UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Structuration du modèle acoustique pour améliorer les performances de la reconnaissance automatique de la parole

(Acoustic Model Structuring for Improving Automatic Speech Recognition Performance)

THÈSE

présentée et soutenue publiquement le 26 Novembre 2014

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Arseniy Gorin

Composition du jury

Rapporteurs :	Tanja Schultz	Prof., Karlsruhe Institute of Technology, CSL
	Yannick Estève	Prof., Université du Maine, LIUM
Examinateurs :	Jean-Luc Gauvain	DR CNRS, Paris Orsay, LIMSI
	Régine André-Obrecht	Prof., Université Paul Sabatier, IRIT
	Guy Perrier	Prof., Université de Lorraine, INRIA, LORIA
Directeur de thèse :	Denis Jouvet	DR INRIA, Nancy, LORIA

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

Mis en page avec la classe thesul.

Résumé

Cette thèse se concentre sur la structuration du modèle acoustique pour l'amélioration de la reconnaissance de la parole par modèle de Markov. La structuration repose sur l'utilisation d'une classification non supervisée des phrases du corpus d'apprentissage pour tenir compte des variabilités dues aux locuteurs et aux canaux de transmission. Du fait de ces variabilités, une unité phonétique dans un contexte donné (c'est-à-dire un triphone) est associée à des paramètres acoustiques dépendants entre autres de l'âge, du sexe et de l'accent du locuteur ainsi que des conditions d'enregistrement et de transmission. Quand les informations associées au locuteur ne sont pas disponibles (ou si on veut aller au delà de la traditionnelle classification homme/femme), il faut mettre en œuvre une classification non supervisée. L'idée est de regrouper automatiquement les phrases prononcées en classes correspondant à des données acoustiquement similaires. Pour la modélisation multiple, un modèle acoustique indépendant du locuteur est adapté pour chaque classe en utilisant les données correspondantes. Pour transcrire un segment de parole, la classe est d'abord estimée, puis le décodage est effectué avec le modèle acoustique correspondant à la classe estimée.

Un grand nombre de classes est souhaitable pour bien représenter les différentes sources de variabilité. Mais, quand le nombre de classes augmente, la quantité de données disponibles pour l'apprentissage du modèle de chaque classe diminue, et cela peut rendre la modélisation moins fiable. Une façon de pallier ce problème est de modifier le critère de classification appliqué sur les données d'apprentissage pour permettre à une phrase d'être associée à plusieurs classes. Ceci peut être obtenu par l'application d'un seuil (marge de tolérance) par rapport à la meilleure distance. Dans la première partie de la thèse, la marge de tolérance est étudiée pour un classifieur GMM (*Gaussian Mixture Model* – mélange de Gaussiennes) indépendant des phonèmes et avec le critère du maximum de vraisemblance. Ensuite, l'approche est appliquée sur un classifieur qui utilise un modèle GMM pour chaque phone et la mesure de divergence de Kullback-Liebler.

L'essentiel de la thèse est consacré à une nouvelle approche qui utilise la classification automatique des données d'apprentissage pour structurer le modèle acoustique : CS-GMM (*Class-Structured GMM*). Ainsi, au lieu d'adapter tous les paramètres du modèle HMM-GMM pour chaque classe de données, les informations de classe sont explicitement introduites dans la structure des GMM en associant chaque composante des densités multi-gaussiennes (ou un sous-ensemble de composantes) avec une classe. Le modèle obtenu a le même nombre de paramètres que le modèle HMM-GMM conventionnel, mais les composantes des densités sont structurées en relation avec les classes des données (par exemple, classes de locuteurs). Pour exploiter efficacement cette structuration des composantes, deux types de modélisations sont proposées. Dans la première approche on propose de compléter cette structuration des densités par des pondérations des composantes gaussiennes dépendantes des classes de locuteurs (*Class-Structured with Class-Dependent mixture Weights* – CS-CDW-GMM). Pour cette modélisation, les composantes gaussiennes des mélanges GMM sont structurées en fonction des classes et partagées entre toutes les classes, tandis que les pondérations des composantes des densités sont dépendantes de la classe. Lors du décodage, le jeu de pondérations des gaussiennes est sélectionné en fonction de la classe estimée. Dans une deuxième approche, les pondérations des gaussiennes sont remplacées par des matrices de transition entre les composantes gaussiennes des densités (comme dans les *Stranded GMMs* – StGMM). Alors que les StGMM étaient originellement initialisés à partir de HMM-GMM conventionnels, le modèle CS-StGMM (*Class-Structured StGMM*) proposé est initialisé à partir d'un modèle structuré.

Les approches proposées dans cette thèse sont analysées et évaluées sur différents corpus de parole qui couvrent différents types de parole (chiffres connectés et données grand vocabulaire d'émissions radio) et différentes sources de variabilité (âge, sexe, accent et bruit). L'utilisation

d'une marge de tolérance permet d'estimer un grand nombre de modèles de classes même avec une quantité limitée des données d'apprentissage. La nouvelle approche basée sur la structuration des composantes gaussiennes des densités utilise beaucoup moins de paramètres. Le premier modèle structuré (CS-CDW-GMM) exploite le décodage habituel de Viterbi, et conduit à des performances similaires à celles des modèles de classes pour la reconnaissance grand vocabulaire, et à de meilleures performances pour la reconnaissance de chiffres connectés prononcés par des enfants et des adultes. Le second modèle structuré (CS-StGMM) utilise un algorithme de décodage plus complexe, mais n'exige pas une classification préalable, et il conduit à de meilleures performances que l'approche CS-CDW-GMM.

Mots-clés: Reconnaissance de la parole , classification non supervisée , modèles de classes de locuteurs , modèles stochastiques de trajectoire , variabilité de locuteur

Abstract

This thesis focuses on acoustic model structuring for improving HMM-based automatic speech recognition. The structuring relies on unsupervised clustering of speech segments of the training data in order to handle speaker and channel variability. Speech variability leads to the fact that a given phonetic unit in a given context (i.e., a context-dependent triphone) is associated to different acoustic features depending mostly on speaker age, gender and accent, as well as on recording conditions.

When speaker-associated information is not available (or if one wants to go beyond the traditional gender-dependent classes), unsupervised clustering techniques have to be applied. The idea is to split the data into acoustically similar classes. In conventional multi-modeling (or class-based) approach, separate class-dependent models are built via adaptation of a speaker-independent model. To transcribe a speech segment, the class is first estimated, and then the corresponding model is selected for decoding.

On the one hand, a large number of classes is desired to represent different sources of variability. On the other hand, when the number of classes increases, less data becomes available for the estimation of the class-based models, and the parameters are less reliable. One way to handle such problem is to modify the classification criterion applied on the training data, allowing a given segment to belong to more than one class. This is obtained by relaxing the classification decision through a soft margin. In the first part of the thesis, soft margin is first investigated for a phone-independent GMM-based classifier and with a maximum likelihood criterion. Then, the soft margin approach is applied on another clustering algorithm, which uses phone-dependent GMMs (or phonetic HMM-GMM) and the Kullback-Liebler divergence measure.

In the main part of the thesis, a novel approach is proposed that uses the clustered data more efficiently in a *Class-Structured GMM* (CS-GMM). Instead of adapting all HMM-GMM parameters separately for each class of data, the class information is explicitly introduced into the GMM structure by associating a given density component (or a subset of components) with a given class. Such model has the same number of parameters as the conventional HMM-GMM, but its density components are structured and associated with global speaker classes. To efficiently exploit such structured HMM-GMM, two different approaches are proposed.

The first approach combines *Class-Structured GMM with Class-Dependent mixture Weights* (CS-CDW-GMM). In this model the Gaussian components are class independent (i.e., they are shared across speaker classes), but they are class-structured, and the mixture weights are class-dependent. In decoding, the set of mixture weights is selected according to the estimated segment class. In the second approach, the mixture weights are replaced by density component transition probabilities (as in Stranded GMM: StGMM). While it was originally proposed to initialize StGMM from conventional HMM-GMM, the proposed *Class-Structured Stranded GMM* (CS-StGMM) is initialized from a class-structured model.

The approaches proposed in this thesis are analyzed and evaluated on various speech data, which cover different types of speech (connected digits and large vocabulary radio broadcast data) and different variability sources (age, gender, accent and noise). Soft classification margin allows to reliably estimate a large number of class-based models even with a relatively limited amount of training data. The new approaches based on class-structuring of the Gaussian components requires less parameters to store and estimate. For decoding, the first structured model (CS-CDW-GMM) relies on conventional and efficient Viterbi search. It performs similar to class-based modeling for large vocabulary speech recognition and better for connected digits produced by speakers of different age and gender. The second structured model (CS-StGMM) uses a more

complex search algorithm for decoding, but it does not require an additional classification pass and it significantly outperforms CS-CDW-GMM.

Keywords: Speech recognition, unsupervised clustering, speaker class modeling, stochastic trajectory modeling, speaker variability

Table of contents

List of figures

List of tables

Synthèse en Français

1 Introduction

Les modèles acoustiques sont l'un des constituants fondamentaux des systèmes de reconnaissance de la parole. Ils modélisent la réalisation acoustique des sons de la langue, et doivent tenir compte des multiples sources de variabilité qui viennent affecter le signal de parole et qui impactent sur les performances de la reconnaissance automatique de la parole (RAP). Une grande partie de la variabilité est due au sexe du locuteur, à son âge et à son accent.

La variabilité résultante des paramètres acoustiques doit être prise en compte par les modèles acoustiques indépendants du locuteur ; or les modèles de Markov cachés avec densités multigaussiennes (HMM-GMM : *Hidden Markov Model with Gaussian Mixture Model observation densities*) ne sont pas capables de représenter précisément des distributions très hétérogènes de paramètres acoustiques. Les limitations des HMM-GMM sont expliquées par les hypothèses d'indépendance conditionnelle assez fortes. Lors du décodage, il n'y a aucune garantie de cohérence de trajectoire (i.e. le chemin optimal peut être associé à des composantes correspondant à des locuteurs ou à des conditions très différentes d'un état à un autre). Deux approches pour améliorer la robustesse des modèles HMM-GMM à la variabilité de signal, sont proposées et étudiées dans cette thèse.

2 Reconnaissance de la parole par HMM

Un système de reconnaissance automatique de la parole utilise trois niveaux d'information pour décoder un message parlé (cf. figure 1). Au niveau des mots on utilise le modèle de langage, qui représente les successions possibles des mots. Cette modélisation est généralement construite à partir de l'analyse de séquences de mots provenant d'un grand corpus textuel.

Le lexique spécifie le vocabulaire et associe à chaque mot une ou plusieurs séquences de phones correspondant à la ou aux prononciations possibles des mots.

Le troisième niveau correspond à la modélisation acoustique, qui traduit la réalisation acoustique de chaque element modélise (phones, silence, bruits, etc).

La modélisation acoustique est au centre de cette thèse. Elle repose sur la paramétrisation du signal, qui consiste souvent à calculer de coefficients cepstraux selon une échelle Mel (MFCC : *Mel Frequency Cepstral Coefficients*). La réalisation statistique des paramètres acoustiques de chaque phone est representée par un modèle de Markov Caché (HMM : *Hidden Markov Model*). Chaque phone est typiquement représenté par 3 états et une densité multigaussienne (GMM : *Gaussian Mixture Model*) est associée à chaque état. Les densités GMM avec un grand nombre de composantes visent à tenir compte des multiples sources de la variabilité qui viennent affecter les signaux de parole (sexe et âge du locuteur, accent, bruits).

Le problème fondamental de la RAP peut être formalisé de la manière suivante. Sachant que

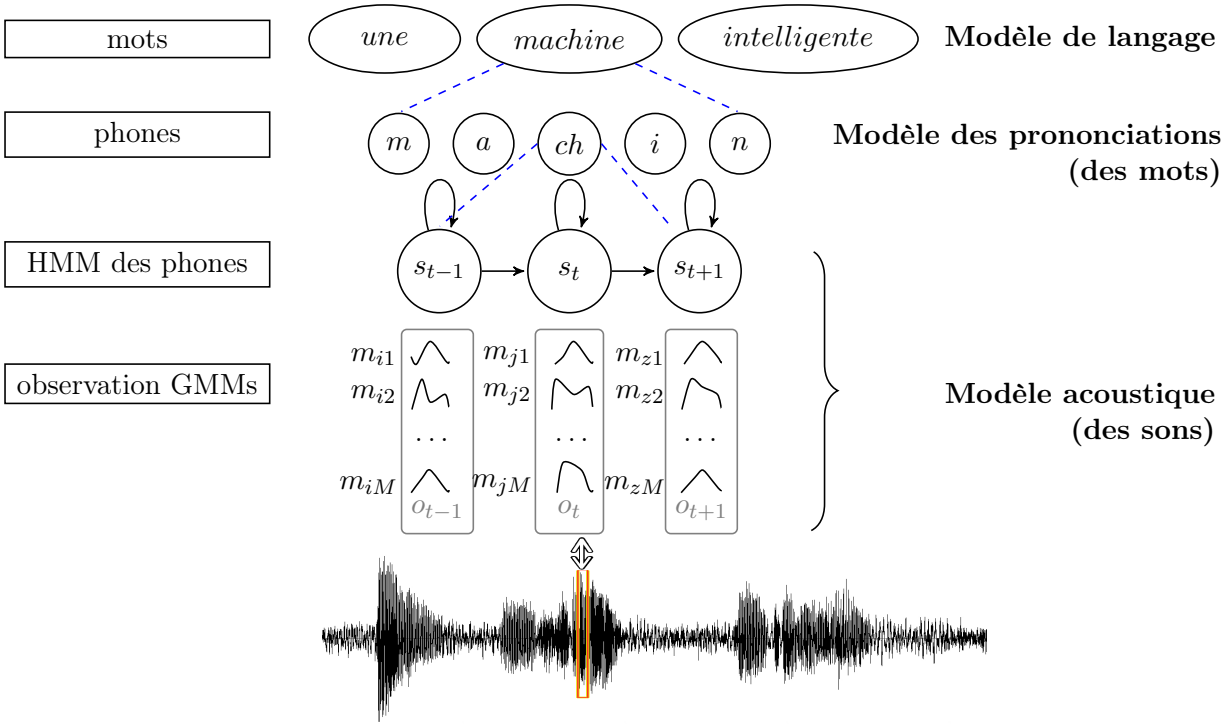

FIGURE 1 – Trois niveaux de l'information d'un message parlé

nous observons une séquence des paramètres acoustiques $\boldsymbol{\mathcal{O}} = (\mathbf{o}_1, \ldots, \mathbf{o}_t, \ldots, \mathbf{o}_T)$, qui correspond à une suite de mots $\boldsymbol{\mathcal{W}}$ d'un langage $\boldsymbol{\mathcal{L}}$, on cherche à déterminer la séquence $\hat{\boldsymbol{\mathcal{W}}}$ la plus probable:

$$\hat{\boldsymbol{\mathcal{W}}} = \arg\max_{\boldsymbol{\mathcal{W}} \in \boldsymbol{\mathcal{L}}} P(\boldsymbol{\mathcal{W}}|\boldsymbol{\mathcal{O}}) = \arg\max_{\boldsymbol{\mathcal{W}} \in \boldsymbol{\mathcal{L}}} \frac{P(\boldsymbol{\mathcal{O}}|\boldsymbol{\mathcal{W}})P(\boldsymbol{\mathcal{W}})}{P(\boldsymbol{\mathcal{O}})} = \arg\max_{\boldsymbol{\mathcal{W}} \in \boldsymbol{\mathcal{L}}} P(\boldsymbol{\mathcal{O}}|\boldsymbol{\mathcal{W}})P(\boldsymbol{\mathcal{W}}) \tag{1}$$

Ici, $P(\boldsymbol{\mathcal{O}}|\boldsymbol{\mathcal{W}})$ correspond à la vraisemblance acoustique et $P(\boldsymbol{\mathcal{W}})$ est la probabilité de la suite de mots $P(\boldsymbol{\mathcal{W}})$. $P(\boldsymbol{\mathcal{W}})$ est fourni par la modèle de langage, tandis que $P(\boldsymbol{\mathcal{O}}|\boldsymbol{\mathcal{W}})$ est calculé grace aux modèles acoustiques (HMM) et au lexique des prononciations.

3 Trajectoires et structuration des modèles acoustiques

Les modèles de Markov cachés avec densités multigaussiennes (HMM-GMM) ne sont pas capables de représenter précisément des distributions très hétérogènes de paramètres acoustiques. Par exemple, le traitement de la variabilité interlocuteur reste un problème pour les systèmes de RAP. Une limitation des HMM résulte du fait que, lors du décodage, il n'y a aucune garantie de cohérence de trajectoire (i.e. le chemin optimal peut-être associé à des composantes de GMM correspondant à des locuteurs ou à des conditions très différentes d'un état à un autre).

Une façon traditionnelle de pallier ce problème consiste à utiliser des modèles acoustiques adaptés à la voix de groupes homogènes de locuteurs (sexe, âge, etc). Cette thèse traite de la situation générale où les données proviennent de locuteurs d'âge et de sexe différents, et dont la classe d'appartenance est inconnue pour le système de reconnaissance. Une classification automatique peut alors être appliquée sur chaque phrase, en supposant que le locuteur ne change pas au cours de la phrase.

Deux approches différentes sont étudiées dans cette thèse. La première approche assez classique consiste en une classification non supervisée et une adaptation des modèles acoustiques pour chaque classe. Malheureusement, la quantité de données disponibles pour l'apprentissage du modèle de chaque classe diminue lorsque le nombre de classes augmente, et cela peut rendre la modélisation moins fiable. La contribution d'une première partie de cette thèse est d'introduire une marge de tolérance pour autoriser une phrase à être associée à plusieurs classes. Nous étudions cette approche pour deux techniques de classification et nous proposons aussi différentes techniques d'optimisation de la marge de tolérance, ainsi que la combinaison de systèmes.

La deuxième partie de cette thèse propose une nouvelle approche de modélisation acoustique, qui utilise la classification des données d'apprentissage pour structurer les composantes gaussiennes des densités GMM. Pour cela on construit les modèles de manière à ce que la $k^{ème}$ composante de chacune des densités soit associée à une même classe de données.

Pour exploiter efficacement cette structuration des composantes, deux types de modélisations sont proposés. Dans la première, dénommée CS-CDW-GMM (*Class-Structured with Class-Dependent mixture Weights*), les composantes gaussiennes des mélanges GMM sont structurées en fonction des classes et sont partagées entre toutes les classes, tandis que les pondérations des composantes des densités sont dépendantes de la classe.

Dans une deuxième approche, nous proposons de combiner la structuration des composantes des densités GMM en fonction des classes avec l'utilisation de matrices de transition entre composantes (MTM : *Mixture Transition Matrices*) comme dans les StGMM (*Stranded GMM*). Dans les StGMM, les matrices de transition MTM définissent les dépendances entre les composantes gaussiennes des densités GMM adjacentes. Tandis que les StGMM étaient originellement initialisés à partir de HMM-GMM conventionnels, la modélisation CS-StGMM (*Class-Structured StGMM*) proposée dans cette thèse combine StGMM et structuration des composantes en classes. Les matrices MTM modélisent alors la probabilité de rester sur des composantes associées à la même classe au cours du temps, ou de passer vers une composante d'une autre classe.

La suite de cette synthèse est organisée de la manière suivante. La section 4 présente la méthode conventionnelle de classification non supervisée pour la modélisation multiple. La section 5 présente l'utilisation d'une marge de tolérance pour la classification automatique, et propose des techniques d'ajustement de ce paramètre, ainsi que la combinaison d'hypothèses de décodage. La section 6 introduit l'approche générale pour la structuration en classes des composantes des densités. La section 7 détaille la structuration en classes avec des pondérations des composantes gaussiennes dépendantes des classes. La section 8.1 rappelle le principe des *S*tranded GMMs conventionels et la section 8.2 détaille la modélisation CS-StGMM proposée qui combine la structuration en classes des composantes et l'utilisation de matrices de transition entre composantes. La section 9 présente les expériences avec les modèles proposés et la section 10 formule les conclusions.

4 Classification non supervisée et modélisation multiple

L'objectif d'une classification non supervisée est de regrouper automatiquement les données d'apprentissage en classes correspondant à des données acoustiquement similaires.

4.1 Les approches

La première approche de classification automatique, utilisée dans cette thèse, repose sur des modèles multigaussiens [Jouvet *et al.*, 2012b]. Dans cette approche, un classificateur GMM générique est utilisé avec le critère de classification basé sur la vraisemblance maximale. La

deuxième approche repose sur des modèles multigaussiens associés aux phones et la divergence de Kullback-Leibler est utilisée pour la classification [Mao *et al.*, 2005].

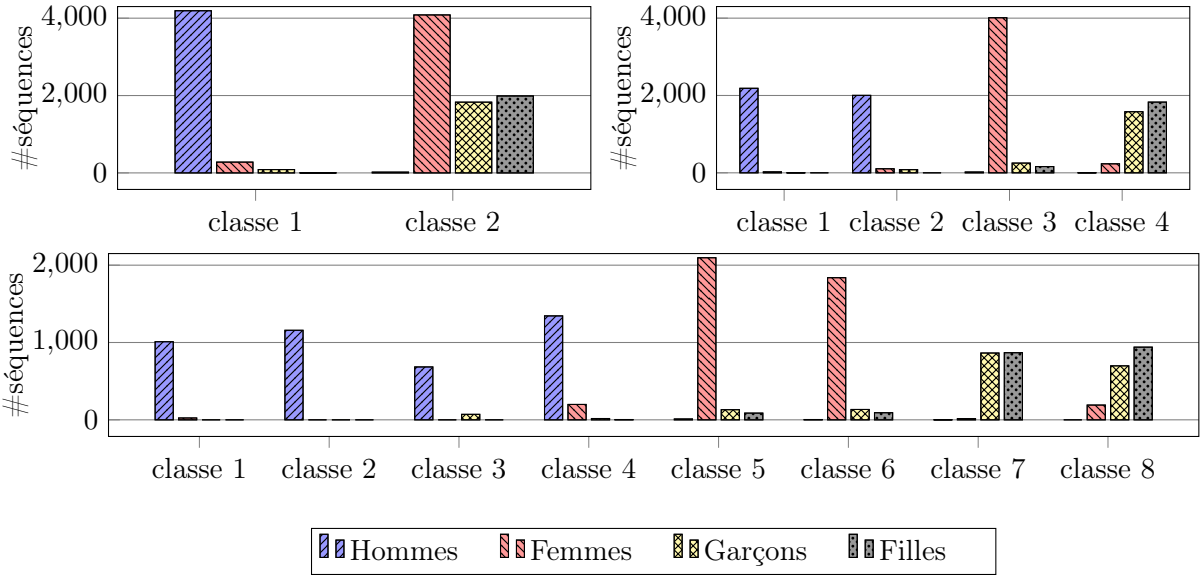

FIGURE 2 – Nombre de séquences par catégorie de locuteurs en fonction du nombre de classes

La figure 2 présente une analyse de la classification GMM des locuteurs du corpus TIDIGITS. Les deux premières classes séparent les données entre hommes d'un coté, et femmes et enfants de l'autre. Avec 4 classes, on voit apparaître une séparation entre les voix de femmes et les voix d'enfants. Par contre, il semble impossible de séparer les voix de garçons des voix de filles, même en augmentant le nombre de classes.

4.2 Evaluation de la modélisation multiple pour la RAP grand vocabulaire

Pour la modélisation multiple, les ensembles de classes des données d'apprentissage (jusqu'a 32 classes) ont été construits en utilisant le classification reposant sur la divergence KL. Cette approche est légèrement meilleure que la classification par maximum de vraisemblance. L'adaptation des modèles de classes repose sur les techniques MLLR, MAP ou la combinaison MLLR+MAP.

Les données d'apprentissage du corpus ESTER2, environ 190 heures, ont servi pour l'estimation des GMMs de classification, ainsi que pour l'estimation des modèles acoustiques des phones associés à chaque classe. Les évaluations ont été menées sur les données françaises du corpus de développement, et correspondent à environ 4h30 de signal audio et 36800 mots. Les modèles acoustiques des phones sont composés de 4500 sénones (états/densités partagés) et chaque densité a 64 composantes gaussiennes.

Les résultats indiqués sur la figure 3 montrent que c'est l'adaptation MLLR+MAP qui donne toujours les meilleures performances. Malheureusement, à partir de huit classes, les performances sont soit les mêmes (données de développement), soit se dégradent (données de test).

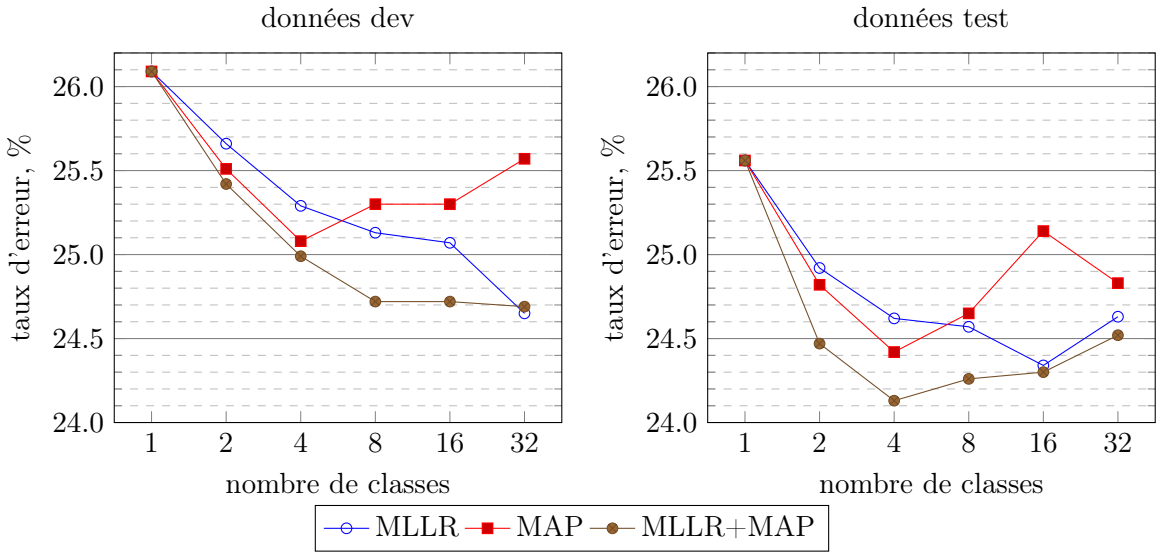

FIGURE 3 – Taux d'erreur (%) en fonction du nombre de classes et de la méthode d'adaptation

5 Contribution en modélisation multiple

Les contributions principales de cette thèse en modélisation multiple sont les suivantes :

— introduction d'une marge de tolérance dans la classification automatique pour améliorer l'apprentissage des modèles acoustiques avec un nombre important de classes;
— proposition de différentes techniques d'optimisation de la marge de tolérance;
— évaluation de la combinaison d'hypothèses (ROVER) pour exploiter au mieux la modélisation multiple.

5.1 Marge de tolérance de classification

L'idée sous-jacente consiste à exploiter de manière optimale les données qui sont à la frontière des classes. En effet les données à la frontière de deux classes peuvent être affectées à l'une ou l'autre des classes, voire aux deux classes. Cela revient à considérer qu'il y a une incertitude sur la frontière. L'introduction d'une marge de tolérance δ permet de gérer une telle incertitude, et d'affecter à plusieurs classes c_k les enregistrements (segments de parole) u qui se trouvent à la frontière des classes :

$$u \in c_k \Leftrightarrow D_{Tot}(p^u||p^{c_k}) \leq \min_{l \in \{1,\ldots,R\}} D_{Tot}(p^u||p^{c_l}) + \delta \qquad (2)$$

ou $D_{Tot}(p^u||p^{c_k})$ est une mesure de divergence entre les distributions des paramètres sur l'enregistrement u et la distribution pour la classe c_k. Lorsque l'on augmente la marge de tolérance δ, de plus en plus de données se trouvent affectées à plusieurs classes, ce qui augmente, en moyenne, la quantité des données associées à chaque classe.

5.2 Optimisation des paramètres

Plusieurs approches ont été étudiées afin d'optimiser la marge de tolérance. Les résultats des évaluations sont présentés sur la figure 4.

1. *"globalOpt"*: pour l'ensemble des classes, une seule valeur de marge de tolérance est choisie, celle qui donne le meilleur résultat (taux d'erreur) sur les données de développement. Le même paramètre est alors utilisé pour le décodage des données de test.
2. *"classOpt"*: pour chaque classe, la marge de tolérance δ choisie est celle qui minimise l'erreur sur les données de développement associées à cette classe.
3. *"classOpt>300"*: si la quantité de données de développement pour certaines classes est suffisante ($>$ 300 secondes), on estime la marge δ qui donne la meilleure performance pour les données de développement associées à cette classe. Sinon, on utilise pour cette classe la marge globalement optimale.

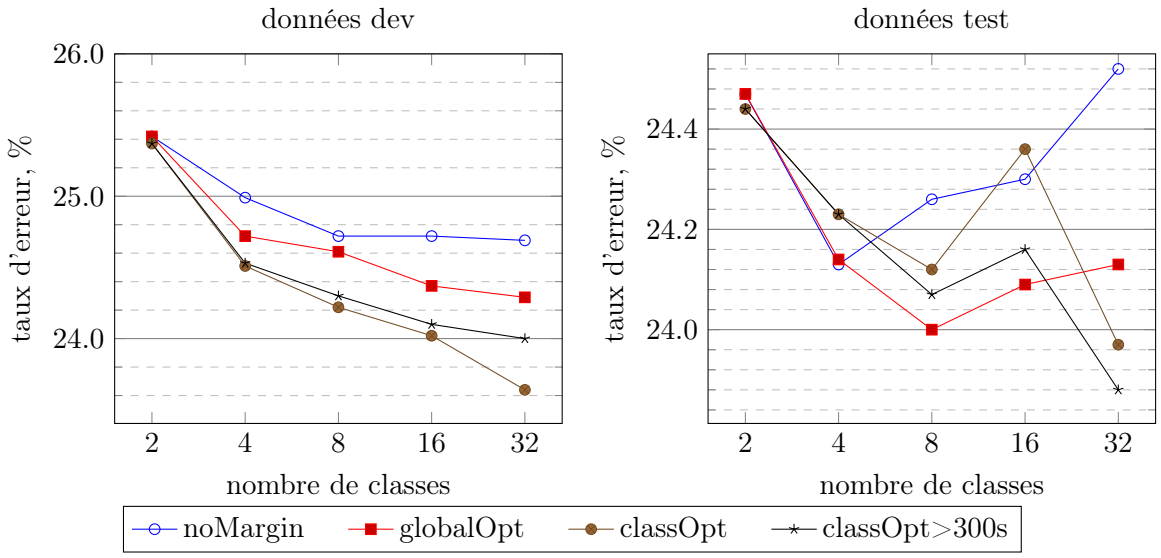

FIGURE 4 – Taux d'erreur (%) en fonction du nombre de classes et de la méthode d'optimisation de la marge

En conclusion, les résultats montrent qu'en introduisant une marge de tolérance raisonnable lors de la classification des données du corpus d'apprentissage, on peut utiliser de manière efficace un nombre important de classes de données, et obtenir des taux d'erreurs significativement meilleurs qu'avec le système de référence. De plus, si les données de développement sont suffisantes, on peut efficacement optimiser la valeur de la marge de tolérance par l'analyse des erreurs de décodage sur les données associées à chaque classe (approche "classOpt>300s").

5.3 Combinaison des modèles de classes

La combinaison des hypothèses de décodage (ROVER) est une technique connue et très efficace. Plusieurs systèmes utilisent ROVER pour combiner les hypothèses fournies par des systèmes qui exploitent différents paramètres acoustiques ou différents modèles de langage. Dans cette thèse il est proposé de combiner les hypothèses calculées avec les modèles correspondant aux N classes les plus "proches" des données à reconnaitre. Les meilleurs résultats sont obtenus avec 32 classes et une marge de tolérance de 0.1 (optimisée sur les données de développement).

Dans cette combinaison, pour chaque segment de parole, les modèles des 5 classes les plus "proches" sont utilisés pour le décodage, et les hypothèses obtenues sont combinées. Le tableau 1 montre les résultats des systèmes de bases indépendants du locuteur (SI : *Speaker-Independent*)

et dépendant du sexe du locuteur (GD : *Gender-Dependent*), et les meilleurs résultats des systèmes de modélisation multiple (classification sans marge), avec optimisation globale de la marge de tolérance (marge globale) ou avec optimisation dépendante de la classe (marge par classe). Les deux dernières lignes correspondent aux résultats des combinaisons ROVER sans et avec utilisation de la marge de tolérance; ils sont significativement meilleurs que tous les autres.

	modèle	détails			taux d'erreur	
		classes	marge	adaptation	dev	test
1	SI référence	LVCSR.4500s.StTel		-	26.09	25.56
2	GD référence	LVCSR.4500s.StTel.GD		MLLR+MAP	25.23	24.46
3	classification (sans marge)	32	none	MLLR	24.69	24.52
4	+ marge globale	32	0.20	MLLR+MAP	24.29	24.13
5	+ marge par classe	32	classOpt>300sec	MLLR+MAP	23.97	23.88
6	ROVER (sans marge)	32	0.00	MLLR+MAP	23.62	23.50
7	+ marge globale	32	0.10	MLLR+MAP	23.46	23.18

TABLE 1 – Comparaison des taux d'erreur des modélisations multiples

6 Structuration des composantes

Comme on l'a montré dans les sections précédentes, l'apprentissage des modèles de classes devient critique lorsque les données associées aux classes sont en quantité limitée. L'introduction d'une marge de tolérance est une solution à ce problème, mais il faut avoir un corpus développement assez grand pour optimiser les paramètres. De plus, les modèles de classes utilisent beaucoup de mémoire, puisqu'on doit estimer autant de modèles acoustiques qu'il y a des classes.

Ici on propose une nouvelle approche, qui repose sur la structuration des gaussiennes des mélanges GMM en fonction des classes et leur partage entre toutes les classes (CS-GMM : *Class-Structured GMM*). La structuration en classes des composantes des densités est faite, de telle manière que initialement, la $k^{ème}$ composante de chaque densité corresponde à une même classe de données (cf. figure 5).

La structuration des composantes gaussiennes est obtenue en initialisant les GMM par concaténation des composantes gaussiennes de GMM de plus faible dimensionalité appris sur les différentes classes. Par exemple, pour fabriquer un modèle avec M composantes gaussiennes associées à Z classes, Z modèles ayant chacun $L = M/Z$ composantes par densité sont appris. Puis, ces composantes sont regroupées dans un mélange global (cf. figure 6 pour les moyennes).

Cette structuration seule n'est pas efficace. En effet, même si les composantes gaussiennes sont associées à certaines classes des locuteurs, cette information n'est pas exploitée par un décodeur classique (i.e. toutes les composantes sont mélangées lors du calcul des probabilités de l'observation). Les sections suivantes proposent deux modifications de la modélisation acoustique, qui permettent d'exploiter efficacement les modèles structurés.

7 Structuration des composantes avec pondérations dépendantes des classes

Dans le premier modèle on propose de partager les composantes gaussiennes entre les différents modèles de classes, et de spécialiser les pondérations des gaussiennes à chaque classe

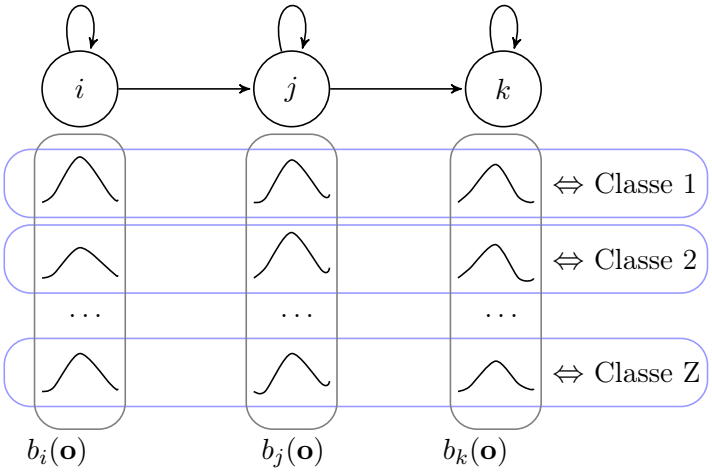

FIGURE 5 – HMM à 3 états et structuration des composantes gaussiennes

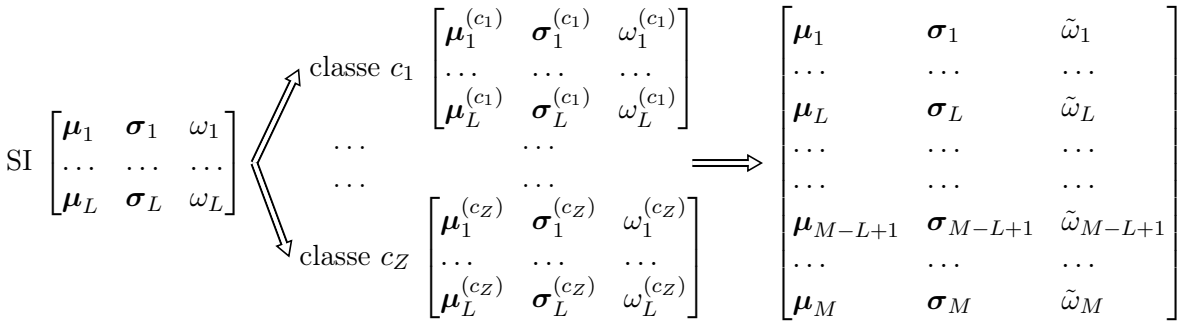

FIGURE 6 – Initialisation du modèle structuré (CS-GMM) à partir de plusieurs modèles associés aux classes

(cf. figure 7). Ainsi, dans un CS-CDW-GMM (*Class-Structured with Class-Dependent Weights GMM*), la densité b_j pour une classe de locuteurs c_i et un état j est définie par

$$b_j^{(c_i)}(\mathbf{o}_t) = \sum_{k=1}^{M} \omega_{jk}^{(c_i)} \mathcal{N}(\mathbf{o}_t | \boldsymbol{\mu}_{jk}, \mathbf{U}_{jk}) \tag{3}$$

où $\mathbf{o}_t$ est un vecteur d'observation pour la trame t, $\omega_{jk}^{(c_i)}$ est la pondération de la composante k pour la classe c_i, et $\mathcal{N}(\mathbf{o}_t|\boldsymbol{\mu}_{jk}, \mathbf{U}_{jk})$ est une densite gaussienne de moyenne $\boldsymbol{\mu}_{jk}$ et de covariance $\mathbf{U}_{jk}$.

Lors du décodage, chaque phrase à reconnaître est d'abord automatiquement classifiée, et assignée à une classe c. Ensuite, le décodage est effectué avec le jeu de pondérations des gaussiennes qui correspond à cette classe.

Lors de l'initialisation du modèle structuré CS-GMM, les pondérations des modèles des classes sont également concaténées (similaire au traitement des moyennes - cf. figure 6), puis renormalisées. Ensuite, les moyennes, les variances et les pondérations sont ré-estimées. Les pondérations, spécifiques à chaque classe, sont apprises à partir des données de la classe correspondante, tandis que toutes les données sont utilisées pour ré-estimer les moyennes et les variances qui sont

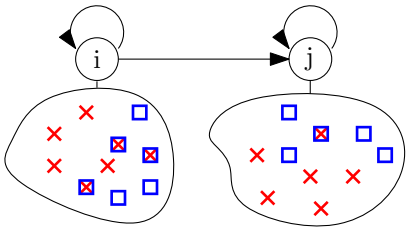

FIGURE 7 – CS-CDW-GMM avec un HMM à 2 états et 2 classes de locuteurs

partagées entre les classes :

$$\omega_{jk}^{(c_i)} = \frac{\sum_{t=1}^{T} \gamma_{jk}^{(c_i)}(t)}{\sum_{l=1}^{M} \sum_{t=1}^{T} \gamma_{jl}^{(c_i)}(t)} \qquad \mu_{jk} = \frac{\sum_{i=1}^{Z} \sum_{t=1}^{T} \gamma_{jk}^{(c_i)}(t) \mathbf{o}_t}{\sum_{i=1}^{Z} \sum_{t=1}^{T} \gamma_{jk}^{(c_i)}(t)} \tag{4}$$

où $\gamma_{jk}^{(c_i)}(t)$ est le compteur Baum-Welch de la $k^{ème}$ composante de la densité b_j, générant l'observation o_t de la classe c_i. Les sommes sur t portent sur toutes les trames des phrases d'apprentissage de la classe concernée. Les variances sont ré-estimées de manière similaire aux moyennes.

Après ré-estimation, les pondérations dépendantes de la classe sont plus grandes pour les composantes associées à la même classe de données ; la figure 8 montre l'exemple des pondérations des classes c_7, c_{17} et c_{27} , moyennés sur les toutes les densités.

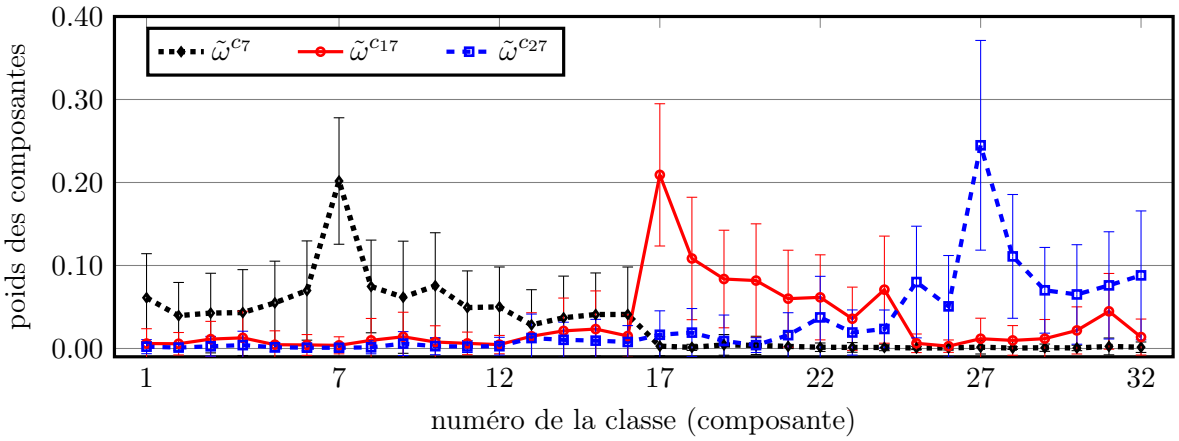

FIGURE 8 – Statistiques des pondérations des composantes pour les classes c_7, c_{17} et c_{27} après ré-estimation (moyennes et écarts-types calculés sur toutes les densités ; Z=32, M=32)

8 Stranded GMM avec structuration en classes des composantes

Les modèles StGMM (*Stranded GMM*), proposés pour la reconnaissance robuste [Zhao and Juang, 2012], reposent sur un enrichissement des densités d'émission des HMM-GMM par l'introduction de dépendances explicites entre les composantes des densités d'états adjacents. Tandis que dans la version proposée par Zhao, les StGMM sont initialisés à partir de HMM-GMM conventionnels, cette thèse propose d'exploiter la structuration des composantes en fonction de classes, pour obtenir des CS-StGMM (*Class-Structured Stranded GMM*).

8.1 Stranded GMM conventionnels

Les modèles StGMM conventionnels font intervenir la suite des états $\boldsymbol{\mathcal{Q}} = (q_1, ..., q_T)$, la suite des vecteurs d'observation $\boldsymbol{\mathcal{O}} = (\mathbf{o}_1, ..., \mathbf{o}_T)$, et la suite des composantes des densités $\boldsymbol{\mathcal{M}} = (m_1, ..., m_T)$, où $m_t \in \{1, ..., M\}$ désigne la composante gaussienne utilisée à l'instant t, et M précise le nombre de composantes par densité.

En comparaison des HMM-GMM conventionnels, les StGMM modélisent les dépendances entre la composante gaussienne m_t utilisée à l'instant t et la composante m_{t-1} utilisée à la trame précédente (cf. les lignes rouges sur la figure 9). La vraisemblance conjointe de la séquence d'observations, de la séquence d'états, et de la séquence de composantes est donnée par :

$$P(\boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\lambda}) = \prod_{t=1}^{T} P(\mathbf{o}_t|m_t, q_t) P(m_t|m_{t-1}, q_t, q_{t-1}) P(q_t|q_{t-1}) \tag{5}$$

où $P(q_t = j|q_{t-1} = i) = a_{ij}$ est la probabilité de transition entre états, $P(\mathbf{o}_t|m_t = l, q_t = j) = b_{jl}(\mathbf{o}_t)$ est la probabilité de l'observation $\mathbf{o}_t$ pour la composante gaussienne $m_t = l$ de la densité associée à l'état $q_t = j$ et $P(m_t = l|m_{t-1} = k, q_t = j, q_{t-1} = i) = c_{kl}^{(ij)}$ est la probabilité de transition entre les composantes gaussiennes. L'ensemble des probabilités de transition entre les composantes détermine les matrices de transition MTM (*Mixture Transition Matrices*) $C^{(ij)} = \{c_{kl}^{(ij)}\}$, avec les contraintes $\sum_{l=1}^{M} c_{kl}^{(ij)} = 1, \forall i, j, k$.

