochastique à la fois pour une tâche d'auto-encodage et une tâche de classification, comme montré dans la figure 1.5. De plus, la combinaison d'ADAGRAD et de la méthode proposée accélère la convergence et augmente la performance de classification par rapport à ADAGRAD seul.



(a) auto-encodeur à 1 couche cachée



(b) classifieur à 2 couches cachées

Figure 1.5: Performance d'apprentissage et de test sur la base MNIST au fil des itérations.

Part I

Introduction

2

Introduction

2.1 Motivation

Hidden Markov Model - Gaussian Mixture Model (GMM - HMM) based automatic speech recognition (ASR) has developed since the eighties [Baker et al., 2009; Rabiner, 1989]. The aim of ASR is to transcribe input speech data into words. Usually, each input raw speech signal segment is encoded into features such as Mel-frequency cepstral coefficients (MFCCs) or perceptual linear predictive coefficients (PLPs). These features discard a large amount of information from waveforms that is considered to be irrelevant. Statistical approaches are then used to learn acoustic and linguistic characteristics from the training data.

Due to the fact that speech is an essential ingredient in communication for human beings, ASR systems have found numerous uses in human-machine interfaces. ASR systems have a wide range of applications including voice control, audio indexing, medical transcription, information retrieval, and speech to text transcription.

So far, ASR systems have mostly been successful in applications for which the speech signal is clean but they are extremely sensitive to changes of acoustic conditions [Acero and Stern, 1990]. This weakness arises especially in far-field or distant microphone conditions [Wölfel and McDonough, 2009; Vincent et al., 2013b]. The task of making ASR systems more robust to the change of acoustic conditions is called robust ASR. The word error rate (WER) deteriorates quickly with noise and reverberation, even though they hardly affect human speech recognition. This is due to several reasons. One of the main reasons is the *mismatch* between training and test data. The *mismatch* caused by channel and speaker variability is beyond the scope of this thesis. In noisy and reverberant conditions, the actual features which are captured in the test environment tend to be blurred or distorted. As a result, the model parameters learned from clean training features are less effective on these features. Another reason is the poor generalization capability of GMM - HMM acoustic modeling. Although with enough components, GMMs can be successfully used to model the probability distribution over the input features associated with each state of an HMM, they are statistically inefficient for modeling features that lie on or near a nonlinear manifold in the data space [Hinton et al., 2012].

2.2 Overview of the problems

Noise and reverberation are two sub-problems that occur in robust ASR. Reverberation appears to be less of a problem and it can be addressed rather well by multi-condition training [Ravanelli and Omologo, 2014; Vincent et al., 2013b] . In this thesis, we focus on the problem of noise robustness. Several approaches have been proposed to address this problem. They can be classified into three categories : front-end, back-end, and hybrid approaches.

Front-end approaches aim to obtain an estimate of clean speech using a speech enhancement technique [Ephraim and Malah, 1984; Boll, 1979]. This allows fast decoding since the speech recognizer is unchanged. The clean speech signal can be estimated either in the spectral domain or in the feature domain. Spectral domain enhancement is often more advantageous because it can utilize both single-channel and multi-channel information [Kolossa et al., 2010; Duong et al., 2010; ?]. One of the disadvantages of this approach is that the estimated clean speech usually contains some distortion. In some cases, this distortion can result in worse ASR performance than without speech enhancement.

By contrast with front-end approaches, back-end approaches [Gales, 1995] aim to reduce the *mismatch* due to speech distortion by updating the parameters of the acoustic model, e.g., GMM-HMM means and variances, so that they fit the input features. Back-end approaches often achieve better performance than front-end approaches. The main disadvantage of these approaches is that they either make the computation cost larger than the front-end approaches or they require a larger amount of data to adapt. Deep neural network (DNN) based acoustic models [Hinton et al., 2012; Li and Sim, 2014] have shown to be more robust to noise when trained in a multi-condition setting. DNNs consist of several layers where several frames of features are taken as inputs and posterior probabilities over the HMM states are produced as outputs. MFCCs or PLPs and their first- and second-order temporal derivatives can be used as features. DNN based acoustic models are usually trained by gradient descent. This results in slow convergence in most cases and this requires tuning of parameters such as the learning rate.

Hybrid methods are a class of approaches that combine front-end and back-end approaches in such a way that both the model parameters and the input features are changed adaptively depending on noise. Hybrid approaches have obtained better performance than the two approaches above [Deng, 2011] . Uncertainty decoding [Deng et al., 2005; Deng, 2011; Astudillo, 2010] is one hybrid method that is the main focus of this thesis. In the framework of uncertainty decoding, speech enhancement is applied to the input noisy signal and the enhanced features are not considered as point estimates anymore but as a Gaussian posterior distribution with time-varying variance. This variance or uncertainty is then used to dynamically adapt the acoustic model on each time frame for decoding. Decoding rules are available in closed form for GMM-HMM models. The uncertainty can be estimated in the spectral domain and then propagated to the feature domain. Most existing spectral-domain uncertainty estimators are fixed by mathematical approximation. Experimentally, they result in inaccurate uncertainty estimation and are still far from the actual distortion between clean and enhanced features.

2.3 Contributions

Motivated by the fact that state-of-the-art uncertainty decoding techniques are not able to accurately estimate uncertainty and by the slow convergence of gradient descent based training for DNNs, we focus in this thesis on solving these two problems. The achieved results have been described in our publications [Tran et al., 2013; Tran et al., 2014a; Tran et al., 2014b; Tran et al., 2015a; Tran et al., 2015b; Tran et al., 2015c]. Our contributions are as follows:

- Contribution to uncertainty handling: we propose a family of learning based methods that allow learning of uncertainty from data in a generative or a discriminative fashion. The proposed methods include: modeling of the full uncertainty covariance matrix, fusion of multiple uncertainty estimators, nonparametric estimation, and discriminative estimation. [Tran et al., 2013; Tran et al., 2014a; Tran et al., 2014b; Tran et al., 2015b; Tran et al., 2015c].

- Contribution to DNN training: we propose a new learning rule that is based on an auxiliary function technique without parameter tuning. Instead of minimizing the objective function, a quadratic auxiliary function is recursively introduced layer by layer which has a closed-form optimum. The monotonic decrease of the new learning rule is proved [Tran et al., 2015a].

2.4 Outline of the thesis

The thesis includes nine chapters organized in four parts:

- Part I: Introduction and overview of robust ASR (Chapters 2 and 3).

- Part II: Uncertainty handling (Chapters 4, 5, 6 and 7).

- Part III: Neural network training (Chapters 8 and 9).

- Part IV: Conclusion and perspectives (Chapter 10).

Chapter 3 presents a literature overview of speech recognition systems, GMM-HMM and DNN based acoustic modeling. It also describes noise robust ASR and some available related datasets.

Chapter 4 presents the state of the art of uncertainty handling in robust ASR, including uncertainty estimation, uncertainty propagation and uncertainty decoding and some of their variants.

Chapter 5 presents the extension of uncertainty propagation to a full uncertainty covariance matrix which is shown to better model uncertainty in the feature domain. A 5% relative word error rate (WER) improvement is reported on Track 1 of the 2nd CHiME Challenge compared to a diagonal uncertainty covariance model.

Chapter 6 proposes a generative learning approach for uncertainty estimation in the spectral domain and in the feature domain. We introduce two possible techniques: linear fusion of multiple existing uncertainty estimators/propagators and nonparametric uncertainty estimation/propagation. We define oracle uncertainties in the spectral and feature domains. The fusion weights and the kernel weights are then learned by minimizing some divergence between the oracle uncertainties and the estimated uncertainties. We show that the minimization problem is an instance of weighted nonnegative matrix factorization (NMF) and we solve it using multiplicative update rules. Experimentally, fusion and nonparametric uncertainty estimation lead to 28% and 29% relative WER improvement compared to conventional decoding, respectively.

Chapter 7 presents a discriminative learning based uncertainty estimator. A gradient based learning rule is derived based on a boosted Maximum Mutual Information (bMMI) objective function. This leads to a 5% relative word error rate reduction compared to the generative nonparametric estimator in Chapter 5.

Chapter 8 presents the state of the art of several neural network architectures including multi-layer perceptrons, convolutional neural networks and recurrent neural networks. Some back-propagation based optimization methods are also revisited in the rest this chapter including stochastic gradient descent (SGD), adaptive gradient (ADAGRAD) descent [Duchi et al., 2011], and second order methods.

Chapter 9 investigates an alternative auxiliary function based optimization rule. Instead of minimizing the objective function, a quadratic auxiliary function is proposed and minimized. Experimental results on the MNIST dataset [LeCun et al., 1998] show that the proposed algorithm converges faster and to a better solution than SGD. In addition, we found a combination of ADAGRAD and the proposed method to accelerate convergence and to achieve better performance than ADAGRAD alone.

Chapter 10 ends the thesis by summarizing the conclusions and presenting future research directions.

3

Overview of robust ASR

The first part of this chapter presents a literature overview of a speech recognition systems including feature extraction, GMM-HMM and DNN based acoustic modeling, lexicon, language modeling, and decoding. The second part describes noise robust ASR and some available related datasets.

3.1 Overview of speech recognition systems

The task of an ASR system is to recognize the word sequence given the speech waveform. A block diagram of a statistical speech recognition system is shown in Figure 3.1. The speech signal captured from a microphone is first converted into a stream of acoustic features by a *feature extraction* module. Secondly, given a sequence of observed features, the recognizer decodes these features using the knowledge obtained from *acoustic and language models*, and a dictionary or *lexicon*, to produce the best hypothesis for the recognized words. A statistical speech recognition system generally finds the most probable word sequence or hypothesis $\mathcal{W}$ for a given sequence of observed features $\mathbf{c}_{1:N}$ by maximizing the posterior probability $P(\mathcal{W}|\mathbf{c}_{1:N})$. This can be

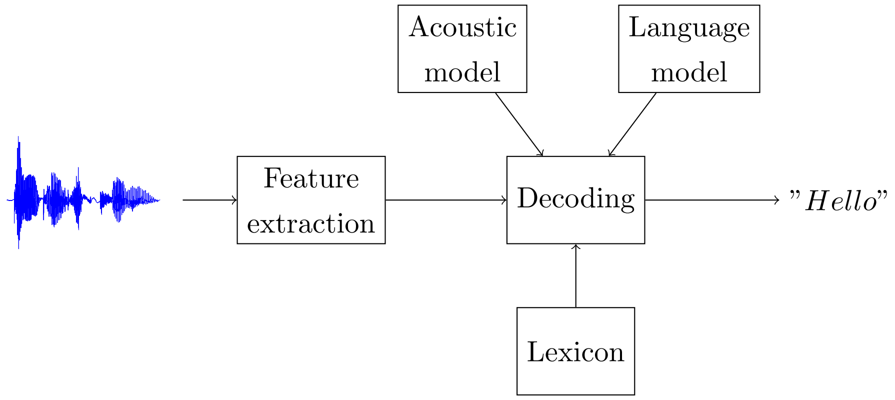


Figure 3.1: Block diagram of an automatic speech recognition system.

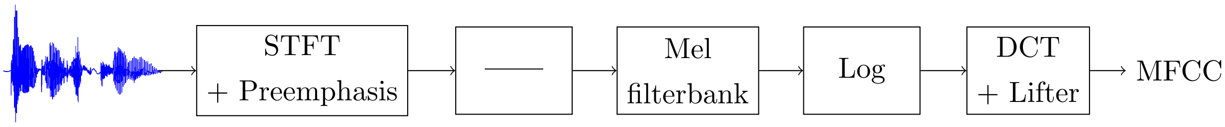


Figure 3.2: Feature extraction based on Mel-frequency cepstral coefficients.

expressed as

$$\hat{\mathcal{W}} = \arg\max_{\mathcal{W}} P(\mathcal{W}|\mathbf{c}_{1:N}). \tag{3.1}$$

Using Bayes's rule for conditional probabilities and noting that, as $\mathbf{c}_{1:N}$ is the observation the term $p(\mathbf{c}_{1:N})$ is constant and does not affect the optimization, we obtain the well-known decoding rule

$$\hat{\mathcal{W}} = \arg\max_{\mathcal{W}} P(\mathbf{c}_{1:N}|\mathcal{W})P(\mathcal{W}) \tag{3.2}$$

where $P(\mathcal{W})$ is the prior probability that the word sequence $\mathcal{W}$ is spoken as defined by the *language model* and $P(\mathbf{c}_{1:N}|\mathcal{W})$ is the likelihood that the sequence of features is observed given the sequence of words $\mathcal{W}$ as defined by the *acoustic model*.

3.1.1 Pre-processing and feature extraction

Mel-frequency Cepstral Coefficients (MFCC) [Davis and Mermelstein, 1980] and Perceptual Linear Prediction (PLP) coefficients [Hermansky, 1990] are commonly used as features in state-of-the-art speech recognition systems. More sophisticated feature representations such as tandem [Hermansky, 2000] and bottleneck [Grezl et al., 2007] can also be used. In this thesis, we focus on the case of MFCC features only but this work could also be extended to, e.g., PLP coefficients. The speech signal is captured by one or multiple microphones and it is converted to digital format. It is usually sampled at 8 or 16 kHz. Features are extracted in several successive steps, as illustrated in Figure 3.2:

- Signal samples are concatenated into frames of 25 ms. This is due to the fact that the vocal tract is considered to be quasi-stationary in each frame. The frames are shifted by 10 ms to increase the smoothness between consecutive frames. Furthermore, in order to avoid discontinuities at the window boundaries, a smoothing window such as the Hamming window [Rabiner and Levinson, 1981] is commonly applied [Rabiner, 1989]. The discrete Fourier transform (DFT) is applied to compute the complex-valued spectrum in each

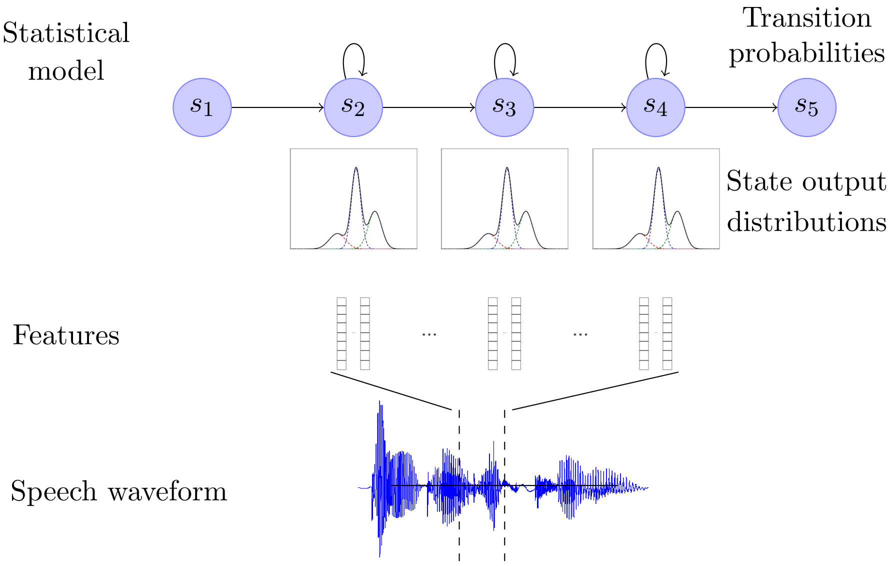


Figure 3.3: HMM acoustic model.

frame. Note that all steps up to here correspond to the Short Time Fourier Transform (STFT). The spectrum is then passed through a pre-emphasis filter to increase the energy of the signal at higher frequencies.

- Then the magnitude spectrum or the power spectrum is derived.
- The magnitude spectrum or the power spectrum is passed through a filter bank whose bins are spaced according to the Mel scale [Davis and Mermelstein, 1980], which approximates the frequency response of the human ear.
- The output of the filter bank is transformed into a logarithmic scale.
- A discrete cosine transform (DCT) is applied to obtain the MFCC observation vectors. The DCT achieves decorrelation so that the feature vectors become compatible with the diagonal covariance matrix assumption used in GMM-HMMs. In the end, the cepstral coefficients are passed through a lifter filter.

3.1.2 Acoustic modeling

Hidden Markov models are the most popular and successful acoustic models used in state-of-the-art speech recognition systems [Rabiner and Juang, 1993; Gales and Young, 2008]. Basically, an HMM is a finite-state machine where each state is associated with an output probability distribution, as shown in Figure 3.3. At each time, when a state is entered according to the transition probabilities between the states, the HMM generates an observation according to the state output distribution. Only the observation can be seen while the state sequence is hidden.

An HMM is usually employed to model one acoustic unit such as a word or a phone. The HMM shown in Figure 3.3 is a left-to-right HMM that is widely used in speech recognition to model a phone. The first and last node represent entry and exit non-emitting states, whereas the other nodes represent states with associated output probability distributions. The connecting arrows between nodes and the self loops represent the allowed transitions between states and have associated probabilities. The output probability distribution $p(\mathbf{c}_n|q)$ for each state q is either represented as a multivariate Gaussian mixture model (GMM) or it can be obtained from the output of a neural network [Bourlard and Wellekens, 1990; Bourlard et al., 1992; Renals et al., 1994; Dahl et al., 2012].

Regarding the GMM model, the ouput probability distribution is specified by

$$p(\mathbf{c}_n|q) = \sum_{m=1}^{M_q} \omega_{qm} \mathcal{N}(\mathbf{c}_n; \boldsymbol{\mu}_{qm}, \boldsymbol{\Sigma}_{qm}) \tag{3.3}$$

where M_q is the number of mixture components, ω_{qm} is the weight of the m_{th} component such that $\sum_{m=1}^{M_q} \omega_{qm} = 1$, and $\boldsymbol{\mu}_{qm}, \boldsymbol{\Sigma}_{qm}$ are the mean and the covariance matrix of the m_{th} component, respectively. The likelihood of the observation sequence $\mathbf{c}_{1:N}$ given the hypothesis $\mathcal{W}$ can be computed using the forward-backward algorithm. All parameters of GMM-HMM models are estimated by maximizing the likelihood of the training data. This can be done by using an *expectation maximization (EM) algorithm* called the *Baum-Welch algorithm.*

Another approach is to use a neural network to model $p(\mathbf{c}_n|q)$ in a hybrid artificial neural network - hidden Markov model (ANN-HMM) structure [Bourlard et al., 1992]. In this case the likelihood of the observation $\mathbf{c}_n$ given state q is broken into two terms $p(q|\mathbf{c}_n)$ and $p(q)$ as

$$p(\mathbf{c}_n|q) \propto \frac{p(q|\mathbf{c}_n)}{p(q)}. \tag{3.4}$$

The ANN aims to predict the state posterior probability $p(q|\mathbf{c}_n)$ given the input features.

Recently, a new hybrid structure between a pre-trained DNN and a context-dependent hidden Markov model space (CD-DNN-HMM) [Hinton et al., 2012] was proposed where the output of a deep neural network (DNN) predicts the log-posterior $\log p(q|\mathbf{c}_n)$. Several consecutive frames are concatenated into a long vector and they are considered as the inputs of the DNN. The log-likelihood of the observation is computed as

$$\log p(\mathbf{c}_n|q) = \log p(q|\mathbf{c}_n) - \log p(q). \tag{3.5}$$

The state priors $\log p(q)$ in equation (3.4) or $p(q)$ in equation (3.5) can be estimated using the state alignments on the training data.

3.1.3 Lexicon and language modeling

A lexicon is one of the modules of a speech recognition system, as shown in Figure 3.1. It defines the allowed vocabulary set for speech recognition and provides one or more pronunciations for each word. Pronunciations are usually given at the phone level. The various pronunciation forms of a word are usually considered as different entries in the lexicon. Speech recognition tasks are often categorized according to their vocabulary size: small vocabulary tasks (less than 10^3 words), medium vocabulary tasks ($10^3 - 10^4$ words), and large vocabulary tasks (more than 10^4 words).

A language model (LM) is also used in speech recognition systems as shown in Figure 3.1. The language model gives the probability of an hypothesis $\mathcal{W} = \mathcal{W}_1, ..., \mathcal{W}_K$ constituting a sequence of words. It represents syntactic and semantic information in the word sequence. The probability $P(\mathcal{W})$ can be computed by using the chain rule as a product of conditional probabilities

$$P(\mathcal{W}) = \prod_{k=1}^{K} P(\mathcal{W}_k | \mathcal{W}_{k-1}, ..., \mathcal{W}_1). \tag{3.6}$$

This considers the full word history which is not very practical. In continuous speech recognition tasks, the vocabulary is too large to allow robust estimation of $P(\mathcal{W})$. One solution is to restrict the history to the preceding $N-1$ words only. This is referred to as an N-gram language model. Typical values are $N = 2$, 3 or 4 which are called bi-, tri-, or four-gram models respectively.

3.1.4 Decoding

To find the most likely word sequence, the state sequence $q_{1:N}$ should be marginalized out:

$$\hat{\mathcal{W}} = \arg\max_{\mathcal{W}} P(\mathcal{W}) \sum_{q_{1:N}} P(\mathbf{c}_{1:N} | q_{1:N}) P(q_{1:N} | \mathcal{W}). \tag{3.7}$$

However, this marginalization turns out to be computationally intractable. Therefore, the sum is replaced by a maximum. Speech recognizers find the word sequence corresponding to the best state sequence

$$\hat{\mathcal{W}} = \arg\max_{\mathcal{W}} P(\mathcal{W}) \max_{q_{1:N}} P(\mathbf{c}_{1:N} | q_{1:N}) P(q_{1:N} | \mathcal{W}). \tag{3.8}$$

This sequence can be computed by the *Viterbi algorithm* [Viterbi, 1967] which is a dynamic programming algorithm.

3.2 Overview of noise-robust ASR

ASR performance dramatically degrades in noisy environments. This is due to the mismatch between training and test data. Many approaches have been proposed to overcome this problem. They can be classified into three categories : front-end, back-end, and hybrid approaches, as illustrated in Figure 3.4.

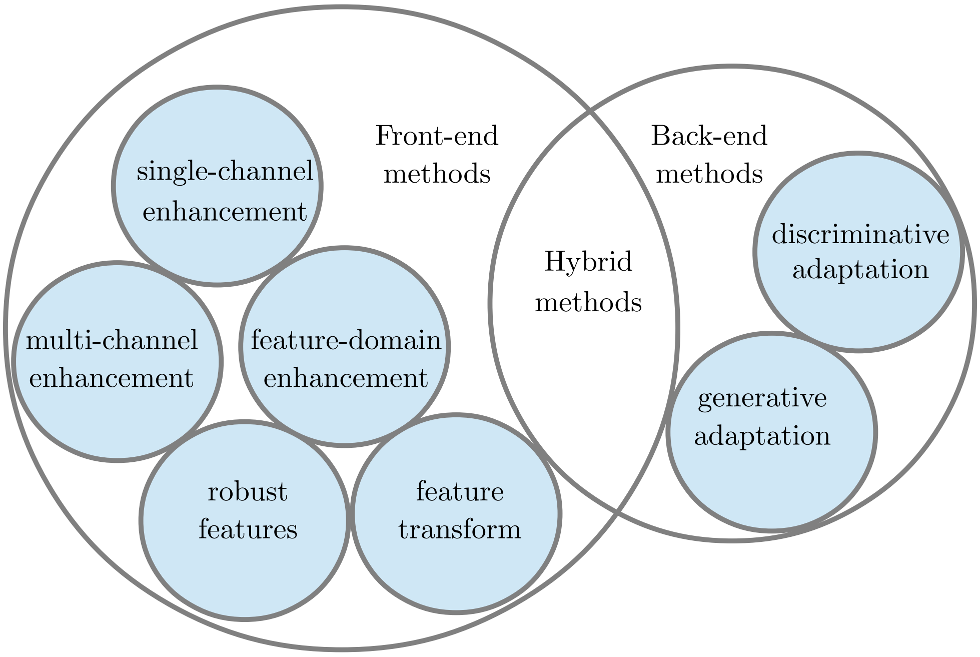


Figure 3.4: Categorization of noise robust ASR methods.

3.2.1 Front-end approaches

Front-end approaches aim to obtain an estimate of clean speech using a speech enhancement technique. This can be done in the spectral domain, or in the feature domain. These approaches utilize either a single microphone or multiple microphones.

Single-channel enhancement

The single-microphone trend started from classical methods such as spectral subtraction [Boll, 1979; McAulay and Malpass, 1980; Ephraim and Malah, 1984] since the 1980s. These methods are based on probabilistic models of speech and noise. The power spectrum of noise is estimated first, then the enhanced speech signal is derived based on a maximum likelihood (ML) or a minimum mean square error (MMSE) criterion. The more accurate the noise estimation, the better the estimation of the enhanced speech signal. These methods often generate some distortion such as *musical noise*. This distortion mainly comes from the fact that it is extremely hard to quickly update the noise spectrum in a non-stationary environment. Improved noise estimation methods were proposed based on minimum statistics [Martin, 2003] and on improved minima controlled recursive averaging [Cohen, 2003]. They were shown to be quite successful with non-stationary noise. However, if speech enhancement is applied only to the test data, the mismatch does not decrease in many cases due to the nonlinear relationship between the spectral domain and the feature domain. When speech enhancement is done in the log-spectral domain [Ephraim and Malah, 1985] instead of the spectral domain, this leads to better mismatch reduction.

Recently, non-negative matrix factorization (NMF) [Lee and Seung, 1999] arised as a successful method to model the spectral characteristics of highly non-stationary speech and noise signals. Firstly, the spectral characteristics of speech and noise are learned from clean speech and noise data in the form of *basis spectra* or *examplar*s [Smaragdis, 2007; Hurmalainen et al., 2011; Gemmeke et al., 2011; Arberet et al., 2010]. Then, given the noisy data, the algorithm infers the hidden clean speech in the ML sense. Learning can be done in a supervised or an unsupervised fashion by minimizing some divergence such as Kulback-Leibler (KL), Itakura-Saito (IS), or squared Euclidean (EUC). NMF techniques can provide a big improvement compared to classical methods especially in a highly non-stationary noise environments [Févotte et al., 2013; Wilson et al., 2008].

Multi-channel enhancement

A complementary way to exploit more information about the captured signal is to use more than one microphone. Multichannel enhancement techniques such as beamforming can take benefit from spatial information [Trees, 2002; Flanagan et al., 1993; Gannot et al., 2001; Doclo and Moonen, 2002]. The direction of arrival of the desired source is assumed to be known in advance. The simplest method is called delay-and-sum beamforming where the signals received by the microphone array are time-aligned with respect to each other and subsequently summed together. One extension of delay-and-sum beamforming is filter-and-sum beamforming where the signals received by the microphone array are filtered before they are summed together. There are many ways to estimate the associated filters using criteria such as minimizing the signal error, maximizing the SNR, improving the perceptual quality as judged by human listeners, maximizing the kurtosis of the output [Kumatani et al., 2012], maximizing the likelihood of the correct ASR hypothesis [Seltzer et al., 2004], or combining beamforming with independent component analysis [Baumann et al., 2003].

Similarly to beamforming, blind source separation techniques assume that two or more sources are typically mixed in the signals reaching the microphones. The desired speech source is to be separated from the other sources such as noise or interfering speakers. This class of techniques is known as *blind* because neither the relative positions of the microphones nor the directions of arrival of the sources are assumed to be known. Separation typically relies on the nongaussianity of the source signals. Measures of nongaussianity are often referred to as *contrast function*s [Comon, 1994]. Several optimization criteria that are typically applied in blind source separation include mutual information [Bell and Sejnowski, 1995], maximum negentropy [Comon, 1994], and maximum likelihood [Cardoso, 1997; Févotte and Cardoso, 2005].

Multichannel NMF [Arberet et al., 2010; Ozerov et al., 2012; Ozerov and Févotte, 2010] combines the benefits of spectral and spatial information. Each source is given a model derived from NMF with the IS divergence, which underlies a statistical model of superimposed Gaussian components. Optimization relies on the joint likelihood of the multichannel data which is maximized using a combination of expectation maximization (EM) updates and multiplicative updates inspired from single-channel NMF [Ozerov and Févotte, 2010]. Multichannel NMF was implemented in particular in the Flexible Audio Source Separation Toolbox (FASST) [Ozerov et al., 2012].

Feature domain enhancement

Feature domain enhancement techniques include stereo-based piecewise linear compensation for environments (SPLICE) [Deng et al., 2000], vector Taylor series (VTS) [Moreno et al., 1996] and the Algonquin method [Frey et al., 2001]. In SPLICE, the noisy features are modeled by a mixture of Gaussian components. In each Gaussian component, a linear relationship is assumed between the noisy speech features and the conditional mean of the clean speech features. An estimate of the clean speech features is derived in the maximum a posterior (MAP) sense. In VTS [Moreno et al., 1996] and Algonquin [Frey et al., 2001], the nonlinear relationship between clean speech, noise, and noisy speech is introduced in the feature domain and the distribution of noisy speech given the distribution of clean speech and noise is computed. The vector Taylor series approximation of this nonlinear function is derived and the MMSE estimate of clean speech given the observed noisy speech is obtained.

Robust features

As a complement or an alternative to speech enhancement, some techniques extract other features than MFCC or PLP, that are more robust to variation of environment. In [Greenberg and Kingsbury, 1997; Kim and Stern, 2009], the input speech signal is processed independently in each frequency band, and the filter bands are designed according to the human ear. In [Greenberg and Kingsbury, 1997], the new representation of speech discards much of the spectro-temporal detail, and focuses on the underlying, stable structure incorporated in the low-frequency portion of the modulation spectrum distributed across critical bands. In [Kim and Stern, 2009], based on the assumption that the noise power changes more slowly than the speech power, a power-bias subtraction technique is applied in each frequency band to remove the slowly varying part of the signal and to make the features less sensitive to noise.

The use of neural networks to extract new features was also studied in the past including tandem features [Hermansky, 2000] and bottleneck features [Grezl et al., 2007]. Tandem features are probabilistic features which are extracted from consecutive feature frames by using a one hidden layer neural network. The neural network is designed by concatenating consecutive feature frames as inputs and considering the log-posterior probabilities of acoustic states as outputs where each element of the output vector corresponds to one phone. The nonlinear activation of the output layer is omitted and replaced by the softmax function. Principal component analysis (PCA) decorrelation is usually applied in order to reduce the dimension of the feature vectors.

The neural network is trained by using the backpropagation algorithm. Bottleneck features [Grezl et al., 2007] are also probabilistic features. The main difference compared to tandem features is that bottleneck features are generated from a multi-layer perceptron with more hidden layers in which one of the internal layers (the one which produces bottleneck features) has a small number of hidden units, relative to the size of the other layers. In this context, bottleneck features provide compressed features rather than a vector of log-posterior probabilities.

Feature transforms

Feature transform approaches have also been proposed to reduce the mismatch due to speech distortion. These approaches operate directly in the feature domain. Cepstral mean and variance normalization [Viikki and Laurila, 1998] are among the first transform methods which were proposed to reduce the variability of the features by equalizing their moments across all environments. More sophisticated transforms were proposed which provide more flexibility by generatively or discriminatively adapting the transform to the environment by feature-space maximum likelihood linear regression (fMLLR) [Gales, 1998] or linear discriminant analysis (LDA) [Häb-Umbach and Ney, 1992]. Such linear transforms greatly increase robustness to channel mismatch, but less so to reverberation and noise which result in nonlinear distortion of the features. Discriminative nonlinear transforms such as feature-space minimum phone error (fMPE) [Povey et al., 2005] have also shown to be effective [Tachioka et al., 2013b].

3.2.2 Back-end approaches

The goal of back-end approaches is to reduce the mismatch due to speech distortion by updating the parameters of the acoustic model, e.g., GMM-HMM means and variances, so that they fit the input features.

Generative adaptation

Multi-condition or multi-style training is the simplest such technique which consists of learning the model parameters from noisy data. To obtain a robust system, many types of noise are included in the training data in the hope that they will cover as many noises as possible in the test data. As a result, a large training set is needed and the trained GMM distributions are broader, so that the model is less discriminative [Deng et al., 2000].

The model can also be adapted to the test conditions in different ways depending on the availability of transcripts for the adaptation data (supervised or unsupervised) and the time when the adaptation data becomes available (offline or online). A straightforward way to adapt a model given some adaptation data would be to retrain the model using the ML criterion. However, since the amount of adaptation data is small, this leads to overfitting. One way to avoid overfitting is to apply simple transformations to the acoustic model parameters such as linear transformations. Maximum likelihood linear regression (MLLR) [Leggetter and Woodland, 1995; Gales, 1998] has been widely used to adapt the mean and/or the covariance of GMM observation densities to the environment. Also, a maximum a posteriori (MAP) approach [Gauvain and Lee, 1994] was proposed in which the model parameters are updated with respect to the ML estimate and the prior distribution of the model parameters. As additional data becomes available, the MAP estimate tends towards the ML estimate.

Parallel model combination (PMC) combines separate noise and speech models to form a corrupted speech model directly for use in the recognition process [Gales and Young, 1996]. Given the input speech models and a noise model, the *mismatch function* is derived in the linear spectral or log-spectral domain then the model parameters of noisy speech are estimated. PMC assumes that the features of both speech and noise follow a Gaussian distribution.

As shown in the PMC method, deriving the distribution of noisy speech given the clean speech model and the noise model is not straightforward. This is due to the nonlinear effect of noise on speech features. VTS [Acero et al., 2000] was proposed as a model compensation approach. VTS linearizes the noisy speech features with a truncated first-order vector Taylor expansion to individually update each model component. So, given the parameters of the input speech model and the noise model, the mean and variances of the GMM model of noisy speech are updated. Similar to PMC, these update formulas assume that a Gaussian clean speech component which is corrupted by noise may be approximated by another Gaussian distribution.

Discriminative adaptation

ML and MAP transformations of HMMs maximize the likelihood or the posterior of the data given the reference transcripts. This leads to models with poor discrimination ability as ML and MAP adaptation do not consider competing hypotheses. In other words, maximizing the likelihood of data is not closely to related minimizing the word error rate in the ASR task. A number of approaches for discriminative adaptation have been investigated using linear transforms to adapt Gaussian means and covariances of GMM models. These transforms are estimated using discriminative criteria such as maximum mutual information (MMI), or minimum phone error (MPE) [Povey and Woodland, 2002; Povey, 2005].

3.2.3 Hybrid approaches

Hybrid compensation approaches consist of using both feature based compensation and model based compensation. There are many possible ways to combine these two strategies.