8.2 Stranded GMM avec structuration des composantes

La modélisation CS-StGMM (*Class-Structured Stranded GMM*) proposée combine la structuration en classes des composantes des densités avec les matrices de transition (MTM) des StGMMs (cf. figure 9).

Pour obtenir cette structuration, le CS-StGMM est initialisé à partir du CS-CDW-GMM décrit dans la section 7. Les moyennes et variances des gaussiennes sont obtenues directement à partir du CS-CDW-GMM et les matrices MTMs sont initialisées avec des distributions uniformes. Les pondérations des gaussiennes du CS-CDW-GMM, qui sont dépendantes des classes, ne sont pas utilisées dans le modèle CS-StGMM.

Quand les CS-CDW-GMM, qui servent à fabriquer les CS-StGMM, sont initialisés à partir de modèles de classes monogaussiens, chaque composante correspond à une classe. Après réestimation, les éléments diagonaux des matrices MTM dominent, ce qui conduit à favoriser la cohérence de la classe lors du décodage d'un segment de parole. Cependant, les éléments non-diagonaux non nuls rendent possibles les contributions d'autres composantes gaussiennes dans le calcul des scores acoustiques. L'avantage des CS-StGMM est qu'ils modélisent explicitement les trajectoires, tout en autorisant des changements de composantes (ou de classes). De plus, le décodage d'une phrase fonctionne en une seule passe; il n'y a pas de classification préalable à faire.

9 Expériences avec les modèles structurés en fonction des classes

Les principales expériences avec les GMMs structurées ont été menées sur le corpus TIDIGIT de chiffres connectés en anglais ; l'un des rares corpus disponibles à contenir à la fois des voix d'adultes et des voix d'enfants. L'ensemble d'apprentissage contient 41224 occurrences de chiffres

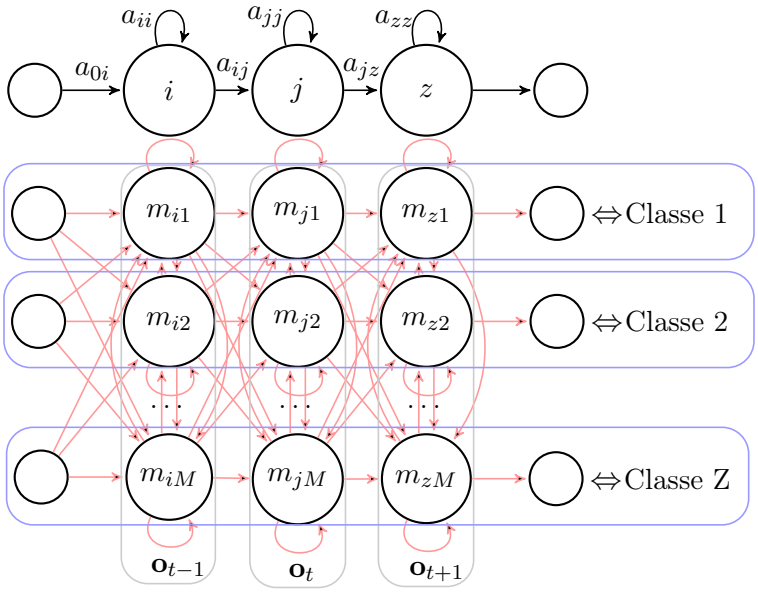

FIGURE 9 – Stranded GMM avec représentation des dépendances entre composantes gaussiennes (lignes rouges) et associations entre classes et composantes gaussiennes pour les CS-StGMM (lignes bleues)

(28329 prononcés par des adultes et 12895 par des enfants). L'ensemble de test contient 41087 occurrences de chiffres (28554 par des adultes et 12533 par des enfants).

Les résultats principaux sont résumés dans le tableau 2. Les systèmes analysés sont: référence (SI GMM), modélisation multiple avec l'adaptation MLLR+MAP (CA-GMM), GMM avec structuration des composantes et pondérations dépendantes des classes (CS-CDW-GMM), *S*tranded GMM conventionels (StGMM) et *S*tranded GMMs avec structuration des composantes (CS-StGMM).

Modèle	Classes	Decodage	Paramètres par état	**Adult**	**Enfant**
SI GMM	1	1 passe	78∗32+32=2528	**1.66**	**1.88**
CA-HMM	4	2 passes	4∗(78∗32+32)=10112	1.32	1.57
CS-CDW-GMM	32	2 passes	78∗32+32∗32=3520	0.80	1.05
StGMM	1	1 passe	78∗32+2∗32∗32=4544	1.11	1.27
CS-StGMM	32	1 passe	78∗32+2∗32∗32=4544	**0.52**	**0.86**

TABLE 2 – Taux d'erreur mot et indication du nombre de paramètres par état. La classe du locuteur est inconnue lors du décodage, et estimée par GMM pour les versions "2 passes"

L'approche CS-CDW-GMM permet une estimation robuste des moyennes et des variances partagées, tout en gardant une dépendance vis-à-vis des classes avec les pondérations des composantes. Avec un nombre limité de paramètres, le taux d'erreur est significativement réduit : 0,80% sur les données des adultes et 1,05% sur les données enfants. L'approche CS-StGMM proposée, qui repose sur la structuration en classes des composantes, améliore encore plus que les CS-CDW-GMM et StGMM conventionnels, et permet d'obtenir un taux d'erreur mot de 0,52% sur les données adultes, et de 0,86% sur les données enfants. A noter que la fabrication de CS-StGMM à partir de CS-CDW-GMM construits sur la base de différents nombres de classes (2,

4, 8 et 16) conduit aussi à des améliorations de performances par rapport au StGMM ; seul le résultat correspondant à 32 classes est indiqué.

Des expériences complémentaires ont été menées sur le corpus NEOLOGOS, qui est beaucoup plus grand (avec environ 1000 locuteurs adultes et 1000 locuteurs enfants) ; elles conduisent aux mêmes conclusions.

10 Conclusion

Dans cette thèse, plusieurs approches pour améliorer la RAP de données hétérogènes ont été étudiées. L'approche conventionnelle repose sur la classification non supervisée des données et l'adaptation des modèles de classes. Malheureusement, la taille limitée des corpus d'apprentissage ne permet pas de fabriquer un grand nombre de modèles de classes.

La première technique étudiée dans cette thèse pour pallier ce problème a consisté à modifier le critère de classification appliqué sur les données d'apprentissage pour permettre à une phrase d'être associée à plusieurs classes. L'approche permet d'améliorer significativement les performances des modèles de classes.

L'approche alternative est plus compacte; elle utilise les classes pour structurer les composantes gaussiennes des densités GMM en associant le $k^{ème}$ composante de chacune des densités à une même classe de locuteurs. On a proposé deux types de modélisations exploitant cette structuration des composantes: avec des pondérations des composantes gaussiennes spécifiques à chaque classe et avec des matrices de transition entre les composantes des densités GMM. Les deux approches sont efficaces, en particulier pour les corpus très limités ayant une grande variabilité acoustique entre locuteurs (par example corpus TIDIGITS).

Pour prolonger ces travaux, nous proposons d'exploiter la marge de tolérance avec d'autres techniques de classification non supervisée (par exemple, avec les i-vecteurs [Zhang *et al.*, 2011]). Une autre piste concerne la structuration des composantes pour d'autres formes de modèles acoustiques (par exemple, pour *Subspace GMMs* [Povey *et al.*, 2011a]). Enfin, l'adaptation des modèles CS-StGMMs est à étudier également.

1

Introduction

Research in *Automatic Speech Recognition* (ASR) has a long history, which includes contributions from many electrical engineers, phoneticians, linguists, computer and data scientists, mathematicians and other researchers. Over the last decades the performance of ASR systems has significantly improved and the recognition tasks have become larger and more realistic. This has also led to the emergence of various ASR-based commercial products, including automatic speech transcription systems, speech and dialog interfaces, etc.

After the first successful application of *Hidden Markov Models* (HMMs), statistical-based approach has become widely used in most of the ASR systems. Significant accuracy improvements of the modern HMM-based systems are achieved due to development of efficient training and adaptation techniques, better acoustic feature processing methods and increased amount of both transcribed speech for acoustic model training and textual data for language modeling. Recently, *Deep Neural Networks* (DNNs) demonstrated significant improvement compared to HMMs. Nevertheless, HMMs are still widely used as a core of many state-of-the-art ASR systems, as many reliable and efficient algorithms for training and parameter tuning are available. Independently on which models are used, state-of-the-art ASR systems are still far from reaching human performance, especially in handling large vocabulary continuous speech produced by a variety of speakers or speech recorded in noisy conditions.

1.1 Content and objectives

Although it is widely applied in speech recognition, the fundamental HMM model is frequently criticized for its inability to accurately model long temporal sequences and highly heterogeneous data. Various modifications have been proposed to achieve more accurate acoustic modeling. In this thesis, some of these modifications are reviewed and some novel approaches are proposed.

1.1.1 General ASR problems addressed in this thesis

The core of the statistical ASR systems is the acoustic model, which allows to distinguish one phonetic unit from another. It relies on the fact that different phonetic units are associated with different acoustic features, which are strongly related to the spectral characteristics of the corresponding signals. At the same time, speech signal contains a large amount of non-phonetic variability, mostly including speaker characteristics (age, gender and accent), recording conditions (noise, reverberation and microphone) and co-articulation effects. *Gaussian Mixture Model* (GMM) is widely used to represent the distribution of the acoustic features associated with an

HMM state. Although part of non-phonetic variability is handled by separately estimated Gaussian components, Speaker-Independent HMM-GMMs are still poor models of the heterogeneous speech signal. The same is true for DNN, although this model is not studied in this thesis.

One problem associated with HMM-GMM is the strong *conditional independence assumptions* of the model parameters. In particular, the observation density is assumed to be dependent only on the associated state. In other words, the density parameters are determined by local frame-level observations. Another problem is so-called *trajectory folding*. This phenomenon is associated with the fact that all GMM components are mixed together in the density estimation. For example, it is known that the acoustic features associated with the same phonetic unit produced by male and female speakers have large variability. At the same time, the components that were fitting mostly the male speaker data in training can be equally used for female speech decoding. Feature normalization (VTLN) can partially reduce this effect, but requires additional computation pass in decoding. Another problem generally associated with most of the statistical-based methods (including HMM) is the performance degradation observed when processing data that are significantly different from training data. A simple example here is the recognition of children speech by a system trained on adult speakers, which becomes dramatically inaccurate compared to the same system used for recognizing adult speech.

1.1.2 Class-based modeling

An efficient strategy for improving ASR robustness with respect to speech variability is to rely on several acoustic models, where each model represents a more homogeneous subset of speech data. A classic example is gender-dependent acoustic modeling, which uses separate models for male and female speakers. These models are typically built by adapting the parameters of a general *Speaker-Independent* (SI) model in order to avoid over-fitting of the resulting models.

Instead of using only two speaker variability classes (such as gender), a larger number of speaker (and possibly recording conditions) classes can be achieved by unsupervised clustering of the speech segments. Unsupervised clustering is also used when no speaker information is available. The main motivation is to improve and validate a general unsupervised framework that can be used for the data that is transcribed, but not annotated with respect to speakers and recording conditions (for example, crawled from the Internet). Throughout this thesis it is assumed that the speaker-related information is not available for training and test data. The objective of the unsupervised clustering-based (or *unsupervised class-based modeling*; later denoted also CB-HMM) approach is to split the set of segments of the training data into acoustically similar classes. In conventional decoding with class-based models, the class corresponding to each segment is estimated and the associated model is selected for decoding.

The class-based approach relies on many model parameters to be estimated and stored. As a result, for a relatively large number of classes the class-based models are not reliably estimated and the ASR performance degrades. To take advantage of a large number of class-based models without accuracy degradation, various methods are considered. Some systems rely on efficient parameter sharing and rapid adaptation techniques to reliably estimate model parameters with a limited amount of data. Others rely on interpolation of the parameters of several class-based models. Another efficient and widely-applied approach relies on combining the hypotheses obtained by decoding with several class-based models.

1.1.3 Trajectory modeling

Although the class-based techniques efficiently reduce the variability to be handled by each acoustic model, other approaches are considered in state-of-the-art ASR systems. Instead of (or together with) data pre-clustering, some of the HMM properties can be improved by introducing additional model dependencies and relaxing the conditional independence assumptions. For example, a broad class of *Segmental Models* (SMs) was developed. The general idea of SMs is to associate each HMM state not with a single observation, but with a sequence of observations (or a segment).

Various types of models were developed within this framework, differing from each other in how the segment is parameterized. Finally, none of these models became as popular, as the conventional HMM. The main reason was that SMs generally required more computations, but did not result in a substantial accuracy improvement. However, increased computational power, enlarged data and refined algorithms for training and adaptation can lead to future growth of popularity of various alternatives to conventional HMM-GMM model structure for ASR. For example, exploring the same idea of parameterizing long contextual information, Artificial Neural Networks (with deep and recurrent architectures) are getting more and more frequently used for replacing conventional HMM with Gaussian mixture density in many state-of-the-art ASR systems. However, studying ANN-based class of models is beyond the scope of this thesis.

1.2 Thesis contributions

This thesis contributes in the following research directions of acoustic modeling for ASR. First, a detailed study of relevant state-of-the-art is done. Then, multi-modeling (or class-based) approach is investigated in combination with soft classification margin, which allows to reliably estimate a large number of classes even with limited data. Finally, a large part of the thesis is devoted to a novel approach based on introducing speaker class information in conventional HMM by structuring the components of the observation density and relaxing some of the HMM conditional independence assumptions.

1.2.1 Soft classification margin for multi-modeling ASR

First, to achieve an efficient *multi-modeling* (or *class-based*) ASR with a large number of unsupervised speaker classes, *soft classification margin* is studied. The margin explicitly increases the amount of class-associated data for model training by allowing a segment to be associated with more than one class based on classification threshold. This approach was developed for a simple phone-independent GMM-based classification in [Jouvet and Vinuesa, 2012] and demonstrated promising performance improvements for radio broadcast speech transcription. In this thesis, a similar idea is applied for a more detailed classification based on phone-dependent GMMs and Kullback-Leibler measure. Moreover, different techniques for tuning the margin parameter on the development data are proposed and studied. Finally, hypothesis combination is applied in addition to soft clustering to further improve the accuracy of class-based ASR.

1.2.2 Class-Structured Gaussian Mixture Model

The second part of the thesis explores a different way to introduce speaker classes in the modeling. Namely, instead of building separate class-based models, a single HMM with *Class-Structured GMM* (CS-GMM) is used. The idea of CS-GMM is to associate each component of

the density with a speaker class. Component structuring alone does not make much sense, as the components are mixed in the density computation. In other words, the trajectory folding problem is not solved. However, the dependency of the component labels on the speaker classes can be exploited to build efficient model structures. In the thesis two such structures are proposed.

The first model based on class-structured architecture relies on *Class-Dependent Weights* (CS-CDW-GMM) with shared class-structured means and variances. Another proposed model structure replaces class-dependent weights by *Mixture Transition Matrices* (MTMs), which explicitly add the dependencies between Gaussian components of the densities associated with adjacent states. Initially, MTMs were used in *Stranded GMM* (StGMM) that was recently proposed as an extension of the HMM-GMM. StGMM relaxes the observation independence assumption and allows to achieve better ASR performance. By replacing mixture weights in CS-CDW-GMM by MTMs, the *Class-Structured Stranded GMM is achieved* (CS-StGMM).

1.2.3 Summary of published work

The novel approaches proposed and investigated in this thesis led to the following scientific publications. [Gorin and Jouvet, 2012] describes the application of soft margin with unsupervised speech data clustering based on phone-dependent GMMs and Kullback-Leibler measure (or KL-based clustering) and hypothesis combination for class-based modeling. [Gorin and Jouvet, 2013] formulates CS-CDW-GMM and describes the corresponding experiments on French radio broadcast data. [Gorin *et al.*, 2014] describes the detailed analysis of StGMM tested on data, which contain different types of non-phonetic variability. [Gorin and Jouvet, 2014b] proposes CS-StGMM and compares this approach with both class-based and CS-CDW-GMM on a small connected digits task. [Gorin and Jouvet, 2014a] completes the study with additional phonetic decoding experiments on a larger database of French telephone speech. Finally, [Gorin and Jouvet, 2014c] demonstrates combination of CS-StGMM approach with feature normalization (VTLN).

1.3 Structure of the thesis

Chapter 2 briefly summarizes the history of the ASR field and reviews the core concepts of statistical speech recognition, such as the general model structure, the main elements of the recognizer, the training and the decoding algorithms. Then, the problems of speech variability and trajectory folding are formulated. Finally, some of the concepts important for understanding the topics of the thesis are discussed, including model adaptation and parameter sharing techniques.

Chapter 3 covers the state-of-the-art in two specific research directions of acoustic modeling for ASR. First, an ensemble of techniques related to multi-modeling (or class-based modeling) is described, including various unsupervised segment clustering techniques and speaker-space models that are based on linear combination of class based models. Then, some alternative acoustic model structures for ASR are reviewed, including various segmental models, dynamic Bayesian models and ANN-based models.

Chapter 4 contains the contributions of this work to unsupervised class-based modeling. More specifically, after a short comparison of three different adaptation techniques for class-based modeling (MLLR, MAP, MLLR+MAP), the tolerance margin is introduced and studied in the framework of KL-based soft clustering with phonetic HMM-GMM. Different ways to tune the margin parameter are proposed and evaluation of the corresponding ASR performance is done on ESTER2 radio broadcast data. Finally, ROVER (Recognizer Output Voting Error Reduction system) combination of the hypotheses achieved with different class-based models is analyzed.

Chapter 5 introduces another important contribution of this thesis. A novel approach based on *Class-Structuring of the GMM components* (CS-GMM) completed with *Class-Dependent mixture Weights* (CS-CDW-GMM) is proposed. First, the initialization, re-estimation and decoding algorithms are formulated. Then, an analysis of class-dependent mixture weights is done. Finally, ASR experiments carried out on French radio broadcast data and on a digit recognition task with adult and child speakers are described.

Chapter 6 describes *Stranded GMM* (StGMM) and presents the corresponding training and decoding algorithms. This model was proposed in [Zhao and Juang, 2012]. This chapter describes the model in greater detail. Namely, the EM algorithm is introduced, and additional analysis of the re-estimated mixture transition matrices is described. Furthermore, ASR experiments carried out on English data with both speaker (age and gender) and channel (non-stationary noise) variability are discussed.

Chapter 7 presents *Class-Structured Stranded GMM* (CS-StGMM) framework. After discussing the motivation for combining class-structuring and component dependencies, the general framework is formulated and ASR experiments are described. For this chapter, the main experiments are conducted on a small English connected digits task. In addition, the best performing approach is also applied on phonetic decoding experiments carried out on a large French telephone speech database.

Each chapter ends with a short conclusion, which summarizes the corresponding approach and discusses the results. Finally, the thesis ends with a global conclusion and a discussion of future work.

2

Automatic speech recognition

This chapter describes basic concepts and the main problems of Automatic Speech Recognition. Section 2.1 briefly summarizes the history of research in the last 50 years, discusses general approaches and some of the first ASR systems. Section 2.2 describes a general framework of statistical speech recognition based on Hidden Markov Models, mostly focusing on the acoustic modeling problem. Section 2.3 introduces variability in the speech signal and trajectory folding that comprise the major challenges of this thesis. Section 2.4 briefly discusses widely used techniques of adaptation and parameter sharing that allow to partially reduce the variability and achieve more accurate modeling. The chapter ends with a short discussion and conclusion.

Contents

2.1 A brief history of speech recognition

Early attempts to automatically recognize human speech go back to early 50's, when people tried to recognize separate digits by comparing acoustic characteristics of the input signal with reference patterns [Davis *et al.*, 1952]. Next 30 years, the main research direction in the field was towards the improvement of pattern matching techniques and building various parametric representations of speech signal. Many key techniques, which are commonly used nowadays, were first introduced in that time [Juang and Rabiner, 2005]; for example, *Dynamic Programming* (DP) algorithms for pattern matching [Viterbi, 1967; Vintsyuk, 1968], *Linear Predictive Coding* (LPC) for representing the speech waveform [Atal and Hanauer, 1971], etc. Many other techniques that

were also intensively explored, somehow vanished with time; for example, some of the early ASR systems attempted to introduce hard-coded phonological and word-boundary rules, which are not considered in most of the state-of-the art systems.

In 1970-80's the main research direction in *Automatic Speech Recognition* (ASR) changes from template matching and rule-based to the statistical modeling framework, which is still actively used in state-of-the-art *Large Vocabulary Continuous Speech Recognition* (LVCSR). In statistical modeling approaches, the speech signal is now represented in terms of probabilistic *Hidden Markov Models* (HMMs). HMMs were described in 60's by Leonard E. Baum and his colleagues [Baum and Petrie, 1966; Baum and Eagon, 1967] and applied to speech about ten years later. The HMM formalism allows to combine the language knowledge together with the temporal acoustic realization of speech sounds in the utterance. As a result, it also motivated the development of other research directions, such as statistical language modeling for ASR [Jelinek, 1990].

At the same time, many research programs were funded and various evaluation campaigns were maintained by the Advanced Research Projects Agency (ARPA) in the USA in order to search for the best strategies for building speech recognition and speech understanding systems and to force competitions between research laboratories. Among other systems developed within the ARPA project was "Harpy" [Lowerre, 1976] by Carnegie Mellon University (CMU), which was quite accurately recognizing speech using a vocabulary of 1011 words and relying on template matching and beam search. Two other systems Hearsay-II [Erman *et al.*, 1980] by CMU and HWIM (Hear What I Mean) by BNN [Dennis H. Klatt, 1977] used lexical decoding network and phonological rules. Some of the first applications of statistical approaches for speech recognition appeared in the DRAGON [Baker, 1975] and IBM TANGORA systems [Jelinek *et al.*, 1975].

First evaluation campaigns were conducted on small-vocabulary data with accurately articulated speech, or read speech. From the end of 80's, with support of Defense Advanced Research Projects Agency (DARPA) and National Institute of Standards and Technology (NIST), various evaluations were carried on a wide range of ASR tasks. This was gradually moving the systems towards recognizing more natural speech, such as radio broadcasts, telephone conversations, television shows, debates, interviews, etc.

Such speech recognition evaluation tasks were causing a massive exchange of ASR technologies between various research laboratories. The HMM-based systems were performing well, especially on continuous speech recognition tasks, which led to their wide usage up to these days. Since the first implementation of HMM in a speech recognizer, many techniques have been developed to improve the recognition accuracy, although the statistical framework still remains the core for most of the recognizers.

Another benefit from such evaluation campaigns was the appearance of publicly available audio transcribed corpora for research purposes. Some popular examples of such corpora in English include (but are not limited to) acoustic-phonetic continuous speech corpus TIMIT [Garofolo *et al.*, 1993], DARPA Resource Management continuous speech corpora (RM) [Price *et al.*, 1988], Air Travel Information Service data (ATIS) [Hemphill *et al.*, 1990], radio broadcast data from the Linguistic Data Consortium (LDC) [Graff *et al.*, 1997], telephone data Switchboard [Godfrey *et al.*, 1992] and Fisher large conversational telephone corpora [Cieri *et al.*, 2004], which capture various types of speech and recording conditions. The summary of such evaluations for English language until 2009 has been recently published by NIST. It shows how the performance of ASR systems was improving over time with respect to the size and the difficulty of the tasks (Figure 2.1).

The active research in ASR also led to the emergence of various publicly available software tools. The most popular examples include the open-source speech recognition en-

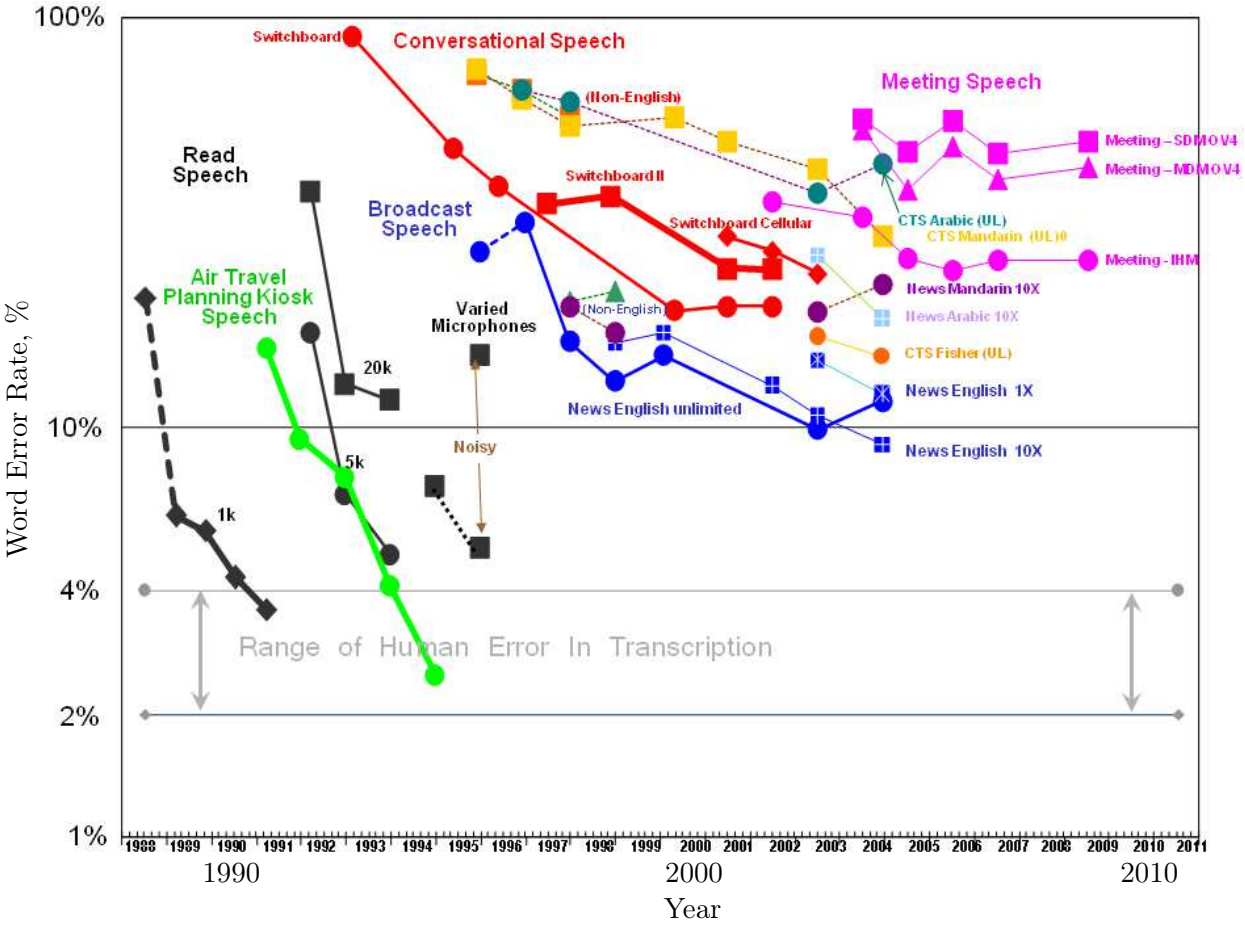

FIGURE 2.1 – NIST Speech-To-Text (STT) benchmark test history
(Source: http://www.itl.nist.gov/)

gines Sphinx [1] initially launched by Carnegie Mellon University (CMU) [Lee *et al.*, 1990; Ravishankar, 1996], HTK [2] by Cambridge University [Young *et al.*, 2006], Julius [3] by Kyoto University[Lee *et al.*, 2001], RWTH [4] by Aachen University [Rybach *et al.*, 2009] and Kaldi [5] by Johns Hopkins University [Povey *et al.*, 2011b]. Together with ASR engines, many other software tools are available. Some popular examples are language modeling tools, such as SRILM [6] by SRI, SLM [7] by CMU, Recurrent Neural Network LM (RNNLM [8]), tools for prosody analysis Praat [9] and WinSnoori [10], and many others.

In a similar way, speech recognition evaluation campaigns were organized for French speech recognition. One of the earliest examples was read speech transcription task from "l'Association

1. http://cmusphinx.sourceforge.net/
2. http://htk.eng.cam.ac.uk/
3. http://julius.sourceforge.jp/en_index.php
4. http://www-i6.informatik.rwth-aachen.de/rwth-asr/
5. http://kaldi.sourceforge.net/
6. http://www.speech.sri.com/projects/srilm/
7. http://www.speech.cs.cmu.edu/SLM/toolkit.html
8. http://www.fit.vutbr.cz/~imikolov/rnnlm/
9. http://www.fon.hum.uva.nl/praat/
10. http://www.loria.fr/~laprie/WinSnoori/

des universités partiellement ou entièrement de langue française" (AUPELF) in 90's [Dolmazon *et al.*, 1997]. More recent evaluations are dealing with radio and TV broadcast data; for example, transcription evaluation campaigns ESTER 2003-2005 [Galliano *et al.*, 2005], ESTER 2007-2009 [Galliano *et al.*, 2009] and ETAPE 2010-2012 [Gravier *et al.*, 2012] supported by the French National Research Agency (ANR).

2.2 Acoustic modeling in statistical speech recognition

Acoustic model is a core of statistical speech recognition systems. It aims to predict the basic phonological units based on the fact that different phones are associated with different acoustic features that are derived from speech data. The following sections explain the general framework of statistical speech recognition and formulate the training and decoding algorithms.

2.2.1 Probabilistic model for speech recognition

This section briefly describes the general modeling problem for ASR. A more detailed description can be found in various literature sources. Some of the most influential and understandable reviews, used for writing this chapter are [Rabiner, 1989], [Gales and Young, 2008] and [Jurafsky and Martin, 2009].

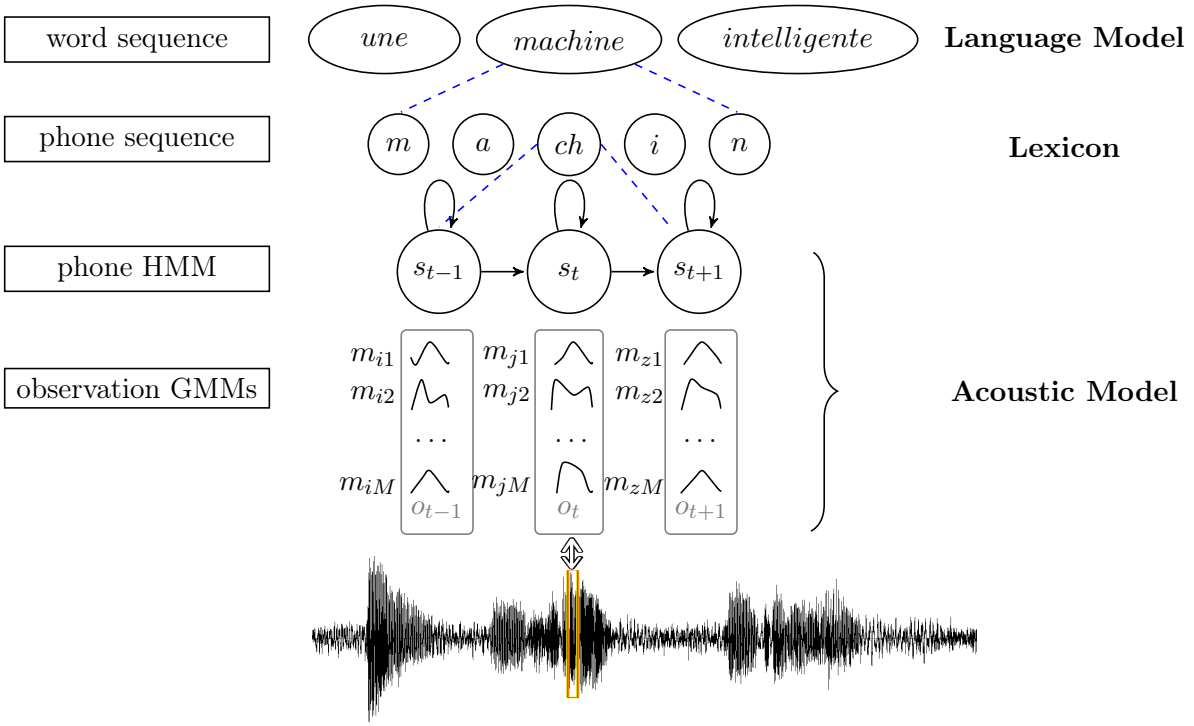

FIGURE 2.2 – Modeling of different levels of information of a spoken sentence by a conventional Hidden Markov Model

Consider first a French sentence "Une machine intelligente" (an intelligent machine), recorded in an audio file. It turns out that such a simple speech message contains various levels of information. These levels and the corresponding models are shown in Figure 2.2. The following sections discuss these levels and the associated models. The discussion starts from the signal level, on

which the *acoustic analysis* is applied to extract useful features. Next, the word sequence level is described. At this level the *language model* is used as a model of the knowledge about the correct word sequence from the language perspective. Then, two intermediate levels of information are considered. The first intermediate level splits each word into sequences of simple phonological units (phones) based on the *lexicon* that also defines different pronunciations of the same word. Finally, the general probabilistic model of ASR and the *acoustic model* are discussed. The acoustic model consists of a *Hidden Markov Model* (HMM) that defines the temporal distribution of the acoustic features for each phone.

Acoustic analysis front end

At the lowest level of the Figure 2.2 we have the digital representation of the acoustic waveform, which represents the amplitude of the sound wave. The signal is first pre-processed (pre-emphasis) and split into short frames (10...25 milliseconds) with some overlap. Then, acoustic features are extracted from each frame. The purpose of the *feature extraction* is, on the one hand, to reduce the data dimensionality for acoustic modeling, and on the other hand to capture the most important parameters of speech, discarding useless information in the signal.

Mel Frequency Cepstral Coefficients (MFCC) are widely used acoustic features for ASR [Davis and Mermelstein, 1980], although some better representations can also be considered. MFCC feature extraction is schematically summarized in Figure 2.3. Standard features for ASR include 12 cepstral features, 1 log energy and their first and second derivatives.

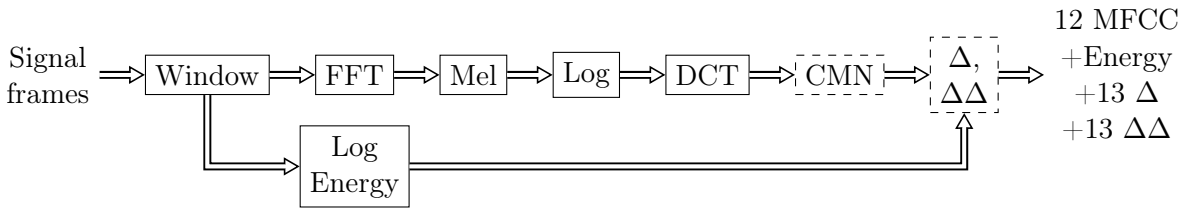

FIGURE 2.3 – Schematic process of standard 39 MFCC features extraction

To compute 12 cepstral features for each time frame of the digital signal, firstly a bell-shaped *Hamming window* is applied to avoid spectral "artifacts" appearing after Fourier transformation of discontinuities of the discrete signal at the frame boundaries. After windowing, the signal is converted from temporal to spectral domain by applying a *Discrete Fourier Transform* (DFT), or a *Fast Fourier Transform* (FFT). To filter most relevant spectral bands, a *Mel filter bank* is applied, followed by a *logarithmic transformation*. Finally, a *Discrete Cosine Transform* (DCT) is applied to derive cepstral features. The features can be further normalized by applying a *Cepstral Mean Subtraction* or a *Cepstral Mean Normalization* (CMN) [Liu *et al.*, 1993]. In addition to 12 cepstral features the *log energy* of the signal has become a standard part of MFCC features for ASR. Finally, for both cepstral and log energy features the first and second *temporal derivatives* are computed to better capture the dynamic properties of speech.

Language model

At the top of the schematic representation (Figure 2.2) the word sequence is viewed from the language perspective. Namely, the text of a spoken sentence is given as a set of words $\boldsymbol{\mathcal{W}} = \{w_1, w_2, \ldots, w_N\}$, which belong to some language $\boldsymbol{\mathcal{L}}$.

As humans, we have good capabilities in working with sequences of words, if the corresponding language is known. We can do an evaluation task, saying whether the sentence is grammatically and semantically correct, or not. Moreover, we can predict some missing words or letters in a text. In both cases we use contextual information and language knowledge. As it was proved in [Shannon, 1951], humans are indeed good predictors of missing words in sentences. When we want a machine to predict the word given a context, or to give an answer about how likely a given word sequence is correct for a given language, we need a *language model* (LM).

A widely-used model of language is *N-gram*, which computes probabilities of each word given the $N-1$ previous words. In the ASR task the LM is frequently represented by a 2^{nd} order *Markov process* - 3-grams, where the probability of a given word w_i depends on the two previous words

$$P(w_i|w_1,\ldots,w_{i-2},w_{i-1}) \approx P(w_i|w_{i-2},w_{i-1}) \tag{2.1}$$

N-grams can be trained from a text corpus by counting word sequences and normalizing counts. This process of model training is also known as *Maximum-Likelihood Estimation* (MLE). In practice, to take into account unobserved 3-grams in the training set and to better generalize on test data, techniques like *backing-off* to 2-grams and 1-grams are applied together with various *smoothing* techniques [Jelinek, 1990].

There were many research works trying to improve the modeling or to use more detailed models of the language knowledge for ASR. Widely applied techniques rely on re-scoring of the ASR results with N-gram LM of higher order and LM adaptation to improve the ASR hypothesis [Bellegarda, 2004]. Although more advanced language modeling can significantly improve ASR performance, these aspects are not related to this thesis. For a detailed description of LM aspects, very good books to start with are [Jelinek, 1997; Jurafsky and Martin, 2009].

Lexicon

So far, two different aspects have been described: the acoustic analysis and the language modeling. To build a direct link between words and observations, two other levels are introduced: *phones* and *model states*. A *phone* is a simple unit of the speech sound. The mapping between words and phones is determined by the *Lexicon*.

The lexicon plays an important role in any ASR system and solves the following tasks:

1. defines the vocabulary, or all possible words that the system can recognize;
2. links the acoustic and language knowledge by associating a sequence of phones $(ph_1,...,ph_N)$ with each word from $\boldsymbol{\mathcal{L}}$
   For example, "machine"→(m,a,ch,i,n);
3. defines the pronunciation variants of each word, which are also useful to model dialects, accents, or "liaison" (an effect of changing the pronunciation of the word ending depending on the following phone appearing in French language).
   For example, the word "deux" can be pronounced as (d,eu) or (d,eu,z) depending on the next word.

General formulation of the probabilistic model for ASR

This section formulates the general probabilistic model of ASR, i.e., the problem of inferring a sequence of words given the observations of acoustic features. Let us denote $\mathbf{o}_t$ the acoustic feature vector (also called *observation*) for time frame t. Also denote $\boldsymbol{\mathcal{O}} = \{\mathbf{o}_1,\ldots,\mathbf{o}_t,\ldots,\mathbf{o}_T\}$

the *observation sequence* associated with a *word sequence* $\boldsymbol{\mathcal{W}}$. The problem of speech recognition can be formulated as finding the most probable sequence of words given the observation sequence.

$$\hat{\boldsymbol{\mathcal{W}}} = \arg\max_{\boldsymbol{\mathcal{W}} \in \boldsymbol{\mathcal{L}}} P(\boldsymbol{\mathcal{W}}|\boldsymbol{\mathcal{O}}) \tag{2.2}$$

where, again, $\boldsymbol{\mathcal{L}}$ denotes a language.

In practice, for *generative models* (like HMM) it is easier to model the *likelihood function* $P(\boldsymbol{\mathcal{O}}|\boldsymbol{\mathcal{W}})$ rather than the *posterior probability* $P(\boldsymbol{\mathcal{W}}|\boldsymbol{\mathcal{O}})$. Equation 2.2 can be rewritten using the *Bayes' rule* and simplified using the fact that when decoding a given utterance the observation sequence probability $P(\boldsymbol{\mathcal{O}})$ remains the same. The most likely sequence of words is then determined as follows:

$$\hat{\boldsymbol{\mathcal{W}}} = \arg\max_{\boldsymbol{\mathcal{W}} \in \boldsymbol{\mathcal{L}}} \frac{P(\boldsymbol{\mathcal{O}}|\boldsymbol{\mathcal{W}})P(\boldsymbol{\mathcal{W}})}{P(\boldsymbol{\mathcal{O}})} = \arg\max_{\boldsymbol{\mathcal{W}} \in \boldsymbol{\mathcal{L}}} \overbrace{P(\boldsymbol{\mathcal{O}}|\boldsymbol{\mathcal{W}})}^{\text{AM likelihood}} \underbrace{P(\boldsymbol{\mathcal{W}})}_{\text{LM prior}} \tag{2.3}$$

Using Equation 2.3 the problem of finding the best word sequence relies on computation of the *acoustic model likelihood* $P(\boldsymbol{\mathcal{O}}|\boldsymbol{\mathcal{W}})$ and of the *language model prior* $P(\boldsymbol{\mathcal{W}})$. These two probabilities have to be maximized together in order to get the best sequence of words. In order to compute $P(\boldsymbol{\mathcal{O}}|\boldsymbol{\mathcal{W}})$ the words can be split into smaller units. The smallest unit in statistical modeling approach is represented by *Hidden Markov Model* (HMM). For a small-vocabulary task each separate HMM can represent a word. However, in most of the practical applications the basic HMM (with 1-5 states) is associated with a phone, a triphone (a phone with different left and right contextual units), or a shared triphone (or senone). As it was described earlier, such mapping between words and phones is the task of the *lexicon*.

2.2.2 Architecture of HMM-based acoustic model

HMM with a continuous observation density function in the form of *Gaussian Mixture Model* (GMM) is one of the most widely-used models for ASR. This section describes the general structure and the parameters of this model. Later, the training and the decoding algorithms are described.

Hidden Markov Model

HMM was successfully applied as a core of statistical ASR in many systems [Rabiner, 1989]. HMM is an extension of Markov chains, which are widely used to model simple sequences. Unlike classical Markov chain, HMM allows us to model hidden events (like phone-dependent states, hidden in the speech signal) and observable events (for example, the acoustic features extracted from the speech signal). One of the most popular types of HMM for modeling the speech and other temporal processes has forward-directed edges and loops (*Bakis model*). Usually the phone model has from 3 to 5 states (Figure 2.4), because the features at the phone edges can significantly differ from the features of the middle part of the same phone.

HMM is defined by the following set of parameters:

1. *State sequence* $\boldsymbol{\mathcal{Q}} = (q_1, \ldots, q_t, \ldots, q_T)$, where $q_t \in \{1, \ldots, N\}$ is a state at time frame t
2. Transition probability matrix $\mathbf{A} = \{a_{ij}\}$, whose elements are called *state transition probabilities* (or simply *transition probabilities*) $a_{ij} = P(q_t = j|q_{t-1} = i)$ for some states

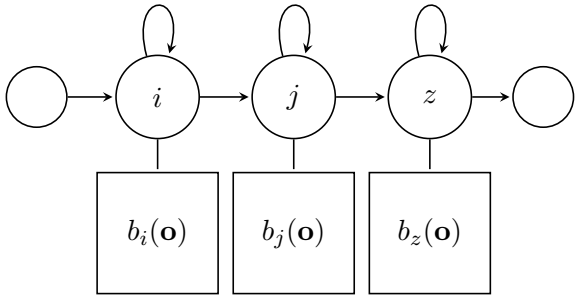

FIGURE 2.4 – A Hidden Markov Model of phone

$i, j \in \{1, .., N\}$ and $\sum_{j=1}^{N} a_{ij} = 1$.