Noise adaptive training is a hybrid strategy that combines noise reduction and multi-style training. First, a GMM-HMM model is pretrained using noisy speech with many types and levels of noise. Noise reduction is applied on the noisy speech to produce ”pseudo-clean” speech. Given enhanced or ”pseudo-clean” training data for each noise type and noise level, the parameters of the model are updated in the ML sense over the entire training data. This optimization is solved by an EM algorithm where the noise type and the noise level are treated as hidden variables [Deng et al., 2000] and the log-likelihood of the entire training set is considered.

The missing data technique [Cooke et al., 2001] can be considered as a hybrid technique. In this technique, the features are classified into two classes: reliable and unreliable. To determine whether a spectral feature is reliable or unreliable, there are two simple heuristics. The first heuristic is that the spectral features which are negative after spectral subtraction are considered as unreliable, and the other features are considered as reliable. The second heuristic consists of using the estimated noise spectrum to derive the local SNR then a threshold is applied to determine reliable features. In the speech recognizer, there are also two ways to treat such features. In the first way, the posterior probability of each state is computed by relying on reliable feature only. In the second way, which is called state-based data imputation, the unreliable features are estimated independently for each state based on the correlation between reliable and unreliable features.

Uncertainty decoding [Deng et al., 2005; Delcroix et al., 2009; Kolossa et al., 2010; Astudillo, 2010; Delcroix et al., 2013a; Astudillo et al., 2013; Nesta et al., 2013] has emerged as a promising hybrid technique whereby speech enhancement is applied to the input noisy signal and the enhanced features are not considered as point estimates but as a *Gaussian posterior distribution with time-varying variance*. This variance or *uncertainty* is then used to dynamically adapt the acoustic model on each time frame for decoding. Decoding rules are available in closed form for GMM-HMMs [Deng et al., 2005]. The uncertainty is considered as the variance of speech distortion. It is derived from a parametric model of speech distortion accounting for additive noise or reverberation and it can be computed directly in the feature domain in which ASR operates [Deng, 2011; Krueger and Haeb-Umbach, 2013; Delcroix et al., 2013b; Liao, 2007; Delcroix et al., 2009] or estimated in the spectral domain then propagated to the feature domain [Kolossa et al., 2010; Astudillo and Orglmeister, 2013; Ozerov et al., 2013; Astudillo, 2010; Srinivasan and Wang, 2007; Kallasjoki et al., 2011; Nesta et al., 2013]. The latter approach typically performs best, as it allows speech enhancement to benefit from multichannel information in the spectral domain.

3.3 Noise-robust ASR datasets

This section presents some speech datasets that can be used for noise robust ASR. A detailed list can be found in [Le Roux and Vincent, 2014].

3.3.1 Categorization

Noise-robust ASR datasets exist for a variety of scenarios, including car, outdoor, domestic, meeting, or office scenarios. These datasets can be categorized according to various attributes. A critical attribute regarding speech enhancement is the number of available microphones. The speech material varies from small-vocabulary read speech (e.g., digits) to large-vocabulary spontaneous speech, and the speakers may be seated in a given position or move across the room. The noise background may be stationary (e.g., air-conditioning, car noise) or nonstationary (e.g., meeting noises, domestic noises) and its level may be low or high. Both reverberation and noise may be simulated or recorded live.

In this thesis, we decided to focus on multichannel, distant microphone recordings involving strong, realistic, nonstationary noise. The list of relevant datasets is then restricted to CHiME [Vincent et al., 2013b], DIRHA [Cristoforetti et al., 2014], DICIT dataset [Brutti et al., 2008], CU-Move dataset [Hansen et al., 2001].

In the rest of this thesis, all experiments are conducted on the CHiME dataset. This dataset consists of two subsets of data corresponding to small vocabulary and medium vocabulary sets. It is detailed in the following Section 3.3.2.

3.3.2 CHiME dataset

The second CHiME Challenge [Vincent et al., 2013b] considered the problem of recognizing voice commands in a domestic environment from recording made using a binaural manikin. There are two tracks: Track 1 for small vocabulary and Track 2 for medium vocabulary. Each track includes three subsets: training, development, and test sets. The utterances are read by speakers and mixed with real domestic background noise as illustrated in Figure 3.5. This figure was taken from the challenge website [1]. The noise included nonstationary noise such as doors opening, appliances being turned on or off, footsteps, and doors banging. These nonstationary noises can change abruptly and are unpredictable. There may be multiple speakers in the room producing overlapping speech; the positions of the competing sound sources can change over time, etc. The recording settings are shown in Figure 3.6.

Due to the limitation of space, the results on the test set will be reported in the main body of this thesis while the full results on both development and test sets can be found in Appendix A.

Track 1: Small vocabulary

The target utterances of Track 1 are taken from the small-vocabulary Grid corpus [Cooke et al., 2006]. Speech consists of 6-word utterances of the form <command> <color> <preposition> <letter> <digit> <adverb>, for example *bin blue at f two now*. The utterances are read by 34 speakers and mixed with real domestic background noise at 6 different SNRs. The task is to report the letter and digit keywords and performance is measured by keyword accuracy. The training set contains 500 noiseless reverberated utterances corresponding to 0.14 h per speaker. The development set and the test set each contain 600 utterances corresponding to 0.16 h per

[1] http://spandh.dcs.shef.ac.uk/projects/chime/PCC/introduction.html

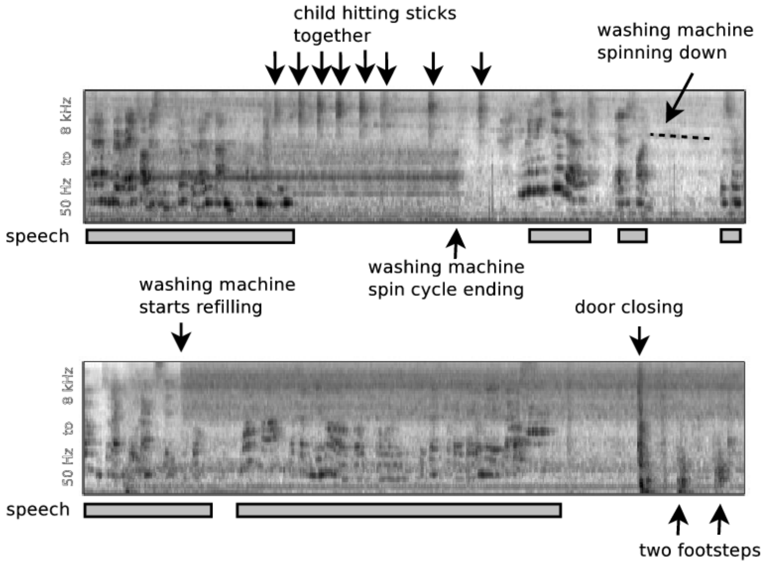


Figure 3.5: Twenty second segment of domestic living room data that forms the background for the PASCAL 'CHiME' challenge.

SNR [Tran et al., 2014a]. The target speaker is allowed to make small head movements within a square zone of +/- 10 cm around a position at 2 m directly in front of the manikin. To simulate the movement, the clean utterances were convolved with time-varying binaural room impulse responses (BRIR).

Track 2: Medium vocabulary

In addition to the above experiments, we evaluated the ASR performance achieved on Track 2 of the challenge. The main difference concerns the vocabulary size. Track 2 is based on the 5000-word Wall Street Journal (WSJ) corpus [Garofalo et al., 2007], which was mixed with real domestic background noise at 6 different SNRs similarly to Track 1. The task is to transcribe the whole utterance and performance is measured in terms of WER. The training set contains 7138 noiseless utterances from 83 speakers totaling 15 hours. The development set contains 410 utterances from 10 speakers, each mixed at 6 different SNRs, totaling 5. The test set contains 330 utterances from 8 speakers, each mixed at 6 different SNRs, totaling 4. The noise properties are similar in all sets, however the noise signals are different.

3.4 Summary

This chapter has reviewed HMM-based automatic speech recognition systems, with a brief description of each module: acoustic model, language model, lexicon, and decoding. To make the system more robust to mismatched conditions, several approaches were proposed including: front-end approaches, back-end approaches, and hybrid approaches. These techniques utilize

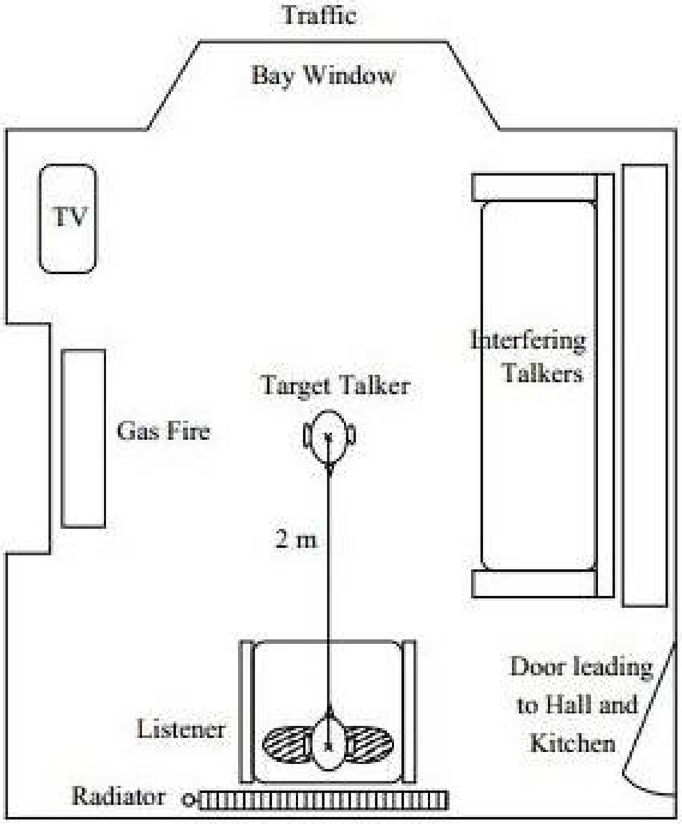


Figure 3.6: CHiME recording settings, taken from [Barker et al., 2013]

either a single microphone or multiple microphones. Hybrid techniques with multiple microphones appear to be the best choice for noise robust ASR and these techniques will be the focus of this thesis. We will focus on MFCC features rather than other features or feature transformation. With several microphones, the system can exploit not only the spectral information of the target source but also the spatial information, which results in better separation in the front-end. Beyond that, with hybrid approaches, not only the features are compensated but also the model parameters are modified resulting in greater mismatch reduction. There are still some open questions which need more investigation. In the next part, I will present the state of the art of uncertainty handling and some proposals to improve its performance.

Part II

Uncertainty handling

4

State of the art

This chapter presents a literature review of existing uncertainty handling approaches. The general framework of uncertainty handling is described in Section 4.1. A brief overview of multichannel source separation is presented in Section 4.2. State-of-the art uncertainty estimation and uncertainty propagation methods are then presented in Sections 4.3 and 4.4. Section 4.5 summarires various forms of uncertainty decoding.

4.1 General framework

Traditional front-end approaches usually aim to obtain a point estimate of the underlying clean speech features, while back-end approaches try to obtain a point estimate of the model parameters. If these estimates are perfect then the overall system will work well. However, this is not true in practice. Recently, uncertainty decoding has emerged as a promising hybrid technique whereby not only a point estimate but the posterior distribution of the clean features given the observed features is estimated and dynamically applied to modify the model parameters in the decoding rule [Kolossa and Haeb-Umbach, 2011], [Droppo et al., 2002], [Deng, 2011], [Deng et al., 2005], [Ion and Haeb-Umbach, 2006].

The uncertainty is considered as the variance of the residual speech distortion after enhancement. It is derived from a parametric model of speech distortion accounting for additive noise or reverberation and it can be computed directly in the feature domain in which ASR operates [Deng, 2011], [Krueger and Haeb-Umbach, 2013], [Delcroix et al., 2013b], [Liao, 2007], [Delcroix et al., 2009] or estimated in the spectral domain then propagated to the feature domain [Kolossa et al., 2010; Astudillo and Orglmeister, 2013; Ozerov et al., 2013; Astudillo, 2010; Srinivasan and Wang, 2007; Kallasjoki et al., 2011; Nesta et al., 2013]. The latter approach typically performs better, as it allows speech enhancement to benefit from multichannel information in the spectral domain.

Figure 4.1 shows the general schema of uncertainty handling including uncertainty estimation in the spectral domain, uncertainty propagation to the feature domain, and uncertainty

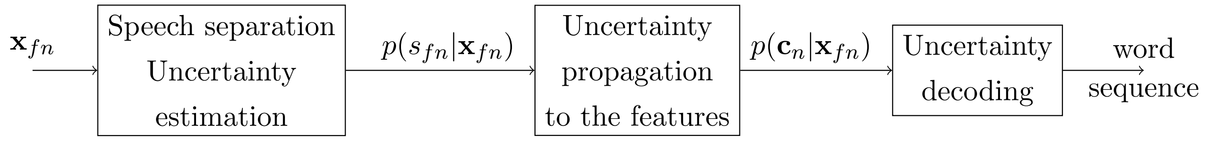


Figure 4.1: Schematic diagram of the state-of-the-art uncertainty handling framework.

decoding with the acoustic model.

4.2 Multichannel source separation

In the multichannel case, let us consider a mixture of J speech and noise sources recorded by I microphones. In the complex short-time Fourier transform (STFT) domain, the observed multichannel signal $\mathbf{x}_{fn}$ can be modeled as [Ozerov et al., 2012]

$$\mathbf{x}_{fn} = \sum_{j=1}^{J} \mathbf{y}_{jfn} \tag{4.1}$$

where $\mathbf{y}_{jfn}$ is the so-called spatial image of the j-th source, and f and n are the frequency index and the frame index, respectively. Each source image is assumed to follow a zero-mean complex-valued Gaussian model

$$p(\mathbf{y}_{jfn}) = \mathcal{N}(\mathbf{y}_{jfn}; \mathbf{0}, v_{jfn}\mathbf{R}_{jf}) \tag{4.2}$$

whose parameters v_{jfn} and $\mathbf{R}_{jf}$ are the short-term power spectrum and the spatial covariance matrix of the source, respectively, which may be estimated using a number of alternative speech enhancement techniques [Kolossa et al., 2010; Ozerov et al., 2012; Nesta et al., 2013]. Once estimated, these parameters are used to derive an estimate of the target speech source by multichannel Wiener filtering

$$\widehat{\boldsymbol{\mu}}_{\mathbf{y}_{jfn}} = \mathbf{W}_{jfn}\mathbf{x}_{fn} \tag{4.3}$$

with

$$\mathbf{W}_{jfn} = v_{jfn}\mathbf{R}_{jf} \left(\sum_{j'} v_{j'fn}\mathbf{R}_{j'f} \right)^{-1}. \tag{4.4}$$

The source spatial image is then downmixed into a single-channel source signal estimate $\widehat{\mu}_{s_{jfn}}$ as

$$\widehat{\mu}_{s_{jfn}} = \mathbf{u}_f^H \widehat{\boldsymbol{\mu}}_{\mathbf{y}_{jfn}} \tag{4.5}$$

where $\mathbf{u}_f$ is a steering vector pointing to the source direction and ^{H} denotes conjugate transposition. In the context of the 2nd CHiME challenge [Vincent et al., 2013a], $I = 2$ and $\mathbf{u}_f^H = [0.5\ 0.5]$ for all f.

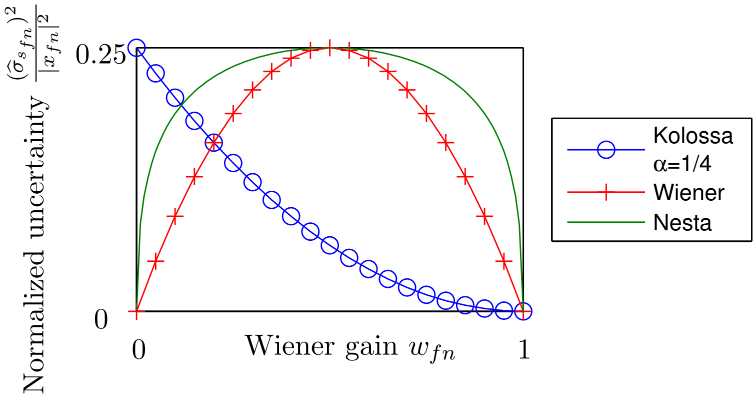


Figure 4.2: Behavior of the uncertainty estimators. The horizontal axis represents the estimated proportion of speech in the observed mixture power spectrum, as defined later in (6.3). The vertical axis is proportional to uncertainty. The uncertainty is normalized by the mixture power spectrum to emphasize the fact that the shape of the estimators doesn't depend on it.

As an alternative to the STFT, quadratic time-frequency representations often improve enhancement by accounting for the local correlation between channels [Ozerov et al., 2012]. Expression (4.3) is not applicable anymore in that case since the mixture signal is represented by its empirical covariance matrix $\widehat{\mathbf{R}}_{\mathbf{xx}_{fn}}$ instead of $\mathbf{x}_{fn}$. A more general expression may however be obtained for the magnitude of the mean as

$$|\widehat{\mu}_{s_{jfn}}| = \left(\mathbf{u}_f^H \mathbf{W}_{jfn} \widehat{\mathbf{R}}_{\mathbf{xx}_{fn}} \mathbf{W}_{jfn}^H \mathbf{u}_f\right)^{1/2} \tag{4.6}$$

which is enough for subsequent feature computation.

4.3 Uncertainty estimation

The goal of uncertainty estimation is to obtain not only a point estimate of the target speech source s_{jfn} represented by the *mean* $\widehat{\mu}_{s_{jfn}}$ of its posterior distribution $p(s_{jfn}|\mathbf{x}_{fn})$ but also an estimate of how much the true (unknown) source signal deviates from it, as represented by its posterior *variance* $\widehat{\sigma}^2_{s_{jfn}}$. Three state-of-the-art estimators are detailed in the following.

Kolossa's estimator

Kolossa et al. [Kolossa et al., 2010] assumed the uncertainty to be proportional to the squared difference between the enhanced signal and the mixture

$$(\widehat{\sigma}^{\text{Kol}}_{s_{jfn}})^2 = \alpha|\widehat{\mu}_{s_{jfn}} - x_{fn}|^2 \tag{4.7}$$

where $x_{fn} = \mathbf{u}_f^H \mathbf{x}_{fn}$ is the downmixed mixture signal and the scaling factor α is found by minimizing the Euclidean distance between the estimated uncertainty and the oracle uncertainty defined hereafter in equation (4.43).

Wiener estimator

Astudillo [Astudillo, 2010] later proposed to quantify uncertainty by the posterior variance of the Wiener filter. In the multichannel case, the posterior covariance matrix of $\mathbf{y}_{jfn}$ is given by [Ozerov et al., 2013]

$$\widehat{\mathbf{\Sigma}}_{\mathbf{y}_{jfn}} = (\mathbf{I}_I - \mathbf{W}_{jfn})\, v_{jfn}\mathbf{R}_{jf} \tag{4.8}$$

with $\mathbf{I}_I$ the identity matrix of size I. The variance of s_{jfn} is then easily derived as

$$(\widehat{\sigma}^{\text{Wie}}_{s_{jfn}})^2 = \mathbf{u}_f^H \widehat{\mathbf{\Sigma}}_{\mathbf{y}_{jfn}} \mathbf{u}_f. \tag{4.9}$$

Nesta's estimator

Recently, Nesta et al. [Nesta et al., 2013] obtained a different estimate based on a binary speech/noise predominance model[2]:

$$(\widehat{\sigma}^{\text{Nes}}_{s_{jfn}})^2 = \widehat{p}_{jfn}(1 - \widehat{p}_{jfn})|x_{fn}|^2 \tag{4.10}$$

where $\widehat{p}_{jfn} = \sqrt{\zeta_{jfn}}/(\sqrt{\zeta_{jfn}} + \sqrt{\zeta_{j^*fn}})$ and

$$\zeta_{jfn} = \mathbf{u}_f^H (v_{jfn}\mathbf{R}_{jf})\mathbf{u}_f \tag{4.11}$$

$$\zeta_{j^*fn} = \mathbf{u}_f^H \left(\sum_{j' \neq j} v_{j'fn}\mathbf{R}_{j'f} \right) \mathbf{u}_f \tag{4.12}$$

are the prior variances of the target speech source j and the other sources, respectively. The behavior of the three estimators is illustrated in Figure 4.2.

Uncertainty estimator with a GMM prior model

In another method [Astudillo, 2013], the prior distributions of the speech and noise signals are assumed to be mixtures of many zero-mean complex Gaussian distributions instead of only one Gaussian component. The prior GMM distributions of speech and noise are learned from clean speech data and noise only data beforehand. As a result, the posterior of the enhanced speech signal for a given pair of Gaussian components (one for speech and one for noise) is computed by the Wiener filter. The posterior of the enhanced speech signal is computed as the expected value of the Wiener posterior over all possible pair of Gaussian components.

4.4 Uncertainty propagation

From now on, we process one target speech source only and we drop index j for notation convenience. The posterior mean $\widehat{\mu}_{s_{fn}}$ and variance $\widehat{\sigma}^2_{s_{fn}}$ of the target speech source are propagated

[2] This formula was initially defined for the variance of $|s_{jfn}|$ [Nesta et al., 2013], however we found it beneficial to use it for the variance of s_{jfn} instead.

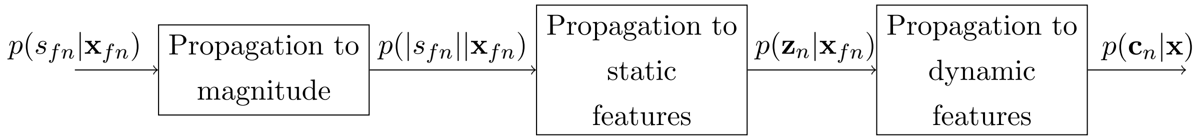

$p(s_{fn}|\mathbf{x}_{fn})$
Propagation to magnitude
$p(|s_{fn}||\mathbf{x}_{fn})$
Propagation to static features
$p(\mathbf{z}_n|\mathbf{x}_{fn})$
Propagation to dynamic features
$p(\mathbf{c}_n|\mathbf{x})$

Figure 4.4: Schematic diagram of uncertainty propagation from the complex-valued STFT domain to the feature domain.

step by step to the feature domain for exploitation by the recognizer. At each step, the posterior is approximated as a Gaussian distribution and represented by its mean and its variance [Astudillo, 2010]. We use 39-dimensional feature vectors $\mathbf{c}_n$ consisting of 12 MFCCs, the log-energy, and their first- and second-order time derivatives. The MFCCs are computed from the magnitude spectrum instead of the power spectrum since this has been shown to provide consistently better results in the context of uncertainty propagation [Kolossa et al., 2011]. Propagation is achieved in three steps as illustrated in Figure 4.4.

4.4.1 To the magnitude spectrum

This section explains how to propagate a complex valued Gaussian distribution to the magnitude domain. As explained above, now we consider MFCCs computed from magnitude spectra $|s_{fn}|$. Since s_{fn} is assumed to be a complex valued Gaussian, the distribution of its amplitude is a Rice distribution. From the close form of the Rice distribution of the amplitude value $|s_{fn}|$, the first and second order moments can be derived easily. Based on the first and second order moments of $|s_{fn}|$, the mean and variance of $|s_{fn}|$ can be computed. In general, the k-th order moment of this distribution has the following closed form [Gradshteyn and Ryzhik, 1995]:

$$\mathbb{E}(|s_{fn}|^k) = \Gamma\left(\frac{k}{2}+1\right)\left(\widehat{\sigma}^2_{s_{fn}}\right)^{\frac{k}{2}} L_{\frac{k}{2}}\left(-\frac{|\widehat{\mu}_{s_{fn}}|^2}{\widehat{\sigma}^2_{s_{fn}}}\right) \tag{4.13}$$

where Γ is the gamma function and $L_{\frac{k}{2}}$ is the Laguerre polynominal. Thus, the first and the second order moments of $|s_{fn}|$ are computed as follows:

$$\mathbb{E}\left(|s_{fn}|\right) = \Gamma\left(\frac{3}{2}\right)\widehat{\sigma}_{s_{fn}} L_{\frac{1}{2}}\left(-\frac{|\widehat{\mu}_{s_{fn}}|^2}{\widehat{\sigma}^2_{s_{fn}}}\right) \tag{4.14}$$

$$\mathbb{E}(|s_{fn}|^2) = \widehat{\sigma}^2_{s_{fn}} + |\widehat{\mu}_{s_{fn}}|^2. \tag{4.15}$$

$L_{\frac{1}{2}}$ is computed in closed form by combining [Gradshteyn and Ryzhik, 1995] and [Rice, 1944] as [Astudillo, 2010]

$$L_{\frac{1}{2}}(q) = e^{\frac{q}{2}}\left((1-q)\, I_0(\frac{q}{2}) + qI_1\left(\frac{q}{2}\right)\right) \tag{4.16}$$

where I_0 and I_1 are order-0 and order-1 Bessel functions. As a result, the estimated mean and the estimated variance of $|s_{fn}|$ are computed as

$$\mu_{|s_{fn}|} = \Gamma\left(\frac{3}{2}\right)\left(\widehat{\sigma}^2_{s_{fn}}\right)^{\frac{1}{2}} L_{\frac{1}{2}}\left(-\frac{|\widehat{\mu}_{s_{fn}}|^2}{\widehat{\sigma}^2_{s_{fn}}}\right) \tag{4.17}$$

$$\sigma^2_{|s_{fn}|} = \mathbb{E}(|s_{fn}|^2) - \mu^2_{|s_{fn}|}. \tag{4.18}$$

4.4.2 To the static MFCCs

The mean and covariance of the magnitude spectrum $|s_{fn}|$ are propagated through MFCC computation including: pre-emphasis filter, Mel filterbank, logarithm, DCT and lifter. The nonlinear transform is given by

$$\mathbf{z}_n = \mathcal{F}\left(|\mathbf{s}_n|\right) = \mathbf{Diag}(\mathbf{l})\mathbf{D}\log(\mathbf{M}\,\mathbf{Diag}(\mathbf{e})|\mathbf{s}_n|) \tag{4.19}$$

where $|\mathbf{s}_n| = [|s_{1n}|, \ldots, |s_{Fn}|]^T$ with F the number of frequency bins, $\mathbf{Diag}(.)$ is the diagonal matrix built from its vector argument, $\mathbf{e}$, $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{l}$ are the $F \times 1$ vector of pre-emphasis coefficients, the $26 \times F$ Mel filterbank matrix, the 12×26 discrete cosine transform (DCT) matrix, and the 12×1 vector of liftering coefficients, respectively. The elements of $\mathbf{e}$ are computed as

$$\mathbf{e}_f = 1 - 0.97e^{-i\omega_f} \tag{4.20}$$

where ω_f is the angular frequency of bin f in $[-\pi; \pi]$. For linear transforms namely pre-emphasis filter, Mel filterbank, DCT and lifter, there is a closed form solution [Astudillo, 2010]. For the logarithm function, the propagation can be achieved by various techniques including VTS, the unscented transform (UT) [Julier and Uhlmann, 2004] or moment matching (MM), also known as the log-normal transform [Gales, 1995; Kolossa et al., 2010; Astudillo, 2010]. The following part will present these existing propagators. For ease of notation, let us define:

$$\mathbf{E} = \mathbf{Diag}(\mathbf{e}) \tag{4.21}$$

$$\mathbf{L} = \mathbf{Diag}(\mathbf{l}). \tag{4.22}$$

VTS based propagation

Given a non-linear function $\mathcal{F}$ (MFCC extraction) of the input vector $|\mathbf{s}_n|$, the output $\mathbf{z}_n$ is approximated by first-order Taylor series expansion [Ozerov et al., 2013] around a given value $|\mathbf{s}^0_n|$

$$\mathbf{z}_n \approx \mathcal{F}\left(|\mathbf{s}^0_n|\right) + \mathcal{J}_{\mathcal{F}}\left(|\mathbf{s}^0_n|\right)\left(|\mathbf{s}_n| - |\mathbf{s}^0_n|\right). \tag{4.23}$$

If $|\mathbf{s}^0_n|$ is the mean vector $\widehat{\boldsymbol{\mu}}_{|\mathbf{s}_n|}$ then $\mathbb{E}\left[\left(|\mathbf{s}_n| - |\mathbf{s}^0_n|\right)\left(|\mathbf{s}_n| - |\mathbf{s}^0_n|\right)^T\right]$ becomes $\widehat{\boldsymbol{\Sigma}}_{|\mathbf{s}_n|}$. As a result, the mean and covariance matrix of the MFCCs $\mathbf{z}_n$ are computed as

$$\widehat{\boldsymbol{\mu}}_{\mathbf{z}_n} = \mathcal{F}\left(\widehat{\boldsymbol{\mu}}_{|\mathbf{s}_n|}\right) = \mathbf{LD}\log(\mathbf{ME}\widehat{\boldsymbol{\mu}}_{|\mathbf{s}_n|}) \tag{4.24}$$

$$\widehat{\boldsymbol{\Sigma}}_{\mathbf{z}_n} = \mathcal{J}_{\mathcal{F}}\left(\widehat{\boldsymbol{\mu}}_{|\mathbf{s}_n|}\right)\widehat{\boldsymbol{\Sigma}}_{|\mathbf{s}_n|}\mathcal{J}_{\mathcal{F}}\left(\widehat{\boldsymbol{\mu}}_{|\mathbf{s}_n|}\right)^T \tag{4.25}$$

where the Jacobian matrix $\mathcal{J}_{\mathcal{F}}\left(\widehat{\boldsymbol{\mu}}_{|\mathbf{s}_n|}\right)$ is given by

$$\mathcal{J}_{\mathcal{F}}\left(\widehat{\boldsymbol{\mu}}_{|\mathbf{s}_n|}\right) = \mathbf{LDDiag}\left(\frac{1}{\mathbf{ME}\widehat{\boldsymbol{\mu}}_{|\mathbf{s}_n|}}\right)\mathbf{ME} \tag{4.26}$$

and the division is performed elementwise.

Moment matching

Another approach is moment matching [Gales, 1995] (MM), also called the log-normal transform. It comes from the fact that, if the input of an exponential function is a normal distribution, then the mean and the covariance of the output can be computed in closed form given the mean and the covariance of the input. By inverting that expression, the mean and the covariance of $\mathbf{z}_n$ can be estimated as

$$\widehat{\boldsymbol{\mu}}_{\mathbf{z}_n} = \mathbf{LD}\left(\log\left(\mathbf{ME}\widehat{\boldsymbol{\mu}}_{|\mathbf{s}_n|}\right) - \frac{1}{2}\log\left(1 + \frac{diag\left(\mathbf{ME}\widehat{\boldsymbol{\Sigma}}_{|\mathbf{s}_n|}\mathbf{E}^T\mathbf{M}^T\right)}{\mathbf{ME}\widehat{\boldsymbol{\mu}}^2_{|\mathbf{s}_n|}}\right)\right) \tag{4.27}$$

$$\widehat{\boldsymbol{\Sigma}}_{\mathbf{z}_n} = \mathbf{LD}\log\left(1 + \frac{\mathbf{ME}\widehat{\boldsymbol{\Sigma}}_{|\mathbf{s}_n|}\mathbf{E}^T\mathbf{M}^T}{\mathbf{Diag}\left(\mathbf{ME}\widehat{\boldsymbol{\mu}}_{|\mathbf{s}_n|}\right)\mathbf{Diag}\left(\mathbf{ME}\widehat{\boldsymbol{\mu}}_{|\mathbf{s}_n|}\right)}\right)\mathbf{D}^T\mathbf{L}^T \tag{4.28}$$

where $diag\left(\cdot\right)$ is the vector of diagonal elements of a matrix. Logarithm, division, and squaring are performed elementwise.

Unscented transform

The unscented transform [Julier and Uhlmann, 2004] is a pseudo-Monte Carlo method which is used to propagate a distribution through a nonlinear transform. It replaces a continuous distribution by a set of sample points called sigma points which are representative of the distribution characteristics [Astudillo, 2010]. The mean and covariance of the output distribution are approximated by transforming this set of sample points.

Regression trees

Regression trees [Srinivasan and Wang, 2007] are another approach to learn the nonlinear transformation of the uncertainty from the linear spectral domain to the cepstral domain. The binary uncertainty in the spectral domain is defined by considering that a time-frequency bin is either reliable or unreliable and it is derived from an ideal binary mask. The input of the regression tree is the set of binary uncertainties in the spectral domain and its outputs are the oracle uncertainties in the cepstral domain.

4.4.3 To the dynamic features

The uncertainty about the static features is propagated to the full feature vector including static and dynamic features. The static features in the preceding 4 frames, in the current frame, and in the following 4 frames are concatenated into a column vector $\bar{\mathbf{z}}_n = [\mathbf{z}_{n-4}^T\ \mathbf{z}_{n-3}^T \cdots \mathbf{z}_{n+4}^T]^T$. The full feature vector $\mathbf{c}_n = [\mathbf{z}_n\ \Delta\mathbf{z}_n\ \Delta^2\mathbf{z}_n]$ can be expressed in matrix form as

$$\mathbf{c}_n = (\mathbf{A} \otimes \mathbf{I}_C)\bar{\mathbf{z}}_n \tag{4.29}$$

where $\otimes$ is the Kronecker product, $\mathbf{I}_C$ is the identity matrix of size $C = 12$, and the matrix $\mathbf{A}$ is given by [Young et al., 2006]

$$\mathbf{A} = \frac{1}{100}\begin{bmatrix} 0 & 0 & 0 & 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & -20 & -10 & 0 & 10 & 20 & 0 & 0 \\ 4 & 4 & 1 & -4 & -10 & -4 & 1 & 4 & 4 \end{bmatrix}. \tag{4.30}$$

The mean and the covariance matrix of the posterior distribution $p(\mathbf{c}_n|\mathbf{x})$ are derived as

$$\widehat{\boldsymbol{\mu}}_{\mathbf{c}_n} = (\mathbf{A} \otimes \mathbf{I}_C)\widehat{\boldsymbol{\mu}}_{\bar{\mathbf{z}}_n} \tag{4.31}$$

$$\widehat{\boldsymbol{\Sigma}}_{\mathbf{c}_n} = (\mathbf{A} \otimes \mathbf{I}_C)\widehat{\boldsymbol{\Sigma}}_{\bar{\mathbf{z}}_n}(\mathbf{A} \otimes \mathbf{I}_C)^T \tag{4.32}$$

where $\widehat{\boldsymbol{\mu}}_{\bar{\mathbf{z}}_n}$ and $\widehat{\boldsymbol{\Sigma}}_{\bar{\mathbf{z}}_n}$ are obtained by concatenating $\widehat{\boldsymbol{\mu}}_{\mathbf{z}_{n-4}}, \ldots, \widehat{\boldsymbol{\mu}}_{\mathbf{z}_{n+4}}$ into a column vector and $\widehat{\boldsymbol{\Sigma}}_{\mathbf{z}_{n-4}}, \ldots, \widehat{\boldsymbol{\Sigma}}_{\mathbf{z}_{n+4}}$ into a block-diagonal matrix. Only the diagonal of $\widehat{\boldsymbol{\Sigma}}_{\mathbf{c}_n}$ is retained [Astudillo, 2010; Astudillo et al., 2014; Kolossa et al., 2010]. In addition, the mean and uncertainty of the log energy are computed separately from the raw signal in the time domain [Astudillo and Kolossa, 2011].

4.4.4 Cepstral mean normalization

Cepstral mean normalization is applied only to the MFCCs, not to the log-energy coefficients. For large enough number of frames, we treat the mean of the MFCCs over time as a deterministic quantity. Therefore, the mean vectors $\widehat{\boldsymbol{\mu}}_{\bar{\mathbf{z}}_n}$ are normalized as usual while the covariance matrices $\widehat{\boldsymbol{\Sigma}}_{\bar{\mathbf{z}}_n}$ are not affected by cepstral mean normalization.

4.5 Uncertainty decoding

4.5.1 Uncertainty decoding

The likelihood of the noisy features given the acoustic model is modified by marginalizing over the underlying clean features as

$$p(\mathbf{x}_n|q) = \int_{\mathbf{c}_n} \frac{p(\mathbf{c}_n|\mathbf{x}_n)p(\mathbf{x}_n)}{p(\mathbf{c}_n)} p(\mathbf{c}_n|q)d\mathbf{c}_n \tag{4.33}$$

where $p(\mathbf{c}_n|q)$ is the clean acoustic model for state q and $p(\mathbf{c}_n|\mathbf{x}_n)$ is the posterior of the clean features computed above. For low distortion levels, this can be approximated up to a multiplicative constant as [Deng et al., 2005; Astudillo and Orglmeister, 2013]

$$p(\mathbf{x}_n|q) \approx \int_{\mathbf{c}_n} p(\mathbf{c}_n|\mathbf{x}_n)p(\mathbf{c}_n|q)d\mathbf{c}_n. \tag{4.34}$$

In the case when $p(\mathbf{c}_n|q)$ is a GMM with M components with weights, means, and diagonal covariance matrices denoted as ω_{qm}, $\boldsymbol{\mu}_{qm}$, and $\boldsymbol{\Sigma}_{qm}$, respectively, the likelihood of the noisy features (4.34) can be computed in closed form as [Deng et al., 2005]

$$p(\mathbf{x}_n|q) \approx \sum_{m=1}^{M} \omega_{qm}\mathcal{N}(\widehat{\boldsymbol{\mu}}_{\mathbf{c}_n}; \boldsymbol{\mu}_{qm}, \boldsymbol{\Sigma}_{qm} + \widehat{\boldsymbol{\Sigma}}_{\mathbf{c}_n}). \tag{4.35}$$

Numerical example of uncertainty decoding

Let us assume that there are two classes q_1 and q_2, each modeled as a univariate Gaussian distribution: $p(x|q_1) \sim \mathcal{N}(x; \mu_1, \sigma_1^2)$ and $p(x|q_2) \sim \mathcal{N}(x; \mu_2, \sigma_2^2)$, respectively. Given a clean observation x, we wish to determine when x is classified into class q_1 or class q_2.

The two distributions $p(x|q_1)$ and $p(x|q_2)$ for the parameter values $\mu_1 = -0.1; \sigma_1^2 = 3; \mu_2 = 5; \sigma_2^2 = 0.01$ are depicted as dashed blue and red curves in Figure 4.5. Suppose that we observed $x = 5$ then this observation will be classified into class q_2 with very high probability. If we cannot observe x anymore but a noisy version $y = 6$, the classifier will class the noisy observation into the wrong class q_1. Indeed, the likelihood of the noisy observation can be computed as

$$p(y|q_1) = \mathcal{N}(y; \mu_1, \sigma_1^2) \approx 5.7 \times 10^{-4} \tag{4.36}$$

$$p(y|q_2) = \mathcal{N}(y; \mu_2, \sigma_2^2) \approx 1.0 \times 10^{-17}. \tag{4.37}$$

Now suppose that, using some uncertainty estimation technique, we obtain the posterior distribution of the clean signal x given the noisy version y as

$$p(x|y) \sim \mathcal{N}(x; \widehat{\mu}_x, \widehat{\sigma}_x^2). \tag{4.38}$$

where $\widehat{\mu}_x = 5.9$ and $\widehat{\sigma}_x^2 = 0.81$. The distribution $p(x|y)$ is depicted as the green curve in Figure 4.5. Then the likelihood of the noisy observation y can be computed by uncertainty decoding as

$$p(y|q_1) \sim \mathcal{N}(\widehat{\mu}_x; \mu_1, \sigma_1^2 + \widehat{\sigma}_x^2) \approx 1.8 \times 10^{-3} \tag{4.39}$$

$$p(y|q_2) \sim \mathcal{N}(\widehat{\mu}_x; \mu_2, \sigma_2^2 + \widehat{\sigma}_x^2) \approx 2.7 \times 10^{-1} \tag{4.40}$$

The two new distributions are depicted as plain blue and red curves in Figure 4.5. Due to adding the variance of $p(x|y)$ to that of the two former distributions, the new distributions are broader. As we can see in Figure 4.5, the observation has higher probability for class q_2 than for class q_1 and will be correctly classified by uncertainty decoding.

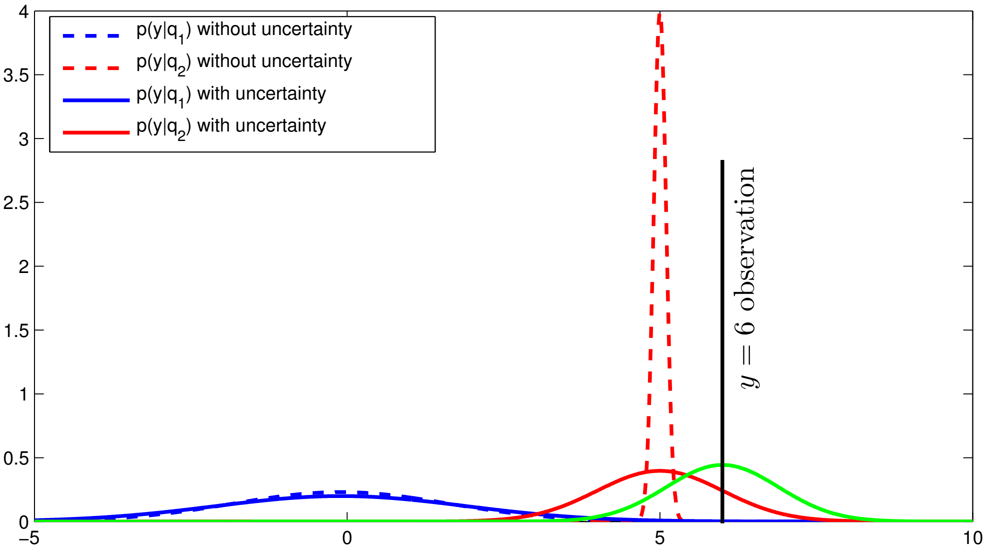


Figure 4.5: Numerical example of the uncertainty decoding.

Computation time

In terms of computation time, the cost of uncertainty estimation and propagation is negligible compared to that of uncertainty decoding. In our implementation, the cost of computing the modified acoustic likelihoods was 1.2 times larger for diagonal uncertainty covariance matrices and 3.1 times larger for full uncertainty covariance matrices than the cost of computing the conventional likelihoods without uncertainty. Furthermore, the impact of this extra cost decreases with larger vocabulary size as the computation time becomes dominated by the decoding of the word graph.

4.5.2 Modified imputation

Modified imputation [Astudillo, 2010] is a variant of uncertainty decoding where the clean feature are estimated depending on the state and mixture component in order to maximize the joint probability of the observation given by the GMM and the estimated uncertainty. It results in the estimated clean features

$$\widehat{\boldsymbol{\mu}}_{\mathbf{c}_n}^{qm} = (\boldsymbol{\Sigma}_{qm}^{-1} + \widehat{\boldsymbol{\Sigma}}_{\mathbf{c}_n}^{-1})^{-1} \left[\boldsymbol{\Sigma}_{qm}^{-1}\boldsymbol{\mu}_{qm} + \widehat{\boldsymbol{\Sigma}}_{\mathbf{c}_n}^{-1}\widehat{\boldsymbol{\mu}}_{\mathbf{c}_n}\right] \tag{4.41}$$

For low distortion level, the likelihood of the estimated clean features is computed by the following equation:

$$p(\mathbf{x}_n|q) \approx \sum_{m=1}^{M} \omega_{qm}\mathcal{N}(\widehat{\boldsymbol{\mu}}_{\mathbf{c}_n}^{qm}; \boldsymbol{\mu}_{qm}, \boldsymbol{\Sigma}_{qm}). \tag{4.42}$$

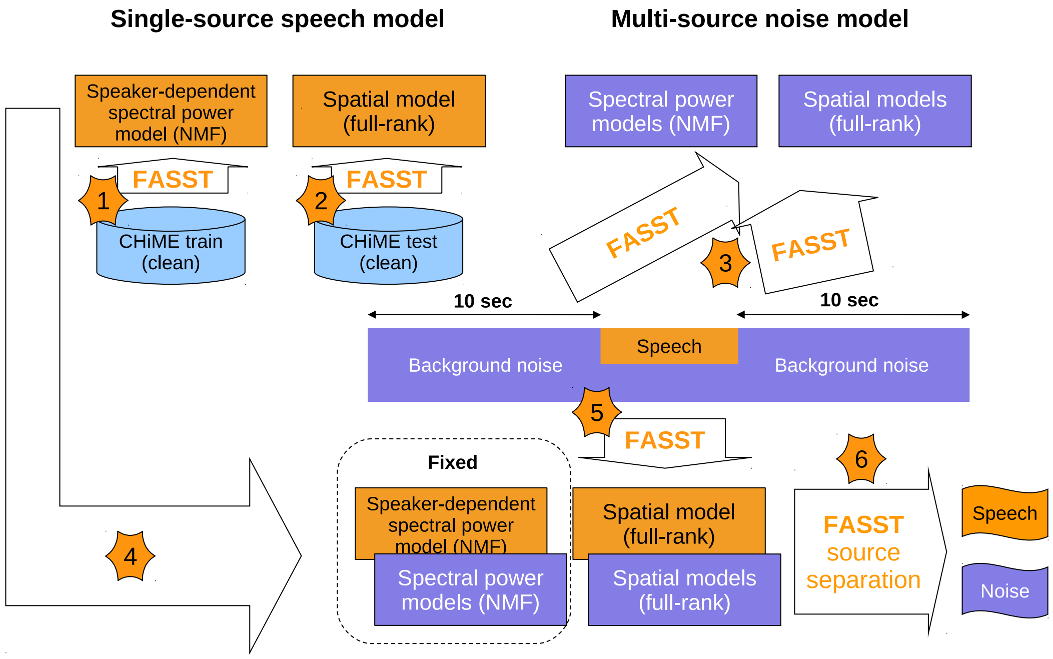

Single-source speech model
Multi-source noise model
Speaker-dependent spectral power model (NMF)
Spatial model (full-rank)
Spectral power models (NMF)
Spatial models (full-rank)
FASST
1
CHiME train (clean)
FASST
2
CHiME test (clean)
FASST
3
FASST
10 sec
10 sec
Background noise
Speech
Background noise
5
FASST
Fixed
Speaker-dependent spectral power model (NMF)
Spectral power models (NMF)
Spatial model (full-rank)
Spatial models (full-rank)
4
6
FASST source separation
Speech
Noise

Figure 4.6: Baseline enhancement system for the CHiME dataset.

4.5.3 Uncertainty training

In the above techniques, the GMM-HMM acoustic model was learned from clean speech. However, a massive amount of data with different noisy conditions are often found in fact. The uncertainty training method in [Ozerov et al., 2013] makes it possible to learn GMM parameters representing clean speech directly from noisy speech with associated dynamic uncertainty. The GMM parameters of clean speech are estimated by maximizing the likelihood of the noisy features in equation (4.34) which takes into account the uncertainty associated with each feature. This is solved by using an EM algorithm. It the E-step, the uncertainty covariance matrices are exploited not only to compute the posterior component probability but also the first and second order moments of underlying clean data. By doing this, uncertainty training actually estimates the distribution of the underlying clean data.

4.6 Baseline system for CHiME

In the following, we used the FASST source separation toolbox as a speech enhancement front end [Ozerov et al., 2012]. This toolbox can model the source spectra by means of multilevel NMF and their spatial properties by means of either rank-1 or full-rank spatial covariance matrices

[Ozerov et al., 2012]. Based on available knowledge such as the speakers identity, the rough target spatial direction, and the temporal location of the target speech utterances within the mixture signal, appropriate constraints can be specified on the model parameters, so as to design a custom speech separation algorithm with little effort.

4.6.1 Speech enhancement baseline

Figure 4.6 illustrates the speech enhancement baseline using the FASST toolbox. We learn an NMF model of the short-term power spectrum (shown as step 1 in Figure 4.6). In the case of Track 1, we learn speaker-dependent NMF model using 500 utterances picked from the noiseless reverberated training set. In the case of Track 2, we learn speaker-independent NMF mode where the training data was collected by using 10% the number of the frames in each utterance in all training set. The NMF basis spectra are initialized by split vector quantization and we used 50 iterations to re-estimate the model using FASST.

We also learn a speaker-independent full-rank spatial covariance model of the target speech source from the noiseless reverberated training set (shown as step 2 in Figure 4.6). In the case of Track 1, 500 utterances are selected from each speaker. In the case of Track 2, tranining data was collected by using 50% the number of the frames in each utterance in all training set. The spatial covariance matrix is randomly initialized and re-estimated using FASST.

The noise is modeled as the sum of two sources. Each source is given a full-rank spatial model and an NMF spectral model. This multi-source noise model is trained on the speech-free background samples (5 s before and 5 s after each utterance) of the mixture signals to be separated (shown as step 3 in Figure 4.6). The model is randomly initialized and trained using FASST. We used 30 iterations for training.

After the spatial models and the NMF spectral models have been trained, the utterance to be separated is modeled as the sum of one speech source and two background noise sources, whose parameters are initialized by those of the corresponding trained models (shown as step 4 and 5 in Figure 4.6). We used 128 NMF components and 32 NMF components for modeling the target source and background noise, respectively. In all experiments, quadratic equivalent rectangular bandwidth (QERB) time-frequency representations [Ozerov et al., 2012] are used to represent the signals. The number of frequency bands is 160, the window size is 24 ms, and overlap is 50%. While the trained NMF basis spectra of the target, the background, and the spatial covariance matrices are kept fixed, the other parameters namely the NMF temporal activation coefficients are reestimated on that noisy utterance using 40 iterations of FASST. Finally, the target speech signal is extracted by multichannel Wiener filtering (shown as step 6 in Figure 4.6). This procedure is applied to all noisy utterances in the training, development, and test sets.

4.6.2 ASR baseline for Track 1

In speech recognition stage for the Track 1 data, the features used are 39-dimensional MFCCs (12 cepstral + log-energy, delta, delta-delta) with cepstral mean subtraction. We use the HTK baseline provided on the CHiME website up to a modification of the ADDDITHER parameter to 25, which governs the amount of noise added to the signal before MFCC calculation, so as to make the MFCCs more robust to zeroes in the speech spectra after source separation.

We use the baseline reverberated acoustic models provided on the CHiME website with a modification of the window length and the step size to 24ms and 12ms, respectively. Speaker-dependent acoustic models are trained on the noiseless reverberated training data using the HTK baseline. Speaker-independent models are learned from all speakers' data and subsequently adapted to each speaker by running 5 additional iterations of Baum-Welch and keeping the weights and variances of the GMM observation probabilities fixed while reestimating their means.

Uncertainty decoding is performed using the HTK baseline with Astudillo's patch[3] for diagonal uncertainty covariances and with our own patch for full uncertainty covariances[4].

4.6.3 ASR baseline for Track 2

For the Track 2, speaker-independent GMM-HMM acoustic models are trained from the reverberated noiseless training set using Kaldi[5]. The feature vectors consist of MFCCs, log-energy, and their first- and second-order derivatives, similarly to above[6]. Uncertainty decoding with diagonal uncertainty covariance matrices is performed using our Kaldi patch[7], which dynamically adapts the GMM observation probabilities as described in Section 4.5.1. Uncertainty decoding with full uncertainty covariance matrices is achieved by retaining the 100-best list obtained with diagonal uncertainty covariance matrices and by recomputing the acoustic scores. The language model is the trigram provided by the Challenge organizers and the optimal language model weight is found on the development set.

4.7 Upper bound on the ASR performance

In order to evaluate the potential of uncertainty decoding, we evaluate its performance with oracle uncertainty.

[3] http://www.astudillo.com/ramon/research/stft-up/

[4] http://full-ud-htk.gforge.inria.fr/

[5] http://kaldi.sourceforge.net/

[6] The considered GMM-HMM does not include advanced feature transforms and training/decoding techniques such as linear discriminant analysis (LDA), maximum likelihood linear transformation (MLLT), feature-space maximum likelihood linear regression (fMLLR), feature-space minimum phone error (fMPE), speaker adaptive training (SAT), discriminative language modeling (DLM), or minimum Bayes risk (MBR) decoding, which were shown to bring the performance of GMM-HMMs close to that of DNN-HMMs [Tachioka et al., 2013b]. The interplay of such techniques with uncertainty decoding is out of the scope of this thesis.

[7] http://ud-kaldi.gforge.inria.fr/

4.7.1 Oracle uncertainty

Oracle uncertainty is the ideal uncertainty of a given estimated feature or estimated spectral. There are two definitions of oracle uncertainty which result in two formulas: the diagonal oracle uncertainty and the full oracle uncertainty. The diagonal oracle uncertainty covariance matrix is computed as the squared difference between the estimated spectral or features and the clean spectral or spectra or features [Deng et al., 2005]. The spectral-domain diagonal oracle uncertainty element at frame index n and spectral bin f is given by

$$\sigma^2_{s_{fn}} = |\widehat{\mu}_{s_{fn}} - s_{fn}|^2. \tag{4.43}$$

Where s_{fn} are the clean STFT coefficients. The spectral-domain diagonal oracle uncertainty element at frame index n and feature index i is given by

$$\sigma^2_{c_{in}} = (\widehat{\mu}_{c_{in}} - c_{in})^2 \tag{4.44}$$

where c_{in} is the clean feature. The full oracle uncertainty covariance matrix in the feature domain is a rank-1 matrix [Ozerov et al., 2013] and it is computed as

$$\mathbf{\Sigma}_{\mathbf{c}_n} = (\widehat{\boldsymbol{\mu}}_{\mathbf{c}_n} - \mathbf{c}_n)(\widehat{\boldsymbol{\mu}}_{\mathbf{c}_n} - \mathbf{c}_n)^T. \tag{4.45}$$

This oracle rank-1 matrix is more informative because it encodes exactly the direction of the difference between the estimated features and the clean features.

4.7.2 Experimental results

First, we evaluate the performance of the speech enhancement front-end only. The acoustic model is trained on noiseless reverberated data. In the decoding stage, we achieved 85.01% keyword accuracy on the Track 1 test dataset after enhancement as shown in Table 4.1. On Track 2, we obtained 53.89% WER on the test dataset as shown in Table 4.2.

Second, we evaluate the performance of uncertainty decoding with these oracle uncertainties. The acoustic model is also trained on noiseless reverberated data as above. The performance in the diagonal uncertainty case (94.57%) is lower than the performance in the full uncertainty case (96.31%) by almost 2% absolute (37% relative) for Track 1. The same trend is obtained for Track 2. The performance in the diagonal uncertainty case was 23.94% WER while that in the full uncertainty case was 18.80% WER.

For comparision, the keyword accuracy in the case when the acoustic model is trained and evaluated on noiseless reverberated data is 96.92% on Track 1. Therefore, full uncertainty decoding using the oracle uncertainty almost reaches the performance achieved on clean data.

4.7.3 Summary

This chapter has reviewed uncertainty handling techniques and how to integrate them into GMM-HMM based speech recognition systems. All methods modeled the uncertainty as the

Uncertainty	Test set						
covariance matrix	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average
no uncertainty	73.75	78.42	84.33	89.50	91.83	92.25	85.01
diagonal	93.58	92.67	94.92	95.25	95.58	95.42	94.57
full	96.33	96.00	96.33	96.50	96.67	96.08	96.31

Table 4.1: Keyword accuracy (%) evaluated with the oracle uncertainties on the Track 1 test set after speech enhancement. Average accuracies have a 95% confidence interval of $\pm 0.8, \pm 0.5, \pm 0.4\%$ for no uncertainty, diagonal and full uncertainty covariance respectively. The full result can be found in Appendix A.1

Uncertainty	Test set						
covariance matrix	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average
no uncertainty	68.47	63.75	56.76	51.03	44.22	39.12	53.89
diagonal	30.9	28.27	24.09	22.14	20.05	18.21	23.94
full	19.49	19.24	18.78	18.14	19.17	18.01	18.80

Table 4.2: WER (%) evaluated on the oracle uncertainties on the Track 2 test set. Average WER have a 95% confidence interval of $\pm 1.1, \pm 0.9, \pm 0.9\%$ for no uncertainty, diagonal and full uncertainty covariance, respectively. The full results can be found in Appendix A.2

variance of a posterior Gaussian distribution. State-of-the-art systems have several limitations. First, the spectral-domain estimation of uncertainty is often obtained by heuristic methods and it turns out that the behaviors of the uncertainty estimators differ. Second, spectral-domain uncertainty is propagated through MFCC computation using an approximate method (VTS, moment matching, unscented transform or regression trees) resulting in a diagonal uncertainty covariance matrix in the feature domain. Nevertheless, uncertainty decoding is potentially efficient to reduce the mismatch in both theoretical and numerical examples. Actually, with oracle uncertainty, modeling the full uncertainty matrix leads to comparable performance to decoding on clean data for a small vocabulary task.

5

Extension of uncertainty propagation to a full covariance matrix

This chapter presents the proposed extension of uncertainty propagation to the full covariance matrix. The motivation of this work is presented in Section 5.1. Subsequently, the main idea is presented in Section 5.2. Last, Section 5.3 shows the results obtained by evaluating on Track 1 of the 2nd CHiME Challenge. This work was published in [Tran et al., 2014a].

5.1 Motivation

Figure 5.1 shows the relation between a diagonal estimated uncertainty covariance matrix and a full oracle uncertainty covariance matrix in the feature domain. Uncertainty on the left side is estimated using the Wiener method and it is propagated to the feature domain using VTS. The oracle uncertainty on the right side is obtained using equation (4.45). It is clear that the diagonal uncertainty covariance matrix *misses some dependencies* between the feature indexes. In addition, the analysis of ASR results with oracle uncertainties in the previous chapter shows that full uncertainty covariance matrices achieve better performance than diagonal uncertainty covariance matrices.

5.2 Extension of uncertainty propagation

To address this issue, we propose to propagate the mean $\widehat{\mu}_{s_{jfn}}$ and the variance $\widehat{\sigma}^2_{s_{jfn}}$ of the target speech source step by step to the feature domain but we retain the dependencies across dimensions. As previously, we use 39-dimensional feature vectors $\mathbf{c}_n$ consisting of 12 MFCCs, the log-energy, and their first- and second-order time derivatives. For legibility, we remove the index j from now on.

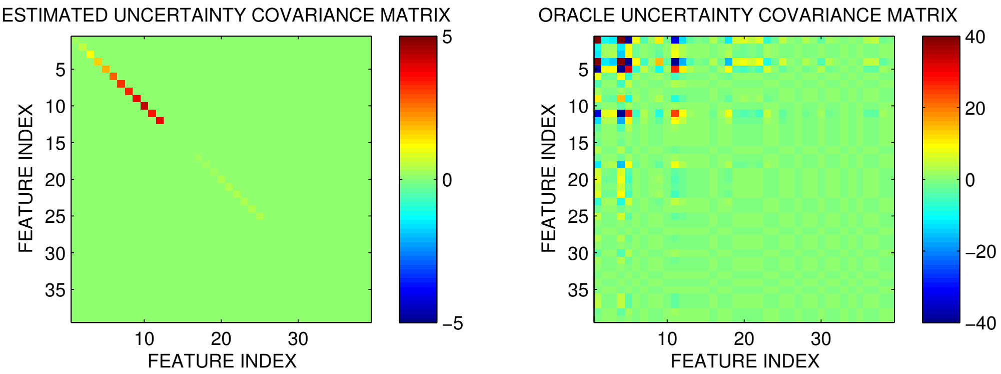


Figure 5.1: Example of diagonal uncertainty covariance matrix on one time frame. This example correspond to frame 50 of the utterance 'bin blue with s nine soon' on the Track 1 dataset. Left side: uncertainty estimated using Wiener + VTS. Right side: oracle uncertainty obtained using equation (4.45).

5.2.1 To the magnitude and the power spectra

The first step is to propagate the uncertainty from the complex-valued spectrum to the magnitude and the power spectra. Assuming that each s_f of the clean speech signal is complex Gaussian-distributed then the distribution of its amplitude is Rice distribution. Let us define the 2×1 vector $\mathbf{v}_{fn} = [|s_{fn}| \ |s_{fn}|^2]^T$. The mean and the covariance matrix of $\mathbf{v}_{fn}$ are given by

$$\hat{\boldsymbol{\mu}}_{\mathbf{v}_{fn}} = \begin{bmatrix} E_1 \\ E_2 \end{bmatrix} \tag{5.1}$$

$$\widehat{\boldsymbol{\Sigma}}_{\mathbf{v}_{fn}} = \begin{bmatrix} E_2 - E_1^2 & E_3 - E_1 E_2 \\ E_3 - E_1 E_2 & E_4 - E_2^2 \end{bmatrix} \tag{5.2}$$

where $E_k = E(|s_{fn}|^k)$ is the k-th order moment of the distribution of $|s_{fn}|$ as given in equation (4.13). The first four moments are obtained as

$$E_1 = \Gamma\left(\frac{3}{2}\right) \widehat{\sigma}_{s_{fn}} L_{\frac{1}{2}} \left(-\frac{|\widehat{\mu}_{s_{fn}}|^2}{\widehat{\sigma}^2_{s_{fn}}}\right) \tag{5.3}$$

$$E_2 = \widehat{\sigma}^2_{s_{fn}} + |\widehat{\mu}_{s_{fn}}|^2 \tag{5.4}$$

$$E_3 = \Gamma\left(\frac{5}{2}\right) \widehat{\sigma}^3_{s_{fn}} L_{\frac{3}{2}} \left(-\frac{|\widehat{\mu}_{s_{fn}}|^2}{\widehat{\sigma}^2_{s_{fn}}}\right) \tag{5.5}$$

$$E_4 = |\widehat{\mu}_{s_{fn}}|^4 + 4|\widehat{\mu}_{s_{fn}}|^2 \widehat{\sigma}^2_{s_{fn}} + 2\widehat{\sigma}^4_{s_{fn}} \tag{5.6}$$

where $L_{\frac{1}{2}}$ and $L_{\frac{3}{2}}$ are given by [Gradshteyn and Ryzhik, 1995; Rice, 1944; Astudillo, 2010]

$$L_{\frac{1}{2}}(q) = e^{\frac{q}{2}} \left((1-q)\, I_0(\frac{q}{2}) + q I_1\left(\frac{q}{2}\right)\right) \tag{5.7}$$

$$L_{\frac{3}{2}}(q) = \frac{1}{3} e^{\frac{q}{2}} \left(\left(2q^2 - 6q + 3\right) I_0\left(\frac{q}{2}\right) + \left(4q - 2q^2\right) I_1\left(\frac{q}{2}\right)\right) \tag{5.8}$$

with I_0 and I_1 denoting the order-0 and order-1 Bessel functions.

The full magnitude and power spectra are concatenated into a $2F \times 1$ vector $\mathbf{v}_n$ as

$$\mathbf{v}_n = [|s_{1n}| \ldots |s_{Fn}| \; |s_{1n}|^2 \ldots |s_{Fn}|^2]^T \tag{5.9}$$

where F is the number of frequency bins. The mean $\widehat{\boldsymbol{\mu}}_{\mathbf{v}_n}$ and the covariance matrix $\widehat{\boldsymbol{\Sigma}}_{\mathbf{v}_n}$ of $\mathbf{v}_n$ are obtained by stacking $\hat{\boldsymbol{\mu}}_{\mathbf{v}_{fn}}$ and $\widehat{\boldsymbol{\Sigma}}_{\mathbf{v}_{fn}}$ in the same order, yielding a block-diagonal covariance matrix with four diagonal blocks.

5.2.2 To the static MFCCs and the log-energy

In the second step, uncertainty is propagated to the vector $\mathbf{z}_n$ consisting of the static MFCCs and the log-energy. This vector may be computed using the nonlinear function $\mathcal{F}$

$$\mathbf{z}_n = \mathcal{F}(\mathbf{v}_n) = \bar{\mathbf{L}}\bar{\mathbf{D}}\log\left(\bar{\mathbf{M}}\bar{\mathbf{E}}\mathbf{v}_n\right) \tag{5.10}$$

where $\bar{\mathbf{E}}$, $\bar{\mathbf{M}}$, $\bar{\mathbf{D}}$ and $\bar{\mathbf{L}}$, are expanded versions of the pre-emphasis matrix, the Mel filterbank matrix, the discrete cosine transform (DCT) matrix, and the liftering matrix, respectively. More specifically, these matrices are defined as

$$\bar{\mathbf{E}} = \begin{bmatrix} \mathbf{Diag}(\mathbf{e}) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_F \end{bmatrix} \tag{5.11}$$

$$\bar{\mathbf{M}} = \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{j}_F \end{bmatrix} \tag{5.12}$$

$$\bar{\mathbf{D}} = \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \tag{5.13}$$

$$\bar{\mathbf{L}} = \begin{bmatrix} \mathbf{Diag}(\mathbf{l}) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \tag{5.14}$$

where $\mathbf{I}_F$ is the identity matrix of size F, $\mathbf{j}_F$ is a $1 \times F$ vector of ones, $\mathbf{Diag}(.)$ is the diagonal matrix built from its vector argument, $\mathbf{e}$ and $\mathbf{l}$ are the vectors of pre-emphasis and liftering coefficients, and $\mathbf{M}$ and $\mathbf{D}$ are the usual Mel filterbank and DCT matrices, respectively. Following the improvement demonstrated by VTS over other techniques in [Ozerov et al., 2013], $\mathcal{F}$ is approximately linearized by its first-order VTS expansion [Ozerov et al., 2013]. The mean and the covariance of $\mathbf{z}_n$ are therefore computed as

$$\widehat{\boldsymbol{\mu}}_{\mathbf{z}_n} = \mathcal{F}(\widehat{\boldsymbol{\mu}}_{\mathbf{v}_n}) = \bar{\mathbf{L}}\bar{\mathbf{D}}\log\left(\bar{\mathbf{M}}\bar{\mathbf{E}}\widehat{\boldsymbol{\mu}}_{\mathbf{v}_n}\right) \tag{5.15}$$

$$\widehat{\boldsymbol{\Sigma}}_{\mathbf{z}_n} = \mathcal{J}_{\mathcal{F}}\left(\widehat{\boldsymbol{\mu}}_{\mathbf{v}_n}\right)\widehat{\boldsymbol{\Sigma}}_{\mathbf{v}_n}\mathcal{J}_{\mathcal{F}}\left(\widehat{\boldsymbol{\mu}}_{\mathbf{v}_n}\right)^T \tag{5.16}$$

with the Jacobian matrix $\mathcal{J}_{\mathcal{F}}\left(\widehat{\boldsymbol{\mu}}_{\mathbf{v}_n}\right)$ given by

$$\mathcal{J}_{\mathcal{F}}\left(\widehat{\boldsymbol{\mu}}_{\mathbf{v}_n}\right) = \bar{\mathbf{L}}\bar{\mathbf{D}}\mathbf{Diag}\left(\frac{1}{\bar{\mathbf{M}}\bar{\mathbf{E}}\widehat{\boldsymbol{\mu}}_{\mathbf{v}_n}}\right)\bar{\mathbf{M}}\bar{\mathbf{E}} \tag{5.17}$$

where the division is performed elementwise.

Uncertainty	Uncertain	Test set						
covariance matrix	features	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average
no uncertainty		73.75	78.42	84.33	89.50	91.83	92.25	85.01
diagonal	static	75.00	79.00	84.75	90.13	91.92	93.67	85.74
	dynamic	75.00	79.00	84.92	90.33	91.92	92.33	85.58
	all	76.93	79.17	85.92	90.00	92.00	93.75	86.29
full	static	76.75	79.33	85.50	90.33	92.33	93.67	86.31
	dynamic	76.75	79.17	85.75	90.33	92.00	93.83	86.30
	all	77.92	80.75	86.75	90.50	92.92	93.75	**87.00**

Table 5.1: Keyword accuracy (%) on the Track 1 dataset achieved by uncertainty decoding of static and dynamic features. Average accuracies have a 95% confidence interval of ±0.8%. The full results can be found in Appendix A.3

5.2.3 To the full feature vector

In the third step, we propagate the uncertainty about the static features to the full feature vector. The static features in the preceding 4 frames, in the current frame, and in the following 4 frames are concatenated into a column vector $\bar{\mathbf{z}}_n = [\mathbf{z}_{n-4}^T \ \mathbf{z}_{n-3}^T \ldots \mathbf{z}_{n+4}^T]^T$. The full feature vector $\mathbf{c}_n = [\mathbf{z}_n \ \Delta\mathbf{z}_n \ \Delta^2\mathbf{z}_n]$ can be expressed in matrix form as

$$\mathbf{c}_n = (\mathbf{A} \otimes \mathbf{I}_C)\bar{\mathbf{z}}_n \tag{5.18}$$

where ⊗ is the Kronecker product, $\mathbf{I}_C$ the identity matrix of size $C = 13$, and the matrix $\mathbf{A}$ is given by (4.30). The mean and the covariance matrix of $\mathbf{c}_n$ are derived as

$$\widehat{\boldsymbol{\mu}}_{\mathbf{c}_n} = (\mathbf{A} \otimes \mathbf{I}_C)\widehat{\boldsymbol{\mu}}_{\bar{\mathbf{z}}_n} \tag{5.19}$$

$$\widehat{\boldsymbol{\Sigma}}_{\mathbf{c}_n} = (\mathbf{A} \otimes \mathbf{I}_C)\widehat{\boldsymbol{\Sigma}}_{\bar{\mathbf{z}}_n}(\mathbf{A} \otimes \mathbf{I}_C)^T \tag{5.20}$$

where $\widehat{\boldsymbol{\mu}}_{\bar{\mathbf{z}}_n}$ and $\widehat{\boldsymbol{\Sigma}}_{\bar{\mathbf{z}}_n}$ are obtained by concatenating $\widehat{\boldsymbol{\mu}}_{\mathbf{z}_{n-4}}, \ldots, \widehat{\boldsymbol{\mu}}_{\mathbf{z}_{n-4}}$ into a column vector and $\widehat{\boldsymbol{\Sigma}}_{\mathbf{z}_{n-4}}, \ldots, \widehat{\boldsymbol{\Sigma}}_{\mathbf{z}_{n+4}}$ into a block-diagonal matrix. The full uncertainty covariance matrix $\widehat{\boldsymbol{\Sigma}}_{\mathbf{c}_n}$ is then exploited to dynamically adapt the recognizer using the uncertainty decoding rule (4.35).