3. Set of *observation likelihood functions* (or *observation densities*) $\mathbf{B} = \{b_j(\mathbf{o}_t)\} = P(\mathbf{o}_t|q_t = j)$ with parameters associated with the corresponding HMM states, $\mathbf{o}_t \in \mathbb{R}^d$ is the observation vector, and d is the dimension of the feature space

4. Initial state distribution, not associated with an observation $\pi_i = P(q_0 = i)$ for each initial state q_0 and $\pi_i = 0$ otherwise

Let us come back to the statistical model earlier defined by the Equation 2.3. In order to compute the likelihood of the word sequence $P(\boldsymbol{\mathcal{O}}|\boldsymbol{\mathcal{W}})$ let us rewrite it in the form of state sequence $P(\boldsymbol{\mathcal{O}}|\boldsymbol{\mathcal{Q}})$, where each state is associated with frame observation vector. The task can now be formulated as finding the state sequence leading to the maximum likelihood of the observation sequence $\boldsymbol{\mathcal{O}}$. Applying Bayes formula leads to the following derivation:

$$\hat{\boldsymbol{\mathcal{Q}}} = \arg\max_{\mathbf{Q}} P(\boldsymbol{\mathcal{Q}}|\boldsymbol{\mathcal{O}}) = \arg\max_{\mathbf{Q}} \frac{P(\boldsymbol{\mathcal{O}}|\boldsymbol{\mathcal{Q}})P(\boldsymbol{\mathcal{Q}})}{P(\boldsymbol{\mathcal{O}})} = \arg\max_{\mathbf{Q}} P(\boldsymbol{\mathcal{O}}|\boldsymbol{\mathcal{Q}})P(\boldsymbol{\mathcal{Q}}) \tag{2.4}$$

It can be seen from the structure of HMM that the two following assumptions take place. They play an important role in further derivations of various algorithms of speech recognition, but put strong limitations on the modeling accuracy of HMM:

1. *The first-order Markov chain assumption*: every state depends only on the previous state: $P(q_t|q_1, \ldots, q_{t-1}, q_{t+1}, \ldots, q_T) = P(q_t|q_{t-1})$

2. *Output independence assumption*: each observation depends only on the state, which produces this observation:
$P(\mathbf{o}_t|q_1, \ldots, q_{t-1}, q_t, \ldots, q_T, \mathbf{o}_1, \ldots, \mathbf{o}_{t-1}, \mathbf{o}_{t+1}, \ldots, \mathbf{o}_T) = P(\mathbf{o}_t|q_t)$

Taking into account these assumptions, the Equation 2.4 is significantly simplified and can be rewritten in terms of the HMM parameters as follows:

$$\hat{\boldsymbol{\mathcal{Q}}} = \arg\max_{\mathbf{Q}} \prod_{t=1}^{T} P(\mathbf{o}_t|q_t)P(q_t|q_{t-1}) = \arg\max_{\mathbf{Q}} \pi_{q_0} \prod_{t=1}^{T} a_{q_{t-1}q_t} b_{q_t}(\mathbf{o}_t) \tag{2.5}$$

In practice and in the remainder of the thesis a single initial state is assumed. Therefore, the initial distribution is excluded from the derivations ($\pi_{q_0} = 1$).

Gaussian mixture probability density function

An important part of HMM is the state *likelihood function* (emission probability) $b_j(\mathbf{o}_t) = P(\mathbf{o}_t|q_t = j)$, which represents the probability of observing the feature vector $\mathbf{o}_t$ given a model state j at time t. Initially, HMM-based systems applied for ASR used discrete likelihood function associated with *discrete vectors*. Later, they were replaced by more accurate continuous density function in the form of the *Gaussian Mixture Model* (GMM). For the last few decades, GMMs were the most widely-used density functions for likelihood computation in ASR. In a *Hidden Markov Model with Gaussian Mixture Observation density* (HMM-GMM, or simply HMM [1]), the observation *probability density function* (pdf) for a model state j is defined as a weighted sum of M multivariate Gaussian functions (or *components of GMM*):

$$b_j(\mathbf{o}_t) = P(\mathbf{o}_t|q_t = j) = \sum_{l=1}^{M} \omega_{jl} \mathcal{N}(\mathbf{o}_t|\boldsymbol{\mu}_{jl}, \boldsymbol{\Sigma}_{jl}) \tag{2.6}$$

where $\mathcal{N}(\mathbf{o}_t|\boldsymbol{\mu}_{jl}, \boldsymbol{\Sigma}_{jl})$ is a Gaussian density with mean vector $\boldsymbol{\mu}_{jl}$ and covariance matrix $\boldsymbol{\Sigma}_{jl}$ (often, only diagonal elements are used). For a feature vector $\mathbf{o}_t$ of a size n:

$$\mathcal{N}(\mathbf{o}_t|\boldsymbol{\mu}_{jl}, \boldsymbol{\Sigma}_{jl}) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}_{jl}|}} \exp\left\{ -\frac{1}{2} (\mathbf{o}_t - \boldsymbol{\mu}_{jl})^T \boldsymbol{\Sigma}_{jl}^{-1} (\mathbf{o}_t - \boldsymbol{\mu}_{jl}) \right\} \tag{2.7}$$

Each component l of a state j has its weight ω_{jl}, and the weights are subject to the constraint

$$\sum_{l=1}^{M} \omega_{jl} = 1 \tag{2.8}$$

Unlike a single Gaussian function, a GMM has more than one peak. Therefore, separated groups of data points can be modeled by different density components. This property is very important for ASR, where different distributions of the acoustic features can correspond to the same phonetic unit because of variability in the speech signal, which is discussed in detail later in Section 2.3.1.

For example, Figure 2.5 compares the estimates of a single Gaussian function and a mixture of two Gaussian functions for some set of data points in a one-dimensional space. In this example, the data are well parameterized by the mixture, while using single Gaussian leads to averaging with a larger variance.

2.2.3 HMM training: forward-backward algorithm

The training task consists in deriving the model parameters given speech signal and corresponding transcript. Consider that the model structure is fixed and known. Therefore, the parameter estimation can be seen as a fundamental problem of training in Machine Learning [Rabiner, 1989; Deng and Li, 2013]. As in most of the practical cases the audio transcriptions contain the corresponding words without any information about state duration and phone boundaries, the training should not only solve the problem of finding the best parameters, but also implicitly align the transcript and the audio.

The *Maximum-Likelihood* (ML) optimization criterion is used to train the HMM parameters. The training algorithm is known as *Expectation-Maximization* (EM) with its efficient *Dynamic*

1. Later in the thesis for simplicity HMM always assumes GMM observation density, unless the form of the density function is explicitly defined (i.e., HMM-ANN, HMM-SVM, HMM-DNN, etc.)

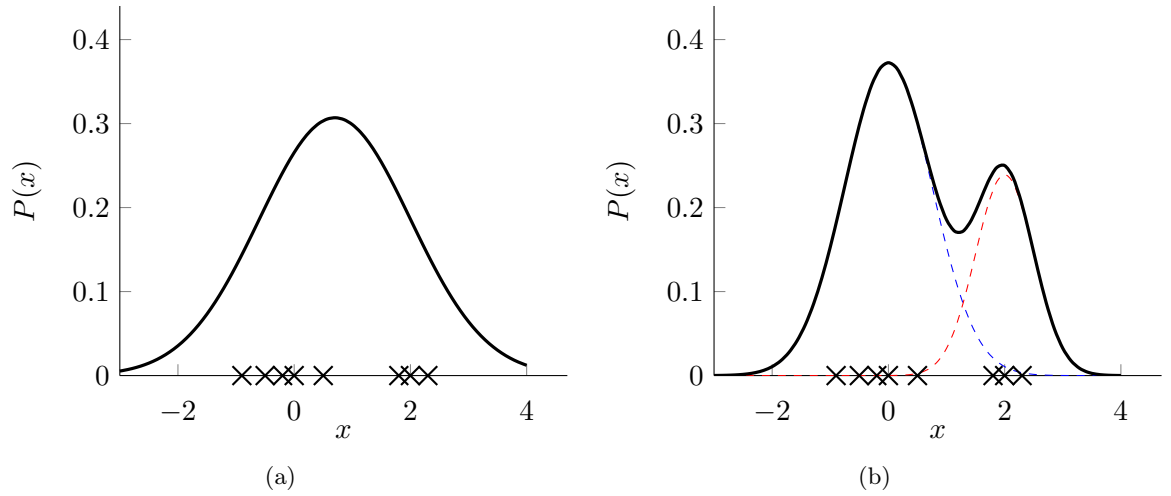

FIGURE 2.5 – EM estimate of some data distribution with a single Gaussian function (a) and a mixture of 2 Gaussian functions (b)

Programming (DP) implementation, known as *Baum-Welch* algorithm. Baum-Welch algorithm appeared in ASR field in the work of [Baum and Petrie, 1966] and has been later generalized to the general Expectation-Maximization for Machine Learning [Dempster *et al.*, 1977]. *Maximum Likelihood Estimation* (MLE) guarantees to find an optimal set of parameters if the model structure is correct, the training data is drawn from the true distribution and the amount of training data is infinite. As in practice none of these assumptions are correct, MLE gives approximate, but "good enough" estimation.

Let us describe in detail the Baum-Welch algorithm for HMM training. Assume the initial model structure is given and unchanged, and the initial set of the parameters $\boldsymbol{\lambda}'$ (described in Section 2.2.2) is defined. The initial parameters are denoted with "prime" symbol.

The goal is to estimate a new set of the parameters $\boldsymbol{\lambda}^*$, which maximize the likelihood of the observed data $P(\boldsymbol{\mathcal{O}}|\boldsymbol{\lambda})$

$$\boldsymbol{\lambda}^* = \arg\max_{\boldsymbol{\lambda}} P(\boldsymbol{\mathcal{O}}|\boldsymbol{\lambda}) \tag{2.9}$$

Such maximization problem can be locally solved by maximizing *Baum's auxiliary function*:

$$Q(\boldsymbol{\lambda}, \boldsymbol{\lambda}') = \sum_{\boldsymbol{\mathcal{Q}}} P(\boldsymbol{\mathcal{Q}}|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \log P(\boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{Q}}|\boldsymbol{\lambda}) \tag{2.10}$$

Accurate detailed derivations of the re-estimation equations for HMM model parameters can be found in [Rabiner, 1989] and [Bilmes, 1998]. The resulting equations can further be efficiently computed by DP-based *Baum-Welch algorithm*.

Baum-Welch consists in computing separately *forward* and *backward* variables, which represent the left and right parts of the sequence.

The forward variable is defined as the probability of the partial observation sequence $\{\mathbf{o}_1, \dots \mathbf{o}_t\}$ and state j at time t:

$$\alpha_t(j) = P(\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_t, q_t = j|\boldsymbol{\lambda}') = [\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij}] b_j(\mathbf{o}_t) \tag{2.11}$$

In a similar way the backward variable is defined as the probability of the observation sequence from the time $t+1$ to the end time T, given the current state i at time t with the model λ'

$$\beta_t(i) = P(\mathbf{o}_{t+1}, \mathbf{o}_{t+2}, \dots \mathbf{o}_T | q_t = i, \boldsymbol{\lambda'}) = \sum_{j=1}^{N} a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j) \tag{2.12}$$

Then, some important variables are defined. First, the probability of being at state i at time t and state j at time $t+1$, given the initial model $\boldsymbol{\lambda'}$ and the observation sequence $\boldsymbol{\mathcal{O}}$ (expected number of transitions from state i to state j observing $\boldsymbol{\mathcal{O}}$):

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda'}) = \frac{\alpha_t(i) a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j)}{\sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_t(n) a_{nm} b_m(\mathbf{o}_{t+1}) \beta_{t+1}(m)} \tag{2.13}$$

Second, the probability of being at the state i at time t, given the observation sequence and the model (expected number of transitions from state i observing $\boldsymbol{\mathcal{O}}$) is defined as follows:

$$\gamma_t(i) = P(q_t = i | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda'}) = \frac{\alpha_t(i) \beta_t(i)}{\sum_{n=1}^{N} \alpha_t(n) \beta_t(n)} = \sum_{j=1}^{N} \xi_t(i,j) \tag{2.14}$$

Using the Equations 2.13 and 2.14, re-estimation formulas for HMM transition probabilities are derived as follows:

$$a_{ij} = \frac{\sum_{t=1}^{T} p(q_{t-1} = i, q_t = j, \boldsymbol{\mathcal{O}} | \boldsymbol{\lambda'})}{\sum_{t=1}^{T} p(q_{t-1} = i, \boldsymbol{\mathcal{O}} | \boldsymbol{\lambda'})} = \frac{\sum_{t=1}^{T} \xi_t(i,j)}{\sum_{t=1}^{T} \gamma_t(i)} \tag{2.15}$$

To derive the re-estimation formulas for the parameters of the Gaussian densities, another random variable is introduced. Let $m_t \in \{1, \dots, M\}$ denote a particular component of the density, observed at time t. Then, the probability of observing the component l of state j at time t is defined as follows:

$$\gamma_t(j,l) = P(q_t = j, m_t = l | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda'}) = \frac{\omega_{jl} \mathcal{N}(\mathbf{o}_t | \boldsymbol{\mu}_{jl}, \boldsymbol{\Sigma}_{jl})}{\sum_{k=1}^{M} \omega_{jk} \mathcal{N}(\mathbf{o}_t | \boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk})} \gamma_t(j) \tag{2.16}$$

Now, the re-estimation formulas for mixture weight, mean and variance values for some state j and component l are derived as follows:

$$\omega_{jl} = \frac{\sum_{t=1}^{T} P(q_t = j, m_t = l, \boldsymbol{\mathcal{O}} | \boldsymbol{\lambda'})}{\sum_{t=1}^{T} P(q_t = j, \boldsymbol{\mathcal{O}} | \boldsymbol{\lambda'})} = \frac{\sum_{t=1}^{T} \gamma_t(j,l)}{\sum_{t=1}^{T} \gamma_t(j)} \tag{2.17}$$

$$\boldsymbol{\mu}_{jl} = \frac{\sum_{t=1}^{T} P(q_t = j, m_t = l, \boldsymbol{\mathcal{O}}|\boldsymbol{\lambda'}) \cdot \mathbf{o}_t}{\sum_{t=1}^{T} P(q_t = j, m_t = l, \boldsymbol{\mathcal{O}}|\boldsymbol{\lambda'})} = \frac{\sum_{t=1}^{T} \gamma_t(j,l) \cdot \mathbf{o}_t}{\sum_{t=1}^{T} \gamma_t(j,l)} \tag{2.18}$$

$$\boldsymbol{\Sigma}_{jl} = \frac{\sum_{t=1}^{T} P(q_t = j, m_t = l, \boldsymbol{\mathcal{O}}|\boldsymbol{\lambda'}) \cdot (\mathbf{o}_t - \boldsymbol{\mu'}_{jl})(\mathbf{o}_t - \boldsymbol{\mu'}_{jl})^T}{\sum_{t=1}^{T} P(q_t = j, m_t = l, \boldsymbol{\mathcal{O}}|\boldsymbol{\lambda'})} = \frac{\sum_{t=1}^{T} \gamma_t(j,l) \cdot (\mathbf{o}_t - \boldsymbol{\mu'}_{jl})(\mathbf{o}_t - \boldsymbol{\mu'}_{jl})^T}{\sum_{t=1}^{T} \gamma_t(j,l)} \tag{2.19}$$

2.2.4 Decoding problem

In the decoding problem given an HMM with the set of parameters $\boldsymbol{\lambda}$ and a sequence of observations $\boldsymbol{\mathcal{O}}$ the task is to compute the most probable sequence of states $\boldsymbol{\mathcal{Q}}$. A widely-used solution is the Viterbi algorithm [Viterbi, 1967]. Let the *Viterbi path score* denote the highest probability along the best path ending at time t in state j:

$$v_t(j) = \max_{q_1, q_2, \ldots, q_{t-1}} P(q_1, \ldots q_{t-1}, q_t = j, \mathbf{o}_1, \ldots, \mathbf{o}_t|\boldsymbol{\lambda}) = \max_i [v_{t-1}(i) a_{ij}] b_j(\mathbf{o}_t) \tag{2.20}$$

Equation 2.20 defines the highest probability of the path, but to actually retrieve the state sequence, *backpointer* values must be computed by taking arg max instead of max in the same equation

$$bp_t(j) = \arg\max_i [v_{t-1}(i) a_{ij}] \tag{2.21}$$

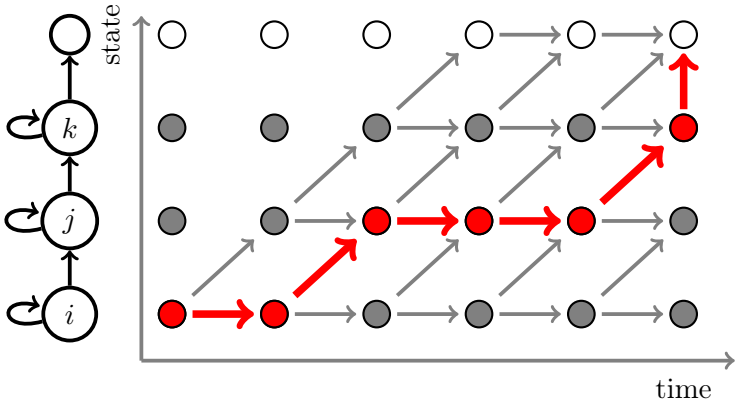

FIGURE 2.6 – Example of Viterbi search for a single phone unit

To actually apply the Viterbi algorithm for ASR, word HMMs are constructed by concatenating phonetic HMMs using non-emitting states. Non-emitting states are also connecting final states of the words with the initial states of the following word candidates. These cross-word transitions are also used to include the language model probabilities.

The number of operations required for the Viterbi decoder is at most equal to N^2T (if all states are connected to all states), where N is the number of states and T is the number of

frames. Generally, if K is an average number of transitions ending in a state (usually $K \ll N$), the Viterbi algorithm requires KNT operations. Nevertheless, as the total number of states to evaluate can be quite large in LVCSR, *beam search* heuristic is implemented. The idea of beam search is to extend the path for the next frame only from the states, for which the path probabilities are higher than a certain threshold that is selected at each frame according to beam width [Lowerre, 1976]. Another decoding algorithm applied for speech recognition is the stack decoding implemented in [Jelinek, 1969].

2.3 Main ASR problems addressed in the thesis

Speech signal has a complex dynamic form. Although the earlier described HMM-based approach is widely applied for ASR, its performance is highly sensitive to speech variability and data mismatch. The following sections review the main sources of variability in the speech signal and describe a particular problem of trajectory folding associated with HMM.

2.3.1 Speech variability

Although today many speech data are available to train large statistical models, the state-of-the-art speech recognition systems are far from ideal. One of the main reasons is the variability of the speech signal. Many studies were done to understand the main sources of speech variability and its impact on recognition errors. Globally, the main sources of speech variability are classified [Kajarekar *et al.*, 1999; Sun and Deng., 1995] as follows:

— recording conditions (environment and transmission channel)
— speaker variability
— co-articulation effects or context of the phonetic units

The channel and environmental variability consist mostly from the microphone, noise, room reverberation, etc. Some relatively simple cases of such variations can be handled in a pre-processing step, or in the feature extraction phase. Dealing with noisy data is yet another challenge of speech recognition and an active research direction.

Another source of variability is caused by differences in pronunciation across different speakers. A detailed classification of speaker variability is described in [Benzeghiba *et al.*, 2007]. Globally, speaker variability is subdivided on inter- and intra-speaker variability.

Inter-speaker variability is caused by differences across different speakers saying the same thing. The most influencing parameters of inter-speaker variability include speaker gender, accent and age. From statistical experiments [Huang *et al.*, 2001], it turns out that gender and accent bring most of the variation (however, children speech was not evaluated). In various works ASR performance on accented speech degrades dramatically if the model is trained on native speech [Lawson *et al.*, 2003]. Gender variability (and most of other inter-speaker variability) mostly comes from the vocal tract size and shape differences [Wegmann *et al.*, 1996; O'Shaughnessy, 2013]. For example, children have shorter vocal tract than adult women and much shorter than adult men, therefore, higher F0 (fundamental frequency).

Intra-speaker variability is caused by the fact that even for the same speaker there are many factors, which can significantly modify the acoustic parameters of the same phonetic units in the same phonetic context. Significant differences appear, for example, due to speaker health condition or emotions. Whispering and shouting also severely modifies the spectral representation of the phone units.

The last, but not the least variability source is co-articulation or context. If the model is trained on separately pronounced phones, it simply cannot be applied for continuous speech

task due to high errors, because it cannot handle the co-articulation effects. The first reason is that a phonetic unit has different acoustic representations in different contexts. Moreover, other speech dynamical factors (speaking rate, co-articulation) can significantly modify the sound representation. Knowing the exact pronunciation of each word can be very useful, but difficult to achieve in practice. For example, in [Saraclar *et al.*, 2000] a "cheating" experiment on the test data of the Switchboard corpus shows that if a fixed known pronunciation for each word is available, the WER drops from 47 % to 27 %. In practice, using different pronunciation variants for each vocabulary word provides a significant improvement [Hain, 2002], but increases the complexity of the decoder and can lead to biased Viterbi scores for words with a larger number of pronunciation variants.

2.3.2 Trajectory, model assumptions and folding problem

The representation of HMM-GMM in a graphical form, as shown in Figure 2.7, will help to understand the notion of speech trajectory described in this section and many of the concepts described further in this thesis.

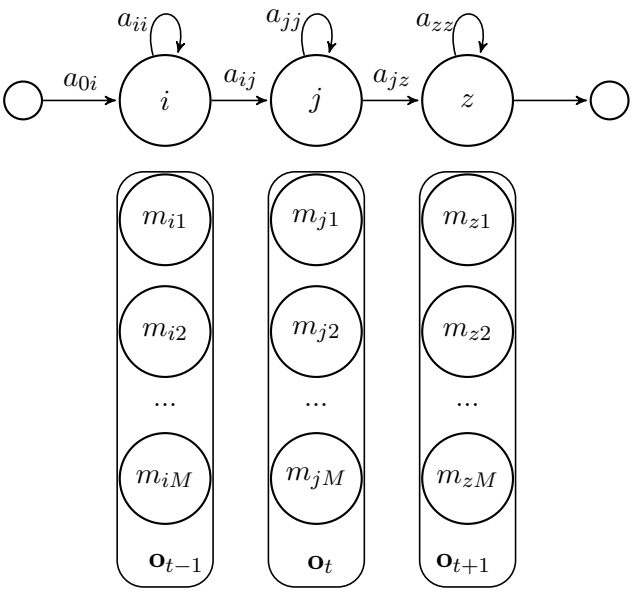

FIGURE 2.7 – Phone HMM with GMM density function

In addition to the sequence of states $\boldsymbol{\mathcal{Q}}$, GMM components can be themselves seen as states with continuous value outputs. The output values of these states depend on the observation vector and on the parameters of each GMM component associated with HMM states. Consider a path through the particular state sequence $\boldsymbol{\mathcal{Q}}$ and the sequence of GMM components $\boldsymbol{\mathcal{M}} = (m_1, \ldots, m_t, \ldots, m_T)$, where each $m_t \in \{1, \ldots, M\}$ denotes the component index of the density associated with time frame t. Further in the thesis such a path (shown schematically in Figure 2.8) is referred to as a *speech trajectory* defined in the acoustic model space.

The joint likelihood of observing the sequence $\boldsymbol{\mathcal{O}}$, a particular state sequence $\boldsymbol{\mathcal{Q}}$ and a component sequence $\boldsymbol{\mathcal{M}}$ is written as follows and simplified by applying the HMM independence

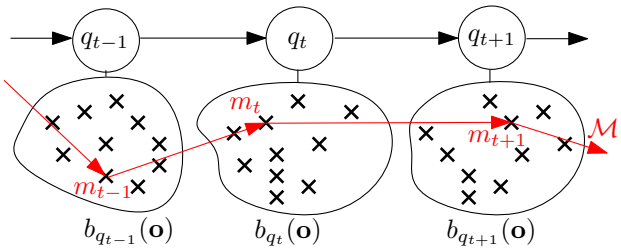

FIGURE 2.8 – Example of speech trajectory in a phone HMM-GMM

assumptions described earlier in Section 2.2.1

$$\begin{aligned} P(\boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\lambda}) &= P(\boldsymbol{\mathcal{O}}|\boldsymbol{\mathcal{M}}, \boldsymbol{\mathcal{Q}}, \boldsymbol{\lambda}) P(\boldsymbol{\mathcal{M}}|\boldsymbol{\mathcal{Q}}, \boldsymbol{\lambda}) P(\boldsymbol{\mathcal{Q}}|\boldsymbol{\lambda}) \\ &= \prod_{t=1}^{T} P(\mathbf{o}_t|m_t, q_t) P(m_t|q_t) P(q_t|q_{t-1}) \end{aligned} \tag{2.22}$$

Summing over all component trajectories leads to a standard likelihood of the state sequence and the observation sequence defined by an HMM

$$\begin{aligned} P(\boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{Q}}|\boldsymbol{\lambda}) &= \sum_{\boldsymbol{\mathcal{M}}} \prod_{t=1}^{T} P(\mathbf{o}_t|m_t, q_t) P(m_t|q_t) P(q_t|q_{t-1}) \\ &= \prod_{t=1}^{T} \left(\sum_{m_t=1}^{M} P(\mathbf{o}_t|m_t, q_t) P(m_t|q_t) \right) P(q_t|q_{t-1}) \\ &= \prod_{t=1}^{T} P(\mathbf{o}_t|q_t) P(q_t|q_{t-1}) \end{aligned} \tag{2.23}$$

In standard HMM-GMM the speech trajectory is hidden, as the pdf defined by $P(\mathbf{o}_t|q_t)$ sums over all components in each state and the observations are independent of each other. In other words, due to the earlier defined output independence assumption of HMM there is no explicit control of trajectory and optimality is not guaranteed. For example, the Viterbi search guarantees to find the best sequence of model states, but the components of GMM are determined only by state-conditioned parameters of Gaussians and the corresponding mixture weights. Various studies discussed the effects, which can be harmful for ASR due to HMM independence assumptions, especially when the data contains different sources of variability (heterogeneous). One such problem is referred to as trajectory folding [Illina and Gong., 1997].

GMM can be seen as a classifier of acoustic features. As different GMMs are associated with different states, each component should in theory represent a given source of variability. For example, consider an HMM with 2 Gaussian components per density. If there are male and female speakers in the training data, it is very likely that the components will be associated with the gender. However, in decoding both components are used for likelihood computation at each frame, as defined in the Equation 2.6. This leads to the folding of all possible trajectories into a single one.

Trajectory folding is an example of problem that appears due to strong conditional independence assumptions of HMM and that causes problems when dealing with heterogeneous data. Next, we will describe some techniques, which allow to do more accurate modeling in such cases.

2.4 Advanced acoustic modeling issues

State-of-the-art HMM-based ASR systems use various techniques to achieve a more accurate modeling accuracy. In this section some of such techniques that are important for further understanding of the thesis are described. First, the concept of efficient parameter sharing is introduced. Then, different types of model adaptation are discussed.

2.4.1 Parameter expansion and sharing

It is desired to build an acoustic model, which is able to represent different sources of variability. To achieve better modeling one can intuitively start thinking about increasing the number of model parameters. For the GMM observation density this simply means increasing the number of Gaussian components. Another idea is to increase the number of HMM state-densities to better capture the contextual variability. To do so, the phone HMM is replaced by *triphone* HMM. Triphone is a unit, which is unique for any possible left and right context of each phone [Lee, 1990]. In practice such direct expansion of both GMM and state space has several problems, which have to be considered:

1. Model learning is more reliable if the model has a small number of parameters. This is especially important if the training data is not large enough
2. Growing up the number of densities without a proper heuristic significantly increases the search space and the computational time. If ASR relies on multi-pass decoding process, the computation time might play an important role
3. Storing the model in memory is now less crucial, as the storage devices are large even for personal computers. However, reducing the model size can still be useful for portable devices

Taking into account these possible problems, it is desirable to find an optimal model structure. On the one hand, this structure should represent many different sources of speech variability. On the other hand, the parameters that are similar across different classes of variability should be shared. Some popular techniques dealing with parameter sharing are further reviewed in this section.

Context-Dependent units and GMM training

A general training pipeline for HMM with *Context-Dependent* (CD) units (phones) with GMM observation densities is described in detail in [Young *et al.*, 1994] and briefly summarized in this section. Training starts from specifying the initial model with *Context-Independent* (CI) units and a single Gaussian component per density. In a *flat start* initialization the state transition probabilities are set uniformly with zeros for not connected nodes. Gaussian parameters are also specified identically. Next, MLE training is done.

To initialize CD states, the CI states and the associated Gaussian densities are copied to cover each possible triphone. The model parameters are then again re-estimated. To train Gaussian Mixture pdf's, a sequential clustering procedure is applied. At each clustering step Gaussian components are expanded (split and perturbed) and re-estimated with MLE after each split. In

theory, such approach assumes very large training data and all triphones frequently observed. In practice, before GMM expansion a state clustering is applied to share states with potentially similar parameters and form *tied states* (or *senones*).

State tying

The tying of CD units is based on a tree-based clustering algorithm [Hwang, 1993; Young *et al.*, 1994]. To build the tree, the approach uses linguistic questions about whether the left/right context of a phone belongs to a certain phonetic class (nasal, fricative, vowel, etc.). After building the tree, the states from each leaf node are combined into tied states, also called senones.

In practice, for LVCSR the number of senones varies from $1k$ to $20k$, depending on the amount of the training data. For comparison, if no tying is used and if we assume all possible triphones, in French language it results in $40^3 = 64k$ possible triphones (considering the set of 40 phones).

Semi-continuous HMM

Another example of parameter sharing is *semi-continuous* (tied-mixture) HMM (SCDHMM) [Hwang and Huang, 1993]. SCDHMM was employed in early versions of the Sphinx recognition toolkit and some other systems. The main idea is to reduce the model by sharing Gaussian mixture parameters (means and variances) across all states, keeping state-dependent sets of mixture weights. Comparing to the continuous GMM density computation (Equation 2.6), the observation density for a SCDHMM at some state j has the following form:

$$b_j(\mathbf{o}_t) = \sum_{l=1}^{M} \omega_{jl} \mathcal{N}(\mathbf{o}_t | \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l) \tag{2.24}$$

SCDHMM seems to be no more as popular, as the modern machines typically handle the continuous HMM computations in reasonable time and there are enough data to estimate density parameters for each model state. However, some recent work demonstrate better results than continuous HMM by introducing some additional dependencies on shared density parameters of SCDHMM (as for example in *Multiple-Codebook Semi-continuous Models [Riedhammer* et al.*, 2012]*)

Subspace GMM

Another efficient parameterization of GMM was proposed in [Povey *et al.*, 2011a; Rose *et al.*, 2011]. The model is called *Subspace GMM* (SGMM). This model is also more compact than HMM-GMM, because the large number of mean vectors is shared over all model states, whereas the parameterization is done by training the projection vectors of low dimension.

SGMM uses state-independent covariance matrix of the *Universal Background Model* (UBM) and computes state-dependent means by linearly projecting the means of UBM. Let us define UBM as a GMM with means and variances $\{\mathbf{m}_l, \boldsymbol{\Sigma}_l\}$, where l denotes the GMM component number. The SGMM mean vector $\boldsymbol{\mu}_{jl}$ for the state j and the component l is computed using *linear subspace projection matrix* $\mathbf{M}_l$ and *projection vector* $\mathbf{v}_j$ for the state j:

$$\boldsymbol{\mu}_{jl} = \mathbf{m}_l + \mathbf{M}_l \mathbf{v}_j \tag{2.25}$$

The mixture weights are computed from the same state-dependent projection vectors using

log-linear model:

$$\omega_{jl} = \frac{\exp \mathbf{w}_l^T \mathbf{v}_j}{\sum_{k=1}^{M} \exp \mathbf{w}_k^T \mathbf{v}_j} \tag{2.26}$$

Finally, the observation density of SGMM is computed using the projected means and mixture weight, and the diagonal elements of the covariance matrix of UBM:

$$b_j(\mathbf{o}_t) = \sum_{l=1}^{M} \omega_{jl} \mathcal{N}(\mathbf{o}_t | \boldsymbol{\mu}_{jl}, \boldsymbol{\Sigma}_l) \tag{2.27}$$

An experimental comparison of continuous, semi-continuous and SGMM on RM task is described in [Riedhammer *et al.*, 2012]. The authors achieved 3.38% *Word Error Rate* (WER) with HMM, 4.66% on SCDHMM and 2.78% on SGMM. The trend is logical, as semi-continuous GMM should perform worse, whereas SGMM frequently outperforms GMM having a smaller number of parameters. SGMM attracts the attention of the modern ASR research community, and this model was recently implemented in the Kaldi ASR toolkit.

2.4.2 Adaptation techniques

It is well-known that *Speaker-Dependent* (SD) ASR systems perform better than the *Speaker-Independent* (SI) ones. At the same time in LVCSR only limited amount of data is available for each speaker even if the speaker identity is known. In such cases, adaptation techniques are very useful. Adaptation consists in modifying the parameters of some general initial model (for example, Speaker-Independent) using a relatively small amount of the adaptation data. The adapted model should not only better represent the adaptation dataset, but also generalize on the unseen data, using the prior knowledge from SI initial model.

Early commercial speech dictation system DRAGON was asking the user to read a sufficient amount of text to adapt the model to the speaker and the microphone (*supervised adaptation*). State-of-the-art ASR systems also rely on modifying the model parameters on-line (*unsupervised* or *semi-supervised adaptation*), improving the performance with time.

To better understand the problem from a technical perspective, let us recall that the conventional MLE training of HMM consists in estimating all model parameters to maximize the likelihood of the observation data (Equation 2.9). A well-known problem of such training is inability of the resulting model to generalize on unobserved data (*overfit*). When starting the derivations of MLE, the data were assumed infinite, although in practice it is not the case.

Assume a large corpus is available for training with many speakers. The speakers are known for training and decoding, but the data available for each speaker is limited. If *Speaker-Dependent* (SD) models are trained with MLE, this would lead to poor estimation. At the same time, even given the variability between different speakers, some of the acoustic features are similar across different speakers. Therefore, SI information might be useful for estimating those model parameters that are not frequently observed from SD set. Such intuitive reasoning forms the basis of most of the adaptation techniques.

Bayesian model adaptation

One of the first applied model adaptation techniques was *Maximum a Posteriori* (MAP) adaptation, based on *Bayesian learning* with prior distribution. It was derived for HMM in the

early 90's [Gauvain and Lee, 1994; Gauvain and Lee, 1992] and still effectively used for adaptation with relatively large amount of data [Deléglise *et al.*, 2009]. MAP adaptation is similar to MLE, except for the fact that prior distribution is used in parameter estimation. This means that instead of maximizing the likelihood of the observed data alone, the following criterion is optimized:

$$\boldsymbol{\lambda}^{MAP} = \arg\max_{\boldsymbol{\lambda}} P(\boldsymbol{\mathcal{O}}|\boldsymbol{\lambda})P(\boldsymbol{\lambda}) \tag{2.28}$$

where $P(\boldsymbol{\lambda})$ denotes prior distribution of the model $\boldsymbol{\lambda}$. In case of HMM the prior distribution is a product of Dirichlet and normal-Wishart densities. These priors guarantee MAP to converge to MLE when the adaptation data set is infinitely large.

Although Bayesian adaptation does not suffer from over-fitting, it still requires a sufficient amount of data for the adaptation. The reason is that the components of Gaussian densities are estimated independently without any constraints. Development of more efficient adaptation techniques, which can take benefit from a small number of the adaptation data, is yet another active field of ASR and Machine Learning research.

Transformation-based model adaptation

When the adaptation data are limited, it is more desirable to share some of the model parameters to estimate. For example, in *Maximum Likelihood Linear Regression* (MLLR) [Leggetter and Woodland, 1994] it is done by estimating only the parameters of a linear transformation. MLLR is frequently applied only to the means of Gaussian densities (although variance re-estimation is also possible).

Let us expand the initial mean vector $\boldsymbol{\mu}_{jl}$ of size n of each state j and component l by adding an offset term μ_0 (frequently equal to 1). Then, we assume the transformed mean vector $\boldsymbol{\mu}_{jl}^{MLLR}$ of the adapted model to be a product of the expanded initial mean vector $\tilde{\boldsymbol{\mu}}_{jl}$ and the transformation matrix $\boldsymbol{\mathcal{A}}_{jl}$ of the size $[n \times (n+1)]$:

$$\boldsymbol{\mu}_{jl}^{MLLR} = \boldsymbol{\mathcal{A}}_{jl} \cdot \tilde{\boldsymbol{\mu}}_{jl} \tag{2.29}$$

The transformation matrix is state- and component- dependent only in theory. In practice it is frequently shared across all states and components of GMM (thus $\boldsymbol{\mathcal{A}}_{jl} = \boldsymbol{\mathcal{A}}$, $\forall j, l$) or across some classes of acoustically similar phones. This makes possible to use only a limited number of utterances for adaptation. To learn the transformation matrix, EM algorithm is used to maximize the likelihood of the adaptation data, computed with the transformed model.

Some modifications of MLLR include constrained MLLR (CMLLR), where the transformation matrices for means and variances are constrained to be the same [Digalakis *et al.*, 1995]. In some recently modifications of CMLLR [Povey and Yao, 2012], the number of estimated coefficients can vary depending on the available data, which leads to improvement of the adapted models given only a few seconds of speech data.

MAP and MLLR are widely applied in state-of-the-art ASR systems. MLLR is typically used when the adaptation data are limited. On the other hand, MAP outperforms MLLR when the data are large enough. Finally, the combination of these two methods usually performs best; transformation matrices are estimated with MLLR and then MAP is applied starting from the transformed models.

Feature transformation

Another approach relies on finding a transformation of the features instead of the model parameters. One example of such feature adaptation is *Vocal Tract Length Normalization* (VTLN) [Panchapagesan and Alwan, 2009; Kim *et al.*, 2004; Cohen *et al.*, 1995]. The intuition of the method is that a large portion of speech variability comes from differences in the length of the vocal tracts of different speakers.

The normalization of the vocal tract length consists in warping the features in the frequency, or in the cepstral domain. The task of estimating the shape and length of the vocal tract from speech data is still an active research direction, and it is indeed a hard task [Lammert *et al.*, 2013; Wakita, 1973]. So, in practice, the vocal tract transformation is assumed to be in the form of piece-wise-linear function. In this case the warping factor α_w is the only parameter to estimate. The optimization is done to maximize the likelihood of the warped observation sequence $\boldsymbol{\mathcal{O}}_\alpha$:

$$\alpha'_{\alpha_\omega} = \arg\max_{\alpha_w} P(\boldsymbol{\mathcal{O}}_{\alpha_w}|\boldsymbol{\lambda}) \tag{2.30}$$

VTLN is also frequently combined with MLLR and MAP to achieve better speaker modeling in both model and feature domains.

Discriminative learning

In MAP, MLLR and VTLN the likelihood of the observation sequence is used to derive the model parameters. The likelihood-based estimation techniques usually converge faster, but the likelihood only indirectly represents the actual task of the model, which is to predict the phone sequence.

Discriminative training aims to optimize the model parameters with respect to the output of the classifier (phones or words for ASR case). Some examples are Maximum Mutual Information (MMI) [Bahl *et al.*, 1986; Woodland and Povey, 2002; Liu *et al.*, 2011], Minimum Classification Error (MCE) [Juang *et al.*, 1997; McDermott *et al.*, 2007], Minimum Word Error and Minimum Phone Error (MPE) [Povey and Woodland, 2002; Gibson and Hain, 2010]. In HMM LVCSR boosted MMI is also applied [Povey *et al.*, 2008].

To facilitate training and comparison of the experiments described in this thesis, discriminative training is not considered.

2.5 Conclusion

This chapter summarized the key aspects and basic techniques of statistical speech recognition. The HMM-based approach has a long history of research, which led to the appearance of various techniques that allow to significantly improve both computation efficiency and the modeling accuracy. The model combines together different sources of information: the acoustic features, the phonetic dictionary and the language. At the same time, efficient training and decoding algorithms based on dynamic programming made application of HMMs possible in early ASR systems of 80's.

Although strong conditional independence assumptions and trajectory folding make the model not robust to speech variability and data mismatch, various techniques have been proposed to address these issues. For example, using Gaussian Mixture observation pdf in combination with efficient parameter sharing and model adaptation has become essential in state-of-the-art

HMM-based ASR systems. Nevertheless, the problem of accurate acoustic modeling with highly heterogeneous data is not yet solved and greatly discussed in the remainder of the thesis.

3

Class-based and trajectory modeling

The notion of *speech trajectory* was defined in Section 2.3.2 as a time-dependent path through the components of Gaussian mixture observation densities. In standard training and decoding formulation, such a trajectory is hidden and all components are used in the same way for likelihood computation because of the strong independence assumptions of HMM. At the same time, the same phonetic units spoken by different speakers lead to significantly different acoustic features. As a result, the component trajectories associated with the same sequence of states but produced by different speakers will significantly differ as well. Conventional HMM is incapable to accurately model such differences and typically leads to the folding of all trajectories and performance degradation when dealing with heterogeneous data. This section reviews two broad classes of state-of-the-art techniques, which aim to achieve more accurate modeling of speaker trajectories in some sense.

Contents

The first set of techniques (discussed in Section 3.1) attempts to introduce speaker class information in the modeling. A straightforward implementation of this idea is done in *multi-modeling* (or *class-based*) ASR. In class-based ASR the temporal trajectory is not modeled directly, but the GMM parameters are adjusted according to the class of the data. Class-based approach is equivalent to using separate HMMs for several homogeneous subsets of data. The homogeneous subsets are either associated with a given speaker variability class (gender, age, etc.) or are

estimated by unsupervised clustering of the data. A *model selection* approach is then used in decoding. Alternatively, in *speaker-space* modeling all class models are used to adjust the resulting model parameters to each speaker separately.

The second broad class of techniques (reviewed in Section 3.2) relies on *explicit trajectory modeling*, which significantly differs from the concept of multi-modeling or speaker-space approaches. These techniques attempt to directly model the local trajectory in decoding, or in the classification process. The trajectory is explicitly represented by introducing additional temporal dependencies in model or in feature space. The objective of these methods is generally to relax some of the HMM conditional independence assumptions and to better parameterize speech trajectories.

3.1 Multi-modeling approach for ASR

Consider first a *Speaker-Independent* (SI) acoustic model trained from heterogeneous data (including different speakers and recording conditions). States of the phone HMM cover the temporal parameters (duration) and allow to introduce the language model probabilities, whereas GMM represents the distribution of the acoustic features associated with each state. As it was mentioned in Section 2.3.1, the acoustic features represent both useful information (to distinguish phones) and information that is irrelevant for phone discrimination (speaker, channel and contextual variability).

The *acoustic space* is defined as the space of all possible values of the acoustic features. GMM approximates the acoustic space by combining several multivariate Gaussian densities. Figure 3.1 schematically represents the HMM, where crosses denote the means of Gaussian densities on the acoustic space associated with each state.

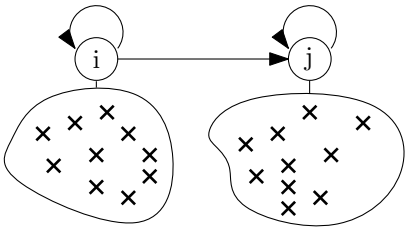

FIGURE 3.1 – Schematic representation of HMM-GMM

Indeed, GMM components carry more information, than just the distribution of the acoustic features for a given phone. Implicitly, these components represent the speaker (or speaker class) characteristics, as well as other non-phonetic variability sources. However, these sources of information are hidden and most of the time not used. For a *Speaker-Dependent* (SD) system working in a fixed and clean recording environment a smaller number of components per density is typically required to accurately represent the distribution of the acoustic features of the phones compared to an SI system covering different speakers and recording conditions, as the speaker variability is reduced. If the speakers are unknown, the variability can also be reduced by unsupervised clustering at the segment level.

The remainder of this section is organized as follows. First, Section 3.1.1 describes the general idea of multi-modeling approach. Then, Section 3.1.2 focuses on two particular algorithms for unsupervised speech data clustering that are further experimentally analyzed and compared in Section 3.1.3. Next, Section 3.1.4 reviews other state-of-the-art speech data clustering techniques. Then, Section 3.1.5 discusses hypothesis combination approach for class-based ASR. Finally,

Section 3.1.6 presents some of the speaker-space models as another way to efficiently use class-based models. The section ends with a short conclusion.

3.1.1 Introduction to multi-modeling approach

Multi-modeling approach relies on a prior knowledge about classes of the training speech segments. Classes can represent speaker age, gender, accent, channel type, etc., or can be constructed by unsupervised clustering of the training data. Separate acoustic *class-based models* are constructed for each variability class by adapting the parameters of a general SI model. Later in the thesis such model is referred to as *HMM with Class-Based GMM*, or simply *Class-Based HMM* (CB-HMM) In decoding the best class-based model is selected for each segment according to knowledge of the class of the segment to be decoded or as the result of an automatic classification. Such approach is referred to as multi-modeling, or class-based with *model selection*.

CB-HMM is schematically represented for the case of two classes C_1 and C_2 in Figure 3.2. Generally, the number of classes is bigger than 2 and it is mainly determined by the amount of data and by the number of different sources of variability. In this example the acoustic space is subdivided by 2 subspaces that can overlap. The class-dependent acoustic models typically represent more homogeneous subsets of data, and this leads to smaller variance. As only one of these models is selected for decoding, the components adjusted on the irrelevant class are not used. Therefore, the trajectory is not mixed between two separate classes of data any more.