5.3 Experimental results

The resulting ASR accuracy on the Track 1 dataset is reported in Table 5.1. Similar trends are observed on development and test data. We recall that, on average over all SNRs in the test set, the baseline accuracy with conventional decoding (no uncertainty) is 85.01%. State-of-the-art uncertainty decoding with diagonal uncertainty covariance on static features (and no uncertainty on dynamic features) increases the accuracy to 85.74%, that is 4% relative error rate reduction with respect to the baseline. Using the full uncertainty covariance and modeling the uncertainty

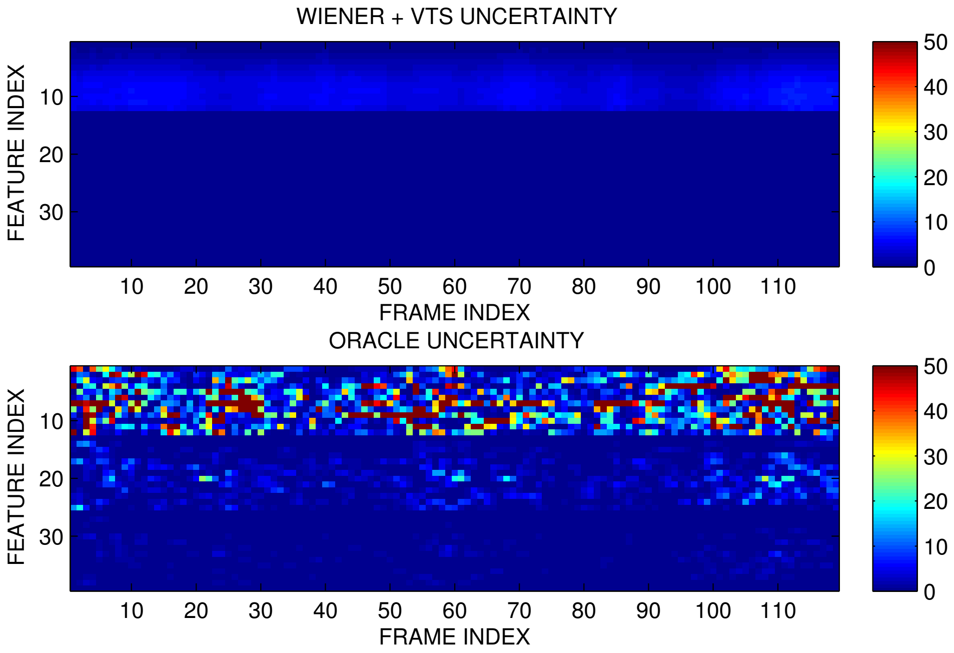


Figure 5.2: Example of estimated uncertainty covariance using Wiener + VTS and full oracle uncertainty covariance matrix over time. This example corresponds to the utterance 'bin blue with s nine soon' on the Track 1 dataset. Top: uncertainty is estimated using Wiener + VTS. Bottom: oracle uncertainty obtained using equation (4.44).

over the dynamic features systematically improve the average performance. The best system using full uncertainty covariance on all features achieves 87.00% accuracy. This corresponds to 13% relative error rate reduction with respect to the baseline, that is three times more than the reduction achieved with diagonal uncertainty covariance for static features.

In an additional experiment (not shown in the table), we evaluated the performance obtained with full uncertainty covariance matrices where the covariance coefficients between the features derived from the magnitude spectrum (MFCCs and their derivatives) and the features derived from the power spectrum (log-energy and its derivatives) have been zeroed out. The resulting ASR performance was equal to that obtained from the full uncertainty covariance matrix which justifies the approximation made by Astudillo [Astudillo and Kolossa, 2011]. As a matter of fact, covariance coefficients far from the diagonal tend to be smaller compared to those close to the diagonal. The covariance coefficients between MFCCs and log-energy also tend to be quite small, despite the fact that the magnitude and power spectra from which they are derived are strongly correlated. This indicates that the estimation of full uncertainty covariance matrices may be simplified by separately estimating and propagating uncertainty for the MFCCs and their derivatives on the one hand and for the log-energy and its derivatives on the other hand.

Performing uncertainty decoding for diagonal uncertainty covariance matrix achieved 86.30%

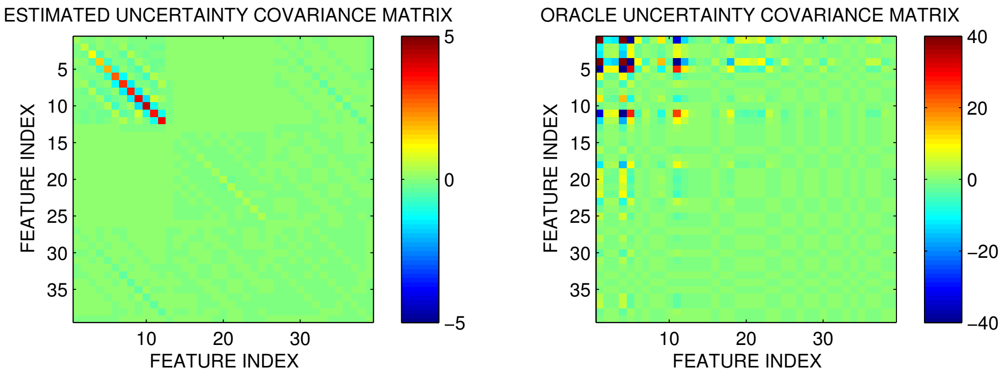


Figure 5.3: Example of full uncertainty covariance matrix on one time frame. This example corresponds to frame 50 of the utterance 'bin blue with s nine soon' on the Track 1 dataset. Left side: uncertainty estimated using Wiener + VTS. Right side: oracle uncertainty obtained using equation (4.45).

accuracy which correspond to 8% relative error rate reduction compared to baseline (enhancement only). Figure 5.2 show an example of estimated diagonal uncertainty covariance matrix over time. The estimated diagonal uncertainties are underestimated in all frames.

An example of estimated full uncertainty covariance matrix is show in Figure 5.3. The off-diagonal elements of the estimated full uncertainty covariance matrix are nonzero except for the elements modeling the uncertainties between the MFCCs and their derivatives. In addition, again, the diagonal elements of the estimated uncertainty covariance matrix look much smaller compare to those of the oracle uncertainty covariance matrix. This underestimation problem is addressed in the next Chapter.

5.4 Summary

In this chapter we presented a technique to propagete uncertainty through MFCC computation that results in a full uncertainty covariance matrix. Experimental results on Track 1 of the 2nd CHiME challenge show that full uncertainty covariance matrices achieved a 5% relative WER reduction compare to diagonal uncertainty matrices. However, modeling the correlation of uncertainty between the MFCCs and the log-energy does not seem to improve ASR performance.

6

Generative learning based uncertainty estimator

This chapter presents the proposed generative learning based uncertainty estimator. The motivation of this work is presented in Section 6.1. The theoretical idea is presented in Section 6.2. Section 6.3 explains how this generative method can learn parameters from the data. The experimental evaluation is presented in Sections 6.4 and 6.5. This work was published in [Tran et al., 2014b; Tran et al., 2015c].

6.1 Motivation

Although the proposed extension to a full uncertainty covariance matrix improves ASR performance, the estimated uncertainty is still inaccurate. As can be seen by comparing Tables 4.1 and 5.1, estimating the full uncertainty covariance matrix improved the accuracy from 85.01% to 87.00% but this is still far from the accuracy of 96.31% achieved by the oracle full uncertainty covariance matrix. It turns out that there is still a big gap between the estimated uncertainty and the oracle uncertainty in general. In particular, as shown in Figures 5.2 and 5.3, the estimated uncertainty is underestimated compared to the oracle one.

Existing methods typically rely on a single uncertainty estimator and propagator fixed by mathematical approximation. As shown in Figure 4.2, the three spectral-domain uncertainty estimators introduced in Section 4.3 have different behaviors. Kolossa's estimator decreases when the estimated speech power spectrum increases. The two other estimators reach a maximum when the estimated power spectra of speech and noise are equal. Nesta's estimator increases more quickly than the Wiener estimator. Motivated by this observation, we now propose a new paradigm where we seek to learn more powerful mappings to predict uncertainty from data. We investigate two such possible mappings: linear fusion of multiple uncertainty estimators/propagators and nonparametric uncertainty estimation/propagation.

6.2 Proposed fusion and nonparametric estimation framework

We now present the proposed fusion and nonparametric estimators. The learning of the corresponding fusion weights and kernel weights is addressed later in Section 6.3. We first focus on uncertainty estimation in the spectral domain and then on uncertainty propagation to the feature domain.

6.2.1 Fused/nonparametric uncertainty estimation

Fusion

A first idea is to fuse multiple uncertainty estimators by linear combination in order to obtain a more accurate estimator. This is a form of early fusion. In the following, we assume that the fusion weights depend on frequency f but that they are independent of the signal-to-noise ratio and the HMM states. Indeed, the signal-to-noise ratio in each time-frequency bin is typically unknown and the uncertainty represents the variance of speech distortion, which depends on the speech enhancement technique but not on the GMM-HMM subsequently used for decoding. Denoting by E the number of estimators, the fused estimator $(\widehat{\sigma}^{\text{fus}}_{s_{fn}})^2$ can be expressed as

$$(\widehat{\sigma}^{\text{fus}}_{s_{fn}})^2 = \sum_{e=1}^{E} \theta^e_{s_f} (\widehat{\sigma}^e_{s_{fn}})^2 \tag{6.1}$$

where $(\widehat{\sigma}^e_{s_{fn}})^2$ is one of the original estimators in (4.7), (4.9), (4.10), and $\theta^e_{s_f}$ are the fusion coefficients. The fusion coefficients are constrained to be nonnegative so that the fused estimator is always nonnegative. Stacking the original uncertainty estimates over all time frames into an $E \times N$ matrix $\widehat{\mathbf{\Lambda}}_{s_f}$ and the fused estimates into a $1 \times N$ vector $\widehat{\boldsymbol{\lambda}}^{\text{fus}}_{s_f}$ for each frequency f, where N is the number of frames, (6.1) can be written in matrix form as

$$\widehat{\boldsymbol{\lambda}}^{\text{fus}}_{s_f} = \boldsymbol{\theta}_{s_f} \widehat{\mathbf{\Lambda}}_{s_f} \tag{6.2}$$

where $\boldsymbol{\theta}_{s_f}$ is the $1 \times E$ vector of fusion coefficients. In order to compensate for possible additive bias in the original uncertainty estimates, we also add a nonnegative frequency-dependent bias. This is simply achieved by adding a row of ones to the matrix $\widehat{\mathbf{\Lambda}}_{s_f}$ and a corresponding coefficient in $\boldsymbol{\theta}_{s_f}$ for the bias value.

Nonparametric mapping

Although the fused uncertainty estimator potentially improves over the original fixed estimators, its shape remains constrained by these original estimators. This motivates us to learn the full shape of the estimator from data in a nonparametric fashion. To do this, we express the uncertainty in the same way as in (6.1), but we replace the existing estimators $(\widehat{\sigma}^e_{s_{fn}})^2$ by *kernels* $(\widehat{\sigma}^e_{s_{fn}})^2$.

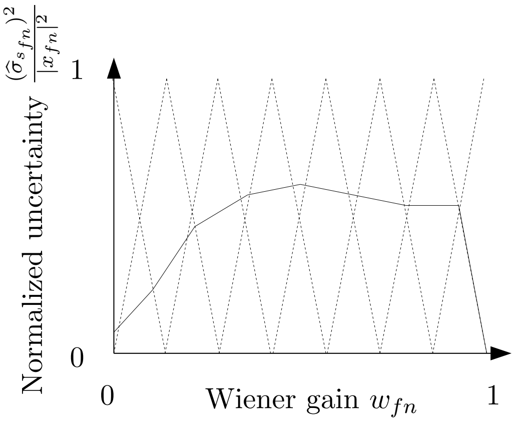


Figure 6.1: $E = 8$ triangular kernels $\mathrm{b}^e(w_{fn})$ (dotted) and example of resulting uncertainty estimator $(\widehat{\sigma}^{\text{fus}}_{sfn})^2/|x_{fn}|^2$ (plain). The horizontal axis represents the estimated proportion of speech in the observed mixture power spectrum, as defined in (6.3). The vertical axis is proportional to uncertainty. The uncertainty is normalized by the mixture power spectrum to emphasize the fact that the shape of the estimator (i.e., the plain curve) doesn't depend on it. Notice that the plain curve is obtained by summing the triangular dotted curves with different nonnegative weights.

As it appears from Section 4.3 and Figure 4.2, most uncertainty estimators share two properties. First, the estimated uncertainty is proportional to the mixture power spectrum. Second, they can be expressed as a function of the Wiener gain, that is the ratio of the speech power spectrum and the mixture power spectrum. In the multichannel case, we define the Wiener gain w_{fn} as

$$w_{fn} = \frac{1}{I}\,\mathrm{tr}(\mathbf{W}_{fn}) \tag{6.3}$$

where $\mathbf{W}_{fn}$ is the multichannel Wiener filter defined in (4.4) and I is the number of channel. By property of the multichannel Wiener filter, w_{fn} is real-valued and between 0 and 1. Based on these two properties, we define the kernels as

$$(\widehat{\sigma}^e_{sfn})^2 = |x_{fn}|^2 \mathrm{b}^e(w_{fn}) \tag{6.4}$$

where $\mathrm{b}^e(.)$ are a set of normalized kernel functions on $[0, 1]$ indexed by $e \in \{1, \ldots, E\}$. In the following, we choose triangular kernels

$$\mathrm{b}^e(w_{fn}) = (E - 1)\max(0, 1 - |(E - 1)w_{fn} - (e - 1)|) \tag{6.5}$$

which results in a piecewise linear mapping. The weights θ^e_{sf} encode the value of the mapping when $w_{nf} = (e - 1)/(E - 1)$. Figure 6.1 shows the shape of the kernels and one example of resulting mapping. The number of kernels E governs the precision of the uncertainty estimates. A bigger E potentially increases accuracy, but too large E results in overfitting.

6.2.2 Fused/nonparametric uncertainty propagation with diagonal covariance

The estimated spectral-domain uncertainties are propagated to the feature domain by VTS. Although this results in better feature-domain uncertainties than unscented transform or moment matching, we found experimentally these feature-domain uncertainties to be still underestimated. This may be due to the initial assumption that spectral-domain uncertainties are independent across time-frequency bins, as well as to the approximations involved in VTS. The estimation of the correlation of uncertainties across time-frequency bins appears to be a difficult far-end goal. Therefore, we propose to learn from data how to correct the estimated uncertainties. Let us consider first the case of a diagonal uncertainty covariance matrix $\widehat{\mathbf{\Sigma}}_{\mathbf{c}_n}$.

Rescaling

A first way of correcting the underestimation is to rescale the coefficients of the estimated uncertainty covariance matrix as [Delcroix et al., 2009]

$$\widehat{\mathbf{\Sigma}}^{\text{scaled}}_{\mathbf{c}_n} = \mathbf{Diag}(\mathbf{g})\widehat{\mathbf{\Sigma}}_{\mathbf{c}_n} \tag{6.6}$$

where $\mathbf{g}$ is a 39×1 vector of nonnegative scaling coefficients.

Fusion

One may also keep several spectral-domain uncertainty estimates by applying the fused estimator in (6.7) with different values of the fusion weights $\theta^e_{s_f}$ and propagate each of them to the feature domain, yielding P feature-domain uncertainty estimates $(\widehat{\sigma}^p_{c_{in}})^2$ indexed by p for each feature index i. These uncertainty estimates may then be fused similarly to (6.7) as

$$(\widehat{\sigma}^{\text{fus}}_{c_{in}})^2 = \sum_{p=1}^{P} \theta^p_{c_i} (\widehat{\sigma}^p_{c_{in}})^2. \tag{6.7}$$

In the following, we assume that the fusion weights $\theta^p_{c_i}$ depend on the feature index i but that they are independent of the signal-to-noise ratio and the HMM state. The fused uncertainty propagator can also be expressed in vector form as

$$\widehat{\boldsymbol{\lambda}}^{\text{fus}}_{c_i} = \boldsymbol{\theta}_{c_i} \widehat{\mathbf{\Lambda}}_{c_i} \tag{6.8}$$

where $\widehat{\boldsymbol{\lambda}}^{\text{fus}}_{c_i}$ is the $1 \times N$ vector of fused estimates, $\boldsymbol{\theta}_{c_i}$ is the $1 \times P$ vector of fusion coefficients and $\widehat{\mathbf{\Lambda}}_{c_i}$ is a $P \times N$ matrix whose elements are $(\widehat{\sigma}^p_{c_{in}})^2$. This expression generalizes (6.6) to the case of multiple feature-domain uncertainty estimates.

Nonparametric mapping

Finally, we can estimate the uncertainty nonparametrically by applying (6.7) where $(\widehat{\sigma}^p_{c_{in}})^2$ are a set of kernels indexed by $p \in \{1, \dots, P\}$. In the following, we choose triangular kernels defined as

$$(\widehat{\sigma}^p_{c_{in}})^2 = (P-1)\max(0, 1 - |(P-1)(\bar{\sigma}_{c_{in}})^2 - (p-1)|) \tag{6.9}$$

where $(\bar{\sigma}_{c_{in}})^2$ is the result of linearly normalizing the nonparametric feature-domain uncertainty estimator $(\widehat{\sigma}_{c_{in}})^2$ to the interval $[0, 1]$ for each feature index i. More precisely, $(\widehat{\sigma}_{c_{in}})^2$ is computed by propagating to the feature domain the nonparametric spectral-domain estimator previously obtained for one value of the weights $\theta^e_{s_f}$ and

$$(\bar{\sigma}_{c_{in}})^2 = [(\widehat{\sigma}_{c_{in}})^2 - (\widehat{\sigma}_{c_{i,\min}})^2]/[(\widehat{\sigma}_{c_{i,\max}})^2 - (\widehat{\sigma}_{c_{i,\min}})^2] \tag{6.10}$$

where $(\widehat{\sigma}_{c_{i,\min}})^2$ and $(\widehat{\sigma}_{c_{i,\max}})^2$ are the minimum and maximum value of $(\widehat{\sigma}_{c_{in}})^2$ observed on the development set for a given i.

6.2.3 Fused/nonparametric uncertainty propagation with full covariance

To exploit the full benefit of uncertainty decoding, a full uncertainty covariance matrix is needed. The extension of (6.8) to full covariance matrices is not trivial, however. Therefore, we first estimate a mapping for the diagonal $\mathbf{diag}(\widehat{\mathbf{\Sigma}}_{\mathbf{c}_n})$, where $\mathbf{diag}(.)$ is the vector consisting of the diagonal entries of its matrix argument, and we apply them to the full matrix $\widehat{\mathbf{\Sigma}}_{\mathbf{c}_n}$ using the following heuristic approach. We compute a vector of equivalent rescaling coefficients as

$$\mathbf{g} = \frac{\mathbf{diag}(\widehat{\mathbf{\Sigma}}^{\text{fus}}_{\mathbf{c}_n})}{\mathbf{diag}(\widehat{\mathbf{\Sigma}}_{\mathbf{c}_n})} \tag{6.11}$$

where the division is performed elementwise, and we apply them to the full matrix as

$$\widehat{\mathbf{\Sigma}}^{\text{fus}}_{\mathbf{c}_n} = \mathbf{Diag}(\mathbf{g})^{1/2}\widehat{\mathbf{\Sigma}}_{\mathbf{c}_n}\,\mathbf{Diag}(\mathbf{g})^{1/2}. \tag{6.12}$$

This approach is applicable to the three methods presented above (rescaling, fusion, and nonparametric mapping) and it ensures that the positive semi-definiteness of the full covariance matrix is preserved.

6.3 Learning of fusion/nonparametric coefficients

The uncertainty estimators presented in the previous section rely on a set of weights. We propose to learn these weights on development data for which the true speech signal is known such that the resulting uncertainty estimates are as close as possible to the oracle uncertainty as defined in Section 4.7.

6.3.1 Weighted divergence measures

For the three proposed approaches (rescaling, fusion, and nonparametric mapping), we optimize the weights on development data by minimizing some measure of divergence between the estimated uncertainties and the oracle uncertainties. There are many possible choices of divergences, including the well-known Itakura-Saito (IS), Kullback-Leibler (KL), and squared Euclidean (EUC) divergences, which belong to the family of β-divergences with $\beta = 0$, 1, or 2, respectively [Kompass, 2007], and more general Bregman divergences. These divergences can be characterized by two main properties: their shape, i.e., how they penalize underestimation and overestimation with respect to each other, and their scale, i.e., how they vary with respect to the scale of the input.

The scale property is particularly important in our context since the scale of speech spectra is extremely variable from one frame to another and the scale of features is extremely variable from one feature index to another. We therefore consider the minimization of the following weighted β-divergence measures

$$\boldsymbol{\theta}_{s_f} = \arg\min_{\boldsymbol{\theta}_{s_f} \geq 0} \sum_n |x_{fn}|^{\alpha - 2\beta} d_\beta \left((\sigma_{s_{fn}})^2 | (\boldsymbol{\theta}_{s_f} \widehat{\boldsymbol{\Lambda}}_{s_f})_n \right) \tag{6.13}$$

$$\boldsymbol{\theta}_{c_i} = \arg\min_{\boldsymbol{\theta}_{c_i} \geq 0} \sum_n (\tilde{\sigma}_{c_i})^\alpha d_\beta \left((\sigma_{c_{in}})^2 | (\boldsymbol{\theta}_{c_i} \widehat{\boldsymbol{\Lambda}}_{c_i})_n \right) \tag{6.14}$$

where $(\sigma_{s_{fn}})^2$ and $(\sigma_{c_{in}})^2$ are the oracle uncertainties in time frame n, $(\boldsymbol{\theta}_{s_f} \widehat{\boldsymbol{\Lambda}}_{s_f})_n$ and $(\boldsymbol{\theta}_{c_i} \widehat{\boldsymbol{\Lambda}}_{c_i})_n$ are the estimated uncertainties in that time frame, $d_\beta(x|y)$ is the β-divergence between two scalars, and $\tilde{\sigma}_{c_i}$ is the standard deviation of the features defined by

$$\tilde{\sigma}_{c_i} = \sqrt{\frac{1}{N} \sum_n c_{in}^2 - \left(\frac{1}{N} \sum_n c_{in} \right)^2}. \tag{6.15}$$

In the following, we consider three particular divergences: the IS divergence

$$d_0(x|y) = \frac{x}{y} - \log \frac{x}{y} - 1, \tag{6.16}$$

the KL divergence

$$d_1(x|y) = x \log \frac{x}{y} - x + y, \tag{6.17}$$

and the squared EUC distance

$$d_2(x|y) = (x - y)^2. \tag{6.18}$$

The exponent α governs the scale of the divergence. For instance, the value $\alpha = 0$ corresponds to a scale-invariant spectral-domain divergence[8]. The spectral-domain divergences corresponding to $\alpha = 1$ and $\alpha = 2$ scale with the magnitude and the squared magnitude of the signal, respectively.

[8] Note that, according to (6.4) and (6.16)–(6.18), the β-divergence in (6.13) scales with $|x_{fn}|^{2\beta}$ hence the normalized β-divergence scales with $|x_{fn}|^\alpha$.

6.3.2 Multiplicative update rules

The optimization problems (6.13) and (6.14) are instances of weighted nonnegative matrix factorization (NMF) [Lee and Seung, 1999]. The fusion coefficients are found by applying the following iterative multiplicative updates [Kompass, 2007]:

$$\boldsymbol{\theta}_{s_f} \leftarrow \boldsymbol{\theta}_{s_f} \odot \frac{\left(\boldsymbol{\zeta}_{s_f} \odot (\boldsymbol{\theta}_{s_f}\widehat{\boldsymbol{\Lambda}}_{s_f})^{\beta-2} \odot \boldsymbol{\Lambda}_{s_f}\right)(\widehat{\boldsymbol{\Lambda}}_{s_f})^T}{\left(\boldsymbol{\zeta}_{s_f} \odot (\boldsymbol{\theta}_{s_f}\widehat{\boldsymbol{\Lambda}}_{s_f})^{\beta-1}\right)(\widehat{\boldsymbol{\Lambda}}_{s_f})^T} \tag{6.19}$$

$$\boldsymbol{\theta}_{c_i} \leftarrow \boldsymbol{\theta}_{c_i} \odot \frac{\left(\boldsymbol{\zeta}_{c_i} \odot (\boldsymbol{\theta}_{c_i}\widehat{\boldsymbol{\Lambda}}_{c_i})^{\beta-2} \odot \boldsymbol{\Lambda}_{c_i}\right)(\widehat{\boldsymbol{\Lambda}}_{c_i})^T}{\left(\boldsymbol{\zeta}_{c_i} \odot (\boldsymbol{\theta}_{c_i}\widehat{\boldsymbol{\Lambda}}_{c_i})^{\beta-1}\right)(\widehat{\boldsymbol{\Lambda}}_{c_i})^T} \tag{6.20}$$

where $\odot$ denotes element-wise multiplication and powers are computed element-wise, $\boldsymbol{\Lambda}_{s_f}$ and $\boldsymbol{\Lambda}_{c_i}$ are the $1 \times N$ vectors of oracle uncertainties, $\boldsymbol{\zeta}_{s_f}$ is the $1 \times N$ vector with entries $|x_{fn}|^{\alpha-2\beta}$, and $\boldsymbol{\zeta}_{c_i}$ is the $1 \times N$ vector whose entries are all equal to $(\tilde{\sigma}_{c_i})^{\alpha}$.

The coefficients $\boldsymbol{\theta}_{s_f}$ and $\boldsymbol{\theta}_{c_i}$ estimated on the development data are then applied to the test data.

6.4 Experimental evaluation on Track 1

In order to assess the proposed framework, we perform a first set of experiments on Track 1 of the 2nd CHiME Challenge.

6.4.1 Estimated fusion/nonparametric coefficients

Figure 6.2 represents the optimal fusion coefficients estimated on the development set for Kolossa's, Wiener, and Nesta's estimators. The resulting spectral-domain estimator in Figure 6.2a is a scaled version of Nesta's at higher frequencies and a combination of Wiener and Nesta's at lower frequencies, while the resulting feature-domain estimator in Figure 6.2b is mostly a scaled version of the KL-fused estimator with some additive bias on the static features.

The weights for the nonparametric estimators are trained on the development set. The optimal parameter choices found on the development set are $E = 200$, $P = 400$, $\alpha = 2$ and $\beta = 1$ in the spectral domain, and $\alpha = 0$ and $\beta = 1$ in the feature domain. The impact of these choices is discussed in Section 6.4.2. Figure 6.3 illustrates the nonparametric mappings learned from the development set. Contrary to Wiener and Nesta's estimators, the learned spectral-domain mapping has an asymmetric shape and it varies with frequency. It is interesting to note that the mapping value at very low frequencies remains large for Wiener gain values close to 1, which is consistent with the fact that there is no speech at these frequencies. The learned feature-domain mapping is more difficult to interpret, as it does not monotonously increase with respect to $(\bar{\sigma}_{c_{in}})^2$ as one would expect. Nevertheless, the learned uncertainty is larger for static

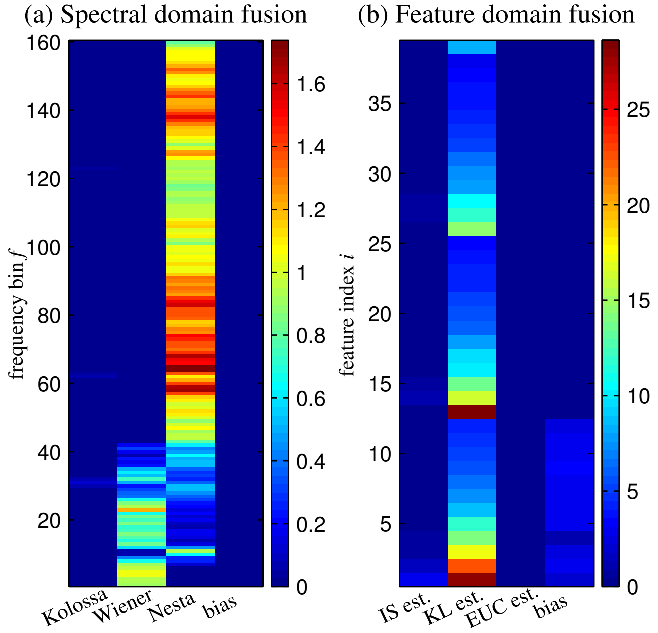


Figure 6.2: (a) Learned spectral-domain fusion coefficients $\boldsymbol{\theta}_{s_f}$ with $\alpha = 0$ and $\beta = 1$ on the Track 1 dataset; the horizontal axis corresponds to the three existing estimators listed in Section 4.3, and "bias" refers to the additive bias as explained in Section 6.2.1. (b) Learned feature-domain fusion coefficients $\boldsymbol{\theta}_{c_i}$ with $\alpha = 0$ and $\beta = 1$ on the Track 1 dataset; the vertical axis is the feature index (in the following order: 12 MFCCs, log-energy, first-order derivatives, and second-order derivatives), the horizontal axis corresponds to the three feature-domain estimators IS est., KL est., and EUC est. resulting from spectral-domain fusion with $\alpha = 0$ and $\beta = 0$, 1, or 2, respectively, and "bias" refers to the additive bias as explained in Section 6.2.1. In both subfigures, darker color corresponds to a larger weight.

MFCCs than for delta and delta-delta MFCCs, which is consistent with the fact that the value range is larger for the former.

6.4.2 ASR results

Fusion and nonparametric mapping

Table 6.1 shows the results achieved with fusion or nonparametric mapping. Similar trends are seen for diagonal and full uncertainty covariances. In the following, we comment the latter only.

Compared to Wiener+VTS alone, feature-domain uncertainty rescaling improves the average accuracy on the test set from 87.00% to 88.11% . This is already a significant improvement, which confirms that the uncertainties estimated by state-of-the-art techniques must be rescaled in order to match the actual uncertainty in the data.

By fusing Kolossa's, Wiener, and Nesta's uncertainty estimators, performance improves to

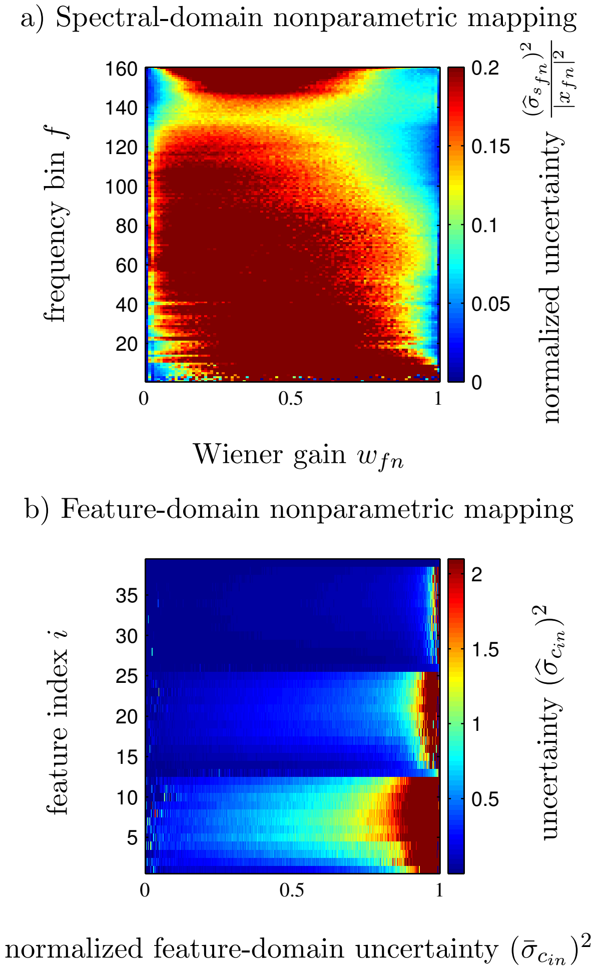


Figure 6.3: (a) Learned spectral-domain nonparametric mapping $\boldsymbol{\theta}_{s_f}$ with $\alpha = 2$, $\beta = 1$, $E = 200$ on the Track 1 dataset; the horizontal axis represents the estimated proportion of speech in the observed mixture power spectrum as defined in (6.3); the color scale is proportional to uncertainty, where darker color means higher uncertainty. (b) Learned feature-domain nonparametric mapping $\boldsymbol{\theta}_{c_i}$ with $\alpha = 0$, $\beta = 1$, $P = 400$ on the Track 1 dataset; the horizontal axis is proportional to the uncertainty estimated by VTS propagation of the estimates in subfigure (a); the color scale represents the learned uncertainty. The horizontal axes in both subfigures and the color scale in subfigure (a) are normalized to emphasize the fact that the shape of the mappings doesn't depend on $|x_{fn}|^2$ and $(\bar{\sigma}_{c_{in}})^2$.

88.20%. Further fusing the IS-fused estimator, the KL-fused estimator and the EUC-fused estimator in the feature domain yields 89.18% accuracy, that is 28% relative error rate reduction compared to conventional decoding and 9% with respect to rescaling.

Using a nonparametric mapping in both the spectral and the feature domains results in 89.42% keyword accuracy, that is 29% relative error rate reduction compared to conventional

Estimation	Propagation	Uncertainty covariance matrix	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average
Wiener	VTS+Scaling	diagonal	78.67	79.50	86.33	90.17	92.08	93.75	86.75
fusion	VTS		78.33	80.17	85.92	90.08	92.08	94.17	86.97
fusion	fusion		80.50	82.17	88.25	91.33	92.50	93.58	88.05
nonparametric	VTS		80.00	81.92	87.25	91.50	92.25	93.08	87.66
nonparametric	nonparametric		81.75	83.50	88.33	91.08	92.75	93.00	**88.40**
Wiener	VTS+Scaling	full	81.75	81.83	88.17	90.50	92.67	93.75	88.11
fusion	VTS		81.00	81.50	87.33	91.00	93.50	94.92	88.20
fusion	fusion		83.17	84.33	89.75	91.17	93.33	93.33	89.18
nonparametric	VTS		82.33	82.58	88.00	92.00	93.33	93.92	88.69
nonparametric	nonparametric		83.78	84.92	88.42	91.25	93.75	94.42	**89.42**

Table 6.1: Keyword accuracy (%) achieved with various fusion or nonparametric mapping schemes on the Track 1 test dataset. This is to be compared to the baseline Wiener+VTS performance in Table 5.1. The full results can be found in Appendix A.4

$P = 400$	Accuracy
$E = 10$	88.82
$E = 20$	88.76
$E = 40$	88.83
$E = 100$	88.90
$E = 200$	**89.00**
$E = 400$	88.83
$E = 1000$	88.67

Table 6.2: Average keyword accuracy (%) on the Track 1 development set for various numbers of kernels for the estimator when number of kernels for the propagator is fixed to 400.

decoding and 2% with respect to fusion. This is about twice larger than the improvements due to uncertainty decoding reported in the state of the art, that are typically on the order of 15% relative or less compared to conventional decoding [Delcroix et al., 2013b; Kallasjoki et al., 2014].

Compared to the oracle results of 94.57% and 96.31% for diagonal and full uncertainty covariance matrices respectively this is also a significant improvement. The proposed nonparametric estimator reduced the gap between no uncertainty and oracle uncertainty by 35% and 39% in the diagonal and full covariance, respectively.

Comparison with ROVER fusion

For comparison, we also evaluate the performance of recognizer output voting error reduction (ROVER) fusion [Fiscus, 1997] on the same data. We estimate spectral-domain uncertainty using Kolossa's, Wiener, and Nesta's estimators and propagate them to the feature domain using

$E = 200$	Accuracy
$P = 10$	88.61
$P = 20$	88.64
$P = 40$	88.78
$P = 100$	88.79
$P = 200$	88.90
$P = 400$	**89.00**
$P = 1000$	88.73

Table 6.3: Average keyword accuracy (%) on the Track 1 development set for various numbers of kernels for the propagator when number of kernels for the estimator is fixed to 200.

Uncertainty covariance matrix	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average
diagonal	79.08	80.75	86.00	90.17	92.08	94.17	87.04
full	81.33	81.75	87.50	91.08	93.75	94.92	88.38

Table 6.4: Keyword accuracy (%) achieved with ROVER fusion on the Track 1's test set. The results of the full dataset can be found in Appendix A.5

VTS. This results in three feature-domain uncertainty estimates. For each of these estimates, we compute the best ASR hypothesis with uncertainty decoding and also compute a confidence measure for each word. We then use ROVER to combine these three hypotheses. The results are shown in Table 6.4. Comparing with Table 6.1, we can see that our fusion/nonparametric framework outperforms ROVER in both the diagonal and the full uncertainty case. This might be explained by the fact that the estimated uncertainties are still underestimated and ROVER does not seem to be able to avoid this problem.

Impact of the number of kernels

We now evaluate the impact of the various parameter choices on the ASR performance. Table 6.2 and Table 6.3 illustrate the impact of the choice of the number of spectral-domain kernels E given the optimal number of feature-domain kernels P, and vice-versa. The best numbers are found to be $E = 200$ and $P = 400$. However, other values yield statistically equivalent ASR performance. The proposed nonparametric mapping framework is therefore robust to the choice of the number of kernels (provided it is in between 10 and 1000) for all α and β.

Impact of the choice of weighted divergences

With the best configuration of number of kernels for the estimator and the propagator found in Section 6.4.2, we evaluate the impact of the choice of weighted divergences. Tables 6.5 and 6.6 report the ASR results for different divergence parameters α and β in the spectral domain and in the feature domain. In either case, this choice has a minor impact on the resulting keyword

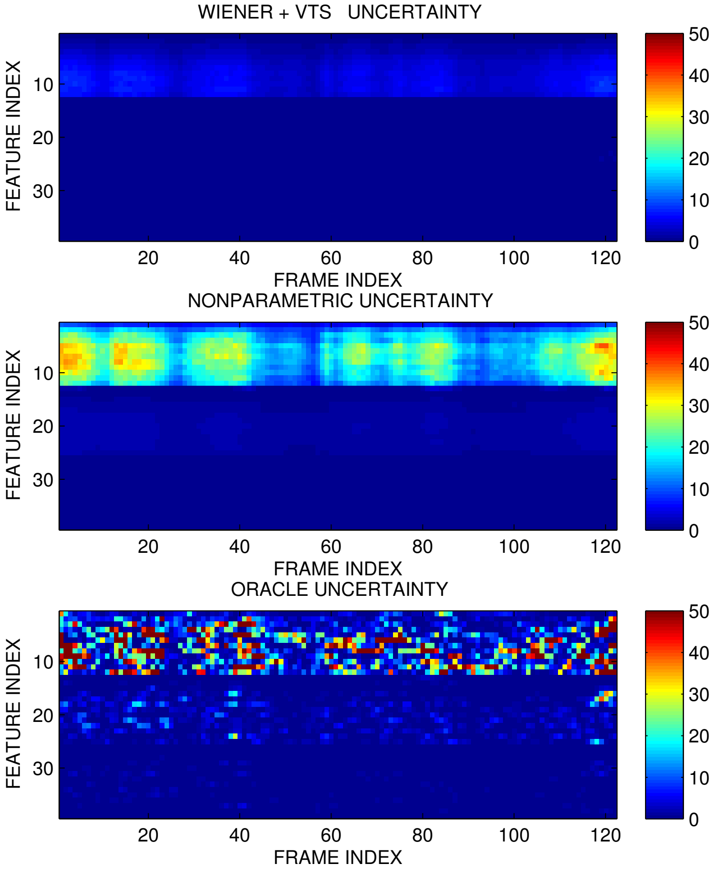


Figure 6.4: Example of uncertainty over time this example corresponds to the utterance "lay blue in d three soon" on the Track 1 dataset . Top: estimated uncertainty using Wiener + VTS. Middle: estimated nonparametric uncertainty, Bottom: oracle uncertainty

accuracy. The best choices appear to be weighted KL-divergences, namely $\alpha = 2$ and $\beta = 1$ in the spectral domain, and $\alpha = 0$ and $\beta = 1$ in the feature domain.

6.4.3 Accuracy of uncertainty estimation

Besides the induced ASR performance, we believe that it is important to evaluate our framework in terms of the resulting uncertainty estimation accuracy. Indeed, it is believed that uncertainty decoding will eventually improve the performance of DNN-based acoustic models by giving additional cues to the DNN about the distortion of the speech input [Seltzer, 2014; Seltzer et al., 2013; Li and Sim, 2014]. Therefore better uncertainty estimation will most probably result in better ASR performance in that context too.

To do so, we measure the weighted β-divergence obtained as the result of solving the minimization problems (6.13) and (6.14) for a given α and β. The results shown in Table 6.7

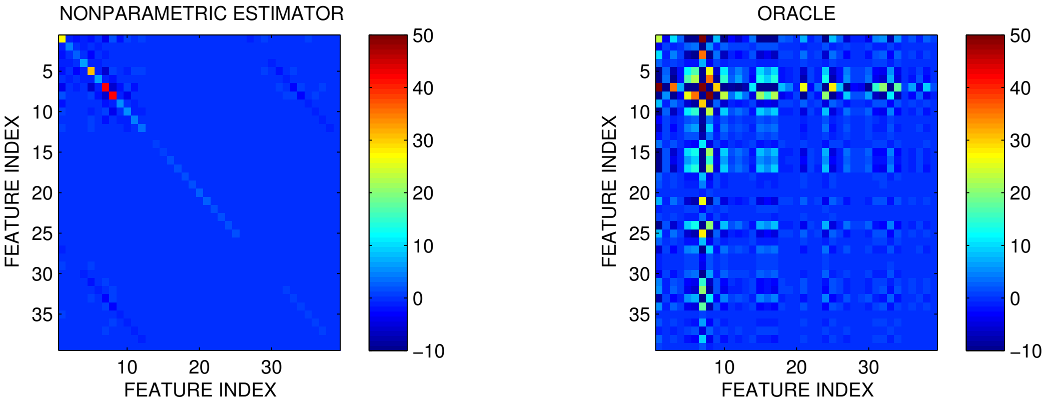


Figure 6.5: Example of full uncertainty covariance matrix. This example coresponds to frame of the utterance on the Track 1 dataset. Left: estimated nonparametric uncertainty. Right: oracle uncertainty.

indicate that fusion and nonparametric mapping improve the accuracy of the estimated uncertainty compared to Wiener + VTS both in the spectral domain and in the feature domain and that nonparametric uncertainty estimation and propagation provides the best results in all cases. Figure 6.4 shows an example of feature-domain nonparametric estimates. Compare to the uncertainty at the top of the figure which is estimated using Wiener + VTS, the nonparametric uncertainty at the middle of the figure addressed much better the underestimation of the diagonal elements. The uncertainties have low values at frame indexes of 24 to 26, 40 to 50, and 84 to 104. This is quite similar to the shape of the corresponding oracle uncertainty. However, the uncertainty values in each particular frame of the nonparametric estimator don't have the same shape on the corresponding oracle uncertainties.

Figure 6.5 shows a comparison between a full uncertainty covariance matrix using obtained nonparametric estimation and the oracle uncertainty covariance matrix. The off-diagonal elements still don't have the same shape as the corresponding oracle uncertainty. This is mainly due to the missing uncertainty correlation between spectral components in the spectral domain and also uncertainty correlation between MFCCs and their time derivatives.

6.5 Experimental evaluation on Track 2

In addition to the above experiments, we evaluated the ASR performance achieved on Track 2 of the 2nd CHiME Challenge [Vincent et al., 2013a].

6.5.1 Experimental setup

The ASR results are shown in Tables 6.8 and 6.9 for GMM-HMM and CD-DNN-HMM based acoustic modeling, respectively. Our nonparametric uncertainty estimation and propagation

Method		fusion			nonparametric		
Uncertainty covariance matrix	β \ α	0	1	2	0	1	2
diagonal	0	85.16	85.94	86.47	86.68	87.16	87.20
	1	86.55	86.65	**86.69**	86.92	87.18	**87.23**
	2	86.18	86.20	86.53	86.50	87.00	87.15
full	0	86.74	87.00	87.16	88.00	88.14	88.25
	1	87.58	87.63	**87.68**	87.93	88.12	**88.29**
	2	87.16	87.23	87.33	87.78	88.04	88.15

Table 6.5: Keyword accuracy (%) on the Track 1 development set for various weighted divergence choices in the spectral domain.

Method		fusion			nonparametric		
Uncertainty covariance matrix	β \ α	0	1	2	0	1	2
diagonal	0	86.49	86.02	85.91	86.85	86.34	86.17
	1	**87.33**	87.00	86.79	**87.95**	87.64	87.20
	2	86.72	86.58	86.27	87.22	87.00	86.68
full	0	88.57	88.51	88.49	88.73	88.65	88.41
	1	**88.73**	88.66	88.56	**89.00**	88.82	88.63
	2	88.63	88.62	88.16	88.84	88.60	88.56

Table 6.6: Keyword accuracy (%) on the Track 1 development set for various weighted divergence choices in the feature domain.

Method		Wiener+VTS			fusion			nonparametric		
Domain	β \ α	0	1	2	0	1	2	0	1	2
spectral	0	0.42	0.66	0.16	0.41	0.61	0.11	**0.10**	**0.14**	**0.07**
	1	0.21	0.65	0.37	0.17	0.32	0.12	**0.14**	**0.27**	**0.04**
	2	0.69	0.32	0.41	0.40	0.16	0.32	**0.30**	**0.12**	**0.24**
feature	0	80.8	70.5	60.9	70.2	65.1	57.8	**66.0**	**60.5**	**55.2**
	1	70.6	60.1	50.7	50.3	55.9	48.5	**45.8**	**51.0**	**46.1**
	2	70.9	65.3	60.6	60.7	61.0	58.8	**56.0**	**58.7**	**55.5**

Table 6.7: Average divergence between the estimated and the oracle uncertainty on the Track 1 development set for various choices of weighted divergence. Bold values indicate the lowest achieved divergence for a given value of α and β.

estimated uncertainty	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average
noisy (no uncertainty)	82.81	78.09	70.15	64.80	53.79	47.02	66.11
enhanced (no uncertainty)	68.47	63.75	56.76	51.03	44.22	39.12	53.89
Wiener + VTS (diagonal)	65.23	61.82	55.18	50.27	43.11	38.79	52.40
nonparametric + VTS (diagonal)	58.12	52.54	49.95	44.42	40.23	35.31	46.76
nonparametric + nonparametric (diagonal)	53.70	48.69	45.72	40.18	37.43	34.18	43.32
Wiener + VTS (full)	63.58	58.85	53.06	48.19	42.09	38.41	50.70
nonparametric + VTS (full)	51.91	46.95	43.48	38.91	35.11	32.55	41.49
nonparametric + nonparametric (full)	46.34	42.75	41.54	37.57	34.21	30.01	**38.74**

Table 6.8: WER (%) achieved on the Track 2 test set with GMM-HMM acoustic models trained on reverberated noiseless data. The full results can be found in Appendix A.6

condition	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average
noisy	50.33	40.82	30.71	24.60	21.43	16.87	30.79
enhanced	41.51	31.46	26.04	21.51	17.82	16.74	25.85

Table 6.9: WER (%) achieved on the Track 2 test set with a CD-DNN-HMM acoustic model trained on enhanced data. The full results can be found in Appendix A.7

framework still significantly outperforms the Wiener + VTS baseline and yields 28% relative WER reduction compared to conventional decoding. This shows that the accuracy of our uncertainty estimators is independent of the vocabulary size. The ASR results for CD-DNN-HMM based acoustic modeling is showed in Figure 6.9 and they clearly outperform GMM-HMM based acoustic modeling. This fusion/nonparametric framework is expected to have some impact on the structure of DNN acoustic models themselves, for which uncertainty decoding has recently started being investigated as one way of giving additional cues to the DNNs about the distortion of the speech input [Seltzer, 2014; Seltzer et al., 2013; Li and Sim, 2014].

6.6 Summary

In this chaper, we presented a fusion/nonparametric framework to learn uncertainty from data. We analyzed and compared the results for both a small vocabulary and a medium vocabulary task. The framework achieved 29% and 28% relative WER reduction on the small vocabulary and the medium vocabulary tasks compared to the baseline system (without uncertainty), respectively. We also compared this framework with ROVER fusion and showed that it outperforms ROVER fusion by 9% relative for the small vocabulary task. In addition, fusion and nonparametric mapping improve the accuracy of the estimated uncertainty compared to Wiener + VTS both in the spectral domain and in the feature domain and nonparametric uncertainty estimation and propagation provides the best results in all cases.

7

Discriminative learning based uncertainty estimator

This chapter presents the proposed method to transform the estimated feature-domain full uncertainty covariance matrix according to a discriminative criterion. The motivation of this work is reported in Section 7.1. The details of this method are presented in Section 7.2. Experimental results are presented in Section 7.3. The part of this work relative to a linear mapping was published in [Tran et al., 2015b].

7.1 Motivation

Existing uncertainty estimation techniques improve ASR accuracy but they still exhibit a gap compared to the use of oracle uncertainty as shown in the previous chapters. This comes partly from the highly non-linear feature transformation and from additional assumptions such as Gaussian distribution and independence between frequency bins in the spectral domain.

In the previous chapter, the proposed uncertainty estimators and propagators were trained such that the trained uncertainty estimates are close to the oracle ones irrespectively of the resulting state hypotheses. Delcroix et al. proposed to train a linear mapping according to a ML criterion instead [Delcroix et al., 2013a; Delcroix et al., 2009]. They applied this approach to diagonal feature uncertainty matrices and they showed significant improvement. These two approaches can be considered as suboptimal because the same importance is given to the correct state hypothesis and to the competing state hypotheses during training.

Recently, maximum mutual information (MMI) [McDermott et al., 2010] and boosted MMI (bMMI) [Povey et al., 2008; Tachioka et al., 2013a] were successfully employed for supervised discriminative adaptation of feature means and diagonal uncertainty matrices by [Delcroix et al., 2011]. However, [Delcroix et al., 2011] estimated the diagonal uncertainty matrix directly in the feature domain as the squared difference between noisy and enhanced features.

7.2 Discriminative uncertainty mapping

In this section, we present a method for state-dependent and state-independent discriminative mapping of the full feature uncertainty covariance matrix. Starting from the diagonal feature-domain uncertainty covariance matrices estimated by the nonparametric technique in Chapter 6, a linear and a nonlinear mapping are trained so as to maximize a discriminative criterion. In the end, a mapping of the full uncertainty covariance matrix is derived.

The discriminative criterion used in this work is bMMI. In the bMMI criterion, wrong state hypotheses are given more weight than the correct state hypothesis. The general case will be presented in Section 7.2.1. The linear and nonlinear mappings will be presented in Sections 7.2.2 and 7.2.3, respectively.

7.2.1 General approach

Focusing on the case of a diagonal uncertainty covariance and state-independent mapping first, the transformed uncertainty for time frame n is given by

$$(\widehat{\boldsymbol{\sigma}}^{\mathrm{t}}_{\mathbf{c}_n})^2 = g\left((\widehat{\boldsymbol{\sigma}}_{\mathbf{c}_n})^2, \boldsymbol{\theta}\right) \tag{7.1}$$

where $(\widehat{\boldsymbol{\sigma}}_{\mathbf{c}_n})^2$ is the vector of diagonal elements of the original uncertainty covariance estimate $\widehat{\boldsymbol{\Sigma}}_{\mathbf{c}_n}$ and g is a state-independent multivariate mapping with parameter set $\boldsymbol{\theta}$. In the following, the estimated uncertainty $\widehat{\boldsymbol{\Sigma}}_{\mathbf{c}_n}$ is the nonparametric uncertainty.

The goal is to find the parameter set $\boldsymbol{\theta}$ that maximizes the posterior probability of the correct recognition hypothesis w.r.t. all hypotheses. The bMMI criterion is given by [Povey et al., 2008]

$$F_{bMMI} = \log\left(\frac{p(\widehat{\boldsymbol{\mu}}_{\mathbf{c}}, \widehat{\boldsymbol{\Lambda}}^{\mathrm{t}}_{\mathbf{c}}|\boldsymbol{q}^*, \boldsymbol{\theta})p(\boldsymbol{q}^*)}{\sum_{\boldsymbol{q}} p(\widehat{\boldsymbol{\mu}}_{\mathbf{c}}, \widehat{\boldsymbol{\Lambda}}^{\mathrm{t}}_{\mathbf{c}}|\boldsymbol{q}, \boldsymbol{\theta})p(\boldsymbol{q})e^{\epsilon A(\boldsymbol{q},\boldsymbol{q}^*)}}\right) \tag{7.2}$$

where $\mathbf{c} = \{\mathbf{c}_n\}_{n=1}^N$ is the feature sequence, $\boldsymbol{q} = \{q_n\}_{n=1}^N$ is one of the hypothesized state sequences, $\boldsymbol{q}^*$ is the correct state sequence, $\widehat{\boldsymbol{\mu}}_{\mathbf{c}}$ is the sequence of estimated feature means, and $\widehat{\boldsymbol{\Lambda}}^{\mathrm{t}}_{\mathbf{c}}$ is the sequence of transformed diagonal uncertainty covariance matrices. Assuming that the training data are so-called "stereo data" consisting of aligned clean and noisy signals, the correct state sequence is computed by forced alignment of the clean model on the clean training data. The term $A(\boldsymbol{q}, \boldsymbol{q}^*)$ is the total number of incorrect states over frames and ϵ is a boosting factor to be chosen. More precisely,

$$A(\boldsymbol{q}, \boldsymbol{q}^*) = \sum_n a(q_n, q_n^*) \tag{7.3}$$

where $a(q, q^*) = 0$ if q and q^* correspond to the same phone or q corresponds to a silence phone and $a(q, q^*) = 1$ otherwise. The summation over $\mathbf{q}$ in the denominator indicates summation over all hypotheses. In practice this sum is restricted to a finite number of terms represented by a lattice or an N-best list. Applying the chain rule, the derivative of the bMMI criterion w.r.t each parameter of the transformation is given by

$$\frac{\partial F_{bMMI}}{\partial \theta_k} = \sum_{i,n} \frac{\partial F_{bMMI}}{\partial (\widehat{\sigma}^{\mathrm{t}}_{c_{in}})^2} \frac{\partial (\widehat{\sigma}^{\mathrm{t}}_{c_{in}})^2}{\partial \theta_k}. \tag{7.4}$$

The derivative of the bMMI criterion w.r.t each element of the transformed uncertainty covariance matrix can be written as

$$\frac{\partial F_{bMMI}}{\partial (\widehat{\sigma}^{\mathrm{t}}_{c_{in}})^2} = \left(\frac{\partial \log p(\widehat{\boldsymbol{\mu}}_{\mathbf{c}}, \widehat{\boldsymbol{\Lambda}}^{\mathrm{t}}_{\mathbf{c}} | \boldsymbol{q}^*, \boldsymbol{\theta})}{\partial (\widehat{\sigma}^{\mathrm{t}}_{c_{in}})^2} - \frac{\partial \log \left(\sum_{\boldsymbol{q}} p(\widehat{\boldsymbol{\mu}}_{\mathbf{c}}, \widehat{\boldsymbol{\Lambda}}^{\mathrm{t}}_{\mathbf{c}} | \boldsymbol{q}, \boldsymbol{\theta}) p(\boldsymbol{q}) e^{\epsilon A(\boldsymbol{q}, \boldsymbol{q}^*)} \right)}{\partial (\widehat{\sigma}^{\mathrm{t}}_{c_{in}})^2} \right) \tag{7.5}$$

Due to

$$\partial p(\widehat{\boldsymbol{\mu}}_{\mathbf{c}}, \widehat{\boldsymbol{\Lambda}}^{\mathrm{t}}_{\mathbf{c}} | \boldsymbol{q}, \boldsymbol{\theta}) = p(\widehat{\boldsymbol{\mu}}_{\mathbf{c}}, \widehat{\boldsymbol{\Lambda}}^{\mathrm{t}}_{\mathbf{c}} | \boldsymbol{q}, \boldsymbol{\theta}) \partial \mathrm{log} p(\widehat{\boldsymbol{\mu}}_{\mathbf{c}}, \widehat{\boldsymbol{\Lambda}}^{\mathrm{t}}_{\mathbf{c}} | \boldsymbol{q}, \boldsymbol{\theta}) \tag{7.6}$$

(7.5) can be written as

$$\frac{\partial F_{bMMI}}{\partial (\widehat{\sigma}^{\mathrm{t}}_{c_{in}})^2} = \left(\frac{\partial \log p(\widehat{\boldsymbol{\mu}}_{\mathbf{c}}, \widehat{\boldsymbol{\Lambda}}^{\mathrm{t}}_{\mathbf{c}} | \boldsymbol{q}^*, \boldsymbol{\theta})}{\partial (\widehat{\sigma}^{\mathrm{t}}_{c_{in}})^2} - \sum_{\boldsymbol{q}} \gamma_{\boldsymbol{q}} \frac{\partial \log p(\widehat{\boldsymbol{\mu}}_{\mathbf{c}}, \widehat{\boldsymbol{\Lambda}}^{\mathrm{t}}_{\mathbf{c}} | \boldsymbol{q}, \boldsymbol{\theta})}{\partial (\widehat{\sigma}^{\mathrm{t}}_{c_{in}})^2} \right) \tag{7.7}$$

where

$$\gamma_{\boldsymbol{q}} = \frac{p(\widehat{\boldsymbol{\mu}}_{\mathbf{c}}, \widehat{\boldsymbol{\Lambda}}^{\mathrm{t}}_{\mathbf{c}} | \boldsymbol{q}, \boldsymbol{\theta}) p(\boldsymbol{q}) e^{\epsilon A(\boldsymbol{q}, \boldsymbol{q}^*)}}{\sum_{\boldsymbol{q}'} p(\widehat{\boldsymbol{\mu}}_{\mathbf{c}}, \widehat{\boldsymbol{\Lambda}}^{\mathrm{t}}_{\mathbf{c}} | \boldsymbol{q}', \boldsymbol{\theta}) p(\boldsymbol{q}') e^{\epsilon A(\boldsymbol{q}', \boldsymbol{q}^*)}} \tag{7.8}$$

are the normalized boosted sequence posteriors. Note that $p(\widehat{\boldsymbol{\mu}}_{\mathbf{c}_n}, (\widehat{\boldsymbol{\sigma}}^{\mathrm{t}}_{\mathbf{c}_n})^2 | q, \boldsymbol{\theta})$ is an m-component GMM given by

$$p(\widehat{\boldsymbol{\mu}}_{\mathbf{c}_n}, (\widehat{\boldsymbol{\sigma}}^{\mathrm{t}}_{\mathbf{c}_n})^2 | q, \boldsymbol{\theta}) = \sum_{m} w_{qm} p(\widehat{\boldsymbol{\mu}}_{\mathbf{c}_n}, (\widehat{\boldsymbol{\sigma}}^{\mathrm{t}}_{\mathbf{c}_n})^2 | q, m, \boldsymbol{\theta}) \tag{7.9}$$

where

$$p(\widehat{\boldsymbol{\mu}}_{\mathbf{c}_n}, (\widehat{\boldsymbol{\sigma}}^{\mathrm{t}}_{\mathbf{c}_n})^2 | q, m, \boldsymbol{\theta}) = \mathcal{N}(\widehat{\boldsymbol{\mu}}_{\mathbf{c}_n}; \boldsymbol{\mu}_{qm}, \boldsymbol{\Sigma}_{qm} + \widehat{\boldsymbol{\Sigma}}^{\mathrm{t}}_{\mathbf{c}_n}). \tag{7.10}$$

Applying the chain rule, the gradient term which is inside the summation in (7.7) can be written as

$$\frac{\partial \log p(\widehat{\boldsymbol{\mu}}_{\mathbf{c}_n}, (\widehat{\boldsymbol{\sigma}}^{\mathrm{t}}_{\mathbf{c}_n})^2 | q_n, \boldsymbol{\theta})}{\partial (\widehat{\sigma}^{t}_{c_{in}})^2} = \sum_{m} \xi_{q_n mn} \delta_{q_n min} \tag{7.11}$$

where $\xi_{q,m,n}$ is the posterior probability of mixture component m

$$\xi_{qmn} = \frac{w_{qm} p(\widehat{\boldsymbol{\mu}}_{\mathbf{c}_n}, (\widehat{\boldsymbol{\sigma}}^{\mathrm{t}}_{\mathbf{c}_n})^2 | q, m, \boldsymbol{\theta})}{\sum_{m'} w_{qm'} p(\widehat{\boldsymbol{\mu}}_{\mathbf{c}_n}, (\widehat{\boldsymbol{\sigma}}^{\mathrm{t}}_{\mathbf{c}_n})^2 | q, m', \boldsymbol{\theta})} \tag{7.12}$$

and

$$\delta_{q,m,i,n} = \frac{1}{2} \left(\frac{(\widehat{\mu}_{c_{in}} - \mu_{qmi})^2}{\sigma^2_{qmi} + (\widehat{\sigma}^{\mathrm{t}}_{c_{in}})^2} - 1 \right) \frac{1}{\left(\sigma^2_{qmi} + (\widehat{\sigma}^{\mathrm{t}}_{c_{in}})^2 \right)}. \tag{7.13}$$

In the end,

$$\frac{\partial F_{bMMI}}{\partial (\widehat{\sigma}^{\mathrm{t}}_{c_{in}})^2} = \sum_{\boldsymbol{q}} (\mathbb{1}_{\mathbf{q}^* = \mathbf{q}} - \gamma_{\mathbf{q}}) \sum_{m} \xi_{q_n mn} \delta_{q_n imn}. \tag{7.14}$$

State-dependent mapping is achieved in a similar way by building a separated mapping for each state.

7.2.2 Linear mapping

We now consider a simple case of transformation g that is a linear rescaling. We enforce the rescaling factors to be nonnegative by taking a squared form of parameters. In this case, the transformed uncertainty is given by

$$(\widehat{\sigma}^{\mathrm{t}}_{c_{in}})^2 = \theta_i^2 (\widehat{\sigma}_{c_{in}})^2 \tag{7.15}$$

where the parameter set $\boldsymbol{\theta}$ is a 39 dimensional vector and i is the parameter index. $(\widehat{\sigma}_{c_{in}})^2$ is the i-th diagonal element of $\widehat{\boldsymbol{\Sigma}}_{\mathbf{c}_n}$. In equation (7.4), the term $\frac{\partial(\widehat{\sigma}^{\mathrm{t}}_{c_{in}})^2}{\partial\theta_k}$ is given by

$$\frac{\partial(\widehat{\sigma}^{\mathrm{t}}_{c_{in}})^2}{\partial\theta_k} = \begin{cases} 2\theta_i (\widehat{\sigma}_{c_{in}})^2 & \text{if } \; i = k \\ 0 & \text{otherwise} \end{cases} \tag{7.16}$$

The gradient is then averaged over all utterances. The scaling factors θ_i are initialized to 1. The bMMI objective function is then optimized using gradient ascent by

$$\theta_i \leftarrow \theta_i + \eta \frac{\partial F_{bMMI}}{\partial \theta_i} \tag{7.17}$$

where η is the step size. After convergence, the rescaled full uncertainty covariance matrix is derived as

$$\widehat{\boldsymbol{\Sigma}}^{\mathrm{t}}_{\mathbf{c}_n} = \mathbf{Diag}(\boldsymbol{\theta}) \widehat{\boldsymbol{\Sigma}}_{\mathbf{c}_n} \mathbf{Diag}(\boldsymbol{\theta}). \tag{7.18}$$

7.2.3 Nonlinear mapping

The transformation g can also be nonlinear. In the following, this nonlinear transformation is modeled as a neural network. $\boldsymbol{\theta}$ is the vector of weights and biases of all layers. The input of the network is a supervector obtained by concatenating the original uncertainties of three consecutive frames. The output is the transformed uncertainty in the current frame $(\widehat{\sigma}^{\mathrm{t}}_{\mathbf{c}_n})^2$. The bMMI objective function is then optimized using stochastic gradient ascent. The term $\frac{\partial(\widehat{\sigma}^{\mathrm{t}}_{\mathbf{c}_n})^2}{\partial\boldsymbol{\theta}}$ in equation (7.4) can be computed by conventional back-propagation. At each iteration, we pick one random utterance corresponding to a random SNR level, then compute the gradient and update the parameters. This procedure is done for all utterances in one epoch. After convergence, the transformed full uncertainty covariance matrix is given by (6.11) and (6.12). For convergence to a better local optimum, the optimization procedure is done in an ML fashion beforehand by optimizing only the numerator in equation (7.2). The parameters which are obtained by optimizing the ML criterion are then used to initialize the estimation according the bMMI criterion.

7.3 Experimental results

For both linear and nonlinear transforms, the boosting factor is set to 0.1. In addition, we remove the frames on which the true state is absent from the expression of the bMMI criterion

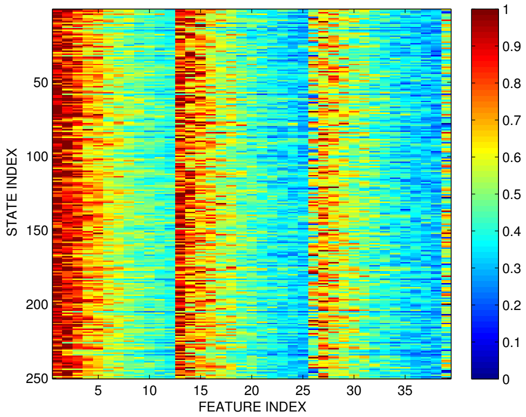


Figure 7.1: State-dependent linear mapping coefficients trained via bMMI.

Method	state dependent	Test set						
		-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average
no uncertainty	no	73.75	78.42	84.33	89.50	91.83	92.25	85.01
nonparametric + bMMI (diag)	no	82.33	83.75	88.50	91.17	92.75	93.0	88.58
nonparametric + bMMI (full)	no	84.17	85.00	88.75	91.33	93.75	94.42	89.57
squared diff + bMMI [Delcroix et al., 2011]	yes	79.92	82.00	87.17	90.67	92.92	93.42	87.68
nonparametric + bMMI(diag)	yes	82.93	83.75	88.50	91.17	92.75	93.33	88.73
nonparametric + bMMI(full)	yes	84.33	85.00	88.92	91.50	93.75	94.50	**89.66**

Table 7.1: Keyword accuracy (%) on the Track 1 test set with discriminative linear mapping. Average accuracies have a 95% confidence interval of ±0.8%. The full results can be found in Appendix A.8

[Povey et al., 2008].

7.3.1 Linear scaling

For the linear transform, we perform gradient ascent with $\eta = 0.1$ using 40 iterations. The resulting ASR performance results are listed in Table 7.1. The baseline without uncertainty handling achieved 85.01% keyword accuracy. Linear discriminative state-independent mapping of the full nonparametric estimator resulted in 89.57% accuracy. This approach outperformed the bMMI approach in [Delcroix et al., 2011] by 16% relative WER reduction[9]. However state-dependent linear mapping did not significantly improve performance.