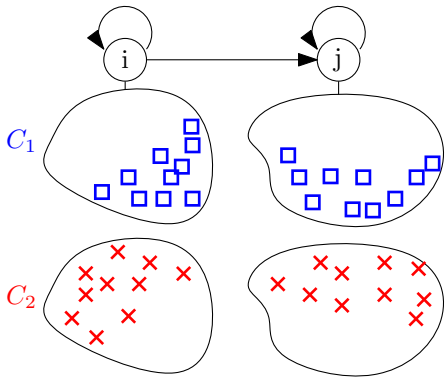

FIGURE 3.2 – HMM with Class-Based HMM for 2 two speaker classes

One problem associated with this approach is how to subdivide the data into classes of homogeneous data. In some training data speaker-related information is available (like gender in NIST evaluation campaigns or in French radio broadcast transcription challenges [Godfrey *et al.*, 1992; Gravier *et al.*, 2012]). Gender-dependent models significantly improve the accuracy compared to the SI system [Illina *et al.*, 2004]. Other types of such a-priori knowledge include the accent (origin of the speakers), the age, the rate of speech, etc. If no prior speaker information is available, or to go beyond traditional gender-dependent classes (i.e. build more classes of variability), unsupervised clustering is applied.

3.1.2 Unsupervised clustering of the speech segments

The idea of automatic clustering for ASR is to find classes of data with similar acoustic characteristics. Assuming that within a speech segment the speaker does not change, the clustering is done at the segment level. The classifier can be formalized as a function f_c, which

maps the elements of the set of segments $\mathbf{U} = \{u_1, ..., u_Z\}$ to elements of the set of class labels $\mathbf{C} = \{c_1, ..., c_R\}$, or $f_c : \mathbf{U} \rightarrow \mathbf{C}$

Together with the acoustic features (observations) associated with the segments $\mathbf{U}$, additional information can be available and used for improving classification. An example of such information is text transcriptions for performing force alignment and extracting phonetic information. Other features that might be used in classification are segment length, speaking rate, SNR level, etc. In the remainder of this section two different approaches for segment classification are described in detail. The first one relies on *ML criterion* and associates a single GMM per class independent on the phonetic content (for simplicity, this approach is referred to as *ML-based classification*). The second approach uses *KL divergence* computed on posterior distribution given a class-independent HMM-GMM (and will be referred to as *KL-based classification*). After describing in detail KL and ML-based classification methods some other techniques for clustering and classification are reviewed.

GMM-based Maximum Likelihood classification

GMM is widely applied for clustering and classification in various Machine Learning tasks. For example, early text-independent speaker recognition and speaker verification systems were successfully using GMMs [Reynolds *et al.*, 2000] for recognizing a speaker from a given set of speakers or for verifying the speaker's identity based on a spoken sentence. The simplest approach for speaker recognition relies on training speaker-dependent GMMs from speaker-labeled training data and estimating the likelihood of the test data with these GMMs. The resulting speaker identity corresponds to the GMM that leads to the maximum likelihood.

A similar idea is applied in GMM-based speech data clustering based on Maximum-Likelihood criterion (denoted as *ML-based clustering*). The objective is to automatically build classes of acoustically similar data regardless of the phonetic content in order to use these data to build *Class-Based HMMs* (CB-HMMs) via adaptation. The application of this technique for class-based ASR is described in [Jouvet *et al.*, 2012b]. Let us review this algorithm in detail. General blocks of the algorithm are shown in Figure 3.3.

First, the standard MFCC acoustic features with first and second derivatives are derived (as described in Section 2.2.1) and the parameters of a single GMM $\mathbf{\Phi}$ with a large number of components $(256, \ldots, 1024)$ are estimated from the full training set with MLE. Next, the GMM is duplicated and the mean values are perturbed ("small shift"). The value of this shift is determined by the component variance σ as $\pm 0.2\sigma$. The resulting GMMs $\mathbf{\Phi}_1$ and $\mathbf{\Phi}_2$ are used for splitting the training data into 2 classes: c_1 and c_2.

To assign a segment u to a class c_k, the likelihood of the observation data of the audio segment $\boldsymbol{\mathcal{O}}_u$ given each class GMM $\mathbf{\Phi}_k$ is computed. The class of the segment is determined by the GMM, which leads to the maximum of the likelihood function. Given R classes and associated GMMs, the assignment criterion is defined as follows:

$$u \in c_k \Leftrightarrow P(\boldsymbol{\mathcal{O}}_u|\mathbf{\Phi}_k) \geq P(\boldsymbol{\mathcal{O}}_u|\mathbf{\Phi}_l), \qquad \forall l \in \{1, \ldots, R\} \tag{3.1}$$

After this class assignment step, the data of each class are used for re-training the class-associated GMM. The classification-training process is repeated until convergence. If the desired number of classes is not achieved, then each class-associated GMM is again duplicated and the classification-training algorithm is again repeated.

The same classification GMMs are used in decoding to identify the class for selecting the best model for each segment of the test set. This means that the same assignment step, as in

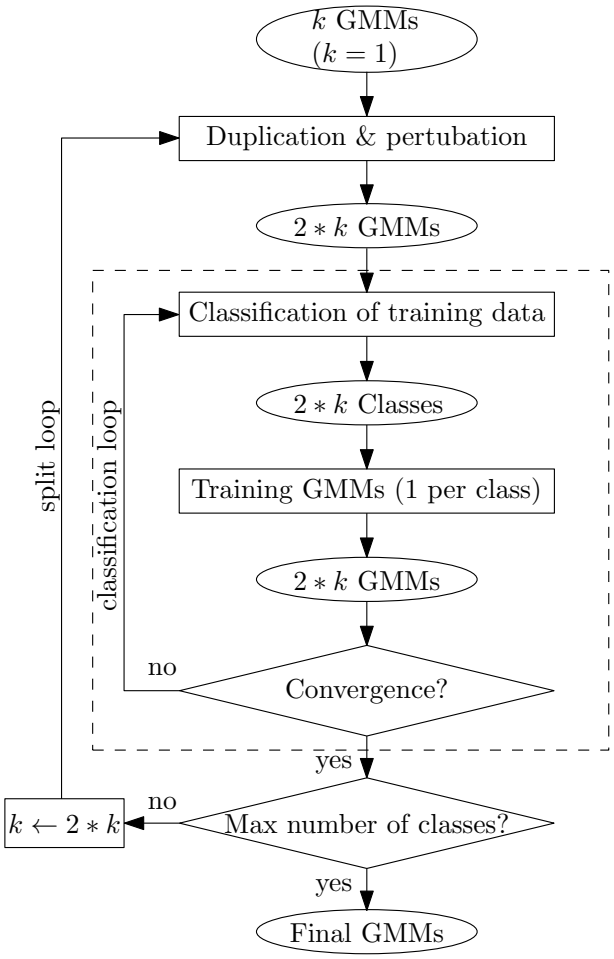

FIGURE 3.3 – GMM Maximum-Likelihood clustering (or simply ML-based clustering)

Equation 3.1 is performed using already trained class-associated GMMs and the observation data of a segment to be decoded. The acoustic model associated with the class leading to the maximum of the likelihood function is used for decoding.

Although clustering of the speech segments is not exactly equivalent to speaker clustering, here and later it is assumed that the main source of variability (estimated at the segment level) comes from the speaker and the described clustering process will be referred to as *speaker clustering* and the resulting classes of segments as *speaker classes*. The experiments also verify this assumption by showing that in the clean recording conditions the segments corresponding to a given speaker are usually assigned to the same class.

The advantage of such GMM-based classification strategy is that no transcription is required to classify the segments. This is an important property, because there is no transcript available for the test data and if a classification method relies on the transcript, an additional decoding pass is required to generate a recognition hypothesis before classification. The disadvantage of the described *ML-based* method is that each data segment is treated equally, as the same GMM is used independently on the phonetic content of the segment. As different phones have different acoustic features, the phonetic variability can impact on the classification results.

Kullback-Leibler classification using phone-dependent GMMs

The distribution of the acoustic features highly depends not only on a speaker class, but also on the phonetic content of the spoken phrase. Therefore, it is desired to use phone-dependent parameters in the clustering and classification algorithms. It was proposed in [Mao *et al.*, 2005] to cluster segments using a general Speaker-Independent (SI) HMM-GMM and classify the segments based on analysis of Gaussian components that are activated in the Baum-Welch posterior computation. *Kullback-Leibler* (KL) divergence measure is used to compare the posterior probabilities estimated from the data to be classified (a segment) and from the data associated with a class. In contrast to earlier described *ML-based* clustering with a single GMM per class, in the proposed technique GMMs of the classifier are phone-dependent. An efficient implementation of this method in a distributed tree-based clustering architecture was described in [Beaufays *et al.*, 2010] and applied for LVCSR demonstrating significant improvements of the accuracy. This section briefly describes the main concepts of this method.

The segment classifier is constructed by concatenating the associated phone HMMs with GMM observation densities. To reduce the number of parameters, in the following formulation of the algorithm and in the corresponding experiments a phone is modeled by a single state. Given N phones and M Gaussian components per density, for each training segment u the averaged over time posterior probabilities are computed for each state $j \in \{1, \ldots, N\}$ and component $l \in \{1, \ldots, M\}$ as follows:

$$\omega c_{jl}^{u} = \frac{1}{T_{uj}} \sum_{t=1}^{T_{uj}} \frac{\omega_{jl} \mathcal{N}(\mathbf{o}_t | \boldsymbol{\mu}_{jl}, \boldsymbol{\Sigma}_{jl})}{\sum_{k=1}^{M} \omega_{jk} \mathcal{N}(\mathbf{o}_t | \boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk})} \tag{3.2}$$

where $\mathcal{N}(\mathbf{o}_t | \boldsymbol{\mu}_{jl}, \boldsymbol{\Sigma}_{jl})$ denotes the Gaussian probability computed with mean vector $\boldsymbol{\mu}_{jl}$ and covariance matrix $\boldsymbol{\Sigma}_{jl}$ and $\mathbf{o}_t$ denotes the observation vector at time frame t from the interval $(1, \ldots, T_{uj})$, which corresponds to all observed occurrences of phone j of the segment u.

Computing the segment weights ωc_{jl}^{u} for the full training set is a time-consuming procedure. However, they are calculated only once and then they are not modified throughout the whole clustering procedure.

To build R classes of data, the following procedure is applied. The idea is to use the computed weights ωc_{jl}^{u} in order to minimize the distance between each segment and its corresponding class by applying *Lloyd iterations*. To apply an iterative classification procedure, initial classes and a similarity measure must be defined. Several approaches are possible for defining initial classes:

- Start from random classes. This is the easiest way, but not the most efficient one;
- Apply *Principal Components Analysis* (PCA) increasing the number of classes by splitting the data set along the largest variance direction, as in [Beaufays *et al.*, 2010];
- Start from classes obtained *from GMM ML-base classification*, as described in the previous section. This initialization method was applied in [Gorin and Jouvet, 2012] and it is used in all experiments with KL-based clustering described in Chapter 4.

Given a set of R initial classes $\mathbf{C} = (c_1, \ldots, c_R)$ and the corresponding set of class-labeled segments, the *KL-based* clustering algorithm proceeds as follows. First, the weights ωc_{jl}^{c} are derived for each phone j, class c and Gaussian component l. To do this, the weights ωc_{jl}^{u} are summed over all segments assigned to the class c, and then normalized with respect to the number of segments that contain phone j. For example, assume for some class c with the associated segments $u_1, \ldots, u_Z$ the phone j appears in N_u segments (in the ideal case $N_u = Z$). Then, the

weight associated with this phone, class c and component l is computed as follows:

$$\omega c_{jl}^{c} = \frac{1}{N_u} \sum_{z=1}^{Z} \omega c_{jl}^{u_Z} \tag{3.3}$$

The acoustic dissimilarity between a segment u from the training set and a class c for a phone j is defined by Kullback-Leibler measure

$$D(p_j^u || p_j^c) = \sum_{l=1}^{M} \omega c_{jl}^{u} \log \frac{\omega c_{jl}^{u}}{\omega c_{jl}^{c}} \tag{3.4}$$

For capturing inter-speaker variability (age, gender, etc.) the acoustic dissimilarity is measured at the segment level and not at the phone level. Let us assume that all phones equally contribute into the resulting segment-level divergence measure. In this case the divergence between a segment u and a class c is computed by averaging the phone-level divergence values (computed by Equation 3.4) over all phones, which were observed in both the segment u and in the class c. If for a segment u and a class c N_g phones were observed, then the total distance is computed as follows:

$$D_{Tot}(p^u || p^c) = \frac{1}{N_g} \sum_{j=1}^{N} D(p_j^u || p_j^c) \tag{3.5}$$

Finally, a segment u is assigned to a class c_k, which leads to the minimum divergence criterion

$$u \in c_k \Leftrightarrow D_{Tot}(p^u || p^{c_k}) \leq D_{Tot}(p^u || p^{c_l}) \qquad \forall l \in \{1, \ldots, R\} \tag{3.6}$$

The following steps of the algorithm are repeated until convergence:

— re-classify the data (Equation 3.6)
— re-estimate the class parameters (Equation 3.3)
— re-compute the class distances (Equations 3.4-3.5)

The advantage of the described KL-based classification is that it takes into account the fact that the distribution of the acoustic features depends not only on the class, but also on the phones. The disadvantage of the method is that, unlike the ML-based classification with a single GMM per class, a transcript is required to estimate the posterior weights for a segment before classification. This means that for the test set, an additional decoding pass is required to generate a hypothesis and to use it for the classification.

More sophisticated methods for computing the final distance for the segment (Equation 3.5) are possible. Instead of simply averaging the phone-level divergence, different phones could be treated differently. For example, some phones can be simply discarded (like less-informative consonants or rarely observed phones). Some preliminary experiments show that using only 2/3 of selected phones in classification lead to a similar performance of the resulting class-based ASR (see Appendix C for details). Another possibility is to replace the average by a weighted sum of phone divergences, thus allowing different phones contribute differently into the resulting segment divergence measure. Such modification is not considered in this thesis, because a separate study is needed to understand how the phone weights should be selected and whether it leads to significant improvement of classification.

3.1.3 Experimental analysis of clustered speech data

This section describes some experiments with unsupervised clustering of speech data. First, the experiments are conducted with clean read speech in order to understand, what kind of speaker variability can be handled by such clustering. Then, class-based ASR experiments are conducted on radio broadcast data in order to evaluate performance improvements and compare *ML-based* and *KL-based* algorithms described in the previous section.

Analysis of some speaker classes after unsupervised clustering

To get an idea on which types of speaker variability are captured by the described techniques, we report here some results obtained when applying the unsupervised clustering on the TIDIGITS training data. The data contain clean digit sequences recorded from speakers of different age groups and gender (see details in Appendix A.2.2).

In this section, the results of ML-based clustering with a single GMM per class are reported. The results of the phone-dependent KL-based clustering provide similar results. The classification GMMs consist of 256 Gaussian components and the acoustic features correspond to standard 12 MFCC + Log Energy + Δ + $\Delta\Delta$ vectors. The resulting distributions of age and gender over 2, 4 and 8 resulting classes are summarized in Figure 3.4.

FIGURE 3.4 – Number of training segments of each age and gender in the resulting 2, 4 and 8 *ML-based classes* of *TIDIGITS* training data

The first clustering step (2 classes) mainly splits male speakers from female and child speakers. The second split (4 classes) allows to separate female speakers from child speakers. It seems impossible to distinguish boys from girls, even with more classes (from 8 up to 32). Similar behavior was observed for KL-based classes.

The conclusion of this analysis is that in clean recording conditions it is possible to build gender and age dependent systems without prior knowledge about the speakers. Although further analysis of which sources of variability are captured by larger number of classes is possible, we rather focus on applying the clustering technique for improving ASR performance.

Comparison of class-based ASR performance with ML-based and KL-based clustering

So far two methods of unsupervised data clustering for ASR were described. The first method relies on a single GMM per class and uses ML criterion (*ML-based clustering*). The second method uses phonetic HMM and KL divergence measure (*KL-based clustering*). In this section a set of LVCSR experiments is described to investigate the proposed approaches in terms of ASR accuracy of the resulting class-based models. The experiments are based on the Sphinx ASR toolkit and diarization tools of LORIA laboratory.

The baseline CD-HMM baseline (*mdl.LVCSR.4500s.StTel*[1]) is based on the LORIA news transcription system trained using radio broadcast recordings containing about 190 hours of speech (*ESTER2.Train* data described in Appendix A.1.1).

The experiment starts from unsupervised clustering of the training data up to 32 classes (the optimal number of classes can only be determined by conducting experiments with different number of classes). ML-based classifier consists of 256 Gaussian components per class and KL-based classifier consists of phone-dependent GMMs with 64 components each. After clustering, the class-based models are constructed by adapting Gaussian mean values of the baseline models with the class-associated data using MLLR.

The evaluation is done on the non-African radios of the development and test data of the ESTER2 evaluation campaign (*ESTER2.Dev.11f* and *ESTER2.Test.17f*) (the details of these datasets are described in Appendix A.1.2). The *Word Error Rates* (WER) associated with *Speaker-Independent* (SI), *Gender-Dependent* adapted with MLLR (GD) and *Class-Based* models adapted with MLLR using *ML-based* and *KL-based* clustering are reported in Figure 3.5 with respect to the number of classes (models) used.

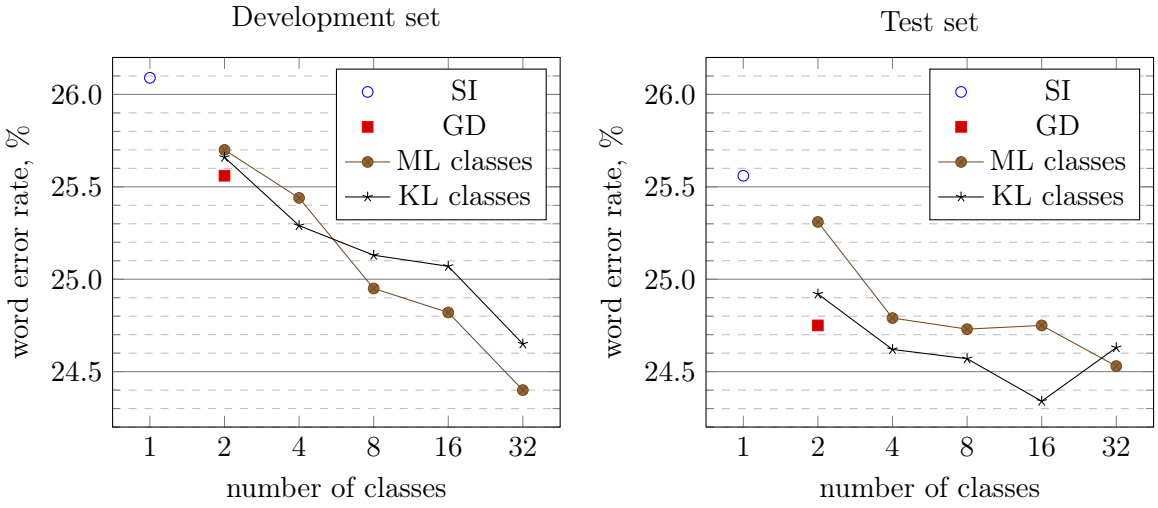

FIGURE 3.5 – Comparing WER for *Speaker-Independent* (SI), *Gender-Dependent* (GD) and *Class-based* models built with ML-based and KL-based classes and MLLR

Overall, the two methods provide similar results on these data. With a larger number of classes the recognition accuracy improves (at least up to a certain number of classes). The overall gain from traditional GD models is statistically significant starting from 8 classes (except for 32 KL

1. 4500 shared densities (senones), 64 Gaussian per density, 39 cepstral features (MFCC+Δ+ΔΔ), separate models for Studio/Telephone quality data (see details in Appendix B.1)

classes for test data).

The method has its limitations, as for example 32 KL-based classes lead to a degradation of the WER on test data. A problem to consider is that growing up the number of classes the number of segments associated with each class is decreased. If not enough data are available for some classes, the corresponding acoustic models are not accurately estimated, which leads to performance degradation of the resulting class-based system.

3.1.4 Some other unsupervised clustering techniques for class-based ASR

In addition to the described GMM ML-based and HMM KL-based clustering techniques, other unsupervised clustering techniques can be considered for class-based ASR. Some of these techniques are reviewed in this section.

Recent works achieve efficient clustering by using *Joint Factor Analysis* (JFA) segment representation with *identity vectors* (i-vectors) [Zhang *et al.*, 2011]. I-vector representation of the speech is widely used in speaker recognition and verification tasks [Dehak *et al.*, 2011]. In this approach the GMM supervector $\mathbf{M}_c$ representing a segment class c (speaker or/and channel variability) is modeled as a transformation of a supervector $\mathbf{M_0}$ of the *Universal Background Model* (UBM) using a normally distributed vector $\mathbf{w}_c$ and a low rank matrix $\mathbf{T}$:

$$\mathbf{M}_c = \mathbf{M_0} + \mathbf{T}\mathbf{w}_c \tag{3.7}$$

The UBM supervector $\mathbf{M_0}$ is obtained by concatenating UBM component mean vectors trained from all data; $\mathbf{T}$ is a fixed but unknown $[MF \times R]$ matrix, representing total variability (eigenspace). Here M denotes the number of GMM components, F is the dimensionality of the feature vector and R denotes the number of eigenvoices (basis supervectors of eigenspace); $\mathbf{w}_c$ is a $[R \times 1]$ vector, which models the coordinates of the class c in the total variability space. Generally $R \ll MF$.

In speaker recognition the training segments are annotated with speaker labels. Therefore, the i-vectors are learned for each speaker from the corresponding observation data. In clustering task the speaker is unknown, but this representation can be used to represent the speaker classes. The classes are achieved by a clustering algorithm similar to the KL-based approach described earlier in Section 3.1.2.

For example, in [Zhang *et al.*, 2011] the i-vectors are built for each segment and a similarity measure is defined for a pair of segments u_i and u_j as the *cosine similarity* of the normalized i-vectors $\hat{\mathbf{w}}_i$ and $\hat{\mathbf{w}}_j$:

$$sim(u_i, u_j) = \hat{\mathbf{w}}_i^T \hat{\mathbf{w}}_j \tag{3.8}$$

A similar approach was applied in [Wang *et al.*, 2011] for eigenvoice clustering with *Cross Likelihood Ratio* (CLR) used as a similarity measure. CLR is widely used for comparing speaker models in diarization tasks [Le *et al.*, 2007]. In conventional diarization approach, for two classes of data c_i and c_j this measure is defined as follows:

$$CLR(c_i, c_j) = \frac{1}{T_i} \log \frac{P(c_i|\boldsymbol{\lambda}_j)}{P(c_i|\boldsymbol{\lambda}_B)} + \frac{1}{T_j} \log \frac{P(c_j|\boldsymbol{\lambda}_i)}{P(c_j|\boldsymbol{\lambda}_B)} \tag{3.9}$$

where T_i and T_j denote the number of frames in the corresponding clusters, $P(c|\boldsymbol{\lambda})$ defines the acoustic likelihood of the data from a class c given a model $\boldsymbol{\lambda}$. Class models are denoted as $\boldsymbol{\lambda}_i$, $\boldsymbol{\lambda}_j$ and background model is denoted as $\boldsymbol{\lambda}_B$. The authors integrate i-vector modeling into CLR framework and apply it for segment clustering.

In another work [Fukuda *et al.*, 2012], the authors demonstrated a flexible tree-based classification framework, in which non-parametric features representing some non-phonetic information about the segment (for example, SNR, speaking rate, duration, etc.) are used for clustering. The approach consists of the following steps, performed at each node of the tree:

1. Split the data associated with the node into two subsets using one of the features;
2. Compute acoustic similarity between the two resulting classes of data. The authors use cosine similarity measure, but Kullback-Leibler divergence can be used as well;
3. Repeat step 1-2 for all non-parametric features and select the one that leads to maximum dissimilarity criterion;
4. Repeat steps 1-3, increasing the classification tree until the desired number of classes is achieved.

The method of [Fukuda *et al.*, 2012] is complementary to the one used in [Beaufays *et al.*, 2010], because it takes advantage of non-parametric features. It might be useful to combine these two approaches. However, experimental comparison of these two methods has not been done. Another advantage of this method is that it allows to understand, which features are more significant for the classification.

This ends the short survey on different speech data clustering and classification techniques. The goal of all these methods is to split the training data into some acoustically similar classes (*clustering task*) and to be able to find a similar class (or set of classes) for a given speech segment (*classification task*). Yet another problem to solve is how to more efficiently use the clustered data.

3.1.5 About model selection and hypothesis combination

So far various methods of speech data clustering have been described without paying much attention on how the classes are used in the decoding phase. In decoding, a segment is represented by a speech signal that comes from an unknown class. Here and later it is assumed that the data are processed by a segmentation algorithm and that the resulting short segments mostly consist of speech without music and other long non-speech events. Thus, it is assumed that the speaker does not change within a segment (segment). This assumption is usually correct, as the accuracy of the speaker change detection by the state-of-the-art systems is relatively high.

Up to now, it was considered that in decoding the best corresponding class is chosen for each sentence using the same distance measure as the one used for clustering of the training data. This strategy is referred to as *model selection* and it is indeed the simplest way to implement the multi-model ASR. Given a segment from the test set, a set of class-based models and some distance (or similarity) measure, various recognition strategies can be considered:

— Select the best model by using the measure, defined for clustering (ML, KL, CLR, i-vector distance, cosine similarity, etc.) and use this model for decoding. This is the simplest and fastest method;

— Do recognition with all models and pick the one that provides the best acoustic likelihood and pick the corresponding hypothesis as an output of the system. Such an approach relies on multiple parallel decodings, so it is computationally expensive, but parallelizable. Also, such posterior-based hypothesis selection is more accurate than the classification-based selection;

— Perform recognition with all models and combine the resulting hypotheses with a voting system, also known as *Recognizer Output Voting Error Reduction* (ROVER) [Fiscus, 1997]. Performance resulting from hypotheses combination is greater if the errors of the

recognizers are different. In practice, it is efficient to combine the systems that are based on different acoustic and language models, use different features and ASR engines [Jouvet and Fohr, 2013a]. Combining the hypotheses resulting from decoding using different class-based models also leads to significant improvements of WER [Gorin and Jouvet, 2012]. This approach is computationally similar to the second one, whereas it can provide better results (see a detailed analysis in Section 4.3);

— As the methods 2 and 3 rely on multiple hypotheses, it is desirable to decrease the number of models to use in decoding. This can be done by firstly *choosing N best models* according to the classification distance measure, as in the 1st method. Then, the N hypotheses are computed using these models. Finally, these hypotheses are combined with voting system;

— Perform a single decoding using the model constructed from several class-based models. This approach is described in greater detail in the following section.

3.1.6 Speaker-space models as implicit speaker trajectory representations

Class-based approach described so far, has several disadvantages:

— *Hard clustering* of the training data implies that each training segment belongs to exactly one class and does not provide any information about the uncertainty of the classification decision (i.e., the classification decision returns only the class label without an associated probability);

— The model selection strategy applied in classification of the segments to be decoded assumes that all models, except for the one that is selected, are irrelevant and not used. At the same time, some speakers can be better represented by a combination of several models;

— The parameters of different class-based models are estimated independently of each other without any parameter sharing. Therefore, the number of parameters to estimate grows linearly with the number of classes, which can become too large for a reliable estimation of the class-based models

This section describes a class of so-called *speaker-space models*, which aim to handle some problems of the hard clustering and of the model selection approach. These techniques have a strong relation to the idea of *class-structured GMM*, which is introduced in the second part of this thesis and plays an important role in this work.

The key idea of speaker-space models is to represent each non-phonetic source of variability (i.e. speaker or environment) as a combination of different classes, or eigenvoices [Gales and Young, 2008]. Figure 3.6 schematically represents the combination of R class-based models with class weights $\{\nu_1^{(sm)}, \ldots, \nu_R^{(sm)}\}$, belonging to a speaker model sm. The combination is done by interpolating the mean vectors of Gaussian densities, while the covariance matrices are shared across different clusters. Specifically, to obtain the mean vector $\hat{\boldsymbol{\mu}}^{(sm)}$ for a speaker model sm given the mean vectors of class models (eigenvoices) $\{\boldsymbol{\mu}^{(1)}, \ldots, \boldsymbol{\mu}^{(R)}\}$ the interpolation is done as follows:

$$\hat{\boldsymbol{\mu}}^{(sm)} = \sum_{r=1}^{R} \nu_r^{(sm)} \boldsymbol{\mu}^{(r)} \tag{3.10}$$

Depending on how the interpolation weights $\nu_r^{(sm)}$ are estimated and how the reference models are obtained, there are various types of speaker-space models in state-of-the-art ASR research, although the idea of mean interpolation remains the same.

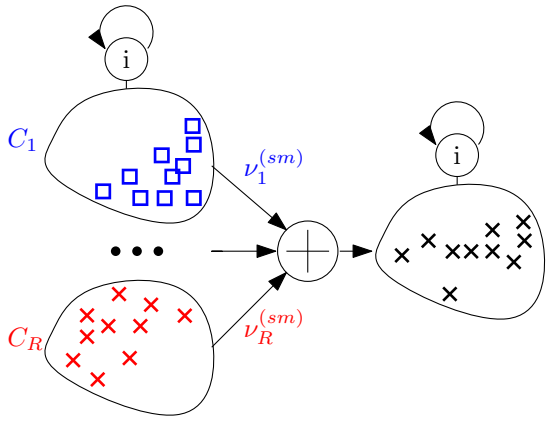

FIGURE 3.6 – Speaker-space model combination of Gaussian mixtures

For example, in the original eigenvoice approach [Kuhn *et al.*, 2000] reference models are represented by eigenvectors from *Principal Component Analysis* (PCA). In *Reference Speaker Weighting* [Mak *et al.*, 2006; Hazen and Glass, 1997] reference models are the means of GMMs of SD models.

In *Cluster Adaptive Training* (CAT) [Gales, 2000] the reference model (called *canonical model*) is generated in an adaptive training manner. Adaptive training starts from an SI canonical model. Then, the speaker-dependent interpolation weight vectors are iteratively estimated for each speaker and the canonical model is re-estimated given these weight vectors. Such re-estimation of the interpolation weights and update of the canonical model is repeated until convergence.

Another example of speaker-space modeling is *Reference Model Interpolation* (RMI), which uses a posteriori selection of reference models before interpolation [Teng *et al.*, 2009].

Speaker-space can be modeled not only in GMM, but also in *Subspace GMM* (SGMM), described in Section 2.4.1. To do this, the computation of the SGMM mean vectors (defined by Equation 2.25) is modified to include the projection of the speaker vector $\mathbf{v}^{(sm)}$ on the speaker subspace matrix $\mathbf{N}_l$ of the subspace l, similar to eigenvoice model:

$$\boldsymbol{\mu}_{jl}^{(s)} = \mathbf{m}_l + \mathbf{M}_l \mathbf{v}_j + \mathbf{N}_l \mathbf{v}^{(sm)} \tag{3.11}$$

Speaker-space models usually significantly outperform the SI models and can provide better results, than class-based models constructed by adapting the SI model with class-associated data. The key advantage of these techniques is that the parameter distribution is not fixed at the class level, but constructed for each particular speaker of the test data.

3.1.7 Conclusion on multi-modeling approach

Let us recall that the problem addressed in this thesis is the general inability to model highly heterogeneous data by conventional HMM-GMM because of the strong conditional independence assumptions leading to trajectory folding. State-of-the-art techniques relying on data clustering allow to reduce the variability in acoustic models and to significantly improve the modeling accuracy.

Two clustering algorithms have been described in detail and experimentally evaluated in this section. It has been shown that unsupervised clustering allows handing age and gender variability and significantly reduce the WER.

While multi-modeling approach allows to use the training data mode efficiently, it does not directly improve the HMM as a model. A better model structure is desired to better handle speaker variability and more accurately represent dynamical properties of the speech. Next section describes some of such alternative models.

3.2 Handling trajectories and speech dynamics in ASR

As it was discussed in Section 2.3.2, the HMM-based framework relies on strong conditional independence assumptions, which are generally incorrect for complex and heterogeneous speech signal. Many authors criticize HMMs for their inability to model long temporal dependencies and claim that the usage of conditional probabilistic models became so popular only because of the lack of knowledge about the speech signal and the insufficiency of training data [Morgan *et al.*, 2005].

Multi-modeling (or class-based) techniques described in the previous section do not solve the problems associated with the HMM, but rather create the conditions, in which the HMM works well enough.

Instead of (or together with) pre-clustering of the data and forming separate sets of model parameters or some model parameter transformations for different sources of variability, one can consider searching for a different model structure to replace the conventional HMM with GMM observation density in order to achieve more accurate recognition performance.

This chapter reviews some history and state-of-the-art approaches related to alternative model structures for speech recognition. Generally, these models attempt to relax some of the conditional independence assumptions and to directly involve dynamic information. For example, dependencies of various forms (both parametric and non-parametric) can be introduced between components of the observation densities (schematically the idea is shown in Figure 3.7).

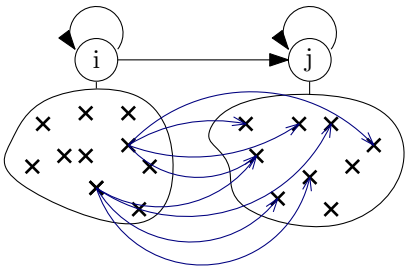

FIGURE 3.7 – Schematic representation of local trajectory model

This section starts by describing techniques applied within the HMM framework in order to better handle the contextual variability of the speech units. Then, a general *speech dynamics model* is described. Next, a broad class of *segmental models* with several extensions is reviewed. The section ends with a brief discussion on using *Artificial Neural Networks* (ANN) in ASR.

3.2.1 Handling contextual variability by HMM-GMM

As it was mentioned in Section 2.3.1, co-articulation and context play an important role in overall variability of the speech signal. Thus, to better discriminate a speech pattern, it is important to understand and to handle contextual variability in the ASR model. An example of contextual variability is the fact that the acoustic characteristics of a phone beginning, middle

and end parts are different. Also, the acoustic features highly depend on the precedent and the following phones because of co-articulation effects.

To handle such contextual differences, state-of-the-art HMM systems use *Context-Dependent* units (*triphones*, or tied triphones). Modeling separately each phone in each context requires more training data, but significantly outperforms the CI model [Lee, 1990]. Furthermore, the triphones allow to more accurately model the different pronunciation variants of the words.

Another widely-applied strategy consists in adding first and second derivatives of the acoustic features (delta features) in the feature vector [Furui, 1986]. The usage of delta features is also frequently criticized as a too simplistic way of modeling the temporal characteristics of the speech [Bridle, 2004]. Features derived from a longer window using Artificial Neural Networks are shown to outperform conventional MFCC with derivatives (this is discussed later in Section 3.2.5).

3.2.2 Multilevel speech dynamics model

The notion of speech dynamics refers to all temporal variations in the speech production process, as defined by [Deng, 2006]. In this work, the author also introduces the *Multilevel Speech Dynamics Model* (MSDM) shown in Figure 3.8. Some existing dynamic models (including HMM) can be derived from MSDM by introducing some simplifications. MSDM is a *Dynamic Bayesian Network* (DBN). DBNs are graphical models with directed edges, which are widely used in machine learning to describe temporal processes with complex dependencies, including the speech signal [Koller and Friedman, 2009; Bilmes, 2004].

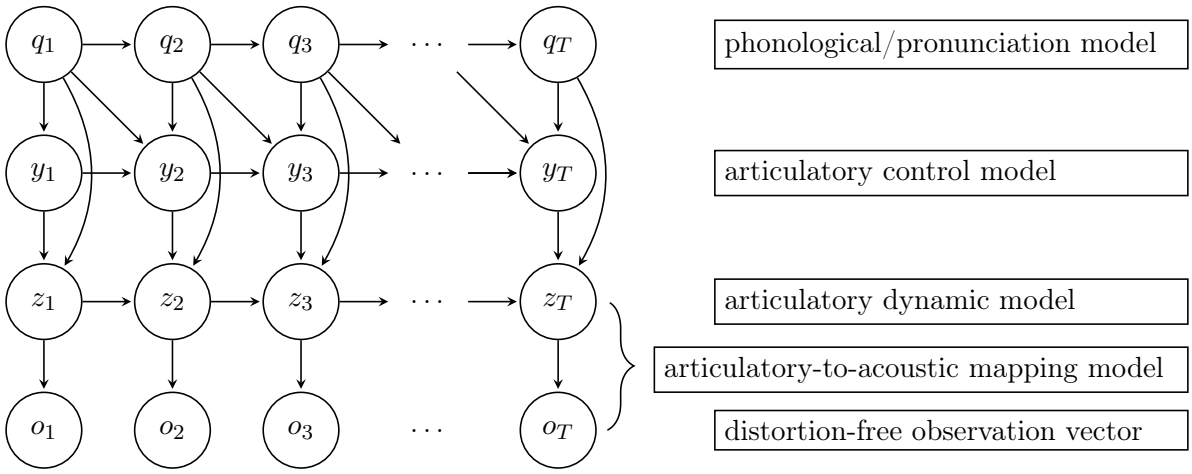

FIGURE 3.8 – Multilevel Speech Dynamics DBN representing the human speech production without a distortion model

In the original implementation by Deng this model also includes a distortion part, which models noise and transmission channel. For simplicity, the channel model is not considered in this section and the observation vector is assumed to be distortion-free. This model consists of 4 layers, and each layer represents a level of speech generation process.

The first level is described by the *phonological/pronunciation* model. In a general case of MSDM this level is represented as a *factorial Markov chain* with multiple paths (tiers). Each path represents an articulatory, or a gesture feature.

The second level contains the *articulatory control*, or the *phonetic target* model, which represents the control signal, which must be sent to the speech motor system to make it achieve the

pronunciation of the desired sequence of phones.

At the third level the *articulatory dynamic* model maps the control sequence into the physical articulatory parameter space. It is approximated as a linear filter.

The last layer represents the acoustic characteristics of the produced sequence. At this level the features are extracted from the speech signal.

Compared to the conventional HMM-GMM structure shown in Figure 2.4, the MSDM has two additional levels of states, which represent the articulatory control and the articulatory dynamic model.

A broad class of models called by Deng *acoustic dynamics models* are further discussed in this section. All these models do not deal with articulatory modeling directly, but attempt to extract hidden temporal dependencies from the observed speech signal.

3.2.3 Segmental models

Segmental Models (SM) belong to a broad class of acoustic speech dynamic models that attempt to break the conditional independence assumptions of the conventional HMM [Ostendorf *et al.*, 1996; Holmes and Russell, 1999].

The emergence of SM was mainly motivated by the fact that in conventional HMM, each state is associated only with a feature vector derived from a single frame. Let us recall that the observation pdf of an HMM is defined as follows

$$b_q(\mathbf{o}) = P(\mathbf{o}|q) \tag{3.12}$$

for state q and observation feature vector $\mathbf{o}$, derived from the corresponding frame. A sequence of states is used for modeling a phone ph (or any discrete unit of speech).

The basic units of SM are defined in a different way. Namely, instead of using the state-associated feature vectors, a sequence of L observations (a *segment* of the length L) $(\mathbf{o}_1, \ldots, \mathbf{o}_L)$ is associated with a phone ph (see right part of Figure 3.9). Thus, the pdf of the SM is defined as follows:

$$b_{ph,L}(\mathbf{o}_1, ..., \mathbf{o}_L) = P(\mathbf{o}_1, ..., \mathbf{o}_L|ph, L) \tag{3.13}$$

The joint distribution of the observation sequence and the segment length given the phone of the SM is defined as follows:

$$P(\mathbf{o}_1, \ldots, \mathbf{o}_L, L|ph) = P(\mathbf{o}_1, \ldots, \mathbf{o}_L|ph, L)P(L|ph) \tag{3.14}$$

The SM segment is described by a *duration distribution* $p(L|ph)$ and a family of *output densities* $\{P(\mathbf{o}_1, \ldots, \mathbf{o}_L, L|ph); L \in \mathcal{L}\}$, where $\mathcal{L}$ denotes all possible observation lengths for the segment ph.

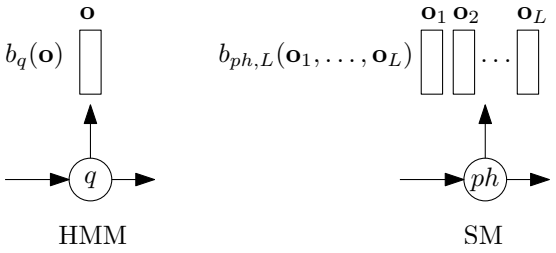

FIGURE 3.9 – Comparison of state generative process of HMM and SM

In the 90's the SMs were actively investigated by the ASR research community. In fact, the model structure proposed by SMs allows to build different acoustic dynamic models. These models are typically classified based on how the observation sequence within the segment is described (the form of the trajectory).

Constrained mean and nonstationary state models

The segment can be represented in the form of a *polynomial function*. Assuming frame-level observations to be conditionally independent given the segment length, each observation in the general case is derived as follows:

$$\mathbf{o}_t = \mathbf{g}_t(\mathbf{\Lambda}_{ph}) + \mathbf{r}_{ph}(t) \tag{3.15}$$

where $\mathbf{g}_t(\mathbf{\Lambda}_{ph})$ is a deterministic function, defined by the segment-dependent parameters $\mathbf{\Lambda}_{ph}$ and $\mathbf{r}_{ph}(t)$ is segment-dependent residual.

Such broad formulation defines a class of models, which differ in how the deterministic function is represented (i.e., parametric, or non-parametric) and how the parameters $\mathbf{\Lambda}_{ph}$ are estimated. Some examples include *Constrained Mean Trajectory* [Ostendorf *et al.*, 1992], and *Nonstationary-state* (or *trended* HMM) [Deng and Aksmanovic, 1994].

Autoregressive and conditional Gaussian models

The trajectory can also be described in the form of the linear recursion of the observation vectors, which takes into account the previous observations with some segment-dependent weights:

$$\mathbf{o}_t = \mathbf{\Lambda}_{ph}(1)\mathbf{o}_{t-1} + \ldots + \ldots \mathbf{\Lambda}_{ph}(L)\mathbf{o}_{t-L} + \mathbf{r}_{ph}(t) \tag{3.16}$$

One form of such model is called *autoregressive* HMMs [Poritz, 1982; Juang and Rabiner, 1985]. When this dependency is reduced to only the previous observation, this model becomes *conditional Gaussian* [Wellekens, 1987]. Autoregressive and Conditional Gaussian models were shown to be useful with only static features, but demonstrated no improvement when the temporal derivatives of the acoustic features were used.

Linear filter model

Another way to represent the segment is to describe the observation vectors in the form of a stochastic *linear dynamical system* as follows:

$$\begin{aligned} \mathbf{z}_{t+1} &= \mathbf{F}_t(ph) \cdot \mathbf{z}_t + \mathbf{w}_t \\ \mathbf{o}_t &= \mathbf{H}_t(ph) \cdot \mathbf{z}_t + \boldsymbol{\nu}_t \end{aligned} \tag{3.17}$$

where $\mathbf{z}_t$ denotes the unobserved state vector, $\mathbf{F}_t(ph)$ and $\mathbf{H}_t(ph)$ are state-dependent matrices, $\mathbf{w}_t$ and $\boldsymbol{\nu}_t$ are uncorrelated vectors.

The model was applied for speech recognition by [Digalakis *et al.*, 1991]. As such parameterization significantly increases the number of model parameters, various tying techniques were investigated. Many existing segmental models can be also viewed as a particular case of the linear filter model.

3.2.4 Segmental mixture and multi-path models

One disadvantage of SMs is that the within-segment trajectories impose strong constraints on the distributions of the acoustic features. This leads to accuracy degradation when the data are highly heterogeneous (i.e., coming from different speakers and recorded in various conditions). The performance of conventional HMM was significantly improved when discrete density was replaced by GMM. Similarly, SM can be extended to the *Segmental Mixture Model* [Gish and Ng, 1993], which has several advantages. First, by using mixtures of state-associated trajectories different sources of information can be compactly represented, as it is done in *multi-path HMMs* [Su *et al.*, 1996; Korkmazskiy *et al.*, 1997]. Second, this model can be used for unsupervised speech data clustering in a similar way as it is done with GMMs, as in [Han *et al.*, 2007]. The approach demonstrated improvements of the clustering in regions of higher articulatory dynamics.

Parameterizing differently the Segmantal Mixture Model component trajectories leads to models with different names and properties. For example, the earlier described Constrained Mean Trajectory model becomes *Stochastic Trajectory Model* (STM) when the single polynomial function associated with the segment is replaced by a mixture of such functions [Gong., 1997; Goldenthal, 1994].