Also, compared with nonparametric uncertainty estimator, linear mapping reduced the WER

[9] Note that the performance reported in this paper differs from [Delcroix et al., 2011] due to the use of a different speech enhancement system.

Method	state dependent	Test set						
		-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average
no uncertainty	no	73.75	78.42	84.33	89.50	91.83	92.25	85.01
nonparametric + bMMI (diag)	no	82.75	84.00	88.50	91.17	93.00	93.17	88.76
nonparametric + bMMI (full)	no	84.55	85.30	88.75	91.33	93.75	94.42	89.68
nonparametric + bMMI(diag)	yes	83.33	84.00	88.50	91.33	93.00	93.75	88.98
nonparametric + bMMI(full)x	yes	84.75	85.50	89.00	91.75	93.75	95.00	**89.95**

Table 7.2: Keyword accuracy (%) on the Track 1 test set with discriminative nonlinear mapping. Average accuracies have a 95% confidence interval of ±0.8%. The full result can be found in Appendix A.9

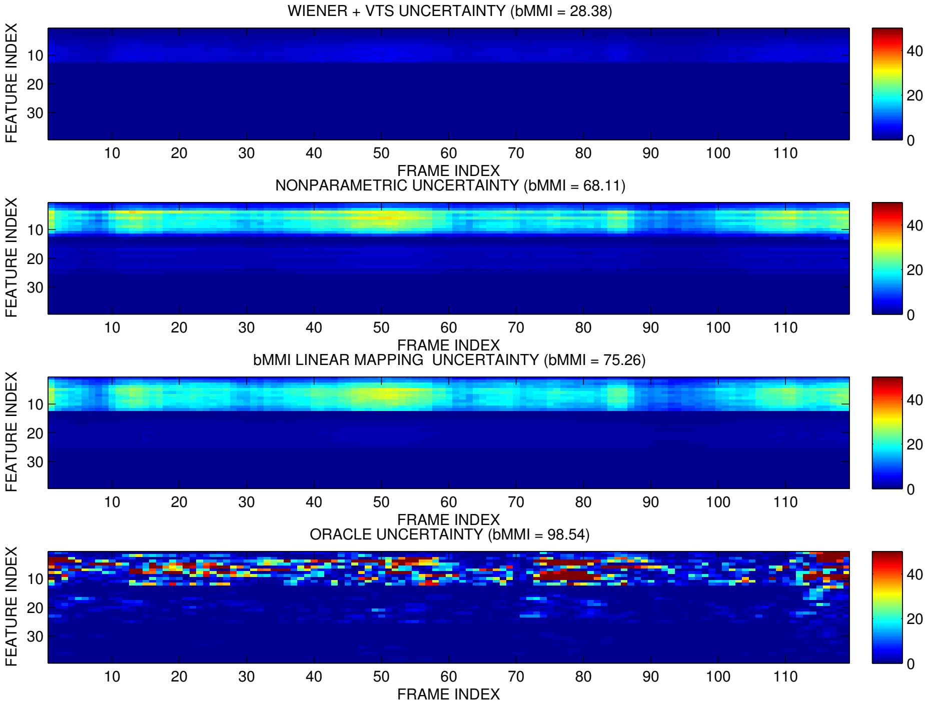


Figure 7.2: Example of uncertainty over time (discriminative linear mapping). This example corresponds to the utterance "bin blue with s nine soon" on the Track 1 dataset. First row: generative estimated uncertainty. Second row: nonparametric uncertainty. Third row: discriminative linear mapping uncertainty. Last row: oracle uncertainty.

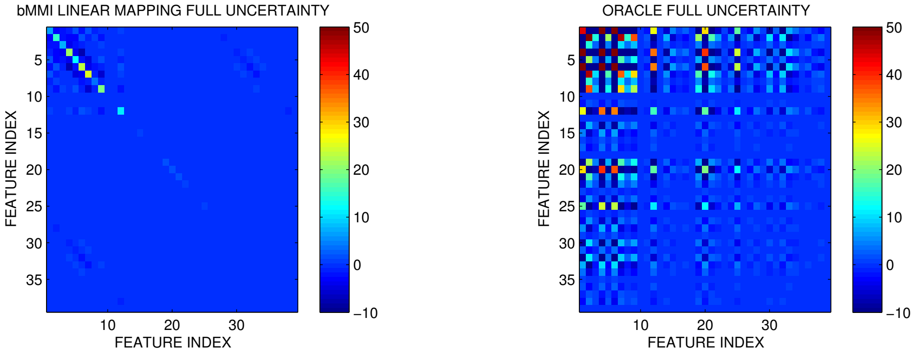


Figure 7.3: Example of full uncertainty covariance matrix (discriminative linear mapping). This example corresponds to frame 60 of the utterance "bin blue with s nine soon" on the Track 1 dataset. Left: discriminative linear mapping full uncertainty covariance matrix. Right: oracle full uncertainty.

1% and 2% relative in the state independent case and state dependent case, respectively.

Figure 7.1 shows the linear mapping coefficients for different states. All entries tend to be smaller than one. Figure 7.2 shows an example of a linear mapping of the nonparametric diagonal uncertainty. The mapped of the nonparametric diagonal uncertainty tends to be slightly smaller compared to the original nonparametric uncertainty but it achieved a higher bMMI score. Figure 7.3 shows an example of a full uncertainty covariance matrix in one frame. It turns out that the discriminative uncertainty estimator can compensate the diagonal elements better than off-diagonal elements. Compensating the underestimation of the off-diagonal elements is still an open problem.

7.3.2 Nonlinear transform

The network has two hidden layers and 100 neurons in each hidden layer. The activation function is a rectified linear unit. The input is normalized to zero mean and unit variance. The neural network was trained with minibatch stochastic gradient ascent. The size of a minibatch corresponds to one utterance. The learning rate was initialized at 0.1 then it is linearly decreased during training. The number of epochs is 100.

This nonlinear mapping improved the accuracy to 89.95% which is 5% relative WER reduction compared to the original nonparametric estimator in the state-dependent full covariance matrix case.

Figure 7.4 shows an example of estimated uncertainty. The estimated uncertainy appears to be sparser than the original nonparametric estimator. Although it doesn't have the same shape as the oracle uncertainty, the corresponding bMMI score is quite close to the oracle uncertainty and better than the one obtained by linear mapping in Figure 7.2.

Figure 7.5 shows an example of full uncertainty covariance matrix in one frame. Similarly

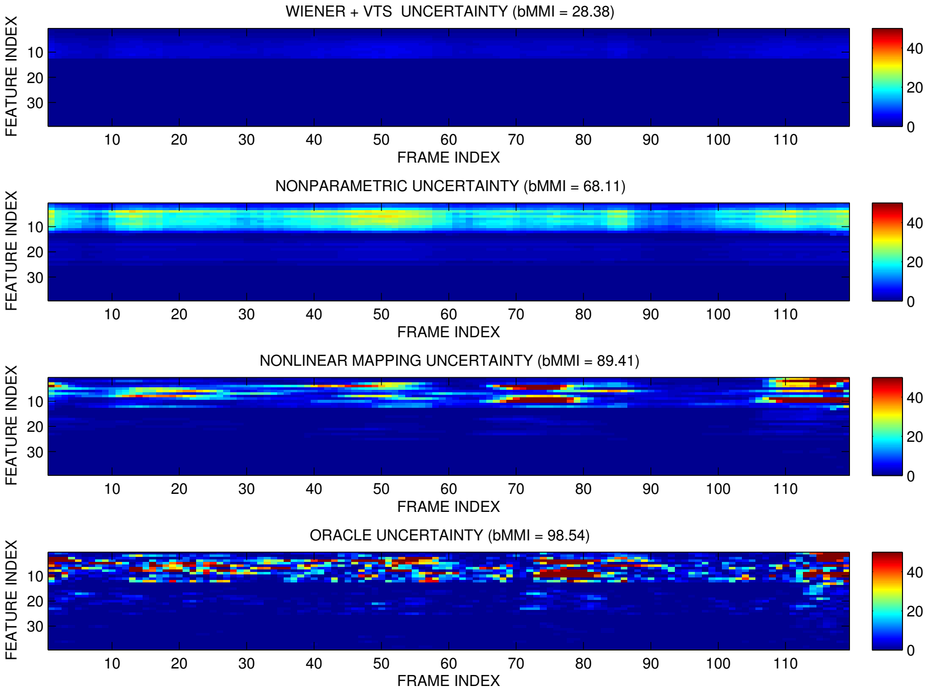


Figure 7.4: Example of uncertainty over time (discriminative). This example corresponds to the utterance "bin blue with s nine soon" on the Track 1 dataset. First row: generatively estimated uncertainty. Second row: nonparametric uncertainty. Third row: discriminative nonlinear mapping uncertainty. Last row: oracle uncertainty.

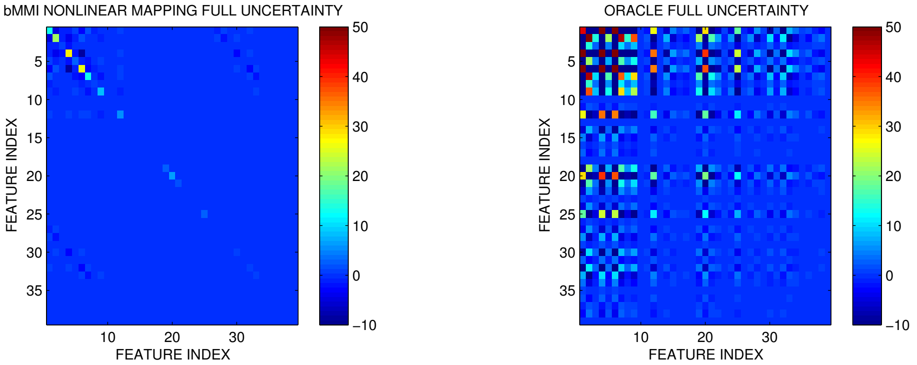


Figure 7.5: Example of full uncertainty covariance matrix (discriminative). This example corresponds to frame 60 of the utterance "bin blue with s nine soon" on the Track 1 dataset. Left: discriminative nonlinear mapping full uncertainty covariance matrix. Right: oracle full uncertainty.

to linear mapping, it appears to compensate the diagonal elements better than the off-diagonal elements.

7.4 Summary

In this chapter, we derived an approach to learn a mapping of the estimated uncertainty covariance matrix so as to maximize the bMMI criterion. Two mapping types (linear and nonlinear) are introduced and evaluated. These mappings improved the WER by 2% and 5% relative on top of the nonparametric framework, respectively. Learning bMMI with full matrices appears to be promising.

Part III

Neural network training

8

State of the art

This chapter presents the state of the art of training for neural networks. Several neural network architectures are described in Section 8.1. Sections 8.2.2, 8.2.3, and 8.2.4 present three state-of-the-art approaches to train neural networks.

8.1 Neural network architectures

Perceptrons

Perceptrons were developed in the 1950s and 1960s [Rosenblatt, 1958]. Figure 8.1 depicts a model of a perceptron. A perceptron takes several binary inputs $z_1, z_2, ...$ and produces a single binary output x. The neuron's output can be set to 0 or 1 and it is determined by whether the weighted sum $\sum_j w_j z_j$ is less than or greater than some threshold value *thre*. A basic mathematical model of one neuron is given by

$$x = \begin{cases} 0 & \text{if } \sum_j w_j z_j \leq thre \\ 1 & \text{if } \sum_j w_j z_j > thre \end{cases} \tag{8.1}$$

where the weights w_j are real numbers expressing the importance of the respective inputs in the output.

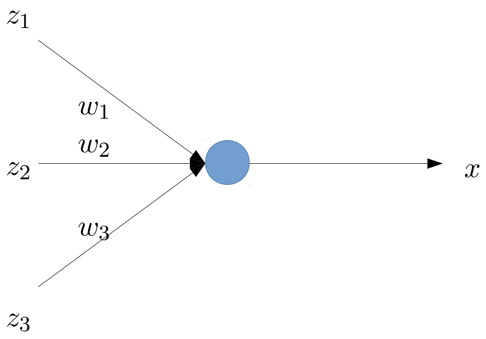


Figure 8.1: The model of a perceptron.

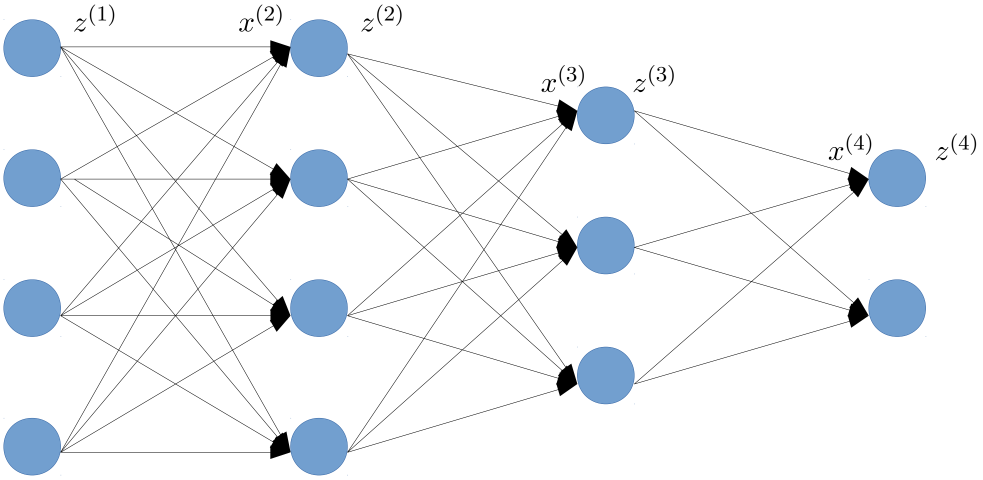


Figure 8.2: Model of a multilayer perceptron with four layers: input, two hidden layers and output layer.

Multilayer perceptron

A multilayer perception (MLP) [Rosenblatt, 1958] is a feedforward neural network consisting of N layers of fully connected perceptrons as shown in Figure 8.2. Let k_n be the number of elements (neurons) in the n-th layer, p the data index. P is number of data samples. Here we define $z_{jp}^{(n)}$ as the input to the j-th element. Let $w_{ij}^{(n)}$ be the weight from the j-th element to the i-th element and $u_i^{(n)}$ be the i-th bias term between the n-th and the $(n+1)$-th layer. The neural network can be defined as

$$x_{ip}^{(n+1)} = \sum_{j=1}^{k_n} w_{ij}^{(n)} z_{jp}^{(n)} + u_i^{(n)} \tag{8.2}$$

$$z_{ip}^{(n+1)} = f(x_{ip}^{(n+1)}) \tag{8.3}$$

where f represents a nonlinear activation function. Possible activation functions include sigmoid

$$f(x) = \frac{1}{1+\exp(-x)} \tag{8.4}$$

tangent hyperbolic

$$f(x) = \frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)} \tag{8.5}$$

or rectified linear unit [Zeiler et al., 2013]

$$f(x) = \max(0, x). \tag{8.6}$$

Softmax output layer

When the outputs of a network are interpretable as posterior probabilities for a categorical target variable, it is highly desirable for those outputs to lie between zero and one and to sum to one.

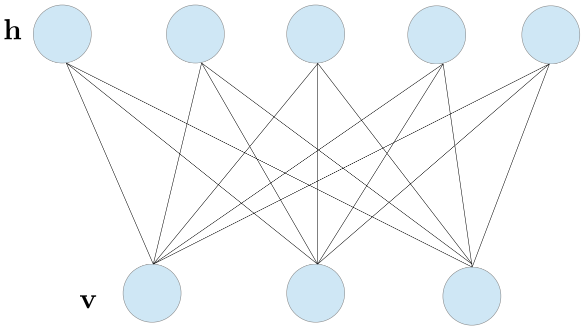


Figure 8.3: Model of an RBM.

A softmax output layer is then used as the output layer in order to convert a K-dimensional pre-activation vector $\mathbf{x}$ into an output vector $\mathbf{z}$ in the range $(0, 1)$:

$$z_i = \frac{\exp(x_i)}{\sum_{j=1}^{K} \exp(x_j)}. \tag{8.7}$$

Maxout network

The maxout model [Goodfellow et al., 2013] is a feed-forward achitecture, such as a multilayer perceptron or convolutional neural network, that uses an activation function called the maxout unit. The maxout unit is given by

$$f_i(\mathbf{x}) = \max_{j \in [1,k]} x_{ij} \tag{8.8}$$

where

$$x_{ij} = \mathbf{w}_{ij}^T \mathbf{z} + u_{ij} \tag{8.9}$$

where $\mathbf{z}$ is the input vector, $\mathbf{w}_{ij}^T$ are a set of trainable weight vectors, and u_{ij} are a set of trainable biases.

Deep belief net

Deep belief nets (DBN) were first proposed by Hinton [Hinton et al., 2006]. A DBN is a generative type of deep neural network, where each layer is constructed from a restricted Boltzmann machine (RBM). A RBM as shown in Figure 8.3 is a generative stochastic artificial neural network that can learn a probability distribution over its inputs. It can also be viewed as an undirected graphical model with one visible layer and one hidden layer with connections between the visible units and the hidden units but no connection between the visible units or the hidden units themselves.

Given training data, training is achieved in two steps. In the first step, by using the so-called contrastive divergence criterion, the RBM parameters are adjusted such that the probability distribution represented by the RBM fits the training data as well as possible. Because this training process does not require labels it is a form of unsupervised training. This is also called ”pre-training”. This pre-training is then repeated *greedily* for all layers from the first hidden layer (after input) to the last hidden layer (before output).

Pretraining in deep neural networks refer to unsupervised training with RBMs. The joint distribution of the hidden layer $\boldsymbol{h}$ and the visible layer $\boldsymbol{v}$ can be written as

$$p(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{\mathbf{Z}} \exp(-\mathbf{E}(\boldsymbol{v}, \boldsymbol{h})) \tag{8.10}$$

where $\mathbf{Z}$ is a normalization constant and $\mathbf{E}(\boldsymbol{v}, \boldsymbol{h})$ is an energy function. For Bernoulli RBMs, the energy function is:

$$\mathbf{E}(\boldsymbol{v}, \boldsymbol{h}) = -\sum_{i=1}^{D}\sum_{j=1}^{K} w_{ij} v_i h_j - \sum_{i=1}^{D} b_i v_i - \sum_{j=1}^{K} a_j h_j \tag{8.11}$$

where w_{ij} denotes the weight of the undirected edge connecting visible node v_i and hidden node h_j, and a and b are the bias terms for the hidden and visible units, respectively. For Gaussian RBMs, assuming that the visible units have zero mean and unit variance, the energy function is:

$$\mathbf{E}(\boldsymbol{v}, \boldsymbol{h}) = \sum_{i=1}^{D} \frac{(v_i - b_i)^2}{2} - \sum_{i=1}^{D}\sum_{j=1}^{K} w_{ij} v_i h_j - \sum_{j=1}^{K} a_j h_j \tag{8.12}$$

An RBM is pre-trained generatively to maximize the data log-likelihood $\log \sum_{\mathbf{h}} p(\boldsymbol{v}, \boldsymbol{h})$ by using so-called contrastive divergence.

When pre-training is finished for one hidden layer, one forward pass of this RBM is executed to get the value of the input to the next hidden layer. When all layer have been pretrained, an ouput layer is added, and given target labels, the network is fined-tuned in a supervised fashion using the back-propagation algorithm [Rumelhart et al., 1986]. This is usually used in DBN or MLP with sigmoid units in order to obtain a better initialization.

Convolutional neural network

Convolutional neural nets (CNN) or Convnets were first proposed by Lecun [LeCun et al., 1998]. A CNN consists of several layers. These layers can be a convolutional layer, a pooling layer, or a fully-connected layer.

In the convolutional layer, given a big matrix of inputs $m \times n$, a small rectangle $d \times h$ rectangle of units called a *local receptive field* connects to the units of the next layer by $d \times h$ trainable weights plus a bias passed though a nonlinear activation function ($d << m, h << n$). Intuitively, it can be understood as a small MLP. This is done to extract information from a group of neighboring units from the previous layer. This comes from the fact that the local units have high correlation. The rectangle of $d \times h$ trainable weights called ’window’ moves around the entire input, with no overlap between moving window position. Each position produces one unit at the next layer. This means that all neighborhood units in the previous layer share the same weights. This comes from the fact that if these *local representations* appears at many places in the input then they should have identical weights. The set of ouputs is called a *feature map*. CNNs can have several *feature maps* with different weights.

The subsampling (or pooling) layer performs local averaging and sub-sampling to reduce the resolution of each feature map. For instance, with one feature map, a local receptive field consisting of a 2×2 matrix can be replaced by only 1 point or unit computed by averaging the four neighboring samples then multiplying the average with a trainable weight plus a bias and passing it through a nonlinear activation function. Note that two consecutive 2×2 matrices do not overlap in the subsampling layer which differs from the convolutional layer. After subsampling, the number of feature maps does not change but the size of feature maps is reduced by a factor of 4 times.

Note that, if there is another convolutional layer after the subsampling layer, then all the steps above remain unchanged except that each output in a feature map is connected to several *local receptive fields* of several previous feature maps.

The fully-connected layer is basically an MLP with full connections from all units to the output layer. All trainable weights and biases are trained by using the back propagation algorithm.

Recurrent neural network

Recurrent neural networks (RNN) have the same architecture as MLP except that certain hidden layers have full recurrent connections with themselves. Intuitively speaking, the output of a hidden layer at the previous time step feeds into the input at the current time step by a trainable weight matrix. RNNs learn a mapping from a sequence input to a sequence output. Theoretically, to train an RNN using backpropagation through time (BPTT), a full sequence of inputs and corresponding outputs must to be used at the same time. However, this is impractical due to the fact that neural networks are usually trained by minibatch stochastic gradient descent. In practice, the gradient is usually approximated by truncating the sequence input to a few time steps. Training recurrent neural networks is known as notoriously difficult due to the vanishing gradient problem. Long short term memory recurrent neural networks [Hochreiter and Schmidhuber, 1997] (LSTM) can be used to avoid this problem. Recurrent neural networks are usually employed for language modeling for instance.

8.2 Training algorithms

8.2.1 Training objective

For regression problems, the conditional probability distribution of the target vector $\boldsymbol{y}$ is assumed to be a K-dimensional Gaussian:

$$p(\boldsymbol{y}_p|\boldsymbol{z}_p, \theta) = \mathcal{N}\left(\boldsymbol{y}_p; \boldsymbol{z}_p, \sigma_{\boldsymbol{y}}^2 \mathbf{I}\right) \tag{8.13}$$

where **I** is the identity matrix and p is the data index. Training to maximize the log-likelihood of the target vector is equivalent to minimizing the squared error between the output of the neural network $\boldsymbol{z}_p$ and the target $\boldsymbol{y}_p$. The objective function of the regression problem is then given by

$$\mathbf{E} = \frac{1}{2}\sum_{ip}(y_{ip} - z_{ip})^2. \tag{8.14}$$

For classification problems, the output $\boldsymbol{y}$ is a multi-dimensional vector with K elements corresponding to K classes. The conditional probability distribution of the target vector $\boldsymbol{y}$ is assumed to be

$$p(\mathbf{y}_p|\boldsymbol{z}_p, \theta) = \prod_{i=1}^{K}(z_{ip})^{y_{ip}} \tag{8.15}$$

Maximizing the log-likelihood of the target vector is equivalent to minimizing the cross-entropy between the output of the neural network $\boldsymbol{z}_p$ and the target $\boldsymbol{y}_p$. The objective function of the classification problem is then given by

$$\mathbf{E} = -\sum_{ip} y_{ip}\log(z_{ip}). \tag{8.16}$$

The parameters θ are updated so as to minimize the squared error or the cross-entropy objective function. The gradient with respect to parameter $w_{ij}^{(n)}$ of each layer can be computed using the chain rule [Rumelhart et al., 1986] as follows:

$$\frac{\partial \mathbf{E}}{\partial w_{ij}^{(n)}} = \sum_{p,i^{(N)},i^{(N-1)},\ldots,i^{(n+1)},i^{(n)}} \frac{\partial \mathbf{E}}{\partial z_{ip}^{(N)}} \frac{\partial z_{ip}^{(N)}}{\partial x_{ip}^{(N)}} \frac{\partial x_{ip}^{(N)}}{\partial z_{ip}^{(N-1)}} \frac{\partial z_{ip}^{(N-1)}}{\partial x_{ip}^{(N-1)}} \cdots \frac{\partial z_{ip}^{(n+1)}}{\partial x_{ip}^{(n+1)}} \frac{\partial x_{ip}^{(n+1)}}{\partial w_{ij}^{(n)}} \tag{8.17}$$

where

$$\frac{\partial \mathbf{E}}{\partial z_{ip}^{(N)}} = \begin{cases} -(y_{ip} - z_{ip}) & \text{in the regression case} \\ -\frac{y_{ip}}{z_{ip}} & \text{in the classification case} \end{cases} \tag{8.18}$$

$$\frac{\partial z_{ip}^{(n)}}{\partial x_{ip}^{(n)}} = f'(x_{ip}^{(n)}) \tag{8.19}$$

$$\frac{\partial x_{ip}^{(n)}}{\partial z_{jp}^{(n-1)}} = w_{ij}^{(n-1)} \tag{8.20}$$

$$\frac{\partial x_{ip}^{(n+1)}}{\partial w_{ij}^{(n)}} = z_{jp}^{(n)} \tag{8.21}$$

This gradient can be computed recursively layer by layer. This is called backpropagation.

8.2.2 Stochastic gradient descent

The most widely used optimization algorithm used for NN training is stochastic gradient descent (SGD). The parameters are updated as follows [Rumelhart et al., 1986]:

$$w_{ij,t+1}^{(n)} = w_{ij,t}^{(n)} - \alpha \, \partial \mathbf{E}_t / \partial w_{ij,t}^{(n)}. \tag{8.22}$$

where α is a fixed learning rate which is set manually and keep very small and it is decayed throught the training process, t is the iteration index and i, j are neuron indexes. SGD can be used in minibatch mode in order to reduce computation cost. In minibatch mode, the gradient is computed from a subset of the full training set of samples usually from 10 to 1000 samples.

8.2.3 Adaptive subgradient method

The adaptive subgradient method ADAGRAD [Duchi et al., 2011] is another popular algorithm whose learning rule is given by

$$w_{ij,t+1}^{(n)} = w_{ij,t}^{(n)} - \alpha \frac{\partial \mathbf{E}_t / \partial w_{ij,t}^{(n)}}{\sqrt{\sum_t (\partial \mathbf{E}_t / \partial w_{ij,t}^{(n)})^2}} \tag{8.23}$$

where α is a fixed learning rate which is set manually and t is the iteration index. Note that these gradient based learning rules can also be applied to $u_i^{(n)}$.

8.2.4 Back propagation training with second order methods

Although minibatch SGD works in theory, it turns out that in practice it can be rather slow. Second order methods can use information from the second order derivative of the objective function in order to accelerate the training process. The most basic method for second order minimization is Newton's method:

$$\mathbf{W}_{t+1}^{(n)} = \mathbf{W}_t^{(n)} - \mathbf{H}_t^{(n)} \nabla \mathbf{E}_t(\mathbf{W}_t^{(n)}) \tag{8.24}$$

where each element of the full Hessian matrix $\mathbf{H}_t^{(n)}$ can be writen as

$$h_{iji'j',t}^{(n)} = (\partial^2 \mathbf{E}_t / \partial(w_{ij,t}^{(n)}) \partial(w_{i'j',t}^{(n)}))^{-1} \tag{8.25}$$

Newton's method may perform better than the simpler minibatch SGD but in high dimensional cases, computing the full Hessian matrix and its inverse are very costly.

This issue can be solved by using Hessian-free methods [Martens, 2010]. The main idea behind these methods is to approximate the objective function with a second-order Taylor expansion, then minimize it using the conjugate gradient method. Using Hessian-free method avoids computing and storing Hessian matrix, and results in a cheaper computation cost.

8.3 Summary

This chapter presented an overview of neural networks, their architecture, objective functions and some state-of-the art optimization methods to train neural networks. In general, neural networks can be trained using SGD whose performance heavily depends on tuning the learning rate while second order methods use a smaller number of iterations to converge, however the computation cost in one iteration is usually higher than that of SGD.

9

Fast neural network training based on an auxiliary function technique

This chapter presents an idea for training a neural network using an auxiliary function method. This work was published in [Tran et al., 2015a].

9.1 Motivation

Deep neural networks have become a hot topic and have been successfully applied for many classification problems such as speech recognition [Seide et al., 2011; Hinton et al., 2012; Veselý et al., 2013], speech separation [Wang and Wang, 2013; Huang et al., 2014; Weninger et al., 2014], robust speech recognition [Seltzer et al., 2013; Renals and Swietojanski, 2014; Weng et al., 2014], language modeling [Mikolov et al., 2010; Arisoy et al., 2012], and image classification [LeCun et al., 1998]. As we have seen in the previous chapter, training algorithm for neural network suffer from certain limitations. In this chapter, we introduce a new learning rule for neural networks that is based on an auxiliary function technique without parameter tuning. Instead of minimizing the objective function, a quadratic auxiliary function is recursively introduced layer by layer which has a closed-form optimum. We prove the monotonic decrease of the new learning rule. Our experiments show that the proposed algorithm converges faster and to a better local minimum than SGD. In addition, we propose a combination of the proposed learning rule and ADAGRAD which further accelerates convergence. Experimental evaluation on the MNIST dataset shows the benefit of the proposed approach in terms of digit recognition accuracy.

9.2 Background

9.2.1 Objective function

In the following, we consider the tangent hyperbolic function and the squared Euclidean loss. The objective function can be expressed as

$$\mathbf{E} = \frac{1}{2}\sum_{p=1}^{P}\sum_{i=1}^{I}(z_{ip}^{(N)} - y_{ip})^2 + \frac{\lambda}{2}\sum_{n=1}^{N}\sum_{i}^{I}\sum_{j}^{J}(w_{ij}^{(n)})^2 \tag{9.1}$$

where I, J are the number of neurons in each layer and the first term is the squared Euclidean distance between the NN output and the target and the second term is a regularization term that avoids overfitting. The problem here is to find a set of $w_{ij}^{(n)}$ and $u_i^{(n)}$ that minimize (9.1).

9.2.2 Auxiliary function technique

Auxiliary function based optimization [de Leeuw, 1994; Heiser, 1995; Becker et al., 1997; Lange et al., 2000; Hunter and Lange, 2004] has recently become popular in certain fields as exemplified by, e.g., the audio source separation techniques HPSS [Ono et al., 2008] and AuxIVA [Ono, 2011]. Following that, to avoid learning rate tuning and derive an effective learning rule, we introduce an auxiliary function technique for NN training. Instead of minimizing the objective function, an auxiliary function is introduced and the minimization procedure is applied to the auxiliary function. Let us express the general optimization problem as:

$$w^{(n)} = \operatorname*{argmin}_{w^{(n)}} \mathbf{E}(w^{(n)}). \tag{9.2}$$

In the auxiliary function technique, an auxiliary function $\mathbf{Q}$ is designed that satisfies

$$\mathbf{E}(w^{(n)}) \leq \mathbf{Q}(w^{(n)}, w_0^{(n)}) \tag{9.3}$$

for all $w^{(n)}$ and all values of the auxiliary variable $w_0^{(n)}$. The equality is satisfied if and only if $w^{(n)} = w_0^{(n)}$. Now, starting from an initial parameter value $w_0^{(n)}$, we can find the optimal value of $w^{(n)}$ that minimizes $\mathbf{Q}(w^{(n)}, w_0^{(n)})$:

$$w_1^{(n)} = \operatorname*{argmin}_{w^{(n)}} \mathbf{Q}(w^{(n)}, w_0^{(n)}). \tag{9.4}$$

As a result

$$\mathbf{E}(w_1^{(n)}) \leq \mathbf{Q}(w_1^{(n)}, w_0^{(n)}) \leq \mathbf{Q}(w_0^{(n)}, w_0^{(n)}) = \mathbf{E}(w_0^{(n)}). \tag{9.5}$$

The procedure can be applied iteratively as shown in Figure 9.1. The inequality (9.5) guarantees the monotonic decrease of the objective function. When the auxiliary function is quadratic, this algorithm converges linearly but at a typically faster rate than SGD [Böhning and Lindsay, 1988]. Also, it does not require any parameter tuning provided that (9.4) can be solved in closed form.

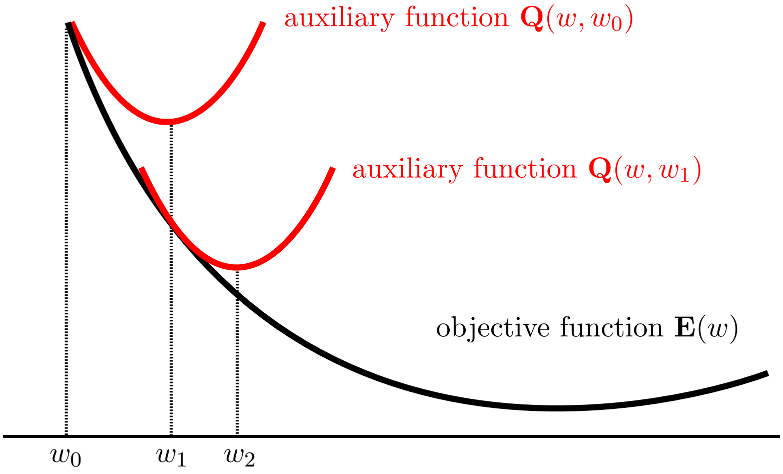

auxiliary function $\mathbf{Q}(w, w_0)$
auxiliary function $\mathbf{Q}(w, w_1)$
objective function $\mathbf{E}(w)$
w_0
w_1
w_2

Figure 9.1: Illustration of the auxiliary function technique.

9.3 Quadratic auxiliary function for neural network

We derive two auxiliary functions at each layer: one relating to the nonlinear activation function (8.3) and one relating to the linear combination (8.2). We then combine these two auxiliary functions into a single minimization scheme.

9.3.1 First quadratic auxiliary function

For simplicity, let us first omit the indices i, p, and n, and derive an auxiliary function for

$$\begin{aligned}\mathbf{E} &= (z-y)^2 \\ &= \tanh^2(x) - 2y\tanh(x) + y^2. \end{aligned} \tag{9.6}$$

The regularization term in (9.1) will be discussed later on. We derive a quadratic auxiliary function using the following lemma.

Lemma 9.3.1 *For any positive real numbers x and x_0 and any real number y, the following inequality is satisfied:*

$$(\tanh(x) - y)^2 \leq ax^2 - 2bx + c = \boldsymbol{Q} \tag{9.7}$$

where

$$a = A_1(x_0) + |y|A_2(-\sigma x_0) \tag{9.8}$$

$$b = y[\sigma x_0 A_2(-\sigma x_0) + \operatorname{sech}^2(x_0)] \tag{9.9}$$

$$\begin{aligned} c = &- A_1(x_0)x_0^2 + \tanh^2(x_0) + |y|A_2(-\sigma x_0)x_0^2 \\ &+ 2y\operatorname{sech}^2(x_0)x_0 - 2y\tanh(x_0) + y^2 \end{aligned} \tag{9.10}$$

and

$$\sigma = \operatorname{sign}(y) \tag{9.11}$$

$$A_1(x_0) = \frac{\operatorname{sech}^2(x_0)\tanh(x_0)}{x_0} \tag{9.12}$$

$$A_2(x_0) = \sup_x \frac{\tanh(x) - \tanh(x_0) - \operatorname{sech}^2(x_0)(x - x_0)}{(1/2)(x - x_0)^2} \tag{9.13}$$

The equality is satisfied if and only if $x = x_0$.

Proof The objective function (9.6) includes two x terms: $\tanh^2(x)$ and $2y\tanh(x)$. According to [de Leeuw and Lange, 2009, Theorem 4.5], when $f(x)$ is an even, differentiable function on $\mathbb{R}$ such that the ratio $f'(x)/x$ is decreasing on $(0, \infty)$, the inequality

$$f(x) \le g(x) = \frac{f'(x_0)}{2x_0}(x^2 - x_0^2) + f(x_0) \tag{9.14}$$

is satisfied.

Also, according to [de Leeuw and Lange, 2009], if a function $f(x)$ is differentiable in x, and

$$A(x_0) = \sup_x \frac{f(x) - f(x_0) - f'(x_0)(x - x_0)}{\frac{1}{2}(x - x_0)^2} \tag{9.15}$$

has a finite positive value, then

$$f(x) \le f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}A(x_0)(x - x_0)^2 \tag{9.16}$$

is satisfied for all x and x_0. By substituting $f(x) = \tanh^2(x)$ into (9.14), and $f(x) = y\tanh(x)$ where y can be positive or negative into (9.16), we have (9.7).

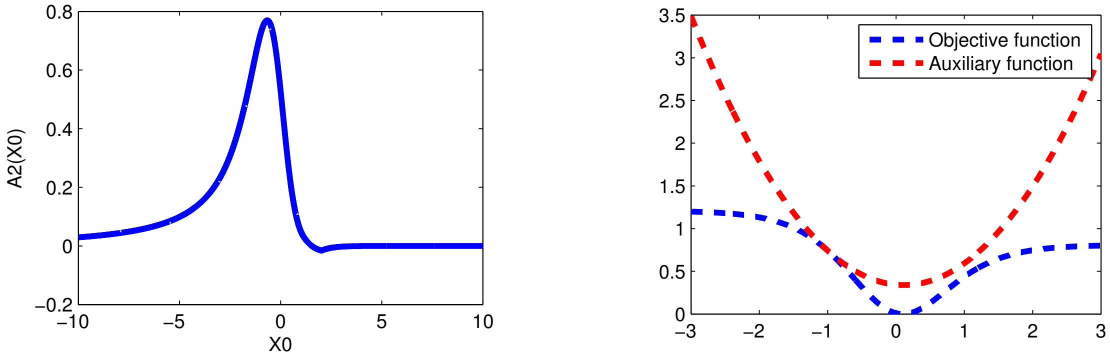
Note that $A_2(x_0)$ cannot computed in closed form. But we can prepare a table of $A_2(x_0)$ in advance. Figure 9.2 shows the shape of $A_2(x_0)$ and an example of the auxiliary function. In other word, note that the regularization term in the objective function (9.1) is a quadratic form of the parameters so that it can be directly used for auxiliary function without effort.

9.3.2 Second auxiliary function for separating variables

Now that we have derived an auxiliary function as a function of the inputs $x_{ip}^{(n)}$ in one layer, we need to propagate it down to the outputs $z_{jp}^{(n-1)}$ of the previous layer. Once again, let us omit the indices i, p, and n, and consider

$$x = \sum_j w_j z_j + u. \tag{9.17}$$

We wish the auxiliary function to decompose as a sum of terms, each relating to one neuron z_j, such that Lemma 9.3.1 can be applied again at the lower layer. Note that plugging (9.17) into (9.7) induces some cross-terms of the form $z_j z_{j'}$. In order to separate the contribution of each z_j additively, we apply the following lemma.



(a)

(b)

Figure 9.2: a) Shape of $A_2(x_0)$; b) Auxiliary function for $x_0 = -1, y = 0.1$.

Lemma 9.3.2 *For $x = \sum_j w_j z_j + u$, the inequality*

$$ax^2 + bx + c \leq \sum_{j=1}^{J} [aJw_j^2(z_j - y_j)^2 + aJ(\beta_j^2 - y_j^2)]$$
$$+ au^2 + bu + c = \mathbf{R} \tag{9.18}$$

is satisfied for any β_j such that $\sum_{j=1}^{J} \beta_j = 0$ where

$$y_j = \frac{\sum_i (2aJ\beta_j - 2au - b)w_j}{\sum_i 2aJw_j^2}. \tag{9.19}$$

The equality is satisfied if and only if

$$\beta_j = w_j z_j - \frac{1}{J}\sum_{j=1}^{J} w_j z_j. \tag{9.20}$$

Proof Generally for any s_j and β_j, minimizing $\sum_{j=1}^{J}(s_j - \beta_j)^2$ under the constraint that $\sum_{j=1}^{J} \beta_j = 0$, we have the inequality

$$\left(\sum_{j=1}^{J} s_j\right)^2 \leq J\sum_{j=1}^{J}(s_j - \beta_j)^2. \tag{9.21}$$

Replace $x = \sum_j w_j z_j + u$ into quadratic form $ax^2 + bx + c$. Applying above inequality to the case where $s_j = w_j z_j$ then take summation of the quadratic form $ax^2 + bx + c$ over i index, we obtain inequality (9.18).

9.3.3 Recursively deriving auxiliary functions

Based on Lemmas 9.3.1 and 9.3.2, we now have two kinds of auxiliary functions for the first term of $\mathbf{E}$ in (9.1) with the following forms:

$$\mathbf{Q}^{(N)} = \sum_p \sum_i a_{i,p}^{(N)} (x_{ip}^{(N)})^2 + b_{i,p}^{(N)} x_{ip}^{(N)} + c_{ip}^{(N)} \tag{9.22}$$

$$\begin{aligned}\mathbf{R}^{(N)} =& \sum_p \sum_i \sum_j a_{ip}^{(N)} J^{(N-1)} (w_{ij}^{(N-1)})^2 (z_{jp}^{(N-1)} - y_{jp}^{(N-1)})^2 \\ &+ a_{ip}^{(N)} (u_i^{(N-1)})^2 + b_{ip}^{(N)} u_i^{(N-1)} + c_{ip}^{(N)} + const\end{aligned} \tag{9.23}$$

where $a_{ip}^{(N)}$, $b_{ip}^{(N)}$, $c_{ip}^{(N)}$, and $y_{jp}^{(N-1)}$ are defined in (9.8), (9.9), (9.10), and (9.19), respectively, $J^{(N-1)}$ is the number of neurons in the $(N-1)$-th layer, and *const* represents a term unrelated to optimization.

The expression of $\mathbf{R}^{(N)}$ is similar to that of the original objective function in that it is a sum of squared error terms of the form $(z-y)^2$. Therefore, we can recursively apply the above two lemmas in decreasing layer order n in a similar fashion as conventional back-propagation and obtain a sequence of auxiliary functions such that

$$\mathbf{E} \leq \mathbf{Q}^{(N)} \leq \mathbf{R}^{(N)} \leq \mathbf{Q}^{(N-1)} \leq \mathbf{R}^{(N-1)} \cdots \tag{9.24}$$

which guarantees the monotonic decrease of the objective function overall.

The optimal values of $w_{ij}^{(n-1)}$ and $u_i^{(n-1)}$ can be obtained by minimizing the sum of $\mathbf{Q}^{(n)}$ and the quadratic regularization term in (9.1). This minimization is costly as it involves some quadratic cross-terms. Noticing that the role of w_j and z_j in (9.17) is symmetric, we can derive a separable majorizing function for $\mathbf{Q}^{(n)}$ which has the same expression as (9.18) where the variables w_j and z_j are switched in (9.18) and (9.19). Each $w_{ij}^{(n-1)}$ and $u_i^{(n-1)}$ can then be separately computed by minimizing the sum of this majorizing function and the regularization term instead.

9.4 Algorithms

9.4.1 Auxiliary function based NN training

In summary, each iteration of the auxiliary function based NN training (AuxNNT) algorithm is described in Algorithm 1. Note that in Algorithm 1, the λ comes from the regularization term as it is explained in the Section 9.3.1.

9.4.2 Hybrid algorithm

One benefit of the proposed AuxNNT method is that it can be combined with any gradient based method such as ADAGRAD [Duchi et al., 2011]. The gradient can be computed at any point

Algorithm 1 Auxiliary function based method (AuxNNT)

```
Require: Initial parameters $w_{ij}^{(n)}$, $u_i^{(n)}$ for all $i$, $j$, $n$
  Compute forward pass using (8.2) and (8.3).
  for $n = N$ to 2
  1. Compute auxiliary function coefficients as follows:
       $\sigma_{ip}^{(n)} = \mathrm{sign}(y_{ip}^{(n)})$
       $a_{ip}^{(n)} = A_1(x_{ip}^{(n)}) + |y_{ip}^{(n)}|A_2(-\sigma_{ip}^{(n)}x_{ip}^{(n)})$
       $b_{ip}^{(n)} = y_{ip}^{(n)}[\sigma_{ip}^{(n)}x_{ip}^{(n)}A_2(-\sigma_{ip}^{(n)}x_{ip}^{(n)}) + \mathrm{sech}^2(x_{ip}^{(n)})]$
       $\beta_{ijp}^{(n)} = w_{ij}^{(n-1)}z_{jp}^{(n-1)} - \frac{1}{J^{(n-1)}}\sum_{j=1}^{J} w_{ij}^{(n-1)}z_{jp}^{(n-1)}$
       $y_{jp}^{(n-1)} = \frac{\sum_i \left(2a_{ip}^{(n)}J^{(n-1)}\beta_{ijp} - 2a_{i,p}^{(n)}u_i^{(n-1)} - b_{ip}^{(n)}\right) w_{ij}^{(n-1)}}{\sum_i 2a_{ip}^{(n)}J^{(n-1)}(w_{ij}^{(n-1)})^2}$
  2. Update the parameters in $(n-1)$-th layer as follows:
       $w_{ij}^{(n-1)} = \frac{\sum_p \left(2a_{ip}^{(n)}J^{(n-1)}\beta_{ijp} - 2a_{ip}^{(n)}u_i^{(n-1)} - b_{ip}^{(n)}\right) z_{jp}^{(n-1)}}{\sum_p 2a_{ip}^{(n)}J^{(n-1)}(z_{jp}^{(n-1)})^2 + \frac{\lambda}{PI^{(n)}}}$
       $u_i^{(n-1)} = \frac{\sum_p \left(-2a_{ip}^{(n)}J^{(n-1)}\sum_j \left(w_{ij}^{(n-1)}z_{j,p}^{(n-1)}\right) - b_{ip}^{(n)}\right)}{\sum_p 2a_{ip}^{(n)}J^{(n-1)}}$
   endfor
```

based on the parameters of the auxiliary function with lower computational effort. We observed in preliminary experiments that, when the change in the parameter values from the previous to the current iteration is small, ADAGRAD results in a greater decrease of the objective function than AuxNNT because the learning rate at the current iteration increases.

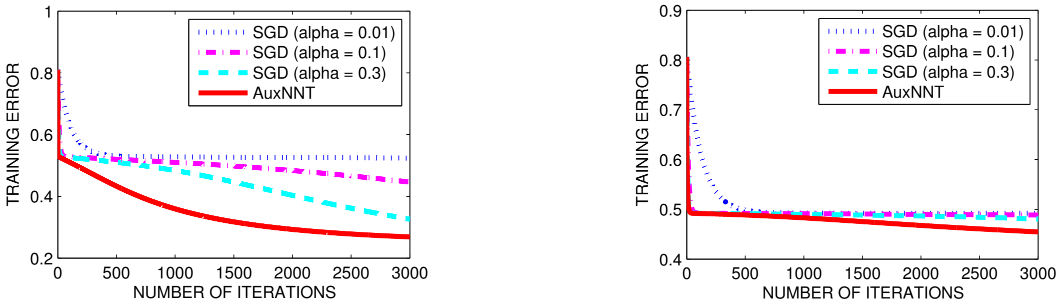
We propose an hybrid approach called Hybrid AuxNNT that takes advantage of both methods. Specifically, when the change in the parameter values is small, several iterations of ADAGRAD are performed. We then select the iteration number for which the gradient is largest and continue with AuxNNT onwards, until the change in the parameter values becomes small again. This hybrid method relies on two tuning parameters: a parameter change threshold ϵ and the number t_{eval} of ADAGRAD iterations. The details of each iteration of this hybrid algorithm are described in Algorithm 2. Note that, contrary to the original ADAGRAD method, not all gradients are accumulated in step 7.

Algorithm 2 Hybrid method (Hybrid AuxNNT)

```
Require: Initial parameters $w_{ij}^{(n)}$, $u_i^{(n)}$ for all $i$, $j$, $n$, $\Delta_k = 0$
Require: global learning rate $\alpha$, threshold $\epsilon$, number of gradient evaluations $t_{eval}$.
  1. Compute forward pass.
  2. Compute auxiliary function coefficients using Algorithm 1.
  3. Update the parameters for all layers using Algorithm 1.
  4. Fold $w_{ij}$ and $u_i$ into a vector $\theta$.
  5. Compute gradient $\partial\mathbf{E}/\partial\theta_k$.
  6. Accumulate square of gradient $\Delta_k \leftarrow \Delta_k + (\partial\mathbf{E}/\partial\theta_k)^2$.
  7. Compute $\delta_{\theta_k} = \theta_{k,\text{previous}} - \theta_{k,\text{current}}$
  if $\sum_k (\delta_{\theta_k})^2 < \epsilon$ then
     for $t = 1$ to $t_{\text{eval}}$ do
        Compute gradient $\partial\mathbf{E}_t/\partial\theta_{k,t}$.
        $\theta_{k,t+1} := \theta_{k,t} + \alpha \frac{\partial\mathbf{E}_t/\partial\theta_{k,t}}{\sqrt{\Delta + \sum_t (\partial\mathbf{E}_t/\partial\theta_{k,t})^2}}$
     end for
     $t_{\max} = \arg\max_{t \in \{1 \ldots t_{\text{eval}}\}} \sum_k (\partial\mathbf{E}_t/\partial\theta_{k,t})^2$
     $\theta_k = \theta_{k,t_{\max}}$.
  end if
  8. Go back to step 1
```

9.5 Experimental evaluation

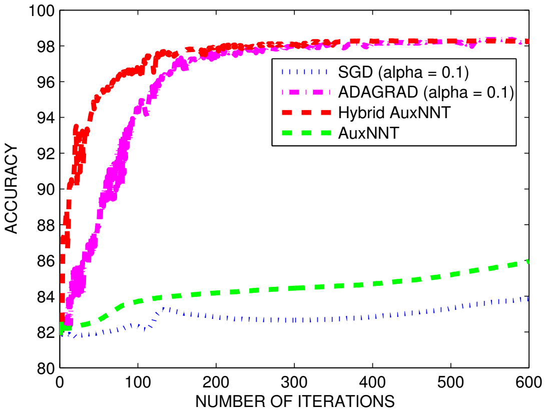
To analyze the effectiveness of the proposed methods, we conducted two experiments on the MNIST handwritten digits dataset [LeCun et al., 1998]. In both experiments, all parameters



(a) 1 hidden layer autoencoder

(b) 2 hidden layers autoencoder

Figure 9.3: Training progess on the MNIST dataset.



(c) 2 hidden layers neural network

Figure 9.4: Testing accuracy on the MNIST dataset for a 2 hidden layes neural network.

were initialized to random numbers drawn uniformly from the interval $-\sqrt{6/(J^{(n-1)}+I^{(n)}+1)}$, $\sqrt{6/(J^{(n-1)}+I^{(n)}+1)}$ where $J^{(n-1)}$ is the the number of inputs feeding into a neuron and $I^{(n)}$ is the the number of units that a neuron feeds into.

In the first experiment, we learned an autoencoder and analyzed the value of the objective function, a.k.a. the training error, over the iterations. Note that the objective function for the autoencoder classically includes a sparsity term which we did not include here. Two autoencoders were built: the first one has one hidden layer with 25 neurons and the second one has two hidden layers with 25 neurons in each hidden layer. The input and output layers have 64 neurons. To generate a training set, we sample 10000 8×8 image patches and concatenate them into a 64×10000 matrix. Figure 9.4 (a) and (b) shows that the AuxNNT method results in monotonic decrease of the training error and converges faster and to a better solution than SGD.

In the second experiment, we analyze the results in terms of classification accuracy. A simple neural network was designed where the input is a 28×28 image folded into a 784 dimensional vector and the output is the 10 dimensional posterior probability vector over the 10 digit classes. For example if the target is the digit “2”, then the second element of the output vector is equal to 1 and the 9 remaining elements are equal to 0. There are two hidden layers with 25 neurons for each layer. When decoding, the recognized digit corresponds to the biggest element in the output vector. The training data contains 10000 image samples. The optimal learning rate was set for ADAGRAD and SGD. Figure 9.3 shows that AuxNNT outperforms SGD and Hybrid AuxNNT outperforms all the other techniques, including ADAGRAD. Using the Hybrid AuxNNT method, we achieved 98.4% accuracy while with ADAGRAD the accuracy was 98.1%.

The computational cost of one iteration of SGD and AuxNNT is equal to 12 s and 30 s, respectively, four-layer networks . To reduce the computation cost of the Hybrid AuxNNT method, we used 1000 samples only to compute the gradient since we found in preliminary experiments that using all data did not significantly affect performance. All data were used to compute the gradient for ADAGRAD, however, since using only 1000 samples was found to degrade ADAGRAD’s performance. The computation cost of one iteration of the Hybrid AuxNNT method is equal to 32 s.

9.6 Summary

A new learning rule was proposed for neural networks based on an auxiliary function technique without parameter tuning. Instead of minimizing the objective function, a quadratic auxiliary function is recursively introduced layer by layer which has a closed form optimum. We also proved the monotonic decrease of the new update rule. Experimental results on the MNIST dataset showed that the proposed algorithm converges faster and to a better solution than SGD. In addition, we found the combination of ADAGRAD and the proposed method to accelerate convergence and to achieve a better performance than ADAGRAD alone. In the future, we will seek to improve the proposed AuxNNT method by using information from previous iterations as well as applying it to robust speech recognition and speech separation tasks.

Part IV

Conclusion and perspectives

10

Conclusion and perspectives

10.1 Conclusion

This thesis investigated the problem of noise robust automatic speech recognition. The first part of the thesis focused on the uncertainty handling framework: new estimation and propagation procedures were proposed that improve the speech recognition performance in a noisy environment. In the second part, a new method was proposed to accelerate the training of a neural network using an auxiliary function technique.

In the first part, three main contributions about uncertainty decoding were made. The first contribution is to model the correlation of uncertainty between the MFCCs which results in a full uncertainty covariance matrix. At the beginning, the uncertainty of the enhanced speech is estimated in the spectral domain by exploiting multichannel speech enhancement. Then, the cross-moments of amplitude and power of the enhanced speech are computed using the Rice distribution. Uncertainty is subsequently propagated through MFCC computation including preemphasis, Mel-filter bank, logarithm, Discrete Cosine Transform (DCT), lifting and time derivation. This is achieved using a first order Vector Taylor Series (VTS) expansion and this results in a full uncertainty covariance matrix. Experimental results on Track 1 of the 2nd CHiME Challenge show that full uncertainty covariance achieves 5% relative WER reduction compared to diagonal uncertainty covariance and 13% relative WER reduction compared to the baseline system (without uncertainty). However, modeling the correlation of uncertainty between the MFCCs and the log-energy does not seem to improve the speech recognition performance.

The second contribution was to propose a generative uncertainty estimator/propagator using either a fusion approach or a nonparametric approach. This contribution constitutes a breakthrough compared to existing uncertainty estimators/propagators, since it recasts the problem of uncertainty estimation/propagation as a machine learning problem. Based on the inherent nonnegative character of uncertainty, the estimation of the fusion weights and the nonparametric kernel weights is done by using multiplicative update rules. The proposed nonparametric estimator achieves 29% and 28% relative WER reduction on Track 1 and Track 2 of the 2nd

CHiME Challenge compared to the baseline system (without uncertainty), respectively. It also outperforms ROVER fusion by 9% relative WER reduction on Track 1. In addition, fusion and nonparametric estimation/propagation improve the accuracy of the estimated uncertainty compared to the Wiener + VTS approach both in the spectral domain and in the feature domain and the nonparametric uncertainty estimator/propagator provides the best accuracy.

The third contribution was to propose a method for discriminative linear and nonlinear mapping of the estimated full feature uncertainty covariance matrix into a transformed full feature uncertainty covariance matrix. Starting from the diagonal feature-domain uncertainty covariance matrices estimated by one of the nonparametric techniques, a linear and a nonlinear mapping are trained so as to maximize the discriminative boosted maximum mutual information (bMMI) criterion using stochastic gradient ascent. Using the learned nonlinear transformation improved the WER by 5% relative compared to the nonparametric framework.

In the second part of the thesis, the contribution was to accelerate the training of multilayer neural networks. To avoid tuning of the learning rate and derive an effective learning rule, we introduced an auxiliary function technique without parameter tuning. Instead of minimizing the objective function, a quadratic auxiliary function is recursively introduced layer by layer which has a closed-form optimum. Based on the auxiliary function behavior, the monotonic decrease of the new learning rule is guaranteed. In addition, we proposed a hybrid approach that takes advantage of both adaptive subgradient methods (ADAGRAD) and the auxiliary function technique. Experimental results on the MNIST dataset showed that the proposed algorithm converges faster and to a better solution than stochastic gradient descent (SGD) for both auto-encoding and classification tasks. We found that the combination of ADAGRAD and the proposed method accelerates the convergence and achieves a better performance than ADAGRAD alone.

10.2 Perspectives

The general theoretical concepts and the experiments presented in this thesis suggest future development of the work in the following directions.

We showed that uncertainty handling can significantly improve the performance of robust ASR systems when the acoustic model is a GMM-HMM. However, some problems remain to be investigated. On the theoretical side, first, DNN based acoustic models were shown to outperform GMM based acoustic models [Hinton et al., 2012]. Unfortunately, there does not exist a closed-form solution for uncertainty decoding in DNNs, yet. Following the initial study recently reported in [Astudillo and Neto, 2011; Abdelaziz et al., 2015], an efficient approach for uncertainty decoding (and uncertainty training too) for DNN based acoustic models is needed. One way is to take inspiration from related works which investigated how to train Support Vector Machines (SVM) from uncertain data [Bi and Zhang, 2004] or feed-forward neural networks from uncertain or missing data [Ghahramani and Jordan, 1994; Tresp et al., 1994; Buntine and Weigend, 2004]. Second, learning the uncertainty using a neural network to transform the Wiener gain or the uncertainties in the spectral domain directly to the feature domain appears promising. This might be expected to mitigate the nonlinear effect of the log computation. Third, estimating the correlation of uncertainty across frames or frequency bins could also help improving the accuracy of the estimated off-diagonal elements of the full uncertainty covariance matrix. On the experimental side, the rigorous optimization and evaluation of the generative and discriminative training criteria with full uncertainty covariance matrices is needed. In addition, validation the benefit of the uncertainty decoding framework on a large vocabulary dataset and in more diverse noise conditions would also be insightful.

Regarding the training of neural networks, we showed that the proposed technique is more efficient than SGD for both auto-encoding and classification tasks. However, there are some open problems. On the theoretical side, first, the performance of the auxiliary function technique must be evaluated for neural networks with different architectures and/or more hidden layers. Second, derivation of the auxiliary function for other objective functions such as the cross-entropy and for other activation functions like maxout and rectified linear units is needed. On the experimental side, the benefit of the auxiliary function technique remains to be evaluated for some specific tasks such as speech enhancement [Huang et al., 2014] or ASR [Hinton et al., 2012].

A

Tables

A.1 Comparison of uncertainty decoding on oracle uncertainties on Track 1

uncertainty	Test set							Development set						
matrix	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average
no	73.75	78.42	84.33	89.50	91.83	92.25	85.01	73.25	78.02	84.33	89.25	91.75	92.18	84.80
diag	93.58	92.67	94.92	95.25	95.58	95.42	94.57	92.92	93.00	94.17	95.67	95.00	95.25	94.33
full	96.33	96.00	96.33	96.50	96.67	96.08	96.31	96.02	96.17	96.00	96.17	96.33	96.08	96.13

Table A.1: Keyword accuracy (%) evaluated with the oracle uncertainties on the Track 1 test set after speech enhancement. Average accuracies have a 95% confidence interval of $\pm 0.8, \pm 0.5, \pm 0.4\%$ for no uncertainty, diagonal and full uncertainty covariance respectively

A.2 Comparison of uncertainty decoding on the oracle uncertainties on Track 2

Uncertainty	Test set							Development set						
covariance matrix	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average
no uncertainty	68.47	63.75	56.76	51.03	44.22	39.12	53.89	72.52	66.69	59.71	54.21	46.32	40.04	56.58
diagonal	30.9	28.27	24.09	22.14	20.05	18.21	23.94	34.92	31.05	27.25	24.78	23.10	19.88	26.83
full	19.49	19.24	18.78	18.14	19.17	18.01	18.80	21.14	21.62	21.38	20.14	22.08	21.41	21.29

Table A.2: WER (%) evaluated on the oracle uncertainties on the Track 2 test set. Average WER have a 95% confidence interval of $\pm 1.1, \pm 0.9, \pm 0.9\%$ for no uncertainty, diagonal and full uncertainty covariance, respectively.

A.3 Comparison of uncertainty decoding of static and dynamic features on Track 1

Uncertainty	Uncertainty	Test set							Development set						
covariance matrix	features	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average
no uncertainty		73.75	78.42	84.33	89.50	91.83	92.25	85.01	73.25	78.02	84.33	89.25	91.75	92.18	84.80
diagonal	static	75.00	79.00	84.75	90.13	91.92	93.67	85.74	74.93	78.75	84.83	89.92	91.83	92.18	85.41
	dynamic	75.00	79.00	84.92	90.33	91.92	92.33	85.58	74.67	78.92	84.75	89.50	91.93	92.48	85.37
	all	76.93	79.17	85.92	90.00	92.00	93.75	86.29	76.13	78.75	85.56	89.68	91.75	93.50	85.89
full	static	76.75	79.33	85.50	90.33	92.33	93.67	86.31	76.40	79.33	85.50	89.75	91.92	92.38	85.88
	dynamic	76.75	79.17	85.75	90.33	92.00	93.83	86.30	76.17	79.25	85.50	89.75	91.92	92.55	85.85
	all	77.92	80.75	86.75	90.50	92.92	93.75	**87.00**	77.92	79.81	86.51	89.93	92.92	93.75	**86.80**

Table A.3: Keyword accuracy (%) on the Track 1 dataset achieved by uncertainty decoding of static and dynamic features. Average accuracies have a 95% confidence interval of ±0.8%.

A.4 Comparison of uncertainty decoding of various fusion or nonparametric mapping schemes

			Test set							Development set						
Estimation	Propagation	Uncertainty covariance matrix	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average
Wiener	VTS+Scaling	diagonal	78.67	79.50	86.33	90.17	92.08	93.75	86.75	78.25	79.17	85.92	89.87	91.80	93.41	86.40
fusion	VTS		78.33	80.17	85.92	90.08	92.08	94.17	86.97	78.33	80.17	85.75	89.92	92.50	93.50	86.69
fusion	fusion		80.50	82.17	88.25	91.33	92.50	93.58	88.05	80.00	81.92	87.25	91.50	92.25	93.08	87.66
nonparametric	VTS		80.00	81.92	87.25	91.50	92.25	93.08	87.66	79.75	81.67	87.17	89.75	91.58	93.50	87.23
nonparametric	nonparametric		81.75	83.50	88.33	91.08	92.75	93.00	**88.40**	80.83	82.00	88.25	90.50	92.67	93.50	**87.95**
Wiener	VTS+Scaling	full	81.75	81.83	88.17	90.50	92.67	93.75	88.11	80.63	81.87	87.35	90.57	92.33	93.75	87.75
fusion	VTS		81.00	81.50	87.33	91.00	93.50	94.92	88.20	80.33	81.33	87.17	91.08	92.25	93.50	87.68
fusion	fusion		83.17	84.33	89.75	91.17	93.33	93.33	89.18	83.33	83.25	88.42	91.50	93.17	93.17	88.73
nonparametric	VTS		82.33	82.58	88.00	92.00	93.33	93.92	88.69	81.42	82.00	87.92	91.75	92.50	93.75	88.22
nonparametric	nonparametric		83.78	84.92	88.42	91.25	93.75	94.42	**89.42**	83.00	83.50	88.67	92.08	93.00	93.75	**89.00**

Table A.4: Keyword accuracy (%) achieved with various fusion or nonparametric mapping schemes on the Track 1 test dataset. This is to be compared to the baseline Wiener+VTS performance in Table 5.1.

A.5 ASR performance with ROVER fusion

	Test set							Development set						
Uncertainty	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average
diagonal	79.08	80.75	86.00	90.17	92.08	94.17	87.04	78.93	80.33	85.75	90.00	92.50	93.50	86.83
full	81.33	81.75	87.50	91.08	93.75	94.92	88.38	80.75	81.50	87.33	91.17	92.25	93.50	87.75

Table A.5: Keyword accuracy (%) achieved with ROVER fusion.

A.6 Comparison of ASR performance on Track 2 of the 2nd CHiME Challenge with GMM-HMM acoustic models

Test condition and	Test set							Development set						
estimated uncertainty	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average
noisy	82.81	78.09	70.15	64.80	53.79	47.02	66.11	86.64	81.22	72.05	66.51	55.86	48.34	68.43
enhanced	68.47	63.75	56.76	51.03	44.22	39.12	53.89	72.52	66.69	59.71	54.21	46.32	40.04	56.58
Wiener + VTS (diag.)	65.23	61.82	55.18	50.27	43.11	38.79	52.40	69.75	64.78	57.98	53.50	45.92	39.12	55.17
nonparametric + VTS (diag.)	58.12	52.54	49.95	44.42	40.23	35.31	46.76	62.32	56.74	52.45	47.39	42.54	36.02	49.58
nonparam. + nonparam. (diag.)	53.70	48.69	45.72	40.18	37.43	34.18	43.32	57.82	52.63	48.77	43.33	39.01	35.02	46.10
Wiener + VTS (full)	63.58	58.85	53.06	48.19	42.09	38.41	50.70	67.80	62.03	56.15	51.19	44.42	39.18	53.46
nonparametric + VTS (full)	51.91	46.95	43.48	38.91	35.11	32.55	41.49	55.32	50.74	45.45	41.39	37.54	33.14	44.10
nonparam. + nonparam. (full)	46.34	42.75	41.54	37.57	34.21	30.01	**38.74**	50.01	46.13	44.12	40.15	36.05	32.43	**41.48**

Table A.6: WER (%) achieved on Track 2 of the 2nd CHiME Challenge with GMM-HMM acoustic models trained on reverberated noiseless data.

A.7 Comparison of ASR performance on Track 2 of the 2nd CHiME Challenge with a DNN acoustic model

Training and test	Test set							Development set						
condition	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average
noisy	50.33	40.82	30.71	24.60	21.43	16.87	30.79	56.82	45.88	36.38	30.71	25.82	22.54	36.36
enhanced	41.51	31.46	26.04	21.51	17.82	16.74	25.85	48.72	37.69	32.35	28.04	24.59	21.18	32.10

Table A.7: WER (%) achieved on Track 2 of the 2nd CHiME Challenge with a DNN acoustic model trained on enhanced data.

A.8 ASR performance with dicriminative uncertainty estimator

Method	state dependent	Test set							Development set						
		-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average
no uncertainty	no	73.75	78.42	84.33	89.50	91.83	92.25	85.01	73.25	78.02	84.33	89.25	91.75	92.18	84.80
nonparametric + bMMI (diag)	no	82.33	83.75	88.50	91.17	92.75	93.00	88.58	81.92	83.33	88.33	91.00	92.50	93.00	88.34
nonparametric + bMMI (full)	no	84.17	85.00	88.75	91.33	93.75	94.42	89.57	83.75	84.18	89.00	92.33	93.08	94.17	89.41
squared diff + bMMI [Delcroix et al., 2011]	yes	79.92	82.00	87.17	90.67	92.92	93.42	87.68	79.50	81.92	87.00	90.50	92.67	93.42	87.50
nonparametric + bMMI(diag)	yes	82.93	83.75	88.50	91.17	92.75	93.33	88.73	82.33	83.67	88.33	91.00	92.50	93.17	88.50
nonparametric + bMMI(full)	yes	84.33	85.00	88.92	91.50	93.75	94.50	**89.66**	83.92	84.28	89.08	92.17	93.67	94.24	89.56

Table A.8: Keyword accuracy (%) on the Track 1 test set with discriminative linear mapping. Average accuracies have a 95% confidence interval of ±0.8%.

Method	state dependent	Test set							Development set						
		-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average
no uncertainty	no	73.75	78.42	84.33	89.50	91.83	92.25	85.01	73.25	78.02	84.33	89.25	91.75	92.18	84.80
nonparametric + bMMI (diag)	no	82.75	84.00	88.50	91.17	93.00	93.17	88.76	82.00	83.00	88.83	91.00	92.92	93.00	88.45
nonparametric + bMMI (full)	no	84.55	85.30	88.75	91.33	93.75	94.42	89.68	83.33	84.50	88.92	91.17	93.50	94.33	89.30
nonparametric + bMMI(diag)	yes	83.33	84.00	88.50	91.33	93.00	93.75	88.98	82.92	83.50	88.50	91.00	92.92	93.33	88.69
nonparametric + bMMI(full)x	yes	84.75	85.50	89.00	91.75	93.75	95.00	**89.95**	84.00	84.83	89.25	92.33	93.92	94.67	89.63

Table A.9: Keyword accuracy (%) on the Track 1 test set with discriminative nonlinear mapping. Average accuracies have a 95% confidence interval of ±0.8%.