Some more recent implementations of multi-path HMMs include *speaker-ensemble HMM* [Ye and Mak, 2012], which essentially consists in using parallel HMMs trained from different speakers. In ensemble HMM, the switch between the different speaker-dependent HMMs is applied at different levels: state, phone or segment. The best performance is achieved at the phone level, significantly improving the WER, but also increasing the model size. Recently, *Synchronous HMM* have been proposed. In this model, instead of making parallel HMMs, an additional layer of substates between HMM states and GMM components is introduced. These substates allow to parameterize long non-phonetic attributes called scenes. The state distribution of Synchronous HMMs is sparse and results in more accurate modeling [Zhao and Juang, 2013].

Other recent works on trajectory modeling extend Segmental Mixture framework by adding dependencies of the segment parameters $\mathbf{\Lambda}_{ph}$ on the current observation vector $\mathbf{o}_t$. For example, the *Semi-Parametric Trajectory Model* was explored in [Sim and Gales, 2007] and provided significant improvements on the spontaneous speech task. In this model, the means and variances of Gaussian densities vary in time based on semi-parametric function of the observation sequence with discriminatively estimated parameters. Such an extension requires more parameters to estimate, but takes an advantage of adjusting the temporal trajectory based on the locally-observed data. A similar idea is exploited in the trajectory model with *temporally varying weight regression*. In this model the posterior features are used for modeling long-term temporal structures by adding temporal dependencies on the mixture weights. The model demonstrates significant improvements in noise-robust and cross-lingual ASR with low-resource languages [Liu and Sim, 2012; Liu and Sim, 2013].

Buried HMM

Another attempt to relax the conditional independence assumptions of HMMs motivated by graphical models led to the emergence of so-called *Buried HMM* (BHMM) [Bilmes, 1999]. BHMM can be seen as a form of autoregressive HMM with Gaussian mixture observation densities completed with additional collections of GMM component dependencies.

Let us recall the joint likelihood defined in Section 2.3.2 for the observation sequence $\boldsymbol{\mathcal{O}} = (\mathbf{o}_1, \ldots, \mathbf{o}_T)$, the HMM state sequence $\boldsymbol{\mathcal{Q}} = (q_1, \ldots, q_T)$ and the GMM component sequence

$\boldsymbol{\mathcal{M}} = (m_1, \ldots, m_T)$

$$P(\boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\lambda}) = P(\boldsymbol{\mathcal{O}}|\boldsymbol{\mathcal{M}}, \boldsymbol{\mathcal{Q}}, \boldsymbol{\lambda})P(\boldsymbol{\mathcal{M}}|\boldsymbol{\mathcal{Q}}, \boldsymbol{\lambda})P(\boldsymbol{\mathcal{Q}}|\boldsymbol{\lambda}) \tag{3.18}$$

This equation is then simplified using HMM conditional independence assumptions (as in Equation 2.23). As a result, the joint likelihood of the observation sequence $\boldsymbol{\mathcal{O}}$ and the state sequence $\boldsymbol{\mathcal{Q}}$ is computed as follows:

$$P(\boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{Q}}|\boldsymbol{\lambda}) = \sum_{\boldsymbol{\mathcal{M}}} P(\boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\lambda}) = \prod_{t=1}^{T} P(q_t|q_{t-1}) \sum_{m_t=1}^{M} P(\mathbf{o}_t|q_t, m_t)P(m_t|q_t) \tag{3.19}$$

where $m_t \in \{1, \ldots, M\}$ denotes the GMM components associated with the state q_t, $P(\mathbf{o}_t|q_t, m_t)$ is the Gaussian probability of the observation $\mathbf{o}_t$ for the component m_t of the state q_t, $P(m_t|q_t)$ denotes the state-dependent component mixture weight and $P(q_t|q_{t-1})$ is the state transition probability.

BHMM generalizes this model by introducing for each time frame t a set of additional dependency variables. Let us introduce the continuous vector $\mathbf{z}_t$ that denotes the entire collection of dependency variables any element of $\mathbf{o}_t$ might use, and a discrete variable v_t indicating the class of $\mathbf{z}_t$. With some independence assumptions, the joint probability of observing the sequence $\boldsymbol{\mathcal{O}}$ and the state sequence $\boldsymbol{\mathcal{Q}}$ becomes as follows:

$$P(\boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{Q}}|\boldsymbol{\lambda}) = \prod_{t=1}^{T} P(q_t|q_{t-1}) \sum_{m_t=1}^{M} \sum_{v_t=1}^{V} P(\mathbf{o}_t|q_t, m_t, v_t, \mathbf{z}_t)P(m_t|q_t, v_t)P(v_t|\mathbf{z}_t) \tag{3.20}$$

where $P(v_t|\mathbf{z}_t)$ denotes the probability of the class v_t given the continuous vector $\mathbf{z}_t$, $P(m_t|q_t, v_t)$ is a discrete probability table, and $P(\mathbf{o}_t|q_t, m_t, v_t, \mathbf{z}_t)$ is a Gaussian pdf defined in the following form:

$$P(\mathbf{o}_t|q_t, m_t, v_t, \mathbf{z}_t) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}_{q_t m_t v_t}|}} \exp\left\{-\frac{1}{2}(\mathbf{o}_t - \mathbf{B}_{q_t m_t v_t} \cdot \mathbf{z}_t)^T \boldsymbol{\Sigma}^{-1}_{q_t m_t v_t}(\mathbf{o}_t - \mathbf{B}_{q_t m_t v_t} \cdot \mathbf{z}_t)\right\}$$

where $\mathbf{B}_{q_t m_t v_t} \cdot \mathbf{z}_t$ denotes the Gaussian mean and $\boldsymbol{\Sigma}_{q_t m_t v_t}$ is the covariance matrix.

Sparse matrices $\mathbf{B}_{q_t m_t v_t}$ consist of the $[n \times s]$ values for a given state q_t, where n is the dimension of the feature vector and s is the size of the vector $\mathbf{z}_t$. With $M = 1$ and $V = 1$ the model becomes equivalent to vector-valued autoregressive HMM. In general case ($V > 1$ and $M > 1$) BHMM can be seen as "mixture of mixtures" HMM model. Multiple structures are determined by the vector $\mathbf{z}_t$. For example, in Figure 3.10 two such model structures are drawn in blue and in red.

Evaluation on a small-vocabulary isolated word recognition task *PHONEBOOK* [Pitrelli *et al.*, 1995] (600 words lexicon is used) demonstrated a significant improvement compared to HMM-GMM with a 10% increase of the model size. However, a relatively simple recognizer is used in this evaluation (context-independent phones with 26 MFCC features: 12 cepstra + log energy with first order derivatives).

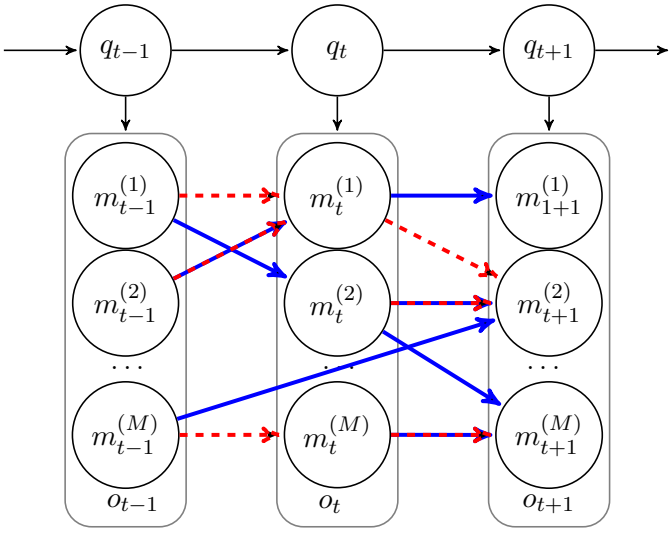

FIGURE 3.10 – An DBN representation of Buried Hidden Markov Model with two assignments of the hidden variables

Stranded Gaussian Mixture HMM

Stranded Gaussian Mixture Hidden Markov Model (later simply referred to as *Stranded GMM*: StGMM) is yet another model, which attempts to add dependencies between the components of the Gaussian mixtures of adjacent states. It was recently proposed by [Zhao and Juang, 2012] in a robust speech recognition system and plays an important role in this thesis.

This model can be seen as an extension of the Conditional Gaussian model, proposed by [Wellekens, 1987] and it has some structural similarities with BHMM. The DBN structure of the StGMM is shown in Figure 3.11.

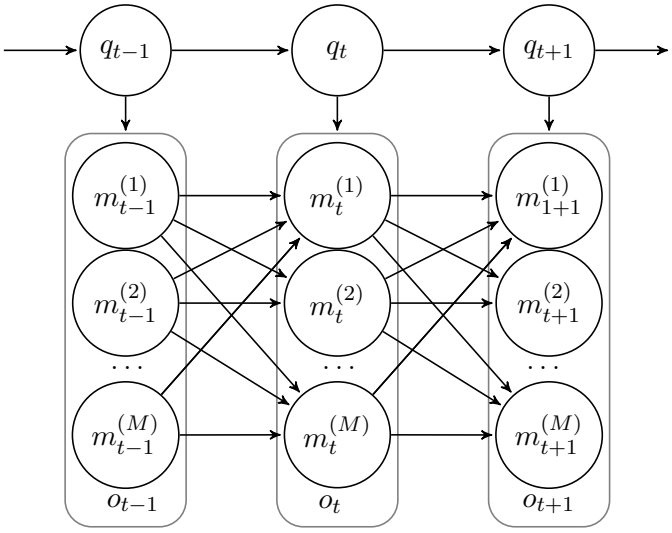

FIGURE 3.11 – DBN representation of the Stranded Gaussian Mixture Model

In StGMM additional dependencies are considered between every GMM component of the current state and every component of the previous state. Unlike BHMM, in StGMM the temporal dependencies are added only on the mixture weights. The joint probability of the observation

sequence $\boldsymbol{\mathcal{O}}$, the state sequence $\boldsymbol{\mathcal{Q}}$ and the component sequence $\boldsymbol{\mathcal{M}}$ of StGMM is written in the following way:

$$P(\boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\lambda}) = \prod_{t=1}^{T} P(\mathbf{o}_t|q_t, m_t)P(m_t|q_{t-1}, q_t, m_{t-1})P(q_t|q_{t-1}) \tag{3.21}$$

which is similar to the distribution of the conventional HMM-GMM except for the fact that the state-conditioned mixture weights $p(m_t|q_t)$ are replaced by the *Mixture Transition Matrices* (MTMs) with probabilities $P(m_t|q_{t-1}, q_t, m_{t-1})$, which leads to different (extended) training and the decoding algorithms.

In [Zhao and Juang, 2012] experiments on the noisy digits speech task demonstrated significant improvements of the StGMM over the conventional HMM. Later in the thesis this model is discussed in greater detail in Chapter 6, where the training and the decoding algorithms are described and the analysis of the model is done. Also, StGMM was used as a basis for the *Class-Structured Stranded GMM* (CS-StGMM), which is developed during this thesis and described in detail in Chapter 7.

3.2.5 Neural Networks for ASR

After speaking about alternative model structures and more accurate temporal trajectory modeling, it would be inappropriate to skip ASR acoustic modeling techniques based on *Artificial Neural Networks* (ANNs). Many research and commercial applications have successfully applied various ANN-based structures in the last years and complete state-of-the-art on this subject is beyond the topic of this thesis. Instead, this section briefly describes the history and some recent advances in this field.

Hybrid HMM-ANN models were applied for ASR earlier in 90's [Bourlard and Morgan., 1994], but did not bring a sufficient improvement compared to state-of-the-art HMM systems. By that time HMM had already successfully adopted the advanced training and adaptation algorithms. As a result, the HMM structure was dominating in public and commercial ASR systems.

Then, ANN became popular for non-linear acoustic feature transformation. Two examples of using ANN in the feature extraction phase are *tandem* [Hermansky, 2000] and *bottleneck* [Grézl *et al.*, 2007] features. Unlike frame-based MFCC features, tandem and bottleneck features use a larger window and allow to model non-linear trajectories. The extracted features are used in conventional HMM-GMM systems and generally outperform the MFCC features.

Recently, *Context-Dependent Deep Neural Networks* (CD-DNNs) were applied for ASR and achieved significant improvements compared to state-of-the-art HMM-GMM systems [Dahl *et al.*, 2010]. The major improvement of the CD-DNN compared to early ANN-HMM systems consists in the fact that CD-DNN uses many hidden layers of perceptrons. This became possible after introducing an efficient unsupervised pre-training algorithm, which consists in training each adjacent pair of layers separately from the other layers as a *restricted Boltzmann machine* (RBM) followed by the conventional *back-propagation* tuning of the multi-layer (or deep) neural network [Hinton *et al.*, 2006].

In CD-DNN for ASR [Hinton *et al.*, 2012] the posterior probabilities of the CD senones (tied states) are estimated by the soft-max output layer of the DNN. This means that DNN does not use the GMM for the posterior estimation, unlike previously discussed tandem or bottleneck approaches. Also, by using several layers and a long context, they can represent long temporal dependencies and approximate various non-linear trajectories of the acoustic features. The architecture of such CD-DNN system is schematically described in Figure 3.12.

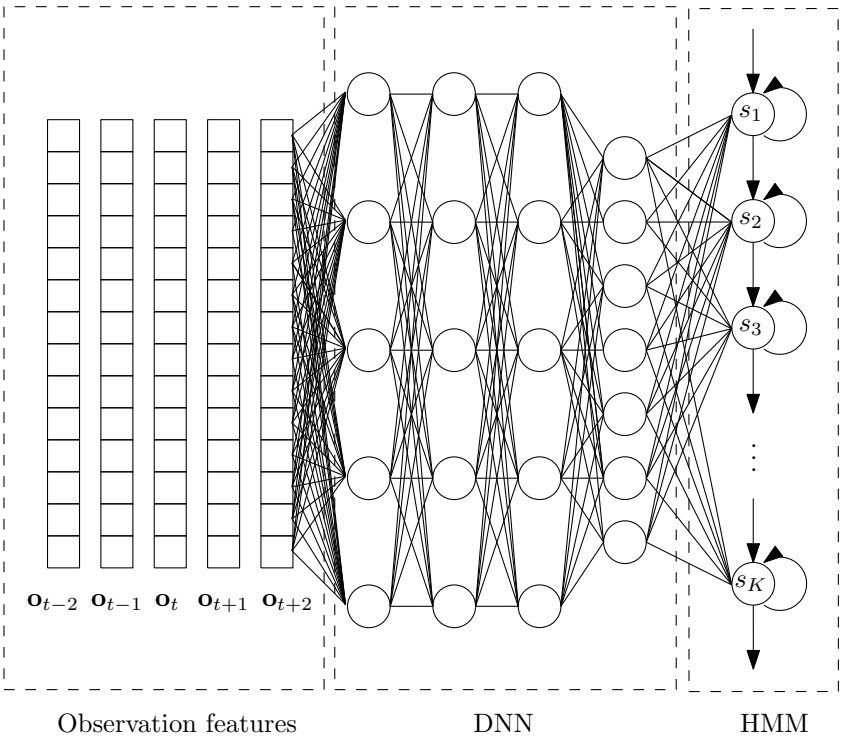

FIGURE 3.12 – Schematic representation of Context Dependent Deep Neural Network architecture for ASR

An example CD-DNN for ASR typically consists of 5-7 hidden layers, each having about 2048 units with a sigmoid non-linearity. The output layer has softmax non-linearity and a number of output units equal to the number of CD-HMM states. *Graphics Processing Units* (GPU) computation architecture allows to efficiently parallelize DNN training and decoding algorithms.

DNN achieved substantial improvements over the best tuned HMMs. At the same time, one problem associated with DNNs is their ignorance of existing theoretical knowledge about speech production. When processing a massive input window associated with a large number of units, the actual physical meaning of these parameters (even layers of parameters) is hidden. Another problem of CD-DNNs is the difficulty to perform rapid speaker adaptation techniques (similar to MLLR for conventional HMM), although some recent works demonstrate successful rapid language adaptation for a multilingual DNN [Vu *et al.*, 2014]. Both these problems are important directions in state-of-the-art research.

Many extensions of DNNs have been recently proposed. For example, to introduce knowledge about feature invariance, *Convolution Neural Networks* (CNNs) were applied and demonstrated a significant improvement over conventional CD-DNN architecture [Abdel-Hamid *et al.*, 2013] in the phonetic decoding task. Another recent example is the use of *Deep Recurrent Neural Networks* (RNN) with a memory block added in the recurrent hidden layer (or so-called *Long Short-Term Memory* (LSTM) Network), which also significantly outperformed the CD-DNN models on a large voice-search dictation data [Sak *et al.*, 2014].

3.3 Conclusion

This chapter has discussed state-of-the-art work relevant to and important for the understanding of the concepts discussed in the remainder of the thesis. Specifically, two different approaches for handling the speaker variability and improving the acoustic model accuracy were discussed.

The first ensemble of techniques referred to as *multi-modeling* or *class-based* ASR relies on building separate sets of parameters (or parameter transformations) from more homogeneous subsets of data (classes). To go beyond conventional gender-, age-, or accent-dependent models, various unsupervised segment clustering techniques are explored. The objective of these methods is to group acoustically similar speech segments in classes and then, to perform model adaptation using class-associated data. In decoding each segment is classified and the corresponding class-based model (or several best models) is selected for speech decoding. Together with model selection strategy an ensemble of *speaker-space models* has been briefly discussed. The advantage of these models is that instead of associating a class to a given speech segment, the model is constructed by interpolating several class-based models with different weights to better fit to each particular speaker.

A different strategy to handle the speaker variability is to investigate *alternative model structures*, which better parameterize long temporal sequences and non-linear speech trajectories. One way to do this is to model each HMM state as a segment composed from the observations of several frames. Various structures of Segmental Models explore different ways to parameterize such segments. Other models explicitly introduce dependencies on the components of the HMM-GMM (for example, Buried HMM and Stranded GMM). Finally, the last group of techniques uses Neural Networks in order to either introduce long context and non-linearities in the acoustic features (tandem and bottleneck features), or to replace GMM probability estimation by the soft-max output layer of a large Neural Network (as in CD-DNN)

Next sections focus on the main contributions of the thesis, on the particular problems associated with class-based modeling and on a novel approach relying on Class-Structuring of the GMM components.

4

Contributions to Class-Based ASR

This chapter discusses multi-modeling (or class-based) ASR introduced earlier in Section 3.1 and describes contributions made to this framework during this thesis. In particular, an efficient method based on a *soft classification margin* is studied in detail. Soft margin allows to improve the accuracy of the adapted class-based models by explicitly increasing the number of speech segments associated with each class.

Multi-modeling approach relies on building separate sets of acoustic model parameters (or parameter transformations) for different classes of speech variability. In a general situation, unsupervised clustering of speech segments is applied. The aim of such clustering is to split the training set into classes of acoustically similar data. As a result, speaker and channel variability is decreased within each class.

After clustering, class-based acoustic models are built by adapting the Speaker-Independent (SI) model parameters using class-associated data. The decoding relies on two passes (at least). In the first pass, the segment to be decoded is classified using the same similarity measure as the one used for the training data clustering. In the second pass, the acoustic model associated with the assigned class is used for decoding the segment.

Many state-of-the-art ASR systems apply an additional decoding pass in order to improve the accuracy of the resulting hypothesis. To do this, the hypothesis from the previous decoding pass is used for transforming the model or the feature parameters (for example, with MLLR or/and VTLN). Then, a new decoding is done with the adapted model parameters. Notice that the third pass decoding is not applied in this thesis in order to facilitate the comparison of different techniques and save time required for the experiments.

Various similarity (or divergence) measures and approaches for clustering speech data have been reviewed in Section 3.1. Two approaches have been discussed in greater detail. The first approach uses *Maximum-Likelihood* (ML) criterion and phone-independent GMMs associated with each class (later referred to as *ML-based* classification or clustering). The second approach relies on *Kullback-Leibler* (KL) divergence measure between the posterior probabilities computed using a phone-dependent speaker-independent GMMs (referred to as *KL-based* classification or clustering). Both approaches lead to a similar accuracy of the class-based ASR. An experimental comparison of two clustering methods for *Large Vocabulary Continuous Speech Recognition* (LVCSR) was described in Section 3.1 with the corresponding WERs summarized in Figure 3.5.

Intuitively, each class-based model should correspond to a characteristic that is invariant at the segment level; for example, speaker, channel, etc. Therefore, on the one hand, it is desirable to build more classes of data in order to reduce the variability of the class-based models and to increase the model accuracy. On the other hand, increasing the number of classes leads to

the fact that less data are associated with each class, which results in unreliable estimation of the class-based models. In order to build many class-based models to capture different speech variability classes and at the same time to reliably estimate the corresponding model parameters, the following approaches can be considered:

1. Collect a large amount of annotated speech data for training. The data must cover most of the variability sources appearing in the test speech data and be large enough to reliably train the acoustic models for each class;
2. Use efficient model and feature adaptation techniques in training;
3. Explicitly increase the number of class-associated segments, or perform soft clustering of the data;
4. Share some of the model parameters of class-based models;
5. Exploit alternative model structures that avoid training separate class-based models for handling heterogeneous speech data;
6. Rely on two-pass decoding process with rapid feature adaptation at the segment level.

This chapter mostly focuses on the approaches 2-3 from this list. Later, Chapters 5 and 7 focus on the approaches 4-5 by introducing *Class-Structured GMM* (CS-GMM). Only increasing the amount of training data (as suggested in the 1st approach) is expensive and has certain limitations; thus, this aspect is not discussed in this thesis. We also do not consider a second pass decoding, mostly to save time in the experiments and to facilitate comparison of different techniques. Nevertheless, performance improvements from a second pass decoding are complementary to the ones achieved by the techniques described in the thesis.

The remainder of the chapter is organized as follows. Section 4.1 describes a detailed study on the adaptation techniques for the class-based acoustic modeling. In particular, the section focuses on *Maximum Likelihood Linear Regression* (MLLR), *Maximum a Posteriori* (MAP) adaptation and their combination. Also, in this section the problem of accuracy degradation associated with an increase of the number of class-based models is experimentally analyzed.

Next, Section 4.2 introduces the *soft classification margin*, which is used in order to explicitly increase the amount of class-associated data to improve the estimation of class-based models. In this context different methods of adjusting (or tuning) the margin parameter are compared on an LVCSR task.

Finally, Section 4.3 analyses various methods of combining the hypotheses recognized with different class-based models with and without soft margin in order to improve the accuracy of the ASR output.

Contents

4.1 Adaptation methods for class-based ASR

Reliable adaptation of class-based models using associated class data is as important as meaningful clustering of the training data. Ideally, the parameters of class-based models are required to fit to the class-associated data and to generalize on unseen data of the same class. However, it is known that conventional training of the acoustic models by estimating all parameters with *Maximum Likelihood Estimation* (MLE) leads to a serious *over-fitting*, if the amount of data is not large enough. To avoid over-fitting and to take advantage of limited data, various adaptation techniques described in Section 2.4.2 can be used.

In this section, the following techniques of acoustic model adaptation for class-based ASR are investigated: Bayesian adaptation (MAP), transformation-based adaptation (MLLR) and their combination (MLLR+MAP).

In MAP adaptation the model parameters are re-estimated in a Bayesian way using SI model parameters depending on the amount of the adaptation data (see Equation 2.28). As a result, in the worst case of MAP (if no data are available), class-based parameters are equal to the parameters of the SI model. In contrast, with infinite adaptation data MAP converges to MLE. Still, for reliable MAP estimation of all HMM parameters a relatively large amount of data is required.

Differently, MLLR estimates only the parameters of one or several linear transformations applied on Gaussian means and variances to maximize the likelihood of the adaptation data. If the adaptation data are limited, the transformation matrix can be shared for all phones, or for some classes of similar phones. Generally, classes of similar phones are constructed by using decision trees with linguistically motivated questions or by relying on data-driven clustering. In most of the experiments described in this thesis the transformation matrix is shared across all triphones with the same base CI phone.

In practice, it is efficient to apply MLLR and MAP together. In this case MLLR is first used to transform the mean vectors, and then MAP is used for estimating the values of means, variances and mixture weights. When the data are limited, variances are not re-estimated.

Investigating different adaptation techniques for class-based modeling

The following set of LVCSR experiments is conducted in order to compare the performance of class-based models constructed with MLLR and MAP adaptation. The radio broadcast recordings containing about 190 hours of speech (*ESTER2.Train* data described in Appendix A.1.1) are automatically clustered using *KL-based* algorithm, as described in Section 3.1.2 (the classifier consists of phone-dependent GMMs each having 64 components). Then, the parameters of CD-HMM baseline (*mdl.LVCSR.4500s.StTel*[1]) are adapted to the class data using MLLR, MAP and MLLR+MAP. The number of components per density is fixed (therefore, increasing the number of class-based models means increasing the total number of model parameters[2]). For decoding, each segment is automatically classified and the corresponding class-associated model is selected. The second pass with semi-supervised adaptation is not applied.

The WERs achieved with class-based ASR with different number of classes on the non-African radios of the development and test data of the ESTER2 evaluation campaign (*ESTER2.Dev.11f*

1. 4500 shared densities (senones), 64 Gaussian per density, 39 cepstral features (MFCC+Δ+$\Delta\Delta$), separate models for Studio/Telephone quality data (see details in Appendix B.1)

2. The number of components for Speaker-Independent and Gender-Dependent baselines was optimized on the development data. So, we are sure that the achieved improvements do not come from a simple increase of the model size

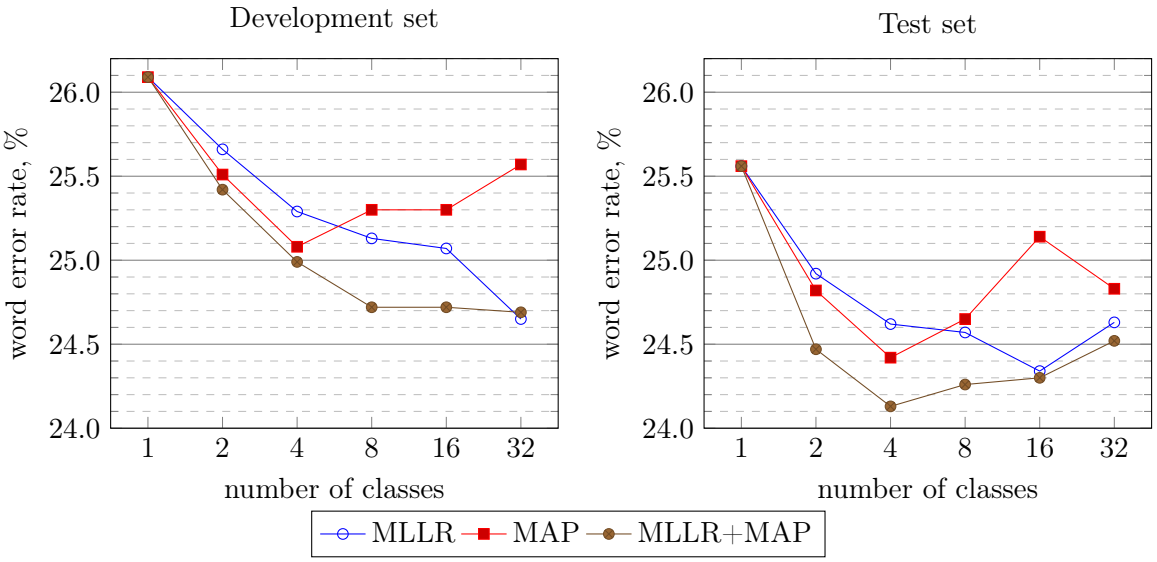

FIGURE 4.1 – Comparing MLLR, MAP and MLLR+MAP for adapting class-based models with *KL-based* classes. Use 190 hours of training data and 4500 senones

and *ESTER2.Test.17f*) are described in Figure 4.1.

First, the experiments demonstrate that MAP adaptation leads to better performances than MLLR for a small number of classes (where enough adaptation data are available). However, with a larger number of classes (starting from 8 classes in this experiment) using MAP only adapted models leads to ASR performance degradation. MLLR+MAP always leads to the best performance. However, when the number of classes gets larger (16-32 classes in the reported experiment), the recognition results do not improve any more (as for the development set), or even degrade (as for the test set).

The experiment shows that MAP adaptation is useful only with large amounts of adaptation data. The intuition of the work described further in this chapter is to increase the amount of data for MAP adaptation to improve the performance of MLLR+MAP adapted models even for a large number of classes.

Impact of enlarging the training data on class-based modeling accuracy

In order to build more class-based models and further improve the ASR accuracy, one can consider enlarging the training data. In order to verify the concept, the ASR experiments with MLLR, MAP and MLLR+MAP adaptation of the class-based models are reproduced with a larger 300 hours training data set (*EEE.Train* described in Appendix A.1.1). Also, *ML-classification* is used instead of *KL-based* classification. A larger dataset allows to train a more detailed SI baseline model with 7500 (*mdl.LVCSR.7500s.StTel*[1]). The WERs of class-based ASR are summarized in Figure 4.2.

Not surprisingly, using a larger training set (300 hours versus 190 hours) and a more detailed model (7500 senones versus 4500 senones) leads to overall better performances. At the same time, the overall analysis of the MLLR, MAP and MLLR+MAP adaptation leads to a similar conclusion. Namely, MLLR+MAP most of the time leads to the best result. With a larger number

1. 7500 shared densities (senones), 64 Gaussian per density, 39 cepstral features (MFCC+Δ+ΔΔ), separate models for Studio/Telephone quality data (see details in Appendix B.1)

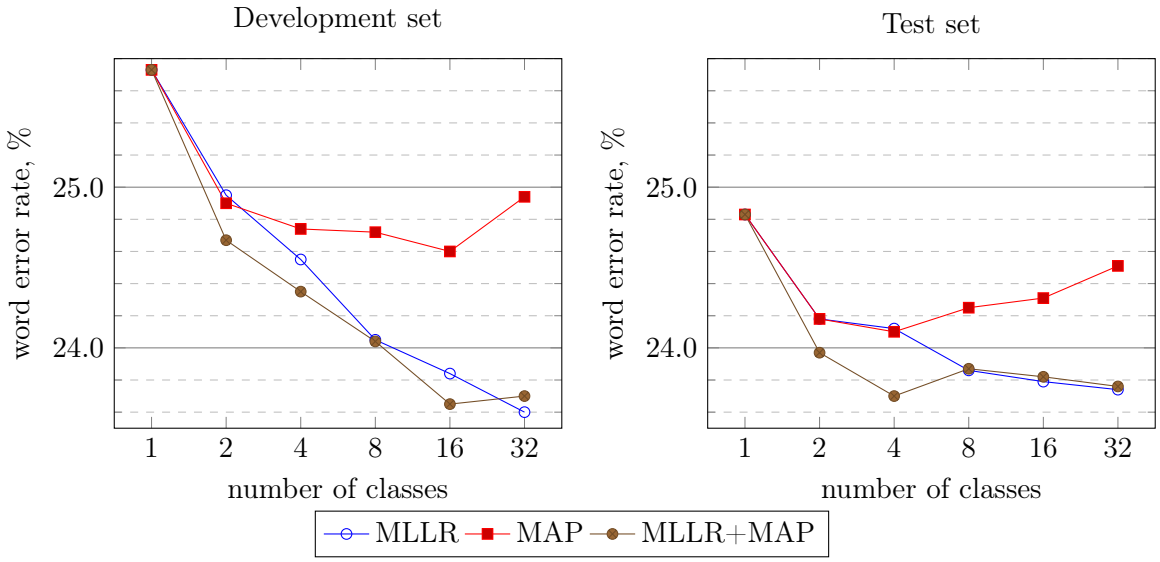

FIGURE 4.2 – Comparing MLLR, MAP and MLLR+MAP for adapting class-based models with *ML-based* classes. Use 300 hours of training data and 7500 senones

of classes the performance gets similar to MLLR.

Next sections deal with the smaller training set (*ESTER2.Train*) and focus on improving the accuracy of MLLR+MAP adaptation for the larger number of class-based models by exploiting a soft classification margin.

4.2 Margin for soft clustering of the speech segments

In order to achieve reliable estimation of a larger number of classes with a relatively limited amount of the training data, it was proposed in [Jouvet and Vinuesa, 2012] to use a soft classification margin in *ML-based* clustering of the training segments to explicitly enlarge the data set associated with each class by allowing one segment to be associated with more than one class. A similar approach was then applied in [Gorin and Jouvet, 2012] for *KL-based* clustering with phone-dependent GMMs as described in Section 3.1.2. This section presents the idea and the analysis of soft classification margin together with different ways to tune this parameter in order to improve the accuracy of the resulting class-based ASR.

4.2.1 Soft classification margin

Soft classification margin aims to increase the number of segments in each class. This is done by associating some segments to more than one class based on a distance threshold (margin) δ. This is schematically represented by Figure 4.3.

To include the margin parameter in *ML-based* classification [Jouvet *et al.*, 2012b], the following modification was proposed. The decision rule (earlier defined by Equation 3.1) that assigns a given segment u to a class c_k, is modified as follows:

$$u \in c_k \Leftrightarrow P(\boldsymbol{\mathcal{O}}_u|\boldsymbol{\Phi}_k) \geq \max_{l \in \{1,\ldots,R\}} P(\boldsymbol{\mathcal{O}}_u|\boldsymbol{\Phi}_l) - \delta, \tag{4.1}$$

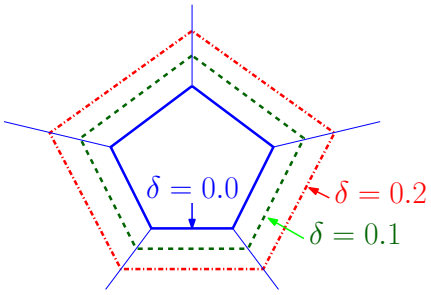

FIGURE 4.3 – The boundaries of a class with different margin parameter δ

where δ denotes the soft classification margin and $P(\boldsymbol{\mathcal{O}}_u|\boldsymbol{\Phi}_k)$ is the likelihood of the observed segment data $\boldsymbol{\mathcal{O}}_u$ given a class GMM $\boldsymbol{\Phi}_k$

It is clear from this definition that δ allows to associate a given segment with more than one class, if the likelihood of the observed data given the corresponding class-associated GMM lies within a range $[\max_l P(\boldsymbol{\mathcal{O}}_u|\boldsymbol{\Phi}_l) - \delta, \max_l P(\boldsymbol{\mathcal{O}}_u|\boldsymbol{\Phi}_l)]$.

Similarly, soft margin is introduced in *KL-based* classification phone-dependent GMMs [Gorin and Jouvet, 2012]. The only difference is that Kullback-Leibler is a divergence measure, whereas Likelihood is a similarity measure. Therefore, the signs in Equation 4.1 must be changed. The assignment criterion for *KL-based* classifier (earlier defined by Equation 3.6) is modified as follows:

$$u \in c_k \Leftrightarrow D_{Tot}(p^u||p^{c_k}) \leq \min_{l\in\{1,\ldots,R\}} D_{Tot}(p^u||p^{c_l}) + \delta \tag{4.2}$$

where $D_{Tot}(p^u||p^{c_k})$ is the KL phone-averaged distance measure between the segment u and the class c_k, defined by Equation 3.5.

4.2.2 Margin parameter and number of class-associated segments

Soft margin allows to explicitly increase the number of segments associated with each class. When $\delta = 0.0$, the resulting classes are equivalent to the classes achieved by conventional hard clustering. In contrast, a very large δ makes all classes contain the full training set (this becomes equivalent to SI model).

Figure 4.4 shows the amount of data associated with each of 8 unsupervised *ML-based* or *KL-based* classes of studio quality data from ESTER2 training set (*ESTER2.Train*) depending on the margin parameter δ. Class labels are sorted with respect to the corresponding size of the classes without applying soft margin.

Increasing the margin δ guarantees the class-associated data to increase (or at least remain the same). The figure also shows that the amount of class-associated data grows differently for different classes, because δ is applied on the actual distance (or similarity) measure between the classes and the segments. It also depends on the classifier and on the estimated parameters of the classes.

4.2.3 Tuning the margin for improving the class-based ASR performance

In the previous section it has been shown how the margin parameter modifies the amount of data associated with classes. However, the size is not as important as the resulting WER achieved with adapted class-based models. Intuitively, the margin should be increased when the

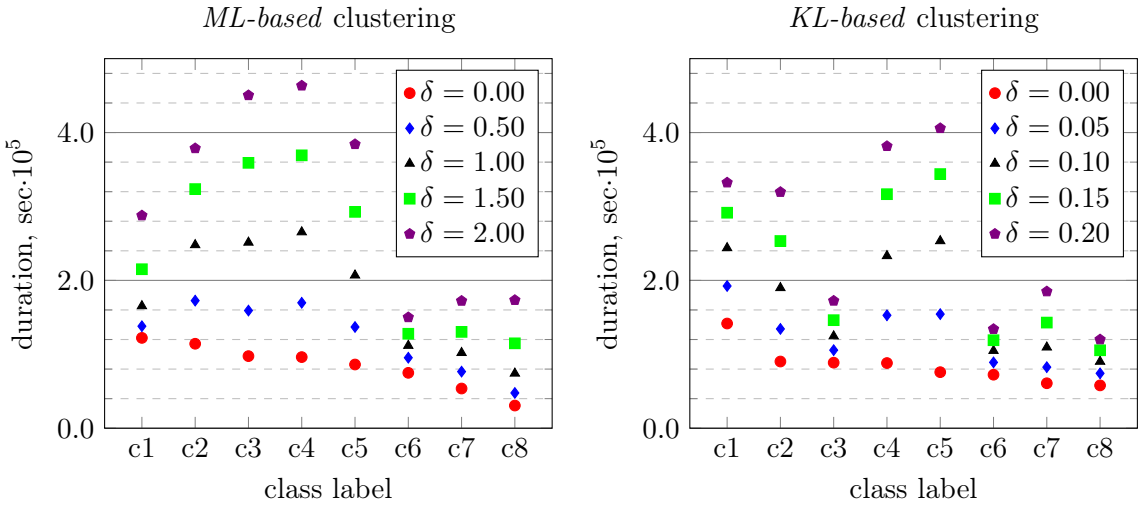

FIGURE 4.4 – Amount of class-associated speech data with respect to the margin value for each of the 8 classes constructed by *ML-based* and *KL-based* clustering

amount of classes gets larger. However, to find the optimal value of the margin, which is the one that leads to the best class-based ASR performance, the development data can be used. The development data are decoded with different sets of class-based models with various margin parameters. Then, the optimal parameters (both the number of classes and the associated margin values) are selected as the ones that lead to the smallest WER.

As in earlier LVCSR experiments MLLR adaptation of the class-models does not lead to ASR performance degradation on the development set (see Figures 4.1 and 4.2), in all following experiments the margin is used only for MAP adaptation after performing MLLR with no margin.

To summarize, the tuning algorithm consists of the following steps. First, the class-based models are constructed by adapting the parameters of SI model with MLLR fitting to the corresponding class-associated subsets clustered without margin ($\delta = 0.0$). Then, starting from the MLLR-transformed class-based models, MAP adaptation is applied to adjust the parameters using the class-associated data clustered with different margin parameters. Next, the development data are classified and decoded with class-based models associated with different margin values.

Finally, by analyzing the errors achieved by class-based ASR with different δ parameter, the following approaches are considered in order to select the best number of classes and the corresponding margin:

— *Global margin tuning* (later referred to as "globalOpt"). The WER is computed for the whole development data. The number of classes and the corresponding single value of δ leading to the minimal WER are considered as optimal;

— *Class margin tuning* (later referred to as "classOpt"). The WERs are analyzed separately in each class and the margin parameter minimizing the class WER is considered as optimal. Therefore, a separate margin value for each class is selected;

— *Class margin tuning for selected classes only.* The WERs are also analyzed in each class. However, the best margin parameter is selected at the class level only if a sufficient amount of the development data (N seconds) is associated with this class (later the method is referred to as "classOpt>Nsec"). Otherwise, the optimal value is selected based on global margin tuning.

These approaches are experimentally investigated later in this section with the corresponding results summarized in Figure 4.5.

Global margin tuning by analyzing errors on the development data

In this approach a single optimal *global margin* value is determined as the one that minimizes the WER of the full development set.

To understand how the accuracy of class-based ASR depends on the margin, the 190 hour training set (*ESTER2.Train* described in Appendix A.1.2) is clustered with *KL-based* algorithm. The clustering is done by using phone-dependent GMMs (separate for studio and telephone quality data) each having 64 components. After clustering, the class-based models are built with MLLR adaptation of Gaussian mean parameters of the CD-HMM baseline (*mdl.LVCSR.4500s.StTel* described in Appendix B.1) on the class-associated data with no margin ($\delta = 0.0$). Then, MAP adaptation of all model parameters is performed with the corresponding class-associated subsets and various margin values. The development and the test data are classified and decoded with the corresponding class-based models. Note that the margin is not used in classification of the development or test data.

Table 4.1 summarizes WERs on non-African subsets of ESTER2 development and test data (*ESTER2.Dev.11f* and *ESTER2.Test.17f*), achieved by class-based ASR with different number of classes and different values of margin.

	Development set					Test set				
Classes	Margin					Margin				
	0.00	0.05	0.10	0.15	0.20	0.00	0.05	0.10	0.15	0.20
2	**25.42**	**25.46**	25.55	25.54	25.65	**24.47**	**24.44**	24.61	24.73	24.85
4	24.99	**24.78**	**24.72**	**24.72**	25.06	**24.13**	**24.16**	**24.14**	24.36	24.49
8	24.72	**24.61**	**24.62**	24.72	24.68	24.26	**24.00**	**23.95**	24.13	24.26
16	24.72	24.50	24.57	**24.37**	24.47	24.30	24.24	**24.02**	**24.09**	**24.10**
32	24.69	24.41	**24.34**	**24.40**	**24.29**	24.52	24.22	**24.06**	**24.01**	**24.13**

TABLE 4.1 – WER of class-based ASR using different number of classes and margin values

Generally, for a larger number of classes the optimal margin value is larger. This is logical, because the amount of class-associated data is roughly inversely proportional to the amount of classes.

Furthermore, the margin optimized on the development set does not always allow to find the best margin for the test set, but it can be considered as "good enough". For example, the best performance on the development data corresponds to 32 classes and $\delta = 0.20$, which leads to 24.29% WER on the development and 24.13% WER on the test data. At the same time, the best result of 23.95% WER could be achieved on the test data with 8 classes and $\delta = 0.10$. However, tuning parameters on the test data is impossible and such a result should be considered as "oracle". The 95% confidence interval in this experiment corresponds to ±0.4% WER on development and ±0.35% WER on test set.

Line "globalOpt" of the Figure 4.5 shows the WER achieved with the strategy of selecting a single margin parameter that minimizes the WER on the development data (*global margin tuning*). In other words, for a given row of the left part of Table 4.1 the column with minimal value for this row is selected. The margin associated with this column is used for the test set.

Applying soft margin always outperforms conventional hard clustering (denoted as "noMargin"). Better performances can be achieved with "classOpt" and "classOpt>300s" that are discussed later in this section.

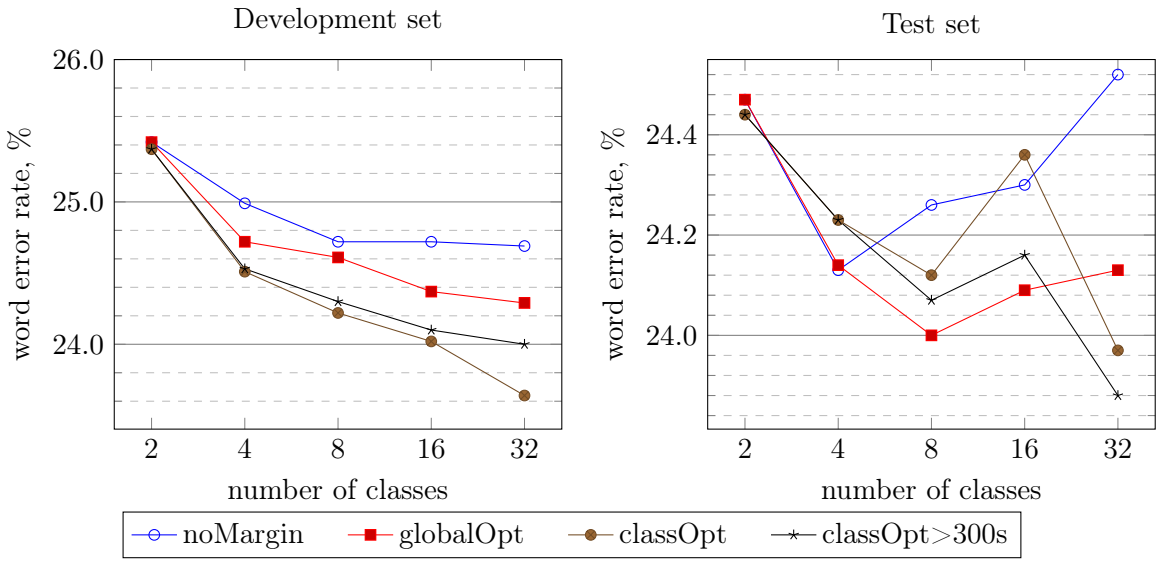

FIGURE 4.5 – Different ways of optimizing soft margin. WER for different number of classes

Class margin tuning by analyzing the errors in each class

The idea of *class margin tuning* is to select δ for each class by separately analyzing the errors of the subsets of development data associated with different classes.