Bibliography

[Abdelaziz et al., 2015] Abdelaziz, A. H., Watanabe, S., Hershey, J. R., Vincent, E., and Kolossa, D. (2015). Uncertainty propagation through deep neural networks. In *Proc. Interspeech*.

[Acero et al., 2000] Acero, A., Deng, L., Kristjansson, T., and Zhang, J. (2000). HMM adaptation using vector Taylor series for noisy speech recognition. In *Proc. ICSLP*, pages 869–872.

[Acero and Stern, 1990] Acero, A. and Stern, R. M. (1990). Environmental robustness in automatic speech recognition. In *Proc. ICASSP*, volume 2, pages 849–852.

[Arberet et al., 2010] Arberet, S., Ozerov, A., Duong, N. Q. K., Vincent, E., Gribonval, R., Bimbot, F., and Vandergheynst, P. (2010). Nonnegative matrix factorization and spatial covariance model for under-determined reverberant audio source separation. In *Proc. ISSPA*, pages 1–4.

[Arisoy et al., 2012] Arisoy, E., Sainath, T. N., Kingsbury, B., and Ramabhadran, B. (2012). Deep neural network language models. In *Proc. NAACL-HLT Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*, pages 20–28.

[Astudillo, 2010] Astudillo, R. (2010). *Integration of Short-Time Fourier Domain Speech Enhancement and Observation Uncertainty Techniques for Robust Automatic Speech Recognition.* PhD thesis, TU Berlin.

[Astudillo and Kolossa, 2011] Astudillo, R. and Kolossa, D. (2011). Uncertainty propagation. In Kolossa, D. and Haeb-Umbach, R., editors, *Robust Speech Recognition of Uncertain or Missing Data - Theory and Applications*, pages 35–62. Springer.

[Astudillo, 2013] Astudillo, R. F. (2013). An extension of STFT uncertainy propagation for GMM-based super-Gaussian a priori models. *IEEE Signal Processing Letters*, 20(12):1163–1166.

[Astudillo et al., 2014] Astudillo, R. F., Braun, S., and Habets, E. A. P. (2014). A multichannel feature compensation approach for robust ASR in noisy and reverberant environments. In *Workshop REVERB*.

[Astudillo et al., 2013] Astudillo, R. F., Kolossa, D., Abad, A., Zeiler, S., Saeidi, R., Mowlaee, P., da Silva Neto, J. P., and Martin, R. (2013). Integration of beamforming and uncertainty-of-observation techniques for robust ASR in multi-source environments. *Computer Speech and Language*, 27(3):837–850.

[Astudillo and Neto, 2011] Astudillo, R. F. and Neto, J. (2011). Propagation of uncertainty through multilayer perceptrons for robust automatic speech recognition. In *Proc. Interspeech*, pages 461–464.

[Astudillo and Orglmeister, 2013] Astudillo, R. F. and Orglmeister, R. (2013). Computing MMSE estimates and residual uncertainty directly in the feature domain of ASR using STFT domain speech distortion models. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(5):1023–1034.

[Baker et al., 2009] Baker, J. M., Deng, L., Glass, J., Khudanpur, S., Lee, C., Morgan, N., and O'Shaughnessy, D. (2009). Research developments and directions in speech recognition and understanding, part 1. *IEEE Signal Processing Magazine*, 26(3):75–80.

[Barker et al., 2013] Barker, J., Vincent, E., Ma, N., Christensen, H., and Green, P. (2013). The PASCAL CHiME speech separation and recognition challenge. *Computer Speech and Language*, 27(3):621–633.

[Baumann et al., 2003] Baumann, W., Kolossa, D., and Orglmeister, R. (2003). Beamforming-based convolutive source separation. In *Proc. ICASSP*, volume 5, pages 357–60.

[Becker et al., 1997] Becker, M. P., Yang, I., and Lange, K. (1997). EM algorithms without missing data. *Statistical Methods in Medical Research*, 6:38–54.

[Bell and Sejnowski, 1995] Bell, A. J. and Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 6(6):1004–1034.

[Bi and Zhang, 2004] Bi, J. and Zhang, T. (2004). Support vector classification with input data uncertainty. *Proc. NIPS*, pages 161–168.

[Böhning and Lindsay, 1988] Böhning, D. and Lindsay, B. G. (1988). Monotonicity of quadratic-approximation algorithms. *Annals of the Institute of Statistical Mathematics*, 40(4):641–663.

[Boll, 1979] Boll, S. (1979). Suppression of acoustic noise in speech using spectral subtraction. In *Proc. ICASSP*, volume 27, pages 113–120.

[Bordes et al., 2009] Bordes, A., Bottou, L., and Gallinari, P. (2009). SGD-QN: Careful quasi-newton stochastic gradient descent. *Journal of Machine Learning Research*, 10:1737–1754.

[Bourlard et al., 1992] Bourlard, H., Morgan, N., Wooters, C., and Renals, S. (1992). CDNN: a context dependent neural network for continuous speech recognition. In *Proc. ICASSP*, volume 2, pages 349–352.

[Bourlard and Wellekens, 1990] Bourlard, H. and Wellekens, C. (1990). Links between markov models and multilayer perceptrons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(12):1167–1178.

[Brutti et al., 2008] Brutti, A., Cristoforetti, L., Kellermann, W., Marquardt, L., and Omologo, M. (2008). Woz acoustic data collection for interactive TV. In *Proc. LREC*.

[Buntine and Weigend, 2004] Buntine, W. and Weigend, A. (2004). Bayesian backpropagation. *Complex systems*, 5(6):603–643.

[Cardoso, 1997] Cardoso, J. F. (1997). Infomax and maximum likelihood for blind source separation. *Neural Computation*, 4(4):112–114.

[Cohen, 2003] Cohen, I. (2003). Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging. *IEEE Transactions Audio, Speech, and Language Processing*, 11(5):466–475.

[Comon, 1994] Comon, P. (1994). Independent component analysis, a new concept? *Signal Processing*, 36:287–314.

[Cooke et al., 2006] Cooke, M., Barker, J., Cunningham, S., and Shao, X. (2006). An audio-visual corpus for speech perception and automatic speech recognition. In *Journal of the Acoustical Society of America*, volume 120, pages 2421–2424.

[Cooke et al., 2001] Cooke, M., Green, P., Josifovski, L., and Vizinho, A. (2001). Robust automatic speech recognition with missing and unreliable acoustic data. *Computer Speech and Language*, 34(3):267–285.

[Cristoforetti et al., 2014] Cristoforetti, L., Ravanelli, M., Omologo, M., Sosi, A., Abad, A., Hagmuller, M., and Maragos, P. (2014). The DIRHA simulated corpus. In *Proc. LREC*.

[Dahl et al., 2012] Dahl, G. E., Yu, D., Deng, L., and Acero, A. (2012). Context-dependent pre-trained deep neural networks for large vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):30–42.

[Davis and Mermelstein, 1980] Davis, S. B. and Mermelstein, P. (1980). Comparison of parametric representations for mono-syllabic word recognition in continuous spoken sentences. *IEEE Transactions on Audio, Speech, and Language Processing*, 28(4):357–366.

[de Leeuw, 1994] de Leeuw, J. (1994). Block relaxation algorithms in statistics. In *Information Systems and Data Analysis*, pages 308–325. Springer.

[de Leeuw and Lange, 2009] de Leeuw, J. and Lange, K. (2009). Sharp quadratic majorization in one dimension. *Computational Stattistics and Data Analysis*, 53:2471–2484.

[Delcroix et al., 2013a] Delcroix, M., Kinoshita, K., Nakatani, T., Araki, S., Ogawa, A., Hori, T., Watanabe, S., Fujimoto, M., Yoshioka, T., Oba, T., Kubo, Y., Souden, M., Hahm, S., and Nakamura, A. (2013a). Speech recognition in living rooms: Integrated speech enhancement and recognition system based on spatial, spectral and temporal modeling of sounds. *Computer Speech and Language*, 27(3):851–873.

[Delcroix et al., 2009] Delcroix, M., Nakatani, T., and Watanabe, S. (2009). Static and dynamic variance compensation for recognition of reverberant speech with dereverberation preprocessing. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(2):324–334.

[Delcroix et al., 2011] Delcroix, M., Watanabe, S., Nakatani, T., and Nakamura, A. (2011). Discriminative approach to dynamic variance adaptation for noisy speech recognition. In *Proc. HSCMA*, pages 7–12.

[Delcroix et al., 2013b] Delcroix, M., Watanabe, S., Nakatani, T., and Nakamura, A. (2013b). Cluster-based dynamic variance adaptation for interconnecting speech enhancement preprocessor and speech recognizer. *Computer Speech and Language*, 27(1):350–368.

[Deng, 2011] Deng, L. (2011). Front-end, back-end, and hybrid techniques for noise-robust speech recognition. In Kolossa, D. and Haeb-Umbach, R., editors, *Robust Speech Recognition of Uncertain or Missing Data - Theory and Applications*, pages 67–99. Springer.

[Deng et al., 2000] Deng, L., Acero, A., Plumpe, M., and Huang, X. D. (2000). Large vocabulary speech recognition under adverse acoustic environments. In *Proc. ICSLP*, pages 806–809.

[Deng et al., 2005] Deng, L., Wu, J., Droppo, J., and Acero, A. (2005). Dynamic compensation of HMM variances using the feature enhancement uncertainty computed from a parametric model of speech distortion. *IEEE Transactions on Audio, Speech, and Language Processing*, 13(3):412–421.

[Doclo and Moonen, 2002] Doclo, S. and Moonen, M. (2002). GSVD-based optimal filtering for single and multimicrophone speech enhancement. *IEEE Transactions on Signal Processing*, 50(9):2230–2244.

[Droppo et al., 2002] Droppo, J., Acero, A., and Deng, L. (2002). Uncertainty decoding with SPLICE for noise robust speech recognition. In *Proc. ICASSP*, pages 56–60.

[Duchi et al., 2011] Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. In *Proc. ICML*, pages 2121–2159.

[Duong et al., 2010] Duong, N. Q., Vincent, E., and Gribonval, R. (2010). Under-determined reverberant audio source separation using a full-rank spatial covariance model. *IEEE Transactions on Audio, Speech and Language Processing*, 18(7):1830–1840.

[Ephraim and Malah, 1984] Ephraim, Y. and Malah, D. (1984). Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator. *IEEE Transactions on Audio, Speech, and Language Processing*, 32(6):1109–1121.

[Ephraim and Malah, 1985] Ephraim, Y. and Malah, D. (1985). Speech enhancement using a minimum mean-square error log-spectral amplitude estimator. *IEEE Transactions on Audio, Speech, and Language Processing*, 33(2):443–445.

[Fiscus, 1997] Fiscus, J. G. (1997). A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). In *Proc. ASRU*, pages 347–354.

[Flanagan et al., 1993] Flanagan, J. L., Surendran, A. C., and Jan, E. E. (1993). Spatially selective sound capture for speech and audio processing. *Speech Communication*, 13(1-2):207–222.

[Frey et al., 2001] Frey, B. J., Deng, L., Acero, A., and Kristjansson, T. (2001). ALGONQUIN:iterating Laplace's method to remove multiple types of acoustic distortion for robust speech recognition. In *Proc. Eurospeech*, pages 901–904.

[Févotte and Cardoso, 2005] Févotte, C. and Cardoso, J. (2005). Maximum likelihood approach for blind audio source separation using time-frequency Gaussian source models. In *Proc. ICASSP*, pages 78–81.

[Févotte et al., 2013] Févotte, C., Le Roux, J., and Hershey, J. R. (2013). Non-negative dynamical system with application to speech and audio. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3158–3162.

[Gales, 1995] Gales, M. (1995). *Model Based Techniques for Noise Robust Speech Recognition.* PhD thesis, Cambridge University.

[Gales and Young, 1996] Gales, M. and Young, S. (1996). Robust continuous speech recognition using parallel model combination. *IEEE Transactions on Speech and Audio Processing*, 4(5):352–359.

[Gales and Young, 2008] Gales, M. and Young, S. (2008). The application of hidden Markov models in speech recognition. *Journal Foundations and Trends in Signal Processing*, 1(3):195–304.

[Gales, 1998] Gales, M. J. F. (1998). Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech and Language*, 12(2):75–98.

[Gannot et al., 2001] Gannot, S., Burshtein, D., and Weinstein, E. (2001). Signal enhancement using beamforming and nonstationarity with application to speech. *IEEE Transactions on Signal Processing*, 49(8):1614–1626.

[Garofalo et al., 2007] Garofalo, J., Graff, D., Paul, D., and Pallett, D. (2007). CSR-I (WSJ0) complete. *Linguistic Data Consortium, Philadelphia.*

[Gauvain and Lee, 1994] Gauvain, J.-L. and Lee, C.-H. (1994). Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing*, 2(2):291–298.

[Gemmeke et al., 2011] Gemmeke, J., Virtanen, T., and Hurmalainen, A. (2011). Exemplar-based sparse representations for noise robust automatic speech recognition. *IEEE Transactions Audio, Speech, and Language Processing*, 19(7):2067–2080.

[Ghahramani and Jordan, 1994] Ghahramani, Z. and Jordan, M. I. (1994). Supervised learning from incomplete data via an em approach. *Proc. NIPS*, page 120–127.

[Goodfellow et al., 2013] Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y. (2013). Maxout networks. In *Proc. ICML*, pages 1319–1327.

[Gradshteyn and Ryzhik, 1995] Gradshteyn, I. S. and Ryzhik, I. M. (1995). *Table of Integrals, Series and Products.* Academic Press.

[Greenberg and Kingsbury, 1997] Greenberg, S. and Kingsbury, B. E. D. (1997). The modulation spectrogram: in pursuit of an invariant representation of speech. In *Proc. ICASSP*, volume 3, pages 1647–1650.

[Grezl et al., 2007] Grezl, F., Karafiat, M., Kontar, S., and Cernocky, J. (2007). Probabilistic and bottle-neck features for LVCSR of meetings. In *Proc. ICASSP*, volume 4, pages 1520–6149.

[Häb-Umbach and Ney, 1992] Häb-Umbach, R. and Ney, H. (1992). Linear discriminant analysis for improved large vocabulary continuous speech recognition. In *Proc. ICASSP*, pages 13–16.

[Hansen et al., 2001] Hansen, J. H. L., Angkititrakul, P., Plucienkowski, J., Gallant, S., Yapanel, U., Pellom, B., Ward, W., and Cole, R. (2001). Cu-move: Analysis corpus development for interactive in-vehicle speech systems. In *Proc. EUROSPEECH*, pages 2023–2026.

[Heiser, 1995] Heiser, W. J. (1995). Convergent computing by iterative majorization: theory and applications in multidimensional data analysis. In *Recent Advances in Descriptive Multivariate Analysis*, pages 157–189. Clarendon Press.

[Hermansky, 1990] Hermansky, H. (1990). Perceptual linear predictive (PLP) analysis of speech. *Journal of the Acoustical Society of America*, 87(4):1738–1752.

[Hermansky, 2000] Hermansky, H. (2000). Tandem connectionist feature extraction for conventional HMM systems. In *Proc. ICASSP*, volume 3, pages 1635–1638.

[Hinton et al., 2012] Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., and Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine*, 29(6):82–97.

[Hinton et al., 2006] Hinton, G., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

[Huang et al., 2014] Huang, P.-S., Kim, M., Hasegawa-Johnson, M., and Smaragdis, P. (2014). Deep learning for monaural speech separation. In *Proc. ICASSP*, pages 1562–1566.

[Hunter and Lange, 2004] Hunter, D. R. and Lange, K. (2004). A tutorial on MM algorithms. *The American Statistician*, 58:30–37.

[Hurmalainen et al., 2011] Hurmalainen, A., Gemmeke, J., and Virtanen, T. (2011). Non-negative matrix deconvolution in noise robust speech recognition. In *Proc. ICASSP*, pages 4588–4591.

[Ion and Haeb-Umbach, 2006] Ion, V. and Haeb-Umbach, R. (2006). Uncertainty decoding for distributed speech recognition over error-prone networks. *Speech communication*, 48:1435–1446.

[Julier and Uhlmann, 2004] Julier, S. and Uhlmann, J. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92:401–422.

[Kallasjoki et al., 2014] Kallasjoki, H., Gemmeke, J. F., and J.Palomäki, K. (2014). Estimating uncertainty to improve exemplar-based feature enhancement for noise robust speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 22(2):368–380.

[Kallasjoki et al., 2011] Kallasjoki, H., Keronen, S., Brown, G. J., Gemmeke, J. F., Remes, U., and Palomäki, K. J. (2011). Mask estimation and sparse imputation for missing data speech recognition in multisource reverberant environments. In *Proc. CHiME*, pages 58–63.

[Kim and Stern, 2009] Kim, C. and Stern, R. (2009). Power-normalized cepstral coefficients (PNCC) for robust speech recognition. In *Proc. Interspeech*, pages 1231–1234.

[Kolossa et al., 2010] Kolossa, D., Astudillo, R., Hoffmann, E., and Orglmeister, R. (2010). Independent component analysis and time-frequency masking for multi speaker recognition. *EURASIP Journal on Audio, Speech, and Music Processing.* Article ID 651420.

[Kolossa et al., 2011] Kolossa, D., Astudillo, R. F., Abad, A., Zeiler, S., Saeidi, R., Mowlaee, P., da Silva Neto, J., and Martin, R. (2011). CHIME challenge: approaches to robustness using beamforming and uncertainty-of-observation techniques. In *Proc. CHiME*, pages 6–11.

[Kolossa and Haeb-Umbach, 2011] Kolossa, D. and Haeb-Umbach, R., editors (2011). *Robust Speech Recognition of Uncertain or Missing data.* Springer.

[Kompass, 2007] Kompass, R. (2007). A generalized divergence measure fon nonnegative matrix factorization. *Neural Computation*, 19(3):780–791.

[Krueger and Haeb-Umbach, 2013] Krueger, A. and Haeb-Umbach, R. (2013). Model based feature enhancement for automatic speech recognition in reverberant environments. In *Proc. ICASSP*, pages 126–130.

[Kumatani et al., 2012] Kumatani, K., McDonough, J., and Raj, B. (2012). Microphone array processing for distant speech recognition: From close-talking microphones to far-field sensors. *IEEE Signal Processing Magazine: Special Issue on Fundamentals of Modern Speech Recognition*, 29:127–140.

[Lange et al., 2000] Lange, K., Hunter, D. R., and Yang, I. (2000). Optimization transfer using surrogate objective functions (with discussion). *Journal of Computational and Graphical Statistics*, 9:1–20.

[Le Roux and Vincent, 2014] Le Roux, J. and Vincent, E. (2014). A categorization of robust speech processing datasets. Technical Report TR2014-116, Mitsubishi Electric Research Labs.

[LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278–2324.

[Lee and Seung, 1999] Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects with nonnegative matrix factorization. *Nature*, 401:788–791.

[Leggetter and Woodland, 1995] Leggetter, C. J. and Woodland, P. C. (1995). Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech and Language*, 9(2):171–185.

[Li and Sim, 2014] Li, B. and Sim, K. C. (2014). An ideal hidden-activation mask for deep neural networks based noise-robust speech recognition. In *Proc. ICASSP*, pages 200–204.

[Liao, 2007] Liao, H. (2007). *Uncertainty Decoding for Noise Robust Speech Recognition.* PhD thesis, Cambridge University.

[Martens, 2010] Martens, J. (2010). Extracting and composing robust features with denoising autoencoders. In *Proc. ICML*.

[Martin, 2003] Martin, R. (2003). Statistical methods for the enhancement of noisy speech. In *Proc. IWAENC*, pages 1–6.

[McAulay and Malpass, 1980] McAulay, R. J. and Malpass, M. L. (1980). Speech enhancement using a soft-decision noise suppression filter. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28:137–145.

[McDermott et al., 2010] McDermott, E., Watanabe, S., and Nakamura, A. (2010). Discriminative training based on an integrated view of MPE and MMI in margin and error space. In *Proc. ICASSP*, pages 4894–4897.

[Mikolov et al., 2010] Mikolov, T., Karafiát, M., Burget, L., Èernocký, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *Proc. Interspeech*, pages 1045–1048.

[Moreno et al., 1996] Moreno, P. J., Raj, B., and Stern, R. M. (1996). A vector Taylor series approach for environment-independent speech recognition. In *IEEE ICASSP*, volume 2, pages 733–736.

[Nesta et al., 2013] Nesta, F., Matassoni, M., and Astudillo, R. (2013). A flexible spatial blind source extraction framework for robust speech recognition in noisy environments. In *Proc. CHiME*, pages 33–40.

[Ono, 2011] Ono, N. (2011). Stable and fast update rules for independent vector analysis based on auxiliary function technique. In *Proc. WASPAA*, pages 189–192.

[Ono et al., 2008] Ono, N., Miyamoto, K., Le Roux, J., Kameoka, H., and Sagayama, S. (2008). Separation of a monaural audio signal into harmonic/percussive components by complementary diffusion on spectrogram. In *Proc. EUSIPCO*, pages 1–4.

[Ozerov and Févotte, 2010] Ozerov, A. and Févotte, C. (2010). Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation. *IEEE Transactions on Speech and Audio Processing*, 18(3):550–563.

[Ozerov et al., 2013] Ozerov, A., Lagrange, M., and Vincent, E. (2013). Uncertainty-based learning of acoustic models from noisy data. *Computer Speech and Language*, 27(3):874–894.

[Ozerov et al., 2012] Ozerov, A., Vincent, E., and Bimbot, F. (2012). A general flexible framework for the handling of prior information in audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(4):1118 – 1133.

[Povey, 2005] Povey, D. (2005). *Discriminative training for large vocabulary speech recognition.* PhD thesis, University of Cambridge.

[Povey et al., 2008] Povey, D., Kanevsky, D., Kingsbury, B., Ramabhadran, B., Saon, G., and Visweswariah, K. (2008). Boosted MMI for model and feature-space discriminative training. In *Proc. ICASSP*, pages 4057–4060.

[Povey et al., 2005] Povey, D., Kingsbury, B., Mangu, L., Saon, G., Soltau, H., and Zweig, G. (2005). fMPE: Discriminatively trained features for speech recognition. In *Proc. ICASSP*, pages 961–964.

[Povey and Woodland, 2002] Povey, D. and Woodland, P. C. (2002). Minimum phone error and i-smoothing for improved discriminative training. In *Proc. ICASSP*, pages 105–108.

[Rabiner and Juang, 1993] Rabiner, L. and Juang, B. H., editors (1993). *Fundamentals of Speech Recognition.* Prentice Hall PTR.

[Rabiner and Levinson, 1981] Rabiner, L. and Levinson, S. E. (1981). Isolated and connected word recognition-theory and selected applications. *IEEE Transactions on Communications*, 29(5):621–659.

[Rabiner, 1989] Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceeding of the IEEE*, 77(2):257–286.

[Ravanelli and Omologo, 2014] Ravanelli, M. and Omologo, M. (2014). On the selection of the impulse responses for distant-speech recognition based on contaminated speech training. In *Proc. interspeech.*

[Renals et al., 1994] Renals, S., Morgan, N., Bourlard, H., Cohen, M., and Franco, H. (1994). Connectionist probability estimators in HMM speech recognition. *IEEE Transactions on Speech and Audio Processing*, 2(1):161–174.

[Renals and Swietojanski, 2014] Renals, S. and Swietojanski, P. (2014). Neural networks for distant speech recognition. In *Proc. HSCMA*, pages 172–176.

[Rice, 1944] Rice, S. (1944). Mathematical analysis of random noise. *Bell System Technical Journal*, 23:282–332.

[Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408.

[Rumelhart et al., 1986] Rumelhart, D., Hintont, G., and Williams, R. (1986). Learning representations by back-propagating errors. *Nature*, (323):533–536.

[Seide et al., 2011] Seide, F., Li, G., and Yu, D. (2011). Conversational speech transcription using context-dependent deep neural networks. In *Proc. Interspeech*, pages 437–440.

[Seltzer, 2014] Seltzer, M. L. (2014). Robustness is dead! Long live robustness! Keynote speech, REVERB Workshop.

[Seltzer et al., 2004] Seltzer, M. L., Raj, B., and Stern, R. M. (2004). Likelihood-maximizing beamforming for robust hands-free speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 12(5):489–498.

[Seltzer et al., 2013] Seltzer, M. L., Yu, D., and Wang, Y. (2013). An investigation of noise robustness of deep neural networks. In *Proc. ICASSP*, pages 7398–7402.

[Smaragdis, 2007] Smaragdis, P. (2007). Convolutive speech bases and their application to supervised speech separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(1):1–14.

[Srinivasan and Wang, 2007] Srinivasan, S. and Wang, D. (2007). Transforming binary uncertainties for robust speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(7):2130–2140.

[Tachioka et al., 2013a] Tachioka, Y., Watanabe, S., Le Roux, J., and Hershey, J. R. (2013a). Discriminative methods for noise robust speech recognition: A chime challenge benchmark. In *Proc. CHiME*.

[Tachioka et al., 2013b] Tachioka, Y., Watanabe, S., Le Roux, J., and Hershey, J. R. (2013b). Discriminative methods for noise robust speech recognition: A CHiME Challenge benchmark. In *Proc. 2nd Int. Workshop on Machine Listening in Multisource Environments (CHiME)*, pages 19–24.

[Tran et al., 2015a] Tran, D. T., Ono, N., and Vincent, E. (2015a). Fast DNN training based on auxiliary function technique. In *Proc. ICASSP*, pages 2160–2164.

[Tran et al., 2014a] Tran, D. T., Vincent, E., and Jouvet, D. (2014a). Extension of uncertainty propagation to dynamic MFCCs for noise-robust ASR. In *Proc. ICASSP*, pages 5507–5511.

[Tran et al., 2014b] Tran, D. T., Vincent, E., and Jouvet, D. (2014b). Fusion of multiple uncertainty estimators and propagators for noise-robust ASR. In *Proc. ICASSP*, pages 5512–5516.

[Tran et al., 2015b] Tran, D. T., Vincent, E., and Jouvet, D. (2015b). Discriminative uncertainty estimation for noise robust ASR. In *Proc. ICASSP*, pages 5038–5042.

[Tran et al., 2015c] Tran, D. T., Vincent, E., and Jouvet, D. (2015c). Nonparametric uncertainty estimation and propagation for noise robust ASR. *IEEE Transactions on Speech and Audio Processing*, 23(11):1835–894.

[Tran et al., 2013] Tran, D. T., Vincent, E., Jouvet, D., and Adiloğlu, K. (2013). Using full-rank spatial covariance models for noise-robust ASR. In *Proc. CHiME*, pages 31–32.

[Trees, 2002] Trees, H. L. V., editor (2002). *Optimum Array Processing.* Wiley-Interscience, New York.

[Tresp et al., 1994] Tresp, V., Ahmad, S., and Neuneier, R. (1994). Training neural networks with deficient data. *Proc. NIPS*, pages 128–135.

[Veselý et al., 2013] Veselý, K., Ghoshal, A., Burget, L., and Povey, D. (2013). Sequence-discriminative training of deep neural networks. In *Proc. Interspeech*, pages 2345–2349.

[Viikki and Laurila, 1998] Viikki, O. and Laurila, K. (1998). Cepstral domain segmental feature vector normalization for noise robust speech recognition. *Speech Communication*, 25(1–3):133–147.

[Vincent et al., 2013a] Vincent, E., Barker, J., Watanabe, S., Le Roux, J., Nesta, F., and Matassoni, M. (2013a). The second 'CHiME' speech separation and recognition challenge: An overview of challenge systems and outcomes. In *Proc. ASRU*, pages 162–167.

[Vincent et al., 2013b] Vincent, E., Barker, J., Watanabe, S., Le Roux, J., Nesta, F., and Matassoni, M. (2013b). The second 'CHiME' speech separation and recognition challenge: Datasets, tasks and baselines. In *Proc. ICASSP*, pages 126–130.

[Viterbi, 1967] Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269.

[Wang and Wang, 2013] Wang, Y. and Wang, D. L. (2013). Towards scaling up classification-based speech separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(7):1381–1390.

[Weng et al., 2014] Weng, C., Yu, D., Seltzer, M. L., and Droppo, J. (2014). Single-channel mixed speech recognition using deep neural networks. In *Proc. ICASSP*, pages 5632–5636.

[Weninger et al., 2014] Weninger, F., Le Roux, J., Hershey, J. R., and Schuller, B. (2014). Discriminatively trained recurrent neural networks for single-channel speech separation. In *Proc. GlobalSIP*, pages 577–581.

[Wilson et al., 2008] Wilson, K. W., Raj, B., Smaragdis, P., and Divakaran, A. (2008). Speech denoising using nonnegative matrix factorixation with priors. In *Proc. ICASSP*, pages 4029 – 4032.

[Wölfel and McDonough, 2009] Wölfel, M. and McDonough, J. (2009). *Distant Speech Recognition.* Wiley.

[Young et al., 2006] Young, S. J., Evermannand, G., Gales, M. J. F., Hain, T., Kershaw, D., Liu, X., More, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., and Woodland, P. (2006). *The HTK Book, version 3.4.* University of Cambridge.

[Zeiler et al., 2013] Zeiler, M. D., Ranzato, M., Monga, R., Mao, M., Yang, K., Le, Q. V., Nguyen, P., Senior, A., Vanhoucke, V., Dean, J., and Hinton, G. (2013). On rectified linear units for speech processing. In *Proc. ICASSP*, pages 3517–3521.

Abstract: This thesis focuses on noise robust automatic speech recognition (ASR). It includes two parts. First, we focus on better handling of uncertainty to improve the performance of ASR in a noisy environment. Second, we present a method to accelerate the training process of a neural network using an auxiliary function technique. In the first part, multichannel speech enhancement is applied to input noisy speech. The posterior distribution of the underlying clean speech is then estimated, as represented by its mean and its covariance matrix or uncertainty. We show how to propagate the diagonal uncertainty covariance matrix in the spectral domain through the feature computation stage to obtain the full uncertainty covariance matrix in the feature domain. Uncertainty decoding exploits this posterior distribution to dynamically modify the acoustic model parameters in the decoding rule. The uncertainty decoding rule simply consists of adding the uncertainty covariance matrix of the enhanced features to the variance of each Gaussian component. We then propose two uncertainty estimators based on fusion to nonparametric estimation, respectively. To build a new estimator, we consider a linear combination of existing uncertainty estimators or kernel functions. The combination weights are generatively estimated by minimizing some divergence with respect to the oracle uncertainty. The divergence measures used are weighted versions of Kullback-Leibler (KL), Itakura-Saito (IS), and Euclidean (EU) divergences. Due to the inherent nonnegativity of uncertainty, this estimation problem can be seen as an instance of weighted nonnegative matrix factorization (NMF). In addition, we propose two discriminative uncertainty estimators based on linear or nonlinear mapping of the generatively estimated uncertainty. This mapping is trained so as to maximize the boosted maximum mutual information (bMMI) criterion. We compute the derivative of this criterion using the chain rule and optimize it using stochastic gradient descent. In the second part, we introduce a new learning rule for neural networks that is based on an auxiliary function technique without parameter tuning. Instead of minimizing the objective function, this technique consists of minimizing a quadratic auxiliary function which is recursively introduced layer by layer and which has a closed-form optimum. Based on the properties of this auxiliary function, the monotonic decrease of the new learning rule is guaranteed.
Keywords: automatic speech recognition, noise robustness, speech enhancement, uncertainty propagation.

Résumé: Cette thèse se focalise sur la reconnaissance automatique de la parole (RAP) robuste au bruit. Elle comporte deux parties. Premièrement, nous nous focalisons sur une meilleure prise en compte des incertitudes pour améliorer la performance de RAP en environnement bruité. Deuxièmement, nous présentons une méthode pour accélérer l'apprentissage d'un réseau de neurones en utilisant une fonction auxiliaire. Dans la première partie, une technique de rehaussement multicanal est appliquée à la parole bruitée en entrée. La distribution a posteriori de la parole propre sous-jacente est alors estimée et représentée par sa moyenne et sa matrice de covariance, ou incertitude. Nous montrons comment propager la matrice de covariance diagonale de l'incertitude dans le domaine spectral à travers le calcul des descripteurs pour obtenir la matrice de covariance pleine de l'incertitude sur les descripteurs. Le décodage incertain exploite cette distribution a posteriori pour modifier dynamiquement les paramètres du modèle acoustique au décodage. La règle de décodage consiste simplement à ajouter la matrice de covariance de l'incertitude à la variance de chaque gaussienne. Nous proposons ensuite deux estimateurs d'incertitude basés respectivement sur la fusion et sur l'estimation non-paramétrique. Pour construire un nouvel estimateur, nous considérons la combinaison linéaire d'estimateurs existants ou de fonctions noyaux. Les poids de combinaison sont estimés de façon générative en minimisant une mesure de divergence par rapport à l'incertitude oracle. Les mesures de divergence utilisées sont des versions pondérées des divergences de Kullback-Leibler (KL), d'Itakura-Saito (IS) ou euclidienne (EU). En raison de la positivité inhérente de l'incertitude, ce problème d'estimation peut être vu comme une instance de factorisation matricielle positive (NMF) pondérée. De plus, nous proposons deux estimateurs d'incertitude discriminants basés sur une transformation linéaire ou non-linéaire de l'incertitude estimée de façon générative. Cette transformation est entraînée de sorte à maximiser le critère de maximum d'information mutuelle boosté (bMMI). Nous calculons la dérivée de ce critère en utilisant la règle de dérivation en chaîne et nous l'optimisons par descente de gradient stochastique. Dans la seconde partie, nous introduisons une nouvelle méthode d'apprentissage pour les réseaux de neurones basée sur une fonction auxiliaire sans aucun réglage de paramètre. Au lieu de maximiser la fonction objectif, cette technique consiste à maximiser une fonction auxiliaire qui est introduite de façon récursive couche par couche et dont le minimum a une expression analytique. Grâce aux propriétés de cette fonction, la décroissance monotone de la fonction objectif est garantie.
Mots-clés: reconnaissance automatique de la parole, robustesse au bruit, rehaussement de la parole, propagation de l'incertitude.