The idea is motivated by the fact that the same margin value increases the amount of data associated with each class differently. For example, in the right part of Figure 4.4 (*KL-based* clustering) it is shown that $\delta = 0.20$ increases the number of frames associated with classes c4 and c5 by more than 4 times, while in classes c6 and c8 the number of frames is increased by less than 2 times.

Figure 4.6 illustrates the WERs computed on non-African development data of ESTER2 (*ESTER.Dev.11f*) separately for studio and telephone quality data associated with different classes. It also shows the amount of development data associated with each class.

The experiment setup is similar to the one described earlier in this section. Namely, the *KL-based* clustering of 190 hour dataset (*ESTER2.Train*) is done up to 4 classes and the parameters of baseline model (*mdl.LVCSR.4500s.StTel*) are adapted with MLLR+MAP using different margin values for the MAP step (MLLR is applied without margin). The analysis of errors depending on different margins leads to the following conclusions:

1. The optimal value of δ is not the same for different classes. For example, according to the right part of Figure 4.6 (telephone quality data), $\delta = 0.15$ leads to minimal WER for class c2, while $\delta = 0.00$ (no margin) provides the best result for c1;
2. The amount of class-associated data is not the same for different classes (development set is not balanced with respect to the classes), and for some classes there are not enough data to evaluate the optimal value of the margin. For example, while 4568 seconds of studio quality data are associated with the class c1, no speech data are available for the class c3 and only 266 seconds of telephone speech are available for c4.

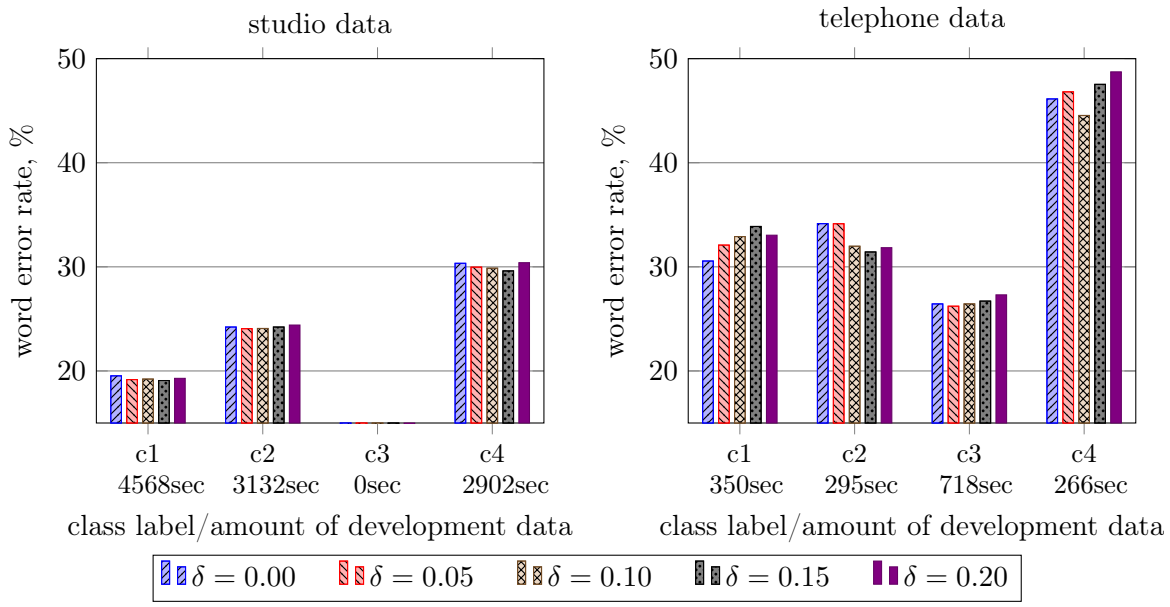

FIGURE 4.6 – WER on the development set in 4 class-based models with different margins

Consequently, although tuning the margin for each class separately seems desirable (because different values are optimal for different classes), in practice a large development set that contains a sufficient amount of data associated with all classes, is required. Together with earlier studied *global margin tuning*, two different types of *class margin tuning* are now examined and the corresponding WERs are shown in Figure 4.5:

1. *Class margin tuning* (line "classOpt"): for a given number of classes and each separate class-model a single margin value is chosen, the one that minimizes the WER on development data.
2. *Class margin tuning for selected classes only* (line "classOpt>300s"): choose an optimal margin for a given class only if more than 300 seconds of development data are available for this class. Otherwise, use the global best value associated with a given number of classes and selected by *global margin tuning* strategy.

Figure 4.7 shows the amount of class-associated training data (expressed in seconds) in some of the 32 *KL-based* classes of studio quality data. The classes corresponding to less than 300 seconds of development data are not represented. The marked values "δ in ClassOpt>300s" show the margin that leads to minimum WER on the associated development data.

Separately tuning the margin value for each class ("classOpt" strategy) significantly improves the WER on the development set, but does not provide better results on the test set. The reason is that some classes do not have enough associated development data to reliably choose the δ parameter based on the error analysis. To better generalize on the test set, a sufficient amount of development data should be associated with the class to be tuned.

To be more confident about the optimality of a chosen δ for a given class, in "classOpt>300s" experiment the class-specific margin is selected only for the classes that have enough development data (more than 300 seconds). This seems to be the best strategy for tuning the parameters, as the performance is close to "classOpt" on development data and better on test data with 32 classes.

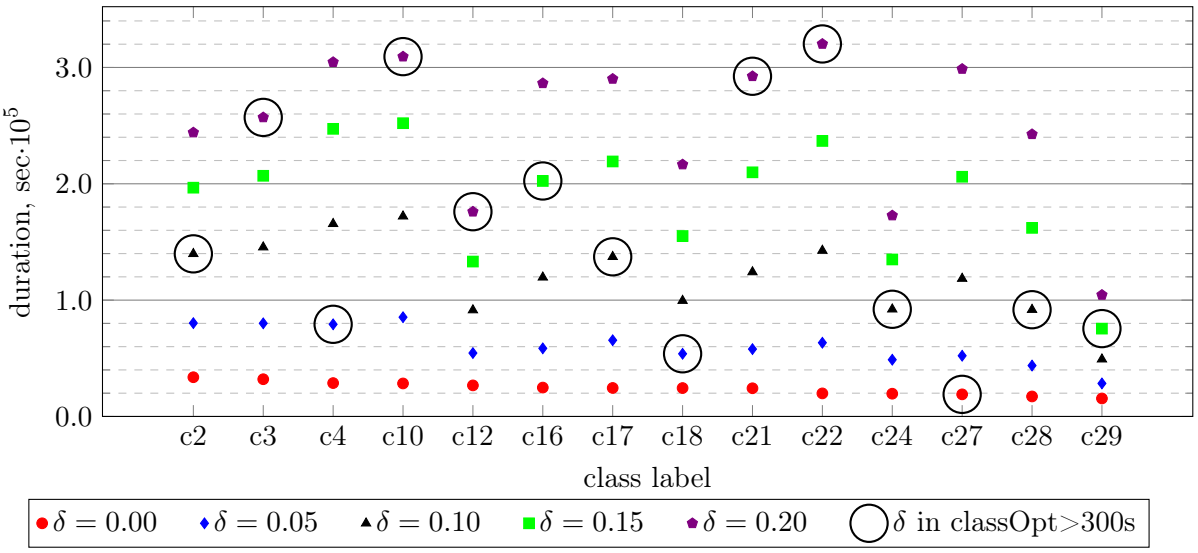

FIGURE 4.7 – Amount of studio quality training data associated with the 32 classes for different margin values δ, and displaying the optimal δ margin for classes associated with enough development data (>300 seconds)

4.3 Hypothesis combination using class-based models

Hypothesis combination is widely used for improving ASR accuracy. For efficient combination the errors made by the various systems that are combined should be different. Usually, the systems to be combined use different acoustic features, language models, or the acoustic model structures.

This section focuses on hypothesis combination using the *Recognizer Output Voting Error Reduction* (ROVER) approach, where the hypotheses to be combined come from different class-based models. The features and the language model are fixed for all recognition experiments. The following two possible approaches are studied:

— All models used for generating hypotheses to be combined, are associated with a single class estimated for a segment. Different margin values are used for estimating the class-based models.

— The models are associated with N most likely classes. A single margin parameter is used for all models.

4.3.1 Hypothesis combination using a single class and different margins

In the first set of experiments the hypotheses are computed using class-based models corresponding to the single estimated class with various margin values. The resulting hypotheses are then combined with ROVER. As the margin is not used in classification of the segments to be decoded, they must be classified only once.

Experiments are conducted using the same ESTER2 data sets, as in the previous experiments with classes-based ASR and soft margin. The 190 hour training set (*ESTER2.Train*) is clustered with *KL-based* algorithm up to 32 classes and the CD-HMM baseline model (*mdl.LVCSR.4500s.StTel*) is adapted using MLLR and MAP with different values of the margin parameter δ.

In this section only 32 classes are considered in order to facilitate the description of the approach. For each segment up to 5 hypotheses are used for combination (correspond to the

margin values 0.00, 0.05, 0.10, 0.15, 0.20). Table 4.2 summarizes the WERs of the resulting combined transcriptions, where different sets of margins are considered. In all sets, the hypothesis associated with $\delta = 0.20$ is included, as this value leads to minimal WER on the development data with 32 classes (see Table 4.1).

Margin values of the combined models	Development set	Test set
0.20 (best globally selected soft margin)	24.29	24.13
0.20 - 0.00 - 0.05 - 0.10 - 0.15	23.95	23.83
0.20 - 0.05 - 0.10 - 0.15	24.13	23.92
0.20 - 0.10 - 0.15	**23.84**	**23.66**

TABLE 4.2 – WERs corresponding to the ROVER combinations of the hypotheses obtained by class-based ASR with 32 *KL-based* classes and different margin for model adaptation

The best result of this experiment is obtained by combining the models, which correspond to 3 best margin values. Adding more models either does not significantly change the resulting accuracy, or even leads to its degradation. Overall performance improvement compared to class-based ASR with model selection and global margin optimization is from 24.29% WER to 23.84% WER on development data (*ESTER2.Dev.11f*) and from 24.13% WER to 23.66% WER on test data (*ESTER2.Test.17f*).

4.3.2 Hypothesis combination using N best classes and a fixed margin

Combining the hypotheses computed with the models associated with the same class and different margin parameters (as described in the previous section) might not be the best possible combination approach due to the following reasons:

— As the class-based models differ only in the amount of margin associated data for MAP estimation, the errors done by these acoustic models are likely to be similar;

— All class-based models with different margin values must be estimated and stored in memory.

In order to use more different models for computing the hypotheses to be combined, the following approach is proposed. Instead of selecting the single most likely class-based model for a segment decoding, a list of N best classes is selected and parallel decoding passes are done with the associated N class-based models. The same margin parameter is used for all candidates. The resulting N hypotheses are then combined with ROVER.

Experiments are conducted using the same 32 class-based models as in the previous section. Figure 4.8 shows the WERs that are achieved when the Nth best model (according to KL divergence measure) is used in decoding. Obviously, the results degrade when N gets larger, because the selected model becomes farther from the optimal one.

The ROVER combination of the resulting hypotheses from the N best class-based models is reported in Figure 4.9. WERs are reported for combinations of 3, 4 and 5 hypotheses that are computed from the corresponding N best class-based models (which separately lead to the WERs described by Figure 4.8). The proposed technique of combining hypotheses from N best class-based models demonstrates better performance, than the earlier described approach relying on combination of the hypotheses achieved with the class-based models associated with the same class and adapted using different margin values.

The best result is achieved, when 5 best hypotheses corresponding to $\delta = 0.10$ are combined. This leads to significantly better WERs: 23.46% on the development set (*ESTER2.Dev.11f*) and

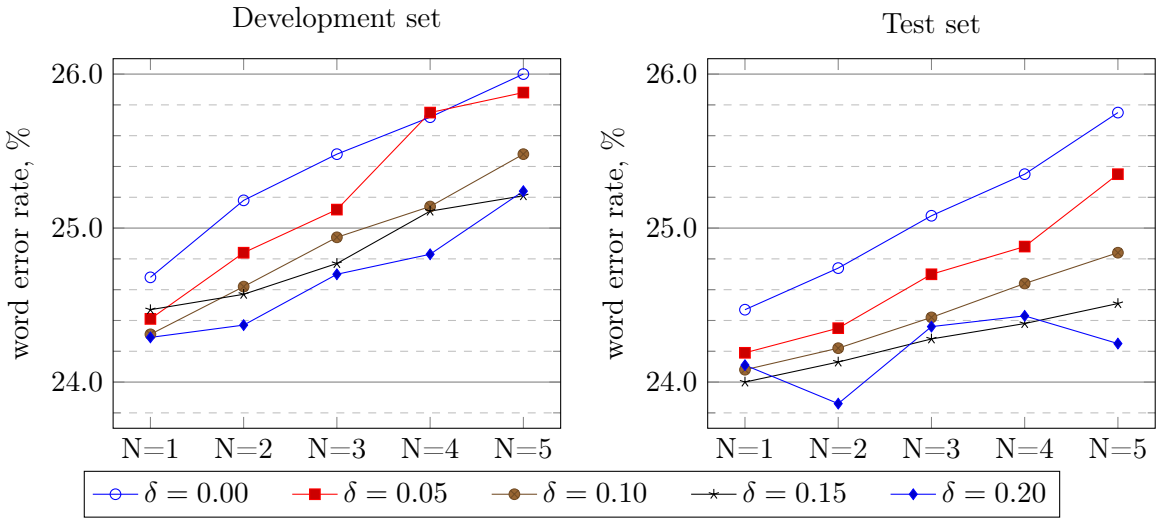

FIGURE 4.8 – WER achieved by decoding with the Nth best model (from 32 class-based models) for different margin values

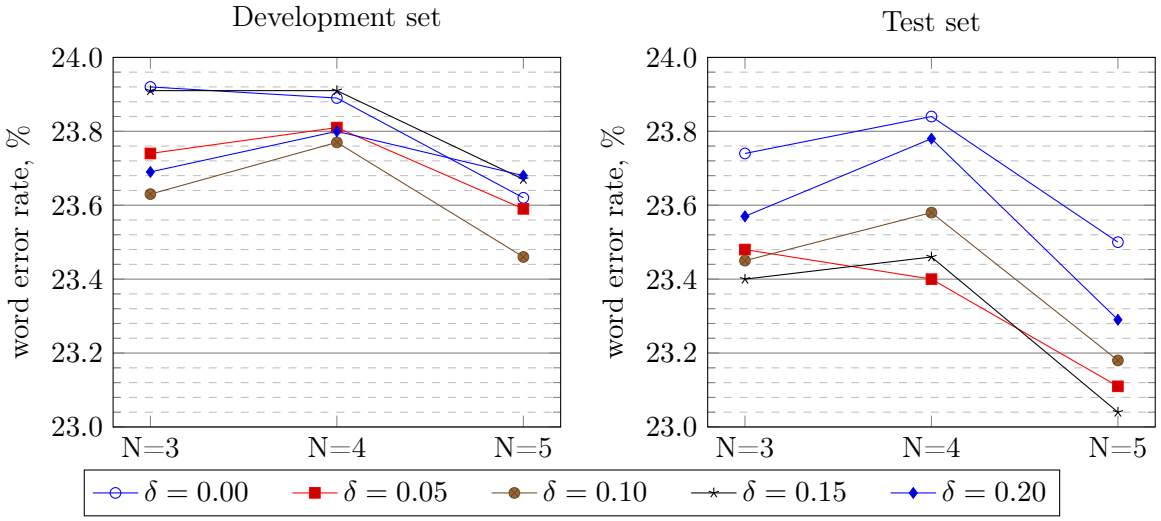

FIGURE 4.9 – WER of the resulting ROVER combined hypotheses achieved with N best models (from 32 class-based models) with different margin values.

23.18% on the test set (*ESTER2.Test.17f*). It is also the best WER that is achieved in the experiments of this chapter.

4.4 Conclusion

This chapter described the multi-modeling approach, based on unsupervised segment clustering. The segments are clustered using either *ML-based*, or *KL-based* algorithms as described in Section 3.1.2. When the training data are clustered, the resulting class-associated subsets of data are used for adapting class-based models. For decoding, the segment classification is done with the same measure as the one used in the corresponding clustering algorithm. This associates each segment with a given class. Finally, the corresponding class-based model is used in decoding.

Several problems associated with such class-based approach have been discussed in this chapter in greater detail. The proposed techniques and the corresponding WERs on *ESTER2* data are summarized in Table 4.3. When the corresponding experiments rely on some parameter tuning (like number of classes, margin value, or number of models for ROVER combination) the best parameters are determined on the development set and the same parameter values are used on the test set.

	Model	Details			WER on ESTER2 data	
		Classes	Margin	Adaptation	Dev	Test
1	SI baseline	LVCSR.4500s.StTel		-	26.09	25.56
2	GD baseline	LVCSR.4500s.StTel.GD		MLLR+MAP	25.23	24.46
3	Clustering	32	none	MLLR	24.69	24.52
4	+ Global margin	32	0.20	MLLR+MAP	24.29	24.13
5	+ Class margin	32	classOpt>300sec	MLLR+MAP	23.97	23.88
6	ROVER	32	0.00	MLLR+MAP	23.62	23.50
7	+ Margin	32	0.10	MLLR+MAP	23.46	23.18

TABLE 4.3 – Comparison of the best WERs of the baseline systems and class-based systems with different margin tuning algorithms and ROVER combination

The first problem of class-based approach is associated with the fact that when the number of classes increases, the amount of data associated with each class decreases. Transformation-based adaptation methods (like MLLR) take benefit from a relatively limited amount of adaptation data. This leads to a constant improvement of class-based models up to quite a large number of classes. The best result when using only MLLR is achieved with 32 classes (line 3 of Table 4.3, see also Figure 4.1 for details), which is a significant gain compared to conventional Speaker-Independent (SI) models, but better than MLLR+MAP adapted Gender-Dependent (GD) models only on the development data (line 2 in Table 4.3).

Although it is known that the combination of MLLR and MAP adaptations should further improve the adapted models, in practice it does not work when the amount of class-associated data decreases too much. To efficiently combine MLLR and MAP adaptations, it is proposed to explicitly increase the number of class-associated segments by assigning some segments to more than one class based on a soft classification margin.

The margin parameter is tuned on the development set. One possibility is to rely on *global margin tuning*; i.e., to select a single value of margin for the corresponding number of classes and apply this value on the test set (line 4 in Table 4.3). A better accuracy is achieved with *class margin tuning for selected classes*, when the margin is separately adjusted for the classes that

contain more than 300 seconds of development data by minimizing the corresponding errors in each class (line 5 in Table 4.3).

The second problem associated with class-based approach is how to more efficiently use the corresponding class-models. Instead of simply selecting the best class and decoding with the corresponding class-model, hypotheses from several class-based models can be combined with the ROVER voting system. The best results are achieved when combining the hypotheses from 5 best classes. With a significant increase of the computational cost (as 5 separate decoding passes are needed) the WER is also significantly improved. Using soft margin for the models to be combined allows to further improve the WER of the final hypothesis (lines 6 and 7 in Table 4.3)

To summarize, unsupervised clustering and multi-modeling approach are significant steps forward compared to conventional gender-dependent modeling. These methods allow to reduce the model variability without prior knowledge about the speaker and the recording conditions. It is also possible to efficiently apply the approach with a relatively limited amount of training data by exploiting soft classification margin. Finally, if computation power allows performing several decoding passes, the approach is significantly improved with ROVER hypothesis combination.

While model adaptation and ROVER combination are known to improve the performance in state-of-the-art systems, this chapter demonstrated an alternative approach of using these techniques with unsupervised classes of segments. Intuitively, the improvements achieved with the proposed techniques are complementary to other known techniques (second pass decoding, additional processing of the acoustic features, hypothesis re-scoring and ROVER combination of the hypotheses of various systems).

5

Class-Structured GMM with Speaker Class-Dependent Mixture Weights

Unsupervised segment clustering and class-based ASR relying on separate models for each variability class allow to reduce the negative effects of non-phonetic variability in the speech signal on the recognition performance. The drawback of such multi-modeling approach is that the number of parameters to estimate and store grows proportionally to the number of classes. Consequently, building a large number of class-based models to cover many sources of variability might be difficult, especially if the training data are relatively limited.

In Chapter 4 it was proposed to use a soft classification margin in order to explicitly increase the amount of class-associated data and to achieve a reliable estimation of class-based models. The disadvantage of such approach is that a large development set is required to accurately tune the margin parameter. Moreover, the approach does not reduce the number of model parameters, as all class-based models are estimated separately and the model parameters are not shared.

This chapter discusses a different principle of using clustered speech data. Instead of adapting separate class-based models it is proposed to learn a single model and to include the speaker class information in the model structure. The goal is to achieve a compact parameterization, which is more robust to speaker variability than conventional HMM-GMM and uses a similar number of parameters. Ideally, it should outperform earlier described class-based approaches using a significantly smaller number of parameters.

The model discussed in this chapter is further referred to as *Class-Structured Gaussian Mixture Model (CS-GMM)*. Structuring consists in associating each k^{th} component of the observation density (or a subset of components) with a given class. Similar to the earlier described class-based modeling, classes are constructed by unsupervised clustering of the training data assuming that within a segment the variability class (for example, speaker) is unchanged. The advantage of CS-GMM is the explicit relationship defined between the GMM components and the speaker class labels. Another advantage is that the knowledge about consistency of the speaker class at the sentence level (instead of the frame level) is introduced. GMM structuring alone is not beneficial, as all Gaussian density components are mixed together in pdf computation. To efficiently exploit CS-GMM it is essential to introduce dependencies of some model parameters on the speaker classes. Two approaches are considered in this thesis.

The first approach described in this chapter consists in combining component structuring with class-dependent mixture weights. This model is referred to as *Class-Structured with Class-Dependent Weights GMM* (CS-CDW-GMM). In this model the Gaussian components are class-independent (shared across classes), but class-structured, and the mixture weights are class-

dependent. In decoding, the set of mixture weights is selected based on a-priori estimated segment class. Schematically, this idea is represented in Figure 5.1.

The second approach consists in replacing the mixture weights by density component transition probabilities resulting in a novel model structure called *Class-Structured Stranded Gaussian Mixture Model* (CS-StGMM). This approach is discussed in greater detail in Chapter 6 and 7.

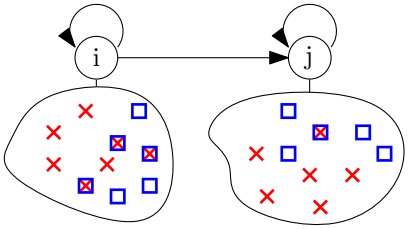

FIGURE 5.1 – Schematic representation of Class-Structured with Class-Dependent Weights GMM (CS-CDW-GMM) for 2 HMM states and 2 speaker classes

This chapter is organized as follows. First, Section 5.1 discusses the motivation for component structuring and formulates the CS-GMM framework more formally. Next, Section 5.2 describes the CS-CDW-GMM with the corresponding training and decoding algorithms. Then, Section 5.3 presents the model analysis and describes the ASR experiments. The chapter ends with a conclusion on the proposed approach.

Contents

5.1 Class-Structured GMM formulation

Class-structuring plays an important role in the remainder of this thesis. This section describes the idea, the initialization and some of the problems associated with this approach.

5.1.1 Motivation for GMM component structuring

Standard training of the HMM-GMM densities starts from a continuous-density HMM with a single Gaussian component per density. Then, the components are copied and the mean values

are perturbed. Next, the model parameters are re-estimated. The split and retrain procedures are repeated until the desired number of Gaussian components per density is achieved. GMM observation densities are required for an accurate modeling of the speaker and environmental variability of the phonetic units. In practice, the described conventional training approach has two disadvantages:

1. GMM components are trained independently, relying only on the associated frames. In contrast, the speaker identity, age, gender, accent or other variability classes are usually invariant within a segment;
2. HMM-GMM suffers from the trajectory folding problem, which means that the components trained on one speaker class are used for probability density computation for another speaker class, as all components are mixed in pdf estimation.

To address both of these problems, information from higher level clustering (such as segment-level unsupervised speaker clustering described in Section 3.1.2) can be included in the model structure. This leads to the idea of Class-Structured GMM formalized in the following sections.

5.1.2 General idea of GMM component class-structuring

The idea of *Class-Structured GMM* (CS-GMM) is to associate each k^{th} component (or a subset of components) of each observation density with a given speaker class label c_k. Again, speaker classes denote the classes achieved by unsupervised clustering of the speech training data at the segment level (although, strictly speaking, this is not an exact speaker clustering).

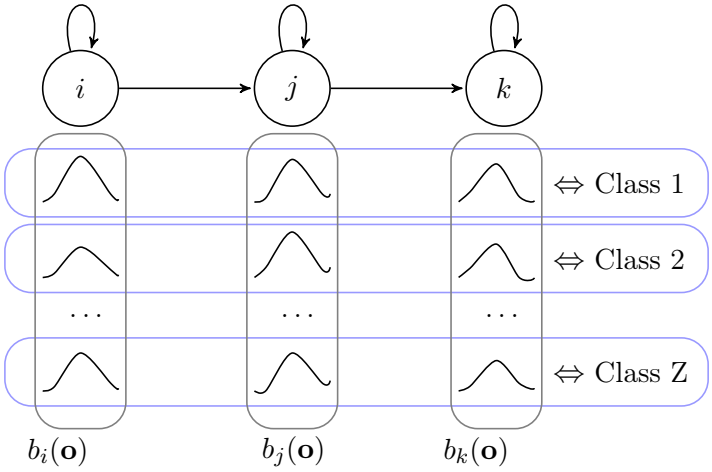

FIGURE 5.2 – An example of 3 HMM states with Class-Structured GMM densities

Figure 5.2 shows an example of 3 HMM states i, j, k with the associated density functions $b_i(\mathbf{o}), b_j(\mathbf{o}), b_k(\mathbf{o})$ defined in the form of GMMs with components structured with respect to the classes $c_1, \ldots, c_Z$. Note that in a general case more than one component can be associated with a class. Comparing the model structure with conventional HMM-GMM (shown in Figure 2.7), the only difference consists in constraining the components of Gaussian components. Considering a single component associated with each class, the number of parameters is exactly equivalent to an HMM-GMM with Z Gaussian components per density, where Z also denotes the number of classes used for GMM structuring. At the same time, unlike the components of the conventional

HMM-GMM that are trained independently based on the statistics from the state-associated frames, the components of the CS-GMM are strongly related to the global segment classes.

5.1.3 Initialization of Class-Structured GMM

Consider the objective is to build the CS-GMM with M components per density and the data are split into Z classes by unsupervised segment-level clustering. The initialization of the CS-GMM for a single GMM pdf is shown in Figure 5.3, where $\boldsymbol{\mu}_l^{(c_i)}, \boldsymbol{\sigma}_l^{(c_i)}$ and $\omega_l^{(c_i)}$ denote the mean, variance and mixture weight associated with the state j, the density component l and the speaker class c_i.

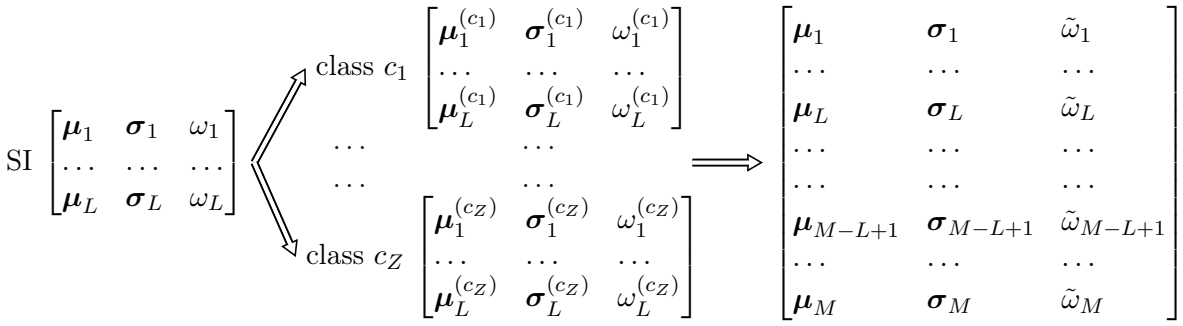

FIGURE 5.3 – Initializing CS-GMM with M components per density from Z classes of speech data (example for a single state)

The initialization algorithm consists of the following steps:

1. Train a *Speaker-Independent* (SI) model with M/Z components per density;
2. Build *Class-Based* models (CB-HMM) each having M/Z components per density by adapting the corresponding SI model on each of Z classes of the training data. Class-based models can also be built directly with standard MLE training if a large amount of data is available;
3. Concatenate the GMM components associated with the same state into a single vector;
4. Re-normalize the mixture weights to satisfy the constraint $\sum_l \omega_l = 1$. This is equivalent to dividing the mixture weights by the number of classes Z.

5.1.4 Discussion on Class-Structured GMM

Intuitively, the optimal number of components per density in the CS-GMM should be similar to the corresponding optimal number of components per density in the conventional HMM-GMM (as the models are represented by the same number of parameters in both cases). At the same time, from the experiments with class-based ASR it is observed that increasing the number of unsupervised classes continues to improve the model accuracy when soft clustering is applied. Such observation leads to the conclusion that the more classes are used for component structuring, the better the accuracy of the resulting model should be. As a result, the number of density components in the class-based models that will be concatenated into a CS-GMM can be relatively small. For example, to initialize a CS-GMM with 32 components per density using 32 classes, the initial class-based models are trained with just a single component per

density. To relax the constraints imposed during the component structuring, the model need to be re-estimated.

However, if one assumes no modifications in the standard training framework, such parameter re-estimation leads to the fact that the GMM components are mixed together in the pdf computation (as defined by Equation 2.6) with the corresponding mixture weights. In other words, although the speaker class-associated parameters are explicitly represented by the subsets of the CS-GMM components, such information would likely be neglected in the parameter re-estimation. Similar reasoning suggests that the decoding framework should be somehow modified in order to use only the relevant components of CS-GMM for a given variability class.

The proposed CS-GMM has some advantages in theory, as it creates a link between the GMM components and the speaker classes. However, in order to use such additional advantage of CS-GMM, additional modifications of the model dependencies are required. The remainder of this chapter focuses on an approach, which relies on conditioning the mixture weights of the CS-GMM on the speaker classes.

5.2 Class-Structured with Class-Dependent Weights Gaussian Mixture Model

An approach to efficiently use the CS-GMM relies on conditioning the weights of Gaussian mixture components (mixture weights) on the speaker classes [Gorin and Jouvet, 2013]. The classes are estimated at the segment level and are used for the initial structuring of the components. Such a model is referred to as *Class-Structured with Class-Dependent Weights GMM* (CS-CDW-GMM). The mixture weights of CS-CDW-GMM are used for weighting the subset of CS-GMM density components, which are more relevant for a given speaker class. The observation density of CS-CDW-GMM is defined as follows:

$$b_j^{(c_i)}(\mathbf{o}_t) = P(\mathbf{o}_t|q_t = j, C_t = c_i) = \sum_{k=1}^{M} \omega_{jk}^{(c_i)} \mathcal{N}(\mathbf{o}_t|\boldsymbol{\mu}_{jk}, \mathbf{U}_{jk}) \tag{5.1}$$

where q_t is the model state at time frame t, C_t is the class of the segment, $\mathbf{o}_t$ is the observation vector, M is the total number of components per mixture and $\mathcal{N}(\mathbf{o}_t|\boldsymbol{\mu}_{jk}, \mathbf{U}_{jk})$ is the Gaussian density with mean vector $\boldsymbol{\mu}_{jk}$ and covariance matrix $\mathbf{U}_{jk}$.

Compared to the standard HMM density defined by Equation 2.6, the only difference is that the mixture weights depend on the class label, which is associated with each segment. A similar form of the density function was used in [Schultz and Waibel, 2001] for multi-language acoustic modeling. The following sections describe the modifications of the training and of the decoding algorithms for CS-CDW-GMM.

5.2.1 Building CS-CDW-GMM model

The Gaussian means and variances of CS-CDW-GMM are initialized from CS-GMM (as described earlier in Section 5.1.3 and shown in Figure 5.3). Two ways of initializing the mixture weights are considered.

The first approach is to initialize class-dependent weights from the weights of the corresponding initial class-based models and append the vectors by small values ε for those components, which are not associated with the corresponding classes [Gorin and Jouvet, 2013]. A schematic representation of such initialization is shown in Figure 5.4.

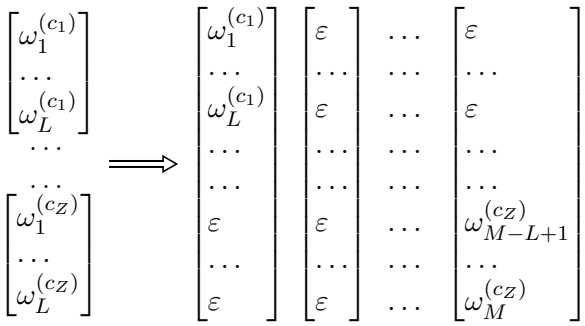

FIGURE 5.4 – Initialization of class-dependent mixture weights with a small probability value ε for the components that are not associated with the corresponding class

The second approach is to define each class-dependent set of mixture weights as a concatenated vector of the weights of the initial class-based models; i.e., initialize the weights from the CS-GMM (see Figure 5.5). This approach assumes no initial constraints on the mixture weights.

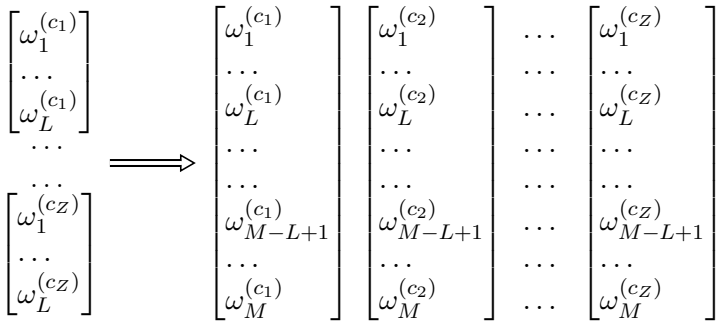

FIGURE 5.5 – Initialization of class-dependent mixture weights without initial constraints

In preliminary experiments it was found that such initialization without initial ε constraint leads to a slightly better modeling, especially with a larger number of classes. Therefore, in all future experiments it is decided to initialize the weights using the second strategy. In fact, some experiments with uniform initialization of the weights demonstrated similar results.

5.2.2 Re-estimation of CS-CDW-GMM

After CS-CDW-GMM initialization, the Expectation-Maximization re-estimation of the model parameters is applied. The re-estimation step is schematically shown in Figure 5.6 for means and mixture weights of a single density component. Re-estimation of variances is similar to the re-estimation of means.

Assuming each segment u of the training data is associated with a class label $C_u \in \{c_1, \ldots, c_Z\}$. So, each observation vector $\mathbf{o}_t$ from some time frame t belonging to the segment u is also associated with the class $C_t = C_u$. To re-estimate the model parameters, Baum-Welch frame-level posterior probabilities (see detailed algorithm description in Section 2.2.3) are computed separately for each class c_i:

$$\gamma_t^{(c_i)}(j,l) = P(q_t = j, m_t = l, C_t = c_i | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}) \tag{5.2}$$

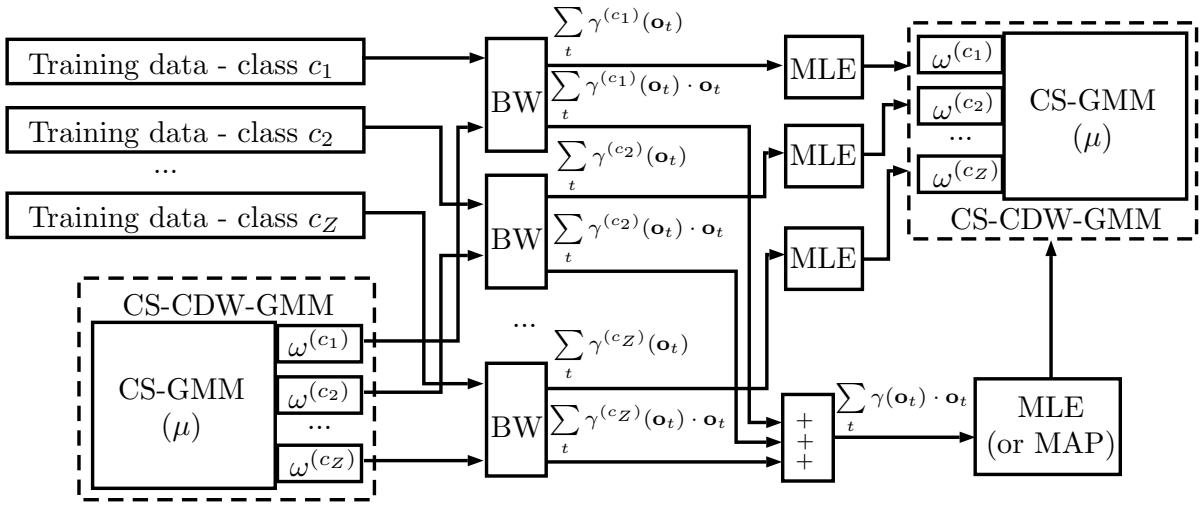

FIGURE 5.6 – Training of CS-CDW-GMM from initialized CS-GMM. The parameter updating procedure is shown for means and mixture weights associated with a single state and density component. BW denotes Baum-Welch algorithm

where q_t is the model state at the time t, m_t is the density component number, $\boldsymbol{\mathcal{O}}$ is the observation sequence and $\boldsymbol{\lambda}$ denotes the model parameters.

ML estimation of the new values of class-dependent mixture weights is done by normalizing the class-dependent posterior probabilities:

$$\omega_{jl}^{(c_i)} = \frac{\sum_{t=1}^{T} \gamma_t^{(c_i)}(j,l)}{\sum_{t=1}^{T} \sum_{k=1}^{M} \gamma_t^{(c_i)}(j,k)} \tag{5.3}$$

As the means and variances are shared across all the speaker classes, the corresponding counts are summed over all classes before normalization. The new values are computed as follows:

$$\boldsymbol{\mu}_{jl} = \frac{\sum_{i=1}^{Z} \sum_{t=1}^{T} \gamma_t^{(c_i)}(j,l) \cdot \mathbf{o}_t}{\sum_{i=1}^{Z} \sum_{t=1}^{T} \gamma_t^{(c_i)}(j,l)} \tag{5.4}$$

$$\boldsymbol{\Sigma}_{jl} = \frac{\sum_{i=1}^{Z} \sum_{t=1}^{T} \gamma_t^{(c_i)}(j,l) \cdot (\mathbf{o}_t - \boldsymbol{\mu}_{jl})(\mathbf{o}_t - \boldsymbol{\mu}_{jl})^T}{\sum_{i=1}^{Z} \sum_{t=1}^{T} \gamma_t^{(c_i)}(j,l)} \tag{5.5}$$

where summation over t means summation over all the frames of all training segments of the class.

In order to better preserve the initial class-structuring, means and variances can also be estimated in a Bayesian way (MAP) with the only difference of an additional summation over classes to take into account the parameter sharing. Preliminary experiments demonstrated that using MAP for Gaussian parameters leads to more stable results. Consequently, in all the experiments done with CS-CDW-GMM, MLE is used for updating the mixture weights and MAP is applied for means and variances.

5.2.3 Decoding with CS-CDW-GMM

Decoding with CS-CDW-GMM is similar to the class-based ASR decoding described earlier in Chapter 4. The process relies on 2 passes. In the first pass, each segment to be recognized is assigned to a class c_i. In the second pass the standard Viterbi decoding is performed using the shared set of means and variances and the set of mixture weights associated with the assigned class c_i. Therefore, the complexity of the decoding with CS-CDW-GMM is equivalent to the complexity of conventional Viterbi with an additional classification pass (as in class-based ASR).

5.3 Model analysis and evaluation

Class-Structured with Class-Dependent Weights Gaussian Mixture Models (CS-CDW-GMM) are experimentally studied in this section. First, an analysis of the re-estimated values of class-dependent mixture weights is done in order to better understand some of the model properties. Second, experiments on an English connected digits task are reported. The objective is to verify how the proposed approach improves the recognition performance on clean data that contains a relatively large amount of speaker variability (mostly speaker age and gender). Finally, the approach is investigated on large vocabulary continuous speech recognition (LVCSR).

5.3.1 Analysis of the class-dependent mixture weights

This section describes the analysis of the mixture weights of re-estimated CS-CDW-GMM. Here, the CS-CDW-GMM is trained using *TIDIGITS*[1] full training set with standard feature extraction. The number of model parameters is similar to the number of parameters of the baseline model trained on a full TIDIGITS training set (*mdl.TIDIGITS.Full*[2]). For initialization of CS-GMM, unsupervised *ML-based* hard clustering (without soft margin as described in Section 3.1.2) is used. Class-based models with a small number of density components are estimated using MLLR+MAP adaptation of the SI model trained from the full training data. Then, the re-estimation of the CS-CDW-GMM parameters is done by applying MLE for the class-based mixture weights and MAP for the shared Gaussian means and variances.

Figure 5.7 shows the class-dependent mixture weights of the CS-CDW-GMM, which is structured with respect to 2 speaker classes. The class-based models used for initializing the CS-CDW-GMM, have 16 Gaussian components per density. This leads to a global model with 32 components per mixture. The resulting re-estimated mixture weights are compared to the mixture weights of a conventional HMM-GMM trained on the same data. In this figure, the values of the mixture weights are averaged over all densities and the corresponding standard deviation values are shown in the bars.

Figure 5.8 shows a similar example of several class-dependent mixture weights averaged over HMM densities (for classes c_7, c_{17} and c_{27}), which correspond to the CS-CDW-GMM initialized from 32 class-based models each having a single Gaussian component per density. The same number of model parameters as in the previous model is used.

These results show that after re-estimation of the CS-CDW-GMM parameters, class-dependent mixture weights are larger for the components that are associated with the corres-

1. Clean digits from different adult male, female and child speakers. Training set: 28329 digits for adult and 12895 digits for child speech. Test set: 28554 digits for adult and 12533 digits for child speech. (see details in Appendix A.2.2)

2. Word-dependent phones (3 state per phone), 32 Gaussian components per density, 39 cepstral features (MFCC+Δ+$\Delta\Delta$), 8kHz down-sampled (see details in Appendix B.2)

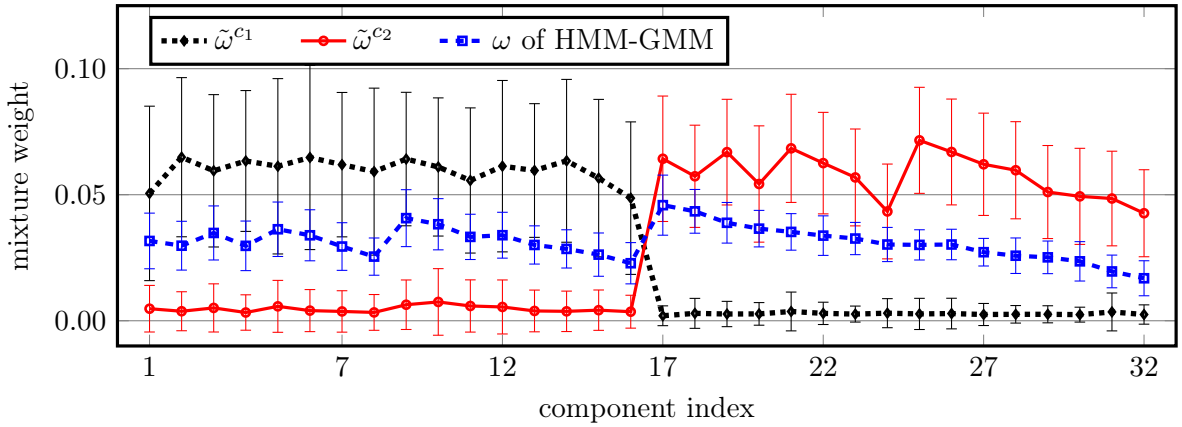

FIGURE 5.7 – Class-dependent mixture weights of CS-CDW-GMM after joint re-estimation compared to the mixture weights of the conventional HMM-GMM. The weights are averaged over densities with corresponding standard deviation values in bars (here the number of classes Z=2 and the number of components per density M=32)

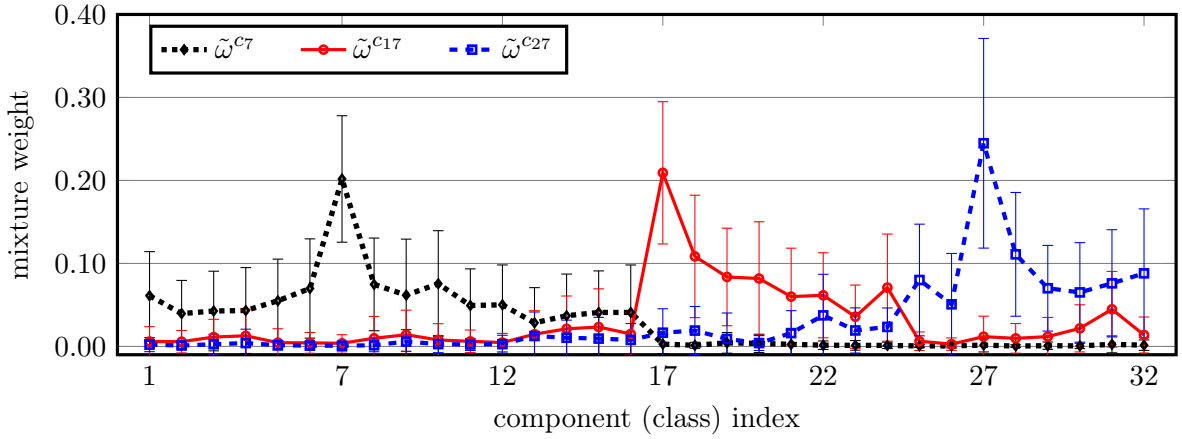

FIGURE 5.8 – Class-dependent mixture weights of CS-CDW-GMM after joint re-estimation. Mixture weights are averaged over densities with corresponding standard deviation values in bars (here the number of classes Z=32 and the number of components per density M=32)

ponding classes of data initially used for structuring. Compared to the mixture weights of the conventional HMM-GMM, the weights of CS-CDW-GMM are not uniformly distributed over the GMM components.

When a larger number of classes are used for structuring the densities (as in Figure 5.8, where 32 classes are used), the neighboring components also have relatively large mixture weights. This happens because of the implementation of *ML-based* classifier (described in Section 3.1.2). As the process relies on the iterative split of the GMM components, it is likely that child classes represent the characteristics of the parent class. For example, in the set of 32 classes, classes 17,...,32 are likely to be more similar to each other, than to the classes 1,...,16, as the corresponding data of these two sets of classes are likely to be associated with initially constructed 2 classes, which mostly separate male speakers from female and child speakers (as shown in Section 3.1.3).

A similar analysis was done in [Gorin and Jouvet, 2013] on large radio broadcast data with a different initialization of mixture weights described earlier in Figure 5.4 (using small probabilities ε for the components that are not associated with the corresponding classes). The analysis led to similar results and conclusions.

5.3.2 ASR experiments on connected digits with age and gender variability

This section describes the evaluation of the CS-CDW-GMM on the TIDIGITS task. Although state-of-the-art systems efficiently handle the task of recognizing clean speech with small vocabulary, TIDIGITS task is still interesting, because a part of the dataset contains recordings from children speakers. As it was discussed in Section 2.3.1, children speech is difficult for ASR. The baseline performance shown in Table 5.1 helps to better understand the problem. Two models with the same number of parameters and using the same features are trained from either adult subset only (*mdl.TIDIGITS.Adult*) or from the full training set (*mdl.TIDIGITS.Full*). Similar to other work with *TIDIGITS* [Burnett and Fanty, 1996], the signal is down sampled to 8 kHz and filtered below 200 Hz and above 3700 Hz to model the telephone channel.

Name	Model description	Adult	Child
mdl.TIDIGITS.Adult	**Training on adult data**	**0.64**	**9.92**
mdl.TIDIGITS.Full	**Training on adult+child data**	**1.66**	**1.88**

Table 5.1 – WERs of the TIDIGITS baseline models estimated using either only adult or full training data

Training on adult data provides the best results for adult speakers, but shows a weak performance on the child subset. When child training data are also used, the conventional HMM-GMM improves on child, but degrades on adult subset. For information, the results that assume that speaker gender and age are known for the training data are presented in Appendix B.2.

In the experiments on TIDIGITS reported in this thesis, let us focus on the situation, when no information about the speaker is available and the models are initially trained from both adult and child speakers (as in *mdl.TIDIGITS.Full* baseline). In the experiments described in this section, up to 32 *ML-based* unsupervised classes of the training speech data are built (see the algorithm in Section 3.1.2). Using different number of classes, the two following types of models are constructed.

First, *Class-Based HMM* (CB-HMM) is built by adapting the parameters of the corresponding SI model with MLLR+MAP (similar to the experiments described in Section 4.1). No margin is applied for CB-HMM, because there is no development set to tune the margin parameter.

Second, the *Conventional HMM with Class-Dependent Weights GMM* (CDW-GMM) is built. In this approach, the Gaussian parameters (means and variances) are initialized from conventional HMM and shared across classes, while the mixture weights are class-dependent. The same re-estimation is applied as for CS-CDW-GMM: class-dependent mixture weights are re-estimated with MLE and shared means and variances are updated with MAP (see Section 5.2.2). CDW-GMM is investigated in order to verify how class-dependent mixture weights alone (without class-structuring) affect the ASR performance.

Finally, earlier described *Class-Structured with Class-Dependent Weights GMM* (CS-CDW-GMM) is trained. CS-GMM is initialized from CB-HMM with a smaller number of Gaussian components per density (32 divided by the number of classes), which is obtained by MLLR+MAP adaptation of the SI models. Then, class-dependent sets of mixture weights are defined and the model parameters are re-estimated. MAP re-estimation is used for shared means and variances using all data and MLE is applied for the mixture weights using the class-associated data. The only difference between CDW-GMM and CS-CDW-GMM is that in the first case the GMM components are trained in conventional way, and in the second case the components are structured with respect to unsupervised classes. In decoding, for all three models the class is determined based on ML-criterion and the corresponding set of mixture weights is selected.

The corresponding WERs are compared in Figure 5.9 separately on adult and child test data. The bars indicate the 95% confidence interval. With a relatively small number of classes (4 classes on adult data and 2 classes on child data), CB-HMM reduces the WER compared to the SI baseline. However, increasing further the number of classes leads to performance degradation. This is not surprising, as with many classes there is not enough data for a reliable estimation of the model parameters.

For structured GMM only mixture weights are re-estimated with class-associated data, whereas all data are used for means and variances re-estimation. Due to such sharing, the model is less sensitive to the number of class-associated data and leads to a continuous improvement of the WER, when the number of classes increases. Interestingly, without component class-structuring, re-estimated class-dependent mixture weights (as in CDW-GMM) do not lead to any performance improvement. Using CS-CDW-GMM with 8 and more classes leads to a significant improvement of WER on both adult and child data compared to the baseline systems. The improvement for adult and child subsets are very similar (comparing the baseline and the result achieved with 32 classes).

Another advantage of the proposed CS-CDW-GMM is a much smaller number of parameters compared to CB-HMM, as the shared means and variances of multivariate Gaussian components contain each 39 values, whereas each mixture weight is parameterized by only one value per component. Table 5.2 summarizes the best performances and the number of parameters corresponding to the 4 proposed approaches.

Component structuring together with class-dependent mixture weights allows to efficiently use class-associated information, while it does not significantly increase the number of parameters with respect to the SI model. Note that in the experiments described in this section the initial model was trained on both adult and child utterances, which was the approach leading to the average best performance of the SI model on the complete test set. However, this significantly degrades the performance on the adult subset (see Table 5.1) compared to training with only adult data. Even after adaptation on adult data, this model does achieve a performance on the adult test data similar to the performance of the model trained on the adult data. The experiment suggests that more accurate modeling can be achieved by training the initial model on the adult subset of the training data and then adapting the model parameters for dealing with child data. In this case, unsupervised clustering and class-structuring are also efficient. See Appendix D.1

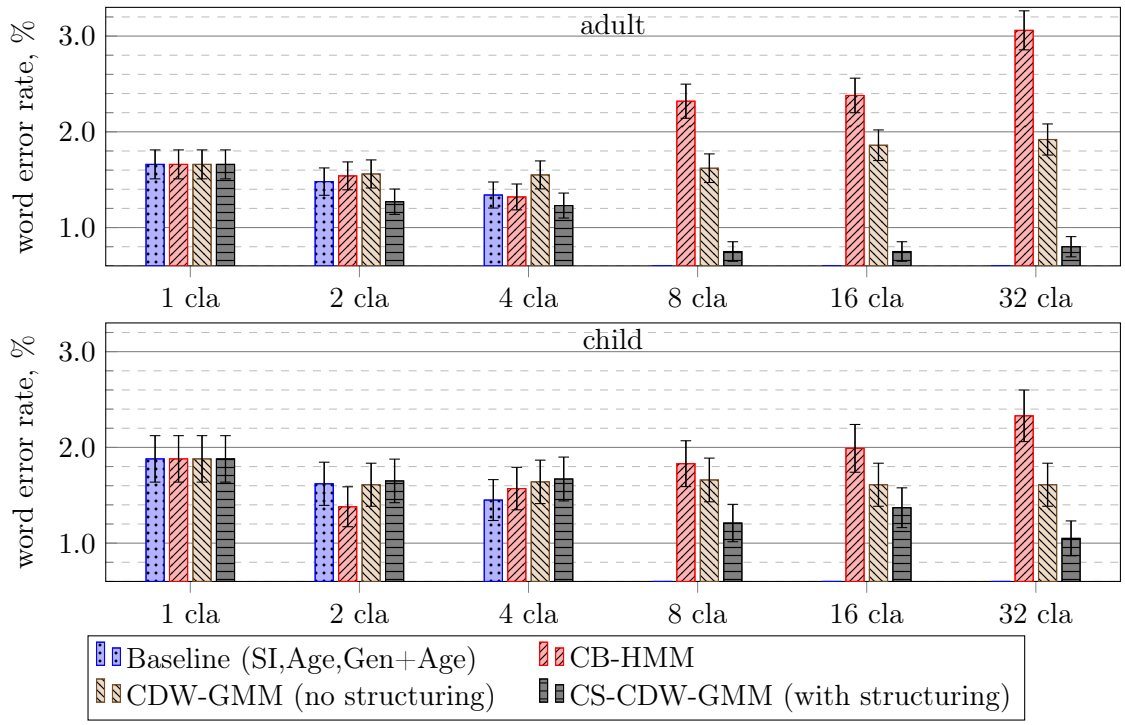

FIGURE 5.9 – WER with 95% confidence intervals for TIDIGITS adult and child data. Performance of baseline *Speaker-Independent* (SI), *Age-Dependent* (Age) and *Gender-Age-Dependent* (Gen+Age) models are compared to the unsupervised *Class-Based HMM* (CB-HMM), *Conventional GMM with Class-Dependent mixture Weights* (CDW-GMM) and *Class-Structured GMM with Class-Dependent mixture Weights* (CS-CDW-GMM)

for further details of this experiment.

5.3.3 Experiments with large vocabulary radio broadcast data

Similar experiments are conducted in this section on an LVCSR task. The model is trained using radio broadcast and TV show recordings containing about 300 hours of speech (*EEE.Train* data set described in Appendix A.1.1 for details). Two *Class-Based models* (CB-HMM) are used for comparison. Both models are built by adapting a CD-HMM baseline (*mdl.LVCSR.7500s.StTel*[1]) with either MAP, or MLLR+MAP. This corresponds to the models used in Section 4.1 and in Figure 4.2.

ML-based unsupervised classes are used for building CS-CDW-GMM. The initialization and re-estimation procedures are exactly similar to those applied in the previous experiments on TIDIGITS data, except for the larger number of components per density and the context-dependent triphones used in LVCSR experiments. Figure 5.10 shows WERs achieved on the non-African radios of the development and test data of the ESTER2 evaluation campaign (*ESTER2.Dev.11f* and *ESTER2.Test.17f*) with conventional class-based models (CB-HMMs) and with the CS-CDW-GMM.

A detailed summarized comparison of the conventional class-based and the proposed class-structured approach with respect to model accuracy and number of parameters per density is

1. 7500 shared densities (senones), 64 Gaussian per density, 39 cepstral features (MFCC+Δ+$\Delta\Delta$), separate models for Studio/Telephone quality data (see details in Appendix B.1)

Model	Decoding	Classes	Parameters/density	Adult	Child
SI HMM	1 pass	1	$78{\cdot}32{+}32{=}2528$	1.66	1.88
Gen+Age HMM	2 pass	4	$4{\cdot}(78{\cdot}32{+}32){=}10112$	1.34	1.45
CB-HMM	2 pass	4	$4{\cdot}(78{\cdot}32{+}32){=}10112$	1.32	1.57
CS-CDW-GMM	2 pass	32	$78{\cdot}32{+}32{\cdot}32{=}3520$	**0.80**	**1.05**

TABLE 5.2 – Summary of the best results and number of parameters per density for the TIDIGITS task achieved with *Speaker-Independent* (SI) and *Gender-Age-Dependent baselines* (Gen+Age), *Class-Based HMM* (CB-HMM) and *Class-Structured with Class-Dependent mixture Weights GMM* (CS-CDW-GMM)

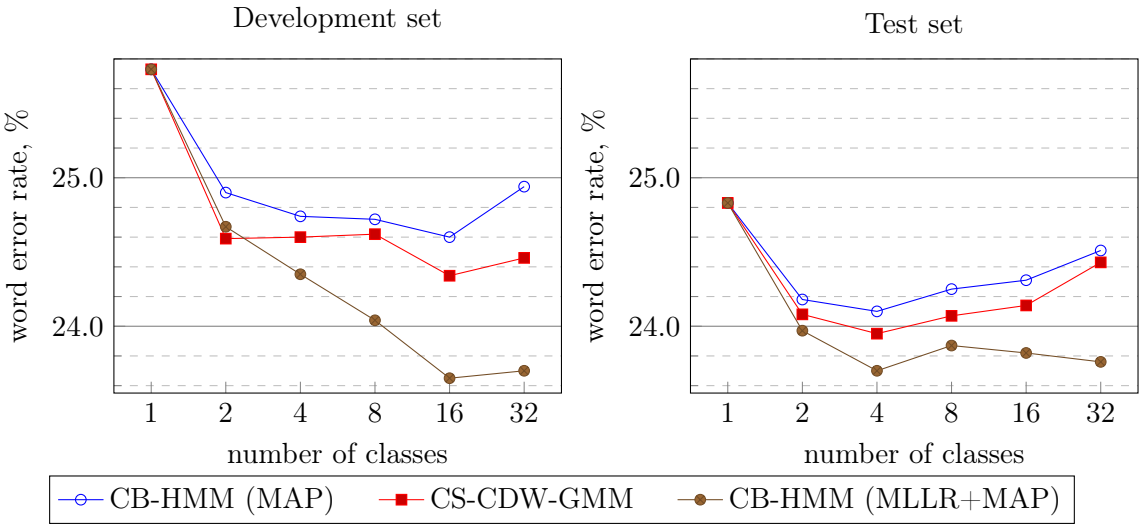

FIGURE 5.10 – WER computed on ESTER2 data with full *Class-Based HMM* (CB-HMM) and *Class-Structured with Class-Dependent Weights GMM* (CS-CDW-GMM)

presented in Table 5.3. Overall, CS-CDW-GMM performs better than MAP adaptation of all model parameters. However, the large amount of data associated with the classes allows to reliably estimate up to 32 class-based models with MLLR+MAP. In the structured approach (as in CS-CDW-GMM), as means and variances are shared across speaker classes, it is not possible (at least in current formulation) to take advantage of MLLR adaptation of the Gaussian parameters. One advantage of CS-CDW-GMM is a significantly reduced number of parameters compared to CB-HMM. As the number of densities and the number of Gaussian components per density are relatively large, such a reduced size might be useful.

Because of the peaked distribution of the mixture weights in CS-CDW-GMM (see Figure 5.8) it is possible to achieve larger WER improvements by increasing the amount of GMM components. For example, using 16 classes and 256 components per density leads to a 23.84% WER on the development and 23.82% on the test data. Such performance of the CS-CDW-GMM is close to 16 classes of full class-based models adapted with MLLR+MAP, while the total number of parameters per density is still almost 3 times smaller for CS-CDW-GMM. Increasing the number of components for conventional (SI) or full class-based HMM did not lead to any improvements.

Classes	Decoding	Parameters per density	WER			
			Dev		Test	
Baseline SI HMM - 64 Gaussian per density						
1	1 pass	**5056**	**25.73**		**24.83**	
Baseline SI HMM - 256 Gaussian per density						
1	1 pass	**20224**	**25.98**		**25.02**	
CB-HMM - 64 Gaussian per density						
			MAP	MLLR+MAP	MAP	MLLR+MAP
2	2 pass	10112	24.90	24.67	24.18	23.97
4	2 pass	20224	24.74	24.35	24.10	23.70
8	2 pass	40448	24.72	24.04	24.25	23.87
16	2 pass	**80896**	**24.60**	**23.65**	**24.31**	**23.82**
32	2 pass	161792	24.94	23.70	24.51	23.76
CS-CDW-GMM - 64 Gaussian per density						
2	2 pass	**5120**	**24.59**	n/a	**24.08**	n/a
16	2 pass	**6016**	**24.34**	n/a	**24.14**	n/a
32	2 pass	**7040**	**24.47**	n/a	**24.43**	n/a
CS-CDW-GMM - 256 Gaussian per density						
8	2 pass	**22016**	**23.90**	n/a	**23.58**	n/a
16	2 pass	**28160**	**23.84**	n/a	**23.82**	n/a

TABLE 5.3 – Comparison of WERs and number of parameters for conventional *Speaker-Independent* (SI) HMM, *Class-Based HMM* (CB-HMM) and *Class-Structured GMM with Class-Dependent Weights* (CS-CDW-GMM)

To summarize, CS-CDW-GMM can be efficiently used for acoustic modeling in LVCSR, in particular if the size of the model is an important issue. The proposed approach performs better than MAP adaptation of class-based models. However, adapting all model parameters with MLLR+MAP allows to significantly reduce the resulting WERs. Also, the peaked distribution of the mixture weights allows to increase the total number of components per density in CS-CDW-GMM and to achieve a WER similar to MLLR+MAP adapted models, while requiring

less parameters (28160 parameters per density for CS-CDW-GMM with 256 Gaussian components and 16 classes versus 80896 parameters for the corresponding CB-HMM with 64 Gaussian components per density).

5.4 Conclusion

This chapter has proposed and investigated component structuring with respect to speaker classes for improved acoustic modeling. *Class-Structured* model allows to build a link between GMM components and segment-level variability classes (such as speaker class) in contrast to implicit frame-based clustering that takes place in training of the components of the conventional HMM-GMM.

Some modifications are required to take advantage of the proposed structuring. Although the components represent different speaker classes, they are mixed in GMM pdf computation with the corresponding weights. In order to efficiently use the class-structured architecture, it is proposed to combine class-structuring with *Class-Dependent Weights* (CS-CDW-GMM). Conditioning only the mixture weights on the speaker classes (and sharing class-structured means and variances across different classes) allows to build a compact, but efficient model. As a result of joint re-estimation of class-dependent mixture weights and shared means and variances, the weights are larger for the components initially associated with the corresponding classes. For decoding, the class is a-priori estimated at the segment level and the corresponding set of mixture weights is selected.

Experiments performed on both small vocabulary *TIDIGITS* and large vocabulary radio broadcast *ESTER2* data show that the model has certain advantages. For small-vocabulary ASR with a relatively limited training data, it allows to build more class-dependent models by efficiently sharing the parameters and to significantly outperform both gender-age-dependent models and class-based models. For large vocabulary data the approach also provides good results and allows to significantly reduce the model size compared to full class-adapted models. Also, unlike the relatively flat distribution of the conventional HMM-GMM mixture weights, the weights of CS-CDW-GMM are sharp. This allows to reliably estimate a larger number of components per density and finally achieve with CS-CDW-GMM performance similar to the full class-based HMM, while having a much smaller total number of parameters. The general framework of the component structuring and the intuitive approach relying on class-dependent mixture weights is further extended and investigated in the following chapters.

6

Explicit speech trajectory modeling with Stranded Gaussian Mixture Model

Explicitly introducing speaker class information by either relying on multi-modeling with *Class-Based HMM* (CB-HMM) (as in Chapter 4) or on *Class-Structuring with Class-Dependent Weights GMM* (CS-CDW-GMM) (as in Chapter 5) approaches improves the ability of the HMM to handle heterogeneous speech data. Both models in different ways add long temporal constraints by either conditioning all model parameters on the segment-level classes, or by structuring the GMM components and conditioning only the mixture weights on these classes.

Such constraints can be seen as an attempt to partially relax the HMM *observation independence* assumption defined earlier in Section 2.2.1. In conventional HMM, the parameters of the observation density depend only on the state that is associated with this density. In CB-HMM and in CS-CDW-GMM the observations are also independent on each other, but they globally depend on the segment classes.

A different approach is considered in this chapter. Instead of conditioning the model parameters on global variables (such as speaker classes), some additional local dependencies between the model observation densities are explicitly introduced. Previously, in Section 2.3.2 various approaches based on *Multilevel Speech Dynamics* (MSDM), *Segmental* (SM), *Segmental Mixture* (SMM), *Artificial Neural Network* (ANN) and other models were described. All these approaches either relax the conditional independence assumptions of HMM-GMM, or use a different model structure (or better training algorithms) in order to better model dynamic properties of speech and consequently to improve the ASR robustness to non-phonetic variability in the speech signal.

A particular model called *Stranded Gaussian Mixture Hidden Markov Model* (or simply *Stranded GMM*: StGMM [1]) is studied in this chapter in greater detail. StGMM has strong relations to *Conditional Gaussian* model [Wellekens, 1987], which was recently extended and applied on a noise-robust ASR task [Zhao and Juang, 2012] achieving a 10% relative improvement of the word error rate. StGMM is an extension of HMM-GMM, which adds dependencies between components of HMM-GMM by expanding the observation density and replacing state-conditioned sets of mixture weights by *Mixture Transition Matrices* (MTMs). MTM consists of transition probabilities between Gaussian components of adjacent states, as schematically shown in Figure 6.1.

The objective of this chapter is to present a detailed theoretical formalization of the model structure, the corresponding training and decoding algorithms, as well as to analyze the model properties and performance. This chapter is organized as follows. Section 6.1 presents the gene-

1. The notation StGMM is used in order to avoid confusion with SGMM, which denotes the *Subspace Gaussian Mixture Model* described in Section 2.4.1.

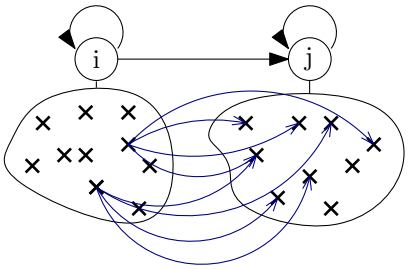

FIGURE 6.1 – Schematic representation of the *Stranded Gaussian Mixture Model* (StGMM)

ral model structure. Section 6.2 describes the EM equations and the efficient Forward-Backward algorithm for StGMM training. Section 6.3 describes the extended Viterbi decoding algorithm. Section 6.4 describes a detailed analysis of the Mixture Transition Matrices and the ASR experiments conducted on two different types of data: clean digits with gender and age variability (*TIDIGITS*[1]) and small-vocabulary speech distorted by non-stationary noise (*CHiME*[2]). Section 6.5 concludes the chapter.

Contents

6.1 Formulation of Stranded Gaussian Mixture Model

Consider the HMM state sequence $\boldsymbol{\mathcal{Q}} = (q_1, \ldots, q_T)$, the observation sequence $\boldsymbol{\mathcal{O}} = (\mathbf{o}_1, \ldots, \mathbf{o}_T)$, and the sequence of components of the observation density $\boldsymbol{\mathcal{M}} = (m_1, \ldots, m_T)$, where every $\mathbf{o}_t$ is an observation feature vector, $m_t \in \{1, \ldots, M\}$ is the index of the GMM density component associated with a state $q_t \in \{1, \ldots, N\}$ at time t, M denotes the number of such components in the mixture and N is the total number of HMM states. The difference of StGMM from the conventional HMM-GMM consists in the fact, that Stranded GMM expands the observation densities of HMM-GMM and explicitly adds dependencies between GMM components of adjacent states replacing the sets of *mixture weights* by *Mixture Transition Matrices* (MTMs) (compare Figure 6.2 versus Figure 2.7).

1. Clean digits from different adult male, female and child speakers (see details in Appendix A.2.2)
2. 1st track of *CHiME* challenge. Short speech commands distorted with non-stationary noise (see details in Appendix A.2.1)

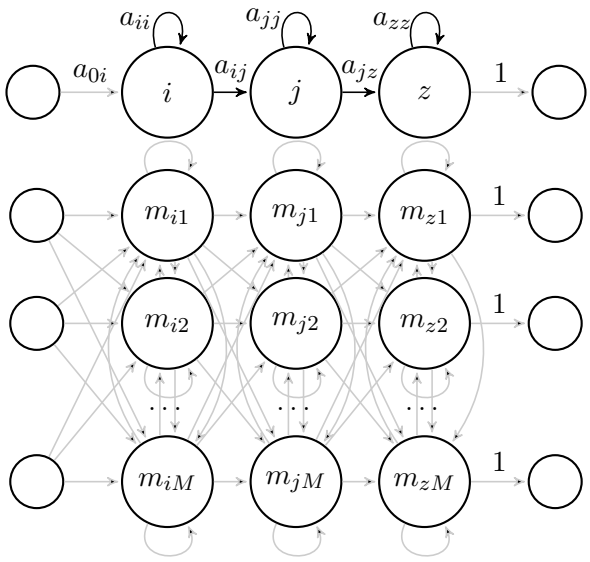

FIGURE 6.2 – Stranded GMM with schematic representation of the component dependencies

The joint likelihood of the observation, state and component sequences is defined as follows:

$$\begin{aligned} P(\boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\lambda}) &= P(\boldsymbol{\mathcal{O}}|\boldsymbol{\mathcal{M}}, \boldsymbol{\mathcal{Q}}, \boldsymbol{\lambda}) P(\boldsymbol{\mathcal{M}}|\boldsymbol{\mathcal{Q}}, \boldsymbol{\lambda}) P(\boldsymbol{\mathcal{Q}}|\boldsymbol{\lambda}) \\ &= \prod_{t=1}^{T} P(\mathbf{o}_t|m_t, q_t) P(m_t|q_{t-1}, q_t, m_{t-1}) P(q_t|q_{t-1}) \end{aligned} \tag{6.1}$$

Most of the terms in this definition of StGMM joint likelihood are similar to those of the conventional HMM defined earlier in Section 2.2.2. The set of StGMM parameters consists of the following components:

1. State transition probability (or simply *transition probability*) $a_{ij} = P(q_t = j|q_{t-1} = i)$, which is equivalent to HMM transition probability;
2. *Component observation likelihood function* $b_{jl}(\mathbf{o}_t) = P(\mathbf{o}_t|q_t = j, m_t = l)$, which denotes the probability of the observation $\mathbf{o}_t$ with respect to a single density component $m_t = l$ associated with a state $q_t = j$.

   The component observation likelihood is defined in the form of Gaussian pdf $\mathcal{N}(\mathbf{o}_t|\boldsymbol{\mu}_{jl}, \boldsymbol{\Sigma}_{jl})$ with mean vector $\boldsymbol{\mu}_{jl}$ and covariance matrix $\boldsymbol{\Sigma}_{jl}$:

   $$\mathcal{N}(\mathbf{o}_t|\boldsymbol{\mu}_{jl}, \boldsymbol{\Sigma}_{jl}) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}_{jl}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{o}_t - \boldsymbol{\mu}_{jl})^T \boldsymbol{\Sigma}_{jl}^{-1}(\mathbf{o}_t - \boldsymbol{\mu}_{jl})\right\} \tag{6.2}$$

   where n denotes the size of the feature vector $\mathbf{o}_t$;
3. *Mixture Transition Matrix* (MTM) $C^{(ij)} = \{c_{kl}^{(ij)}\}$ associated with two adjacent states i and j and composed of *mixture transition probabilities*. Mixture transition probability $c_{kl}^{(ij)} = P(m_t = l|q_{t-1} = i, q_t = j, m_{t-1} = k)$ defines the probability of moving from the component $m_{t-1} = k$ of the density associated with the state $q_{t-1} = i$ to the component $m_t = l$ of the density associated with the state $q_t = j$. MTM replaces state-associated mixture weights $\omega_{jl} = P(m_t = l|q_t = j)$ of the conventional HMM-GMM. As $c_{kl}^{(ij)}$ defines

a probability, the following constraint takes place:

$$\sum_{l=1}^{M} c_{kl}^{(ij)} = 1, \qquad \forall i, j, k. \tag{6.3}$$

6.2 Stranded GMM training algorithm

State transition probabilities, means and variances of StGMM are initialized from conventional HMM-GMM with the same number of components per density. The rows of Mixture Transition Matrices (MTMs) are initialized from the corresponding values of HMM-GMM mixture weights (i.e., initially $\forall i, k \; c_{kl}^{(ij)} = \omega_{jl}$). Then, the StGMM parameters are re-estimated in an iterative *Expectation-Maximization* (EM) manner. Experiments with flat (uniform) initialization of MTMs demonstrated similar performance of the resulting re-estimated models. It is also important to note that in the experiments with StGMM conducted in this work only 2 MTMs are used for each state (i.e., cross-phone MTMs are shared). Such sharing is not done in [Zhao and Juang, 2012], but this allows to significantly reduce the amount of model parameters.

This section presents the re-estimation formulas for updating the parameters of StGMM. The training task is formalized as follows. Given an initial set of model parameters $\boldsymbol{\lambda}'$ estimate new model parameters $\boldsymbol{\lambda}^*$, which maximize the likelihood of the observation training data. Maximizing the likelihood of the observation data given the model parameters of the Markov chain is equivalent to maximizing the auxiliary function $Q(\boldsymbol{\lambda}, \boldsymbol{\lambda}')$:

$$\boldsymbol{\lambda}^* = \arg\max_{\boldsymbol{\lambda}} Q(\boldsymbol{\lambda}, \boldsymbol{\lambda}') \tag{6.4}$$

The Q-function and the EM equations for the parameters of the StGMM are derived in the following section. Then, an efficient dynamic programming algorithm for parameter estimation is described. In order to simplify the reading, some mathematical details are omitted in this chapter. For all details see Appendix F.4.

6.2.1 Model parameters estimation. Derivation from Q-function

Considering the joint likelihood defined by Equation 6.1, the Q-function is defined as follows:

$$Q(\boldsymbol{\lambda}, \boldsymbol{\lambda}') = E_{\boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}} | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}'} [\log P(\boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}})] = \sum_{\boldsymbol{\mathcal{Q}}} \sum_{\boldsymbol{\mathcal{M}}} P(\boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}} | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \log P(\boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}} | \boldsymbol{\lambda}) \tag{6.5}$$

where summation over $\boldsymbol{\mathcal{Q}}$ and $\boldsymbol{\mathcal{M}}$ means summation over all possible sequences of states and density components.

Substituting the joint likelihood defined by Equation 6.1 and the model parameters into Equation 6.5 and replacing the logarithm of product by the sum of logarithms, the Q-function

is rewritten with respect to the model parameters as follows:

$$\begin{aligned} Q(\boldsymbol{\lambda}, \boldsymbol{\lambda}') &= \sum_{j=1}^{N}\sum_{l=1}^{M}\sum_{t=1}^{T} P(q_t = j, m_t = l|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \log b_{jl}(\mathbf{o}_t) \\ &+ \sum_{i,j=1}^{N}\sum_{k,l=1}^{M}\sum_{t=1}^{T} P(q_{t-1} = i, q_t = j, m_{t-1} = k, m_t = l|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \log c_{kl}^{(ij)} \\ &+ \sum_{i,j=1}^{N}\sum_{t=1}^{T} P(q_{t-1} = i, q_t = j|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \log a_{ij} \\ &= Q_B(\boldsymbol{\lambda}, \boldsymbol{\lambda}') + Q_C(\boldsymbol{\lambda}, \boldsymbol{\lambda}') + Q_A(\boldsymbol{\lambda}, \boldsymbol{\lambda}') \end{aligned}$$

Then, by independently maximizing $Q_A(\boldsymbol{\lambda}, \boldsymbol{\lambda}')$, $Q_C(\boldsymbol{\lambda}, \boldsymbol{\lambda}')$ and $Q_B(\boldsymbol{\lambda}, \boldsymbol{\lambda}')$ re-estimation equations for all model parameters can be derived using the method of Lagrange multipliers. The resulting equations for updating the state transition probabilities a_{ij}, mixture transition probabilities $c_{kl}^{(ij)}$, Gaussian means $\boldsymbol{\mu}_{jl}$ and variances $\boldsymbol{\Sigma}_{jl}$ are as follows (see full derivation in Appendix F.4):

$$a_{ij} = \frac{\sum_{t=1}^{T} P(q_{t-1} = i, q_t = j|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}')}{\sum_{t=1}^{T} P(q_{t-1} = i|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}')} \tag{6.6}$$

$$c_{kl}^{(ij)} = \frac{\sum_{t=1}^{T} P(q_{t-1} = i, q_t = j, m_{t-1} = k, m_t = l|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}')}{\sum_{t=1}^{T} P(q_{t-1} = i, q_t = j, m_{t-1} = k|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}')} \tag{6.7}$$

$$\boldsymbol{\mu}_{jl} = \frac{\sum_{t=1}^{T} P(q_t = j, m_t = l|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \cdot \mathbf{o}_t}{\sum_{t=1}^{T} P(q_t = j, m_t = l|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}')} \tag{6.8}$$

$$\boldsymbol{\Sigma}_{jl} = \frac{\sum_{t=1}^{T} P(q_t = j, m_t = l|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \cdot (\mathbf{o}_t - \boldsymbol{\mu}_{jl})(\mathbf{o}_t - \boldsymbol{\mu}_{jl})^T}{\sum_{t=1}^{T} P(q_t = j, m_t = l|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}')} \tag{6.9}$$

6.2.2 Forward-backward algorithm for parameter estimation

In order to efficiently compute the updated values of the StGMM model parameters according to EM training, the forward-backward algorithm is used. It can be seen as an extension of the standard Baum-Welch algorithm for HMM training (described in Section 2.2.3). The forward variable $\alpha_t(j, l)$ is defined as the joint probability of observing the sequence $(\mathbf{o}_1, \ldots, \mathbf{o}_t)$ and the mixture component $m_t = l$ in the state $q_t = j$ at time t. It is recursively computed as follows:

$$\alpha_t(j, l) = P(\mathbf{o}_1, \ldots, \mathbf{o}_t, q_t = j, m_t = l|\boldsymbol{\lambda}') = \sum_{i=1}^{N}\sum_{k=1}^{M} \alpha_{t-1}(i, k) a_{ij} c_{kl}^{(ij)} b_{jl}(\mathbf{o}_t) \tag{6.10}$$

In a similar way, the backward variable $\beta_t(i,k)$ is computed as the likelihood of the rest of the sequence knowing state $q_t = i$ and component $m_t = k$ for the time frame t:

$$\beta_t(i,k) = P(\mathbf{o}_{t+1}, \ldots, \mathbf{o}_T | q_t = i, m_t = k, \boldsymbol{\lambda}') = \sum_{j=1}^{N} \sum_{l=1}^{M} a_{ij} c_{kl}^{(ij)} b_{jl}(\mathbf{o}_{t+1}) \beta_{t+1}(j,l) \tag{6.11}$$

To derive the parameter re-estimation formulas, some additional variables are defined. The first one is the probability of being in state $q_{t-1} = i$ with mixture component $m_{t-1} = k$ and state $q_t = j$ with mixture component $q_t = l$ given the observation sequence $\boldsymbol{\mathcal{O}}$ computed as follows:

$$\begin{aligned} \zeta_t(i,j,k,l) =& P(q_{t-1} = i, q_t = j, m_{t-1} = k, m_t = l | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \\ =& \frac{\alpha_{t-1}(i,k) a_{ij} c_{kl}^{(ij)} b_{jl}(\mathbf{o}_t) \beta_t(j,l)}{P(\boldsymbol{\mathcal{O}} | \boldsymbol{\lambda}')} \\ =& \frac{\alpha_{t-1}(i,k) a_{ij} c_{kl}^{(ij)} b_{jl}(\mathbf{o}_t) \beta_t(j,l)}{\sum\limits_{i,j=1}^{N} \sum\limits_{k,l=1}^{M} \alpha_{t-1}(i,k) a_{ij} c_{kl}^{(ij)} b_{jl}(\mathbf{o}_t) \beta_t(j,l)} \end{aligned} \tag{6.12}$$

Another important value is the probability of being in state $q_t = j$ with mixture component $m_t = l$ given the observation sequence $\boldsymbol{\mathcal{O}}$ defined as follows:

$$\begin{aligned} \gamma_t(j,l) &= P(q_t = j, m_t = l | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \\ &= \frac{\alpha_t(j,l) \beta_t(j,l)}{P(\boldsymbol{\mathcal{O}} | \boldsymbol{\lambda}')} = \frac{\alpha_t(j,l) \beta_t(j,l)}{\sum\limits_{i=1}^{N} \sum\limits_{k=1}^{M} \alpha_t(i,k) \beta_t(i,k)} \end{aligned} \tag{6.13}$$

Using Equation 6.12 and Equation 6.13 in combination with the previously derived equations for the model parameters (i.e., Equations 6.6, 6.7, 6.8 and 6.9) leads to re-estimation formulas for StGMM parameters in terms of counts that are computed with the forward and backward variables:

$$c_{kl}^{(ij)} = \frac{\sum\limits_{t=1}^{T} P(q_{t-1} = i, q_t = j, m_{t-1} = k, m_t = l | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}')}{\sum\limits_{t=1}^{T} P(q_{t-1} = i, q_t = j, m_{t-1} = k | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}')} = \frac{\sum\limits_{t=1}^{T} \zeta_t(i,j,k,l)}{\sum\limits_{t=1}^{T} \sum\limits_{z=1}^{M} \zeta_t(i,j,k,z)} \tag{6.14}$$

$$a_{ij} = \frac{\sum\limits_{t=1}^{T} P(q_{t-1} = i, q_t = j | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}')}{\sum\limits_{t=1}^{T} P(q_{t-1} = i | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}')} = \frac{\sum\limits_{t=1}^{T} \sum\limits_{k,l=1}^{M} \zeta_t(i,j,k,l)}{\sum\limits_{t=1}^{T} \sum\limits_{u=1}^{N} \sum\limits_{v,z=1}^{M} \zeta_t(i,u,v,z)} \tag{6.15}$$

$$\boldsymbol{\mu}_{jl} = \frac{\sum_{t=1}^{T} P(q_t = j, m_t = l|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \cdot \mathbf{o}_t}{\sum_{t=1}^{T} P(q_t = j, m_t = l|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}')} = \frac{\sum_{t=1}^{T} \gamma_t(j,l) \cdot \mathbf{o}_t}{\sum_{t=1}^{T} \gamma_t(j,l)} \tag{6.16}$$

$$\boldsymbol{\Sigma}_{jl} = \frac{\sum_{t=1}^{T} P(q_t = j, m_t = l|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \cdot (\mathbf{o}_t - \boldsymbol{\mu}_{jl})(\mathbf{o}_t - \boldsymbol{\mu}_{jl})^T}{\sum_{t=1}^{T} P(q_t = j, m_t = l|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}')} = \frac{\sum_{t=1}^{T} \gamma_t(j,l) \cdot (\mathbf{o}_t - \boldsymbol{\mu}_{jl})(\mathbf{o}_t - \boldsymbol{\mu}_{jl})^T}{\sum_{t=1}^{T} \gamma_t(j,l)} \tag{6.17}$$

To summarize, the extended Baum-Welch algorithm allows to efficiently compute the re-estimation values of the StGMM model parameters increasing the likelihood of the observed training data. It requires more computations and memory than the Baum-Welch for the conventional HMM-GMM, as the forward and backward variables are computed for each state and each density component.

6.3 Viterbi decoding for Stranded GMM

For decoding with StGMM the Viterbi algorithm is also extended. Namely, the *Viterbi path score* (defined in Section 2.2.4 for the conventional HMM) is extended in order to not only compute the best state sequence ending at time frame t in state j, but also to keep these values for each separate component of the observation densities:

$$\begin{aligned} v_t(j,l) &= \max_{q_1, q_2, \cdots, q_{t-1}} P(q_1, \ldots, q_{t-1}, q_t = j, m_t = l, \mathbf{o}_1, \ldots, \mathbf{o}_t|\boldsymbol{\lambda}) = \\ &= \max_i \sum_{k=1}^{M} v_{t-1}(i,k) a_{ij} c_{kl}^{(ij)} b_{jl}(\mathbf{o}_t) \end{aligned} \tag{6.18}$$

The *backpointer* is used to retrieve the best state sequence:

$$bp_t(j) = \arg\max_i \sum_{k,l=1}^{M} v_{t-1}(i,k) a_{ij} c_{kl}^{(ij)} b_{jl}(\mathbf{o}_t) \tag{6.19}$$

Assuming the worst scenario (i.e., all states connected to all states, which is not the case in general HMM topology for ASR), the Viterbi search requires N^2T operations, where N denotes the number of model states and T is the sequence length. The same estimation for StGMM leads to $(NM)^2T$ operations, where M denotes the number of components per density. In a general scenario, when K is an average number of transitions ending in a state (usually $K \ll N$), the conventional Viterbi requires KNT and StGMM Viterbi requires KNM^2T operations.

Also, in practice only few components dominate in pdf computation and the MTMs have a sparse structure. Therefore, by using efficient data structures and a non-aggressive score pruning it is possible to reduce the complexity of StGMM Viterbi to the level of the conventional HMM decoding.

6.4 Model analysis and evaluation

The following sections describe the analysis of the parameters of *Mixture Transition Matrices* (MTMs) and present the ASR performance evaluation with the *Stranded Gaussian Mixture Model* (StGMM) on two types of data with different speech variability. Experiments are first reported on the clean digits data, where most of the variability comes from the differences in speaker age and gender (*TIDIGITS*). Then, evaluation is reported on a small-vocabulary speech corrupted by various types of non-stationary noise (the data from the 1st track of the *CHiME* 2013 challenge).

6.4.1 Analysis of the mixture transition matrices

Mixture Transition Matrices (MTMs) introduce dependencies between components of the densities associated with adjacent states. The aim of this section is to experimentally investigate the distribution of the MTMs. For this experiment, the StGMM parameters are initialized from the HMM-GMM baseline with 32 Gaussian components per density (*mdl.TIDIGITS.Full*[1]) trained on clean connected digits (*TIDIGITS* training data). MTM rows are initialized from the corresponding mixture weight values. Then, all model parameters are re-estimated. Only 2 MTMs are used for each state (i.e., cross-phone MTMs are shared).

By analyzing the resulting model parameters, it was observed that MTMs, which correspond to *inter-state* transitions (i.e., $C^{(ij)}$, $i \neq j$) are substantially different from the MTMs of the *intra-state* transitions, or loops (i.e., $C^{(ii)}$). For example, Figure 6.3 shows the averaged over all HMM states inter-state and intra-state MTMs of the earlier described StGMM.

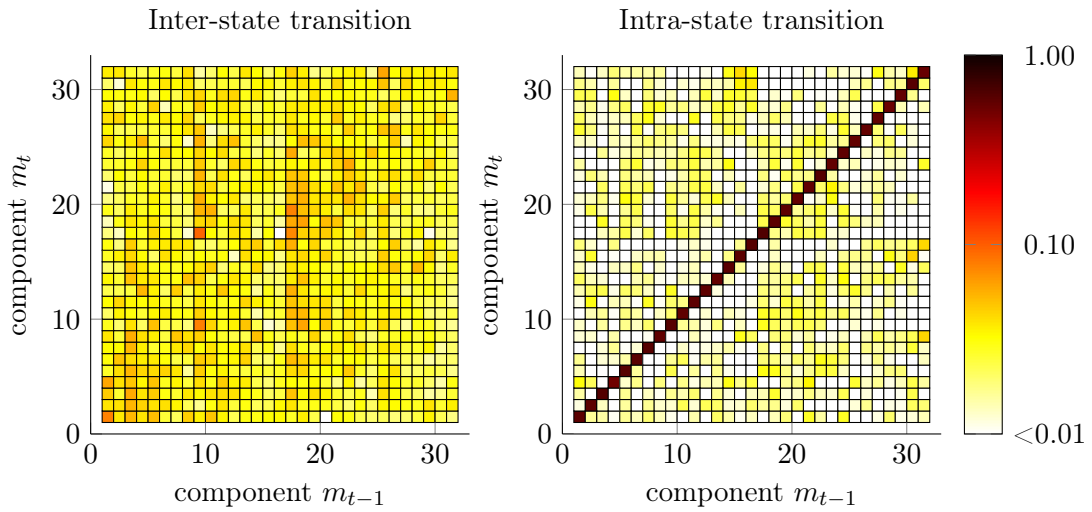

FIGURE 6.3 – Inter-state and intra-state Mixture Transition Matrices (MTMs) of StGMM trained from *TIDIGITS* data and averaged over states

Intra-state MTMs tend to have large values on diagonal (in average). This means, that the same component of the density is likely to be dominating for adjacent frames if the state label is unchanged. This can be explained by the fact that the adjacent frames associated with the same density are likely to have similar acoustic characteristics.

1. Word-dependent phones (3 state per phone), 32 Gaussian components per density, 39 cepstral features (MFCC+Δ+ΔΔ), 8kHz down-sampled (see details in Appendix B.2)

The average value of the diagonal elements of the intra-state MTMs is 0.61 ± 0.02. The distribution of both inter-state and intra-state MTMs is sparse. Similar analysis was repeated using noisy data (*CHiME*) showing the corresponding values of diagonal elements 0.67 ± 0.02. At the same time, the inter-state transitions can take place from any component to any other component (as the GMM components are trained independently and there are no constraints between components of different model states). This leads to the flat distribution of averaged values of the corresponding MTMs.

Both sparsity of all MTMs and sharp distribution of intra-sate MTMs make training and decoding problems challenging. Later in Section 6.4.3 it is experimentally shown, that sharp intra-state transitions constrain too much the trajectory and can lead to ASR performance degradation when the data contains non-stationary noise.

6.4.2 ASR experiments on connected digits with age and gender variability

As it was said earlier, recognizing connected digits is a relatively easy task. The challenge associated with *TIDIGITS* comes from the fact that the data contain recordings of both child and adult speakers [1]. If the models are trained using only the adult subset, the performance on child data is low. When learning the model on both adult and child data, a significant improvement is achieved on child data, but degradation is observed on the adult data.

In this section, an HMM baseline (same as in Section 6.4.1) trained on the full training set (*mdl.TIDIGITS.Full*) is compared to the StGMM with the same number of densities and Gaussian components. While the previous section focused on the analysis of the MTMs of this model, this section compares the corresponding WER improvements achieved with StGMM. In both cases all training data are used. To analyze the impact of different StGMM parameters on the resulting model accuracy two models are built: one with only MTM parameters re-estimated, and another one with all model parameters (MTM, μ and σ) re-estimated. Table 6.1 summarizes the WERs corresponding to the SI baseline HMM (*mdl.TIDIGITS.Full*), gender and age-dependent HMM (*mdl.TIDIGITS.Full.GenAge*) and the two StGMMs.

Model - 32 Gaussian per density	**Adult**	**Child**
mdl.TIDIGITS.Full	**1.66**	**1.88**
mdl.TIDIGITS.Full.GenAge	1.34	1.45
StGMM: MTM	**1.09**	**1.35**
StGMM: MTM+μ+σ	**1.11**	**1.27**

TABLE 6.1 – WERs on *TIDIGITS* data achieved with SI baseline HMM (*mdl.TIDIGITS.Full*), gender and age-adapted HMM (*mdl.TIDIGITS.Full.GenAge*) and two StGMMs (with only MTMs and with all parameters re-estimated)

Re-estimating only MTMs leads to a significant performance improvement. Re-estimating also means and variances does not significantly change the resulting performance. Compared to the conventional HMM-GMM, fully re-estimated StGMM improves the WER from 1.66% to 1.11% on adult and from 1.88% to 1.27% on child speech. Both improvements are statistically significant with respect to the 95% confidence interval. The StGMM performance is even better than the one achieved with the gender-age adapted baseline (*mdl.TIDIGITS.Full.GenAge*).

1. Training set: 28329 digits for adult and 12895 digits for child speech. Test set: 28554 digits for adult and 12533 digits for child speech

6.4.3 ASR experiments on noisy data

The improvements achieved with StGMM in recognizing clean speech with age and gender variability motivate further investigation of this model for different types of speech variability. This section presents a comparison of the StGMM with conventional HMM-GMM performances when processing speech corrupted by non-stationary noise. The noisy data is taken from 1st track of *CHiME* challenge (see details in Appendix A.2.1). The baseline model is again the SI HMM with 32 Gaussian components per density trained on the full training set (*mdl.CHiME.noisy* mentioned earlier in Section 6.4.1 and described in detail in Appendix B.3). This baseline is used for performance comparison and for initializing the parameters of the StGMM.

Table 6.2 shows the details of keyword recognition accuracy [1] on the *CHiME* development data [2] achieved with the HMM-GMM baseline and with different types of StGMM. The column "StGMM training" shows whether only MTMs or all model parameters are re-estimated. The column "Intra-state MTM" shows whether intra-state MTMs were re-estimated, or fixed (explained later).

Model		-6dB	-3dB	0dB	3dB	6dB	9dB	Average
baseline mdl.CHiME.noisy		**55.75**	**60.08**	69.58	77.67	80.08	84.25	**71.24**
StGMM training	Intra-state MTM							
MTM	trained	53.75	58.92	67.50	75.26	79.75	84.17	69.89
MTM	fixed	**57.08**	**61.08**	69.17	77.25	80.00	85.17	**71.63**
MTM+μ+σ	fixed	**57.83**	**62.08**	69.58	77.33	80.17	85.17	**72.03**

TABLE 6.2 – Keyword recognition accuracy (%) on the development set of *CHiME* 2013 task. The comparison is done for different approaches of StGMM training

For this set of experiments it was observed that sharp diagonal distribution of intra-state MTMs significantly hurts the performance of the recognizer (this corresponds to "Intra-state MTM - trained" row of Table 6.2). A simple "work around" approach is to keep MTMs for intra-state transitions unchanged after initialization (not re-estimate them). In this case, StGMM outperforms GMM with larger gains in the noisy part (corresponds to the rows "Intra-state MTM - fixed"). Further improvement is achieved for this dataset when means and variances are jointly re-estimated with inter-state MTMs.

As the relative improvements obtained on *CHiME* data are not as substantial as the improvements achieved on *TIDIGITS* in the previous section, the statistical significance of the results must be verified. This is done using the McNemar test [Gillick and Cox, 1989]. This test consists in analyzing the errors produced by two systems and computing the probability P of how likely the improvement is achieved by chance. Comparing HMM-GMM and StGMM in the experiments with noisy *CHiME* data leads to $P = 0.017$, which means that the results are indeed statistically significant with respect to 95% confidence interval.

In addition to the experiments described in this section, further tests on *CHiME* data are performed with more advanced feature extraction, including *speech enhancement*. Speech enhancement reduces noise in both training and test data and allows to significantly improve the ASR performance. The experiments described in Appendix E show that speech enhancement can

1. In CHiME evaluation the results are usually reported in terms of keyword accuracy
2. 600 sentences for each SNR level: -6dB, -3dB, 0dB, 3dB, 6dB, 9dB

be combined with StGMM modeling framework achieving performance improvements similar to those obtained in the experiments described in this section.

6.5 Conclusion

This chapter described the framework and the analysis of *Stranded Gaussian Mixture Model* (StGMM), which is an extension of conventional HMM-GMM with additional temporal dependencies between components of the observation densities. By replacing the mixture weights associated with a given state and a component number by *Mixture Transition Matrices* (MTMs), the conditional independence assumption of the observation densities defined for HMM is no more valid. Consequently, StGMM is a more accurate model of speech dynamics than conventional HMM-GMM.

For StGMM training, the model parameters are initialized from the conventional HMM-GMM with the same number of components per density. The rows of the *Mixture Transition Matrices* (MTMs) are initially equally initialized from the corresponding mixture weights, or simply defined uniformly. Next, all model parameters are re-estimated using the extended Baum-Welch algorithm derived in this chapter. Decoding of StGMM can also be seen as an extended version of the conventional Viterbi decoding algorithm of HMM-GMM. Although it requires M^2 times more computations (M denotes the number of components per density), the complexity can be reduced by using the fact that only few components generally dominate in the pdf computation.

The analysis of the model parameters also shows that the MTM distributions associated with inter-state and intra-state transitions are different. While the sparse MTMs of the inter-state transitions averaged over states do not follow any specific pattern and in average are distributed uniformly, the intra-state MTMs have large values in diagonal. This fact might lead to high sensitivity of the StGMM performance to a mismatch between the training and testing data.

With respect to the ASR performance with StGMM, two types of signal variability were studied. First, gender and age in clean speech were investigated with the *TIDIGITS* data. Then, the same experiments were carried on speech corrupted by non-stationary noise, as in *CHiME* task. Although StGMM was originally proposed for robust speech recognition of noisy data, the experiments described in this chapter demonstrated that it provides the largest improvement on the clean speech produced by adult and children speakers. When the signal is corrupted by non-stationary noise, StGMM improves the performance not as greatly and only if intra-state MTMs are not re-estimated.

Although they require an increased computation power, such extended models as StGMM, are certainly interesting for future research. One interesting direction is to introduce speaker information in the StGMM similar to how it was done in Class-Structured with Class-Dependent mixture Weights GMM (see Chapter 5). This idea is developed in the next chapter.

7

Class-Structuring for explicit speech trajectory modeling

This chapter combines various approaches described earlier in this thesis and proposes a novel model referred to as *Class-Structured Stranded Gaussian Mixture Model* (CS-StGMM).

In order to achieve a compact representation, which at the same time uses the speaker class information, it was proposed in Chapter 5 to train an HMM-GMM, whose density components are structured with respect to speaker classes. The model is referred to as *Class-Structured GMM* (CS-GMM). Structuring consists in associating each density component (or a set of components) with a speaker class. In order to efficiently use the class-structuring it was proposed to condition the mixture weights on the speaker classes. The resulting model is referred to as *Class-Structured with Class-Dependent Weights GMM* (CS-CDW-GMM).

The proposed CS-CDW-GMM has two limitations inherited from class-based modeling. First, the set of mixture weights associated with each class is fixed at the utterance level. On the one side, this is logical, as it is assumed that the speaker class is unchanged within an utterance. On the other side, a better parameterization might be achieved by additionally allowing the component weights to be adjusted based on local observations, as it is done in *Stranded GMM* described in Chapter 6. The second problem of both CB-HMM and CS-CDW-GMM is that an additional classification pass is required in decoding to select the corresponding class-based model.

Class-Structured Stranded Gaussian Mixture Model (CS-StGMM) proposed in this chapter attempts to combine the component class-structuring (as in CS-GMM) with additional temporal dependencies (as in StGMM). Schematically this idea is shown in Figure 7.1 for two HMM states and two classes used for the model structuring.

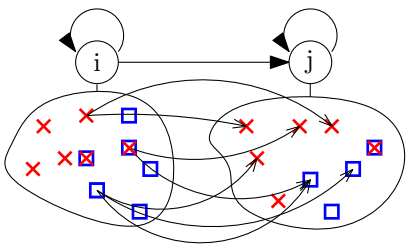

FIGURE 7.1 – Schematic representation of *Class-Structured Stranded GMM* (CS-StGMM)

The remainder of this chapter is organized as follows. Section 7.1 describes the CS-StGMM framework and the corresponding model initialization. Section 7.2 describes the ASR experiments on a small *TIDIGITS* database [1] and additional ASR experiments on the larger *NEOLOGOS* database [2]. The chapter ends with conclusion.

Contents

7.1 Formulation of Class-Structured Stranded GMM

The idea of *Class-Structured Stranded Gaussian Mixture Model* (CS-StGMM) is to structure the components of the conventional Stranded GMM, such that initially the k^{th} component of each density corresponds to a class of data (Figure 7.2). From another point of view, this idea is exactly equivalent to replacing the state- and class-dependent sets of mixture weights of the *Class-Structured with Class-Dependent Weights GMM* described in Chapter 5 by the *Mixture Transition Matrices* (MTMs) of the conventional *Stranded GMM* (StGMM) described in Chapter 6.

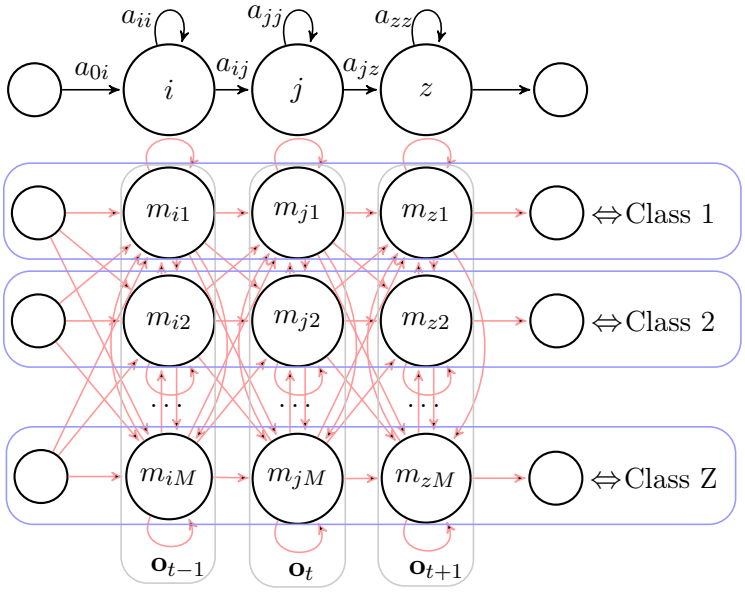

FIGURE 7.2 – *Class-Structured Stranded GMM* (CS-StGMM). Here each k^{th} component is associated with a separate class of data

In Section 2.3.2 the *speech trajectory* was defined as a path through a given sequence of HMM states $\boldsymbol{\mathcal{Q}}$ and a sequence of corresponding GMM components $\boldsymbol{\mathcal{M}}$. In the conventional HMM this trajectory is hidden and defined only by the frame-level observations due to the

1. Clean digits from different adult male, female and child speakers (see details in Appendix A.2.2)
2. 1000 adult and 1000 child speakers. French read telephone speech with different dialects and telephone network quality. 5M phones of training data and 780k phones of test data (see details in Appendix A.2.3)

conditional independence assumptions. In CS-CDW-GMM the trajectory is modeled globally and implicitly at the utterance level through the class-dependent weights that typically assign low values to the weights of the components that are irrelevant to the associated class. In contrast, in StGMM the trajectory is modeled locally and explicitly by introducing dependencies of the GMM components associated with a given frame on the components used in a previous frame. The proposed structure of CS-StGMM attempts to introduce in the modeling both global trajectory constraints by structuring the density components with respect to the speaker classes and local trajectory constraints by introducing the component transition probabilities.

Technically, CS-StGMM differs from StGMM only by the initialization of the model parameters. While the initialization of the conventional StGMM parameters is done from the conventional HMM-GMM with the corresponding number of components, CS-StGMM is initialized from a class-structured model. One possibility is to initialize the CS-StGMM parameters from the CS-GMM (i.e., just after the component structuring). Another possibility is to initialize CS-StGMM from CS-CDW-GMM (i.e., after structuring and re-estimation of the model parameters with class-dependent mixture weights). In this case, each set of class-dependent mixture weights associated with an HMM state is replaced by two *Mixture Transition Matrices* (MTMs): one matrix for inter-state transitions and another matrix for intra-state transitions. Finally, the CS-StGMM parameters are re-estimated with the extended Baum-Welch algorithm described in Section 6.2.

The advantage of initializing the StGMM parameters from CS-CDW-GMM instead of initializing them from CS-GMM is that in the first case the Gaussian means and variances are already re-estimated in a global manner (pre-trained) and intuitively it seems enough to re-estimate only MTMs. The ASR experiments described in the next section demonstrate that StGMM is indeed sensitive to the initialization procedure. As a result, initializing the model with well-trained parameters allows to achieve overall best ASR performance.

The advantage of CS-StGMM is that it explicitly parameterizes speech trajectories and still allows to automatically switch between different components (speaker classes). As the class-dependent mixture weights are replaced by class-independent MTMs, the utterance classification algorithm is no more used in decoding. When the initial structuring of CS-GMM is done from class-based models with 1 Gaussian per density, each component corresponds to a separate class. After EM re-estimation of all parameters, the diagonal elements of MTMs are dominating, which leads to the consistency of the class within utterance decoding. At the same time, non-diagonal elements allow other Gaussian components to contribute to the acoustic score computation.

7.2 Model analysis and evaluation

Earlier in Section 5.3.2 the *TIDIGITS* dataset was used for evaluation of both class-based and class-structuring approaches. This section describes the experiments on this data with *Class-Structured Stranded Gaussian Mixture Model* (CS-StGMM) comparing the performance with CS-CDW-GMM. Then, experiments performed on a larger and more realistic *NEOLOGOS* corpus of French telephone speech are reported and discussed.

7.2.1 Experiments on TIDIGITS data

The number of states and components per density, as well as the feature extraction, are similar to the earlier described baseline (*mdl.TIDIGITS.Full*[1]). Both adult and child speakers are used

1. Word-dependent phones (3 state per phone), 32 Gaussian components per density, 39 cepstral features (MFCC+Δ+$\Delta\Delta$), 8kHz down-sampled

for training the initial model and for re-estimating the CS-StGMM. Figure 7.3 summarizes the WERs achieved on adult and child test data with the following models:

1. The bars "CS-CDW-GMM" show the performance of CS-CDW-GMM achieved in Section 5.3.2. For this model, the *ML-based* clustering of the full training data is performed and the structuring is done using MLLR+MAP class-adapted models with $32/Z$ components per density, where Z denotes the number of classes. After structuring, class-dependent mixture weights are initialized and the model parameters are re-estimated with MLE for class-dependent weights and MAP for Gaussian means and variances;
2. Two types of "CS-StGMM" results are reported. They differ in how the parameters of CS-StGMM are initialized:
   (a) *initialized with CS-GMM*. For this model the structuring is done in the same way as for the CS-CDW-GMM. However, instead of using class-dependent mixture weights, MTMs are exploited and the parameters of StGMM are re-estimated with MLE. Experiments have shown that using MAP for Gaussian means and variances re-estimation led to similar results as the models fully re-estimated with MLE;
   (b) *initialized with CS-CDW-GMM*. For this model the initialization is done from the re-estimated CS-CDW-GMM. Class-dependent weights are replaced by class-independent MTMs, which are initialized uniformly in this case.

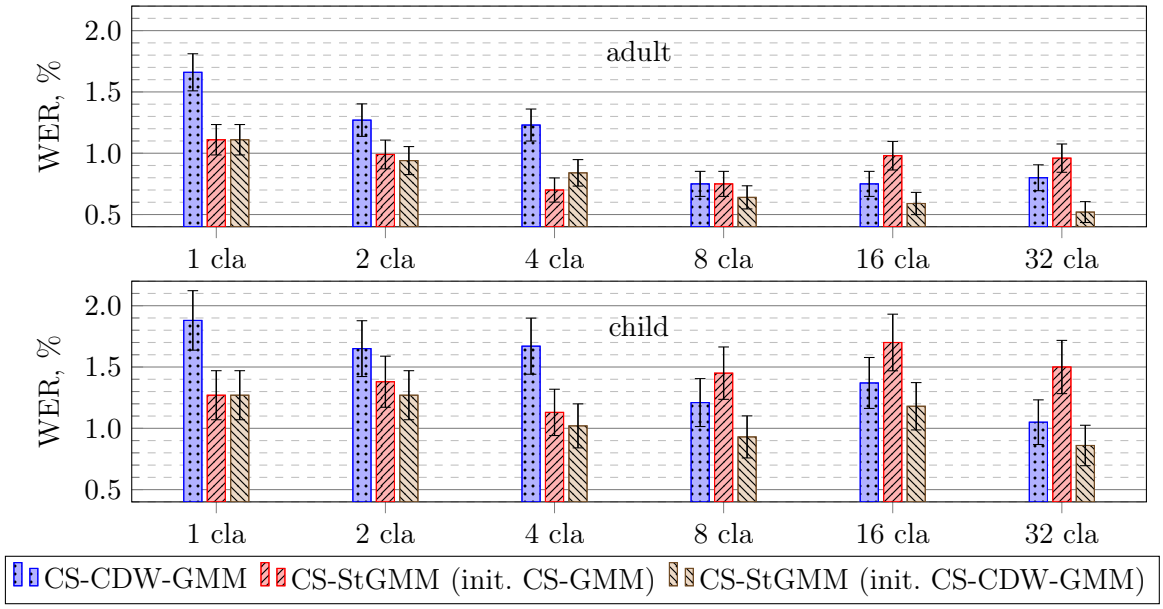

FIGURE 7.3 – WERs on *TIDIGITS* data achieved with the *Class-Structured with Class-Dependent Weights GMM* (CS-CDW-GMM) and two differently initialized *Class-Structured Stranded GMMs* (CS-StGMM). For 1 class these models correspond to the conventional HMM-GMM and conventional Stranded GMM respectively. The bars show the 95% confidence interval

Analysis of the ASR errors shows that initialization of StGMM is crucial for the resulting model accuracy. Initialization of CS-StGMM from CS-GMM with a relatively large amount of classes (starting from 8) leads to performance degradation. Initializing CS-StGMM from CS-CDW-GMM with different number of classes is always leading to WER improvement compared to the WER achieved with CS-CDW-GMM. Table 7.1 summarizes the best configurations.

Model	Classes	Decoding	Parameters/density	**Adult**	**Child**
HMM baseline	1	1 pass	78∗32+32=2528	**1.66**	**1.88**
CB-HMM	4	2 pass	4∗(78∗32+32)=10112	1.32	1.57
CS-CDW-GMM	32	2 pass	78∗32+32∗32=3520	0.80	1.05
StGMM	1	1 pass	78∗32+2∗32∗32=4544	1.11	1.27
CS-StGMM	32	1 pass	78∗32+2∗32∗32=4544	**0.52**	**0.86**

TABLE 7.1 – Summary of the best results along with the number of required decoding passes and the number of model parameters per density. Compared models are the conventional HMM-GMM baseline, Class-Based models (CB-HMM), Class-Structured with Class Dependent Weights GMMs (CS-CDW-GMM), conventional Stranded GMM (StGMM) and Class-Structured Stranded GMM built from 32 classes (CS-StGMM)

While conventional StGMM improves from 1.66% to 1.11% on adult and from 1.88% to 1,27% on child data, compared to the conventional HMM trained on full train data (adult+child), the Class-Structured StGMM (SC-StGMM) improves further achieving 0.52% WER on adult and 0.86% on child data and outperforms both CB-HMM and CS-CDW-GMM. Also, it relies on a single pass decoding (no utterance classification is required before decoding).

Figure 7.4 shows averaged over states inter-state and intra-state MTMs of CS-StGMM initialized with 32 classes. While the inter-state MTMs of conventional StGMM (shown in Figure 6.3) do not follow any specific pattern, the elements of inter-state MTMs of CS-StGMM are larger on the diagonal. The value of diagonal elements of inter-state MTMs of CS-StGMM is 0.15 ± 0.07.

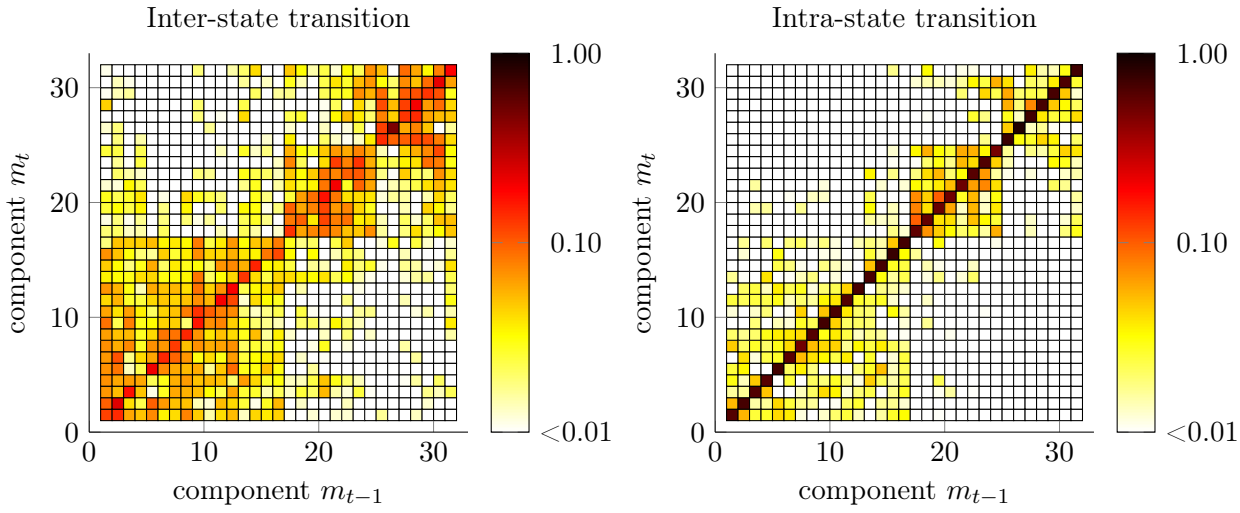

FIGURE 7.4 – Inter-state and intra-state Mixture Transition Matrices (MTMs) of CS-StGMM trained from *TIDIGITS* data and averaged over states

It is shown in the Appendix D.1 that excluding child speakers from the initial model training leads to better performances of the resulting adapted models. The initial class-based models used for structuring of CS-StGMM can also be trained from adult data only and then adapted with class-associated data that contain both adult and child speakers. This results in a better baseline and further improves the CS-StGMM performance. Appendix D.2 describes the experiments with CS-StGMM on *TIDIGITS* data, in which the models are initially trained on adult speakers.

Overall, the best performance is also achieved with CS-StGMM.

7.2.2 Experiments on NEOLOGOS data

The experiments described in this section verify the best performing approaches on a larger and more realistic data. In particular, the performance of CS-CDW-GMM and CS-StGMM are compared with the conventional HMM-GMM in phonetic decoding experiments on *NEOLOGOS* data described in Appendix A.2.3.

There are not many publications that deal with this database. Part of NEOLOGOS was used for phonetic decoding in the context of rapid speaker adaptation with Reference Model Interpolation (RMI) research [Wenxuan *et al.*, 2007]. High PER [1] shows that the task is far from being solved. The main challenges are caused by diversity of speakers and recording conditions, together with a variety of non-speech acoustic events and fillers (as speakers were doing the recordings mainly from home).

For the reported experiments, a set of 30 phonemes is used for both training and evaluation. In the chosen phoneme set, the apertures of the vowels are not considered; i.e., the open and the close /o/ are merged in a single unit, same for the open and the close /e/, as well as for the open and the close /ø/. In addition, 6 fillers, a silence and a short pause units are used in modeling. The baseline model (*mdl.NEOLOGOS.CI.pho*) consists of context-independent units modeled by 3 states. Each density has 32 Gaussian components. A 3-gram phone language model is derived from the training data. For evaluation purposes the development and test data were forced-aligned with the model trained on large vocabulary radio broadcast data and adapted on *NEOLOGOS* data. The word insertion penalty and the language model weight were optimized on development set. Table 7.2 summarizes the corresponding Phone Error Rates on *NEOLOGOS* test data [2].

CS-CDW-GMM is initialized with 32 class-based models constructed using MLLR+MAP adaptation of the SI model with a single Gaussian per class-model. The data classes are built by *ML-based* clustering of the full training data set. After initialization, the parameters are re-estimated with MAP for shared structured means and variances and MLE for mixture weights. CS-StGMM is initialized from CS-CDW-GMM, and all model parameters are then re-estimated. The corresponding Phone Error Rates of the baseline, the CS-CDW-GMM and the CS-StGMM are summarized in Table 7.2.

Model	Classes	Decoding	All	Adult	Child
HMM baseline	1	1 pass	42.42	41.16	55.55
CS-CDW-GMM	32	2 pass	**41.36**	**40.20**	**53.43**
CS-StGMM	32	1 pass	**41.14**	**40.03**	**52.75**

Table 7.2 – Phone Error Rate on *NEOLOGOS* test data

The 95% confidence interval is about ±0.11% for adult and ±0.38% for child test sets. Similar to other experiments, CS-StGMM significantly outperforms CS-CDW-GMM and does not require an additional classification pass in decoding (1 pass decoding).

The improvements achieved on NEOLOGOS task are not as large as the ones obtained with TIDIGITS. The first reason is that phonetic decoding is more challenging than digits recognition.

1. With RMI speaker adaptation approach and applying re-scoring with 4-gram LM, they achieved 37.0% Phone Error Rate on 50 adult speakers and 61.8% on 100 child speakers

2. 712773 phones for adult and 68238 phones for child speakers

Moreover, the recordings in NEOLOGOS database contain long non-speech events (for example, cough and spontaneous untranscribed commentaries of the speakers) and even background speech (for example, parents suggesting to children speakers how to read a prompt). Overall quality of the NEOLOGOS recordings is lower than the quality of TIDIGITS audios. Therefore, a large percentage of errors is rather systematic for this database.

7.3 Conclusion

This chapter described the proposed approach, which combines *Class-Structured GMM* (CS-GMM) with *Stranded GMM* (StGMM) to improve ASR performance when dealing with heterogeneous speech data. The resulting *Class-Structured Stranded GMM* (CS-StGMM) has the advantages of both models.

In conventional HMM-GMM (and conventional StGMM) the components are trained independently based only on the local frame-level observations. In contrast, by structuring the GMM components, the utterance-level speaker class information is introduced. At the same time, CS-StGMM has advantages over CS-CDW-GMM. While in CS-CDW-GMM the speech trajectory is constrained by the mixture weights globally conditioned on the utterance class, in CS-StGMM the trajectory is more accurately parameterized by the *Mixture Transition Matrices* (MTMs). This not only improves the modeling accuracy, but also allows to do an efficient single pass decoding instead of relying on the utterance-level classification (as in CS-CDW-GMM or CB-HMM).

The performance of CS-StGMM depends on the initialization of the model parameters. The overall best performance is achieved when CS-StGMM is initialized from the re-estimated CS-CDW-GMM. The performance evaluated on the small *TIDIGITS* task shows significant improvements compared to all approaches investigated in this thesis. Performance improvement is also verified on a larger and more realistic data set (*NEOLOGOS*), where a significant improvement is also observed.

8

Conclusion and future work

8.1 Conclusion

This thesis described a study on improving ASR performance by using speaker class information in the statistical acoustic modeling framework. Large non-phonetic variability in the speech signal results in a low performance of conventional HMM model. Speaker-dependent or speaker class-dependent models are known to reduce the amount of such variability to be handled by each model and to significantly improve the ASR performance. For example, a conventional approach in many ASR systems relies on building separate models for speakers of different gender. In a general situation (considered in this thesis) the speaker-related information is assumed to be unknown. In such a case unsupervised clustering is used for splitting the data into acoustically similar classes. Together with the clustering itself, this work attempts to find a more efficient way to use clustered speech data.

A broad analysis of state-of-the-art ASR demonstrates that generally two different concepts are applied to improve robustness of ASR to speech variability. One research direction deals with multi-modeling and model adaptation techniques and relies on several acoustic models. Another direction is towards improving the model structure by introducing additional dependencies or more efficiently parameterizing the distribution of the features. The thesis in some way deals with both approaches and finally attempts to combine them.

First, conventional multi-modeling approach is studied. This approach consists in unsupervised clustering of the speech utterances of the training data and in adapting the class-based models. The analysis demonstrated that even with compact transformation-based adaptation (MLLR) it is difficult to achieve a significant improvement with a relatively large number of classes due to lack of class-associated data. For handling this problem, soft clustering was investigated in combination with hypothesis combination. Soft clustering explicitly increases the amount of class-associated data by allowing one utterance to be associated with more than one class based on a tolerance classification margin. Soft margin is tuned on the development data to minimize the WER of the class-based recognition. The method allows to involve Bayesian adaptation (MAP) combined with MLLR even with a large number of classes, which leads to performance improvements. Finally, combining the hypotheses achieved with different class-based models further improves the performance.

Although class-based ASR leads to significant performance improvements, it has certain disadvantages. In particular, the number of parameters to store and estimate grows proportionally to the number of classes. Moreover, a relatively large development set is required to tune the soft margin parameter. Consequently, in the second part of the thesis a different way of involving the

speaker class information in the acoustic modeling is proposed. Instead of adapting the model parameters for each class separately, it is proposed to use a single HMM, whose GMM components are structured with respect to the speaker classes. The structuring consists in associating a given density component (or a subset of components) with a speaker class. While in the conventional HMM the observation densities (Gaussian mixtures) are associated only with single frame-based features and trained independently on each other, the structuring introduces long utterance-level constraints and leads to a strong relation between the GMM components and the speaker classes.

In order to efficiently use the structured architecture, additional dependencies are added for the component weights. Two conceptually different forms of such dependencies are proposed in this thesis. The first type of dependencies is inspired by the class-based approach. The difference is that only mixture weights are class-dependent, while Gaussian means and variances are shared across the speaker classes, but class-structured. The second type of component dependencies investigated is more related to the explicit trajectory modeling (a special type of segmental modeling). In particular, the weights of Gaussian components are replaced by *Mixture Transition Matrices* (MTMs) of so-called *Stranded GMM* (Stranded GMM). In this model the Gaussian weights do not globally depend on the speaker classes, but locally depend on the components from the previous observation density.

The experimental investigation of the techniques proposed in this thesis leads to various conclusions, which greatly depend on the speech data amount and on the requirement to the ASR system. For achieving lower errors on a large vocabulary non real-time speech transcription system with large training and development datasets, the most efficient approach would involve class-based modeling with tuned soft margin, and an additional adaptation pass in decoding, followed by hypothesis combination. For systems trained on a limited data with large speaker variability (such as variability between child and adult speakers), the proposed class-structuring is beneficial due to the efficient model parameterization. Using component transition probabilities instead of class-dependent mixture weights is also more desirable, as no segment classification is required in decoding. As a result, the approach is robust to potential errors in segment classification, and the resulting WER is lower.

8.2 Future work

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

8.2.1 Segment clustering and unsupervised classification

The KL-based clustering framework can be further extended. The simple phone selection approach described in Appendix C suggests that similar ASR performances can be achieved by using only a subset of phones. Possibly, a better phone selection strategy can be considered. Also, phone selection can be replaced by phone weighting depending on how useful or reliable these phones are for classification. Another improvement in class-based modeling is about including non-parametric features (similar to [Fukuda *et al.*, 2012]), such as SNR, speaking rate, etc. One more research direction is to investigate soft classification margin in combination with other speaker classification techniques (for example, based on i-vectors [Zhang *et al.*, 2011]).

8.2.2 Class-structuring with class-dependent weights

Class-structuring proposed in this thesis is a novel and yet not well-studied approach for dealing with heterogeneous speech data. Motivated by the performance improvements achieved with *Class-Structured GMM with Class-Dependent Weights* (CS-CDW-GMM), possible extensions of this model must be considered.

LVCSR experiments described in Section 5.3.3 demonstrated that the performance of CS-CDW-GMM with a larger number of components per density (256 in the reported experiments) becomes similar to class-based modeling (with 64 components), whereas increasing the number of components in conventional class-based ASR only degrades the performance. This leads to the idea that CS-CDW-GMM might lead to better performances with a larger number of model parameters, as the sharp distribution of class-dependent mixture weights significantly reduces the number of components that are actually used for PDF computation.

Another issue is to combine CS-CDW-GMM with a local estimation of the class, instead of full segment-level classification. For example, the class can be computed at the phone level using phone-depended GMMs.

Finally, the idea behind the component structuring can be applied for different models; for example, in Subspace GMM [Povey *et al.*, 2011a] described in Section 2.4.1.

8.2.3 Conventional and Class-Structured Stranded GMM

Stranded GMM (StGMM) and the corresponding *Class-Structured Stranded GMM* (CS-StGMM) are probably the most interesting parts of this thesis. Some theoretical and algorithmic improvements must be considered in order to study all details of these models.

All the experiments conducted in this thesis are based on Sphinx ASR toolkit. The implementation of StGMM was done only for phonetic decoding and decoding with small vocabulary (like digits). There are certain difficulties in implementing the algorithm in the lexical tree decoding used in Sphinx for LVCSR. The significant amount of computations can also be reduced by storing the component scores in a list data structure and introducing a threshold for discarding meaningless components.

Another detail to implement is a specific handling of the filler units. Ideally, propagation of the component score in the filler must be bypassed; i.e., done without using *Mixture Transition Matrices* (MTMs). Although theoretical motivation for such step is clear (as the component dependencies are somewhat random for the filler units), a naive practical implementation relying on replacing the MTM by a uniform matrix for the fillers has not resulted in a sufficient improvement.

Finally, the adaptation of StGMM is not well-studied. In theory, adapting the means and variances of StGMM with MLLR is similar to the conventional HMM-GMM adaptation. In practice, it seems that the MTMs have more discriminative power than the mixture weights. As a result, the dependencies modeled by MTMs are no more valid after linear transformation of the Gaussian means.

A

Experimental datasets

The ASR experiments described in this thesis are conducted on different datasets in order to investigate the robustness of the studied approaches to different sources of non-phonetic variability (for example, speaker age or noise). Also, datasets with different vocabulary size and different amount of training data are used. The following sections summarize all datasets used in this thesis.

A.1 French radio broadcast corpora

French radio broadcast data are used for *Large Vocabulary Continuous Speech Recognition* (LVCSR) experiments. The recordings contain different speakers (mostly adult) and recording conditions (studio and telephone quality data). The recordings also contain various non-speech events (music, background noise). For training, two datasets of different size are considered (190 hours and 300 hours of speech).

A.1.1 French radio broadcast training data

Table A.1 summarizes two training datasets used in this thesis for LVCSR. Later in this section they are described in a greater detail.

Subset name	Short description	Size
ESTER2.Train	ESTER2 campaign training set	190h
EEE.Train	Training data of ESTER2, ETAPE and EPAC	300h

Table A.1 – French radio broadcast training sets

ESTER2 training data (ESTER2.Train)

This dataset is associated with the second ESTER evaluation campaign [Galliano *et al.*, 2009]. It contains broadcast news recordings from France Inter (Inter), Radio France International (RFI), France Culture, Radio Classique, news and TV shows with foreign accented speech (radio Africa, Congo and TVME). Overall, there is about 190 hours of audio in this training set.

ESTER2+ETAPE+EPAC training data (EEE.Train)

This larger training dataset is obtained by combining parts of several training datasets from the ESTER2 and ETAPE [Gravier *et al.*, 2012] evaluation campaigns and from EPAC [Yannick Estève *et al.*, 2010] project. ETAPE training data consist of 7.5 hours of radio and 18 hours of TV data, which include news and debate shows. The EPAC corpus was obtained by manually transcribing in the EPAC project a part of non-transcribed ESTER 1 corpus. These are recordings from France Inter (30 hours), France Culture (40 hours) and RFI (25 hours) radios.

A.1.2 French radio broadcast evaluation data

For evaluating LVCSR systems, ESTER2 development and test sets are used. The original ESTER2 development set (ESTER2.Dev.20f) consists of 6 hours and test set (ESTER2.Test.26f) includes 7 hours of audio. In most of the experiments African radios were excluded from the evaluation. Tables A.2 and A.3 summarize the complete information about resulting development and test subsets of ESTER2 data.

Radio	Date	Duration	Dev.20f	Dev.11f
RFI	2007/07/07	1h	•	•
RFI	2007/07/10	20min	•	•
Inter	2007/07/10	20min	•	•
Inter	2007/07/11	20min	•	•
Inter	2007/07/12	20min	•	•
Inter	2007/07/16	1h	•	•
Inter	2007/07/23	40min	•	•
TVME	2007/07/15	15min	•	•
TVME	2007/07/16	15min	•	•
TVME	2007/07/17	15min	•	•
TVME	2007/07/18	15min	•	•
Africa 1	2007/06/18	15min	•	-
Africa 1	2007/06/19	15min	•	-
Africa 1	2007/06/14	15min	•	-
Africa 1	2007/06/28	15min	•	-
Africa 1	2007/06/13	15min	•	-
Africa 1	2007/06/25	15min	•	-
Africa 1	2007/06/08	15min	•	-
Africa 1	2007/06/26	15min	•	-
Africa 1	2007/07/10	20min	•	-
Length			7h20min	5h00min

Table A.2 – ESTER2 full and non-African development sets

A.2 Other datasets used in the thesis

Besides LVCSR experiments, some recognition tests are performed on connected digits, small-vocabulary tasks and with phonetic decoding of read speech. Table A.4 summarizes the corresponding datasets that are described in detail in the following sections.

Radio	Date	Segment	Test.26f	Test.17f
RFI	2008/01/18	20:30-20:40	•	•
RFI	2008/01/22	09:30-09:40	•	•
RFI	2008/01/22	20:30-20:40	•	•
RFI	2008/01/23	20:30-20:40	•	•
RFI	2008/01/24	20:30-20:40	•	•
RFI	2008/01/25	20:30-20:40	•	•
RFI	2008/01/28	20:30-20:40	•	•
Inter	2007/12/18	19:00-19:20	•	•
Inter	2007/12/20	19:00-19:20	•	•
Inter	2007/12/21	19:00-19:20	•	•
Inter	2008/01/17	10:00-11:00	•	•
Inter	2008/01/18	10:00-11:00	•	•
Inter	2008/01/24	19:20-20:00	•	•
TVME	2007/12/19	21:35-21:50	•	•
TVME	2007/12/21	21:35-21:50	•	•
TVME	2008/01/07	21:35-21:50	•	•
TVME	2008/01/08	21:35-21:50	•	•
Africa 1	2008/02/04	19:00-19:10	•	-
Africa 1	2008/02/05	19:00-19:10	•	-
Africa 1	2008/02/06	12:00-12:10	•	-
Africa 1	2008/02/07	12:00-12:10	•	-
Africa 1	2008/02/08	12:00-12:10	•	-
Africa 1	2008/02/09	12:00-12:10	•	-
Africa 1	2008/02/10	12:00-12:10	•	-
Africa 1	2008/02/11	12:00-12:10	•	-
Africa 1	2008/02/12	12:00-12:10	•	-
Length			7h20min	5h50min

TABLE A.3 – ESTER2 full and non-African test sets

Dataset	Short description	Vocabulary	Speakers	Train	Dev	Test
CHiME	6-word commands with noise	35 keywords	34	17k sentences	3.6k sentences	3.6k sentences
TIDIGITS	connected digits	digits 0-9 and "o" letter	326 (225 adult, 101 child)	41.2k digits (28.3k adult, 12.9k child)	–	41.1k digits (28.6k adult, 12.5k child)
NEOLOGOS	telephone speech used for phonetic decoding	3.3k words (not used in phonetic decoding)	2000 (1000 adult, 1000 child)	5M phones (4.4M adult, 0.6M child)	13.6k phones (10.7k adult, 2.9k child)	781.0k phones (712.8k adult, 68.2k child)

TABLE A.4 – Short summary of datasets and their size.

A.2.1 Noisy speech data (CHiME)

The 1st track of CHiME challenge [Vincent *et al.*, 2013] aims to evaluate the ASR performance when dealing with various recording conditions and channel distortions on a small-vocabulary task. Non-stationary noise in CHiME is added to the utterances and the microphone movements are modeled. In the challenge, the task is to recognize a digit and a letter tokens in sequences of 6 words of the following format (with an example sequence below):

<cmd:4>	<color:4>	<prep:4>	**<letter:25>**	**<numb:10>**	<adv:4>
SET	RED	IN	*B*	*THREE*	NOW

Overall, there are 10 possible keywords for the digits and 25 keywords for the letters. The training set consists of 17000 utterances, which come from 34 different speakers (500 utterances per speaker). The development and the test sets contain 3600 utterances (600 for each SNR level). The utterances are corrupted by various types of non-stationary background noise with SNR from -6 to 9 dB.

A.2.2 Connected digits data with children speech (TIDIGITS)

TIDIGITS connected digits task [Leonard and Doddington, 1993] is used for experiments focusing mostly on recognizing child and adult speech. There are 225 adult speakers (111 male, 114 female) and 101 child speakers (50 boys and 51 girls). In total, the full training data set consists of 41224 digits (28329 for adult and 12895 for child speech). The test set consists of 41087 digits (28554 for adult and 12533 for child).

A.2.3 Large telephone speech corpus for phonetic decoding (NEOLOGOS)

The French database NEOLOGOS consists of 3 databases that were recorded over fixed telephone network [Charlet *et al.*, 2005]. First, IDIOLOGOS1 contains 1000 adult speakers of different gender and accent. Each speaker produces 50 sentences. Second, each of 200 selected speakers from IDIOLOGOS1 additionally produces 450 sentences for IDIOLOGOS2 database. Finally, PAIDIOLOGOS contains 1000 children speakers, each producing 37 sentences.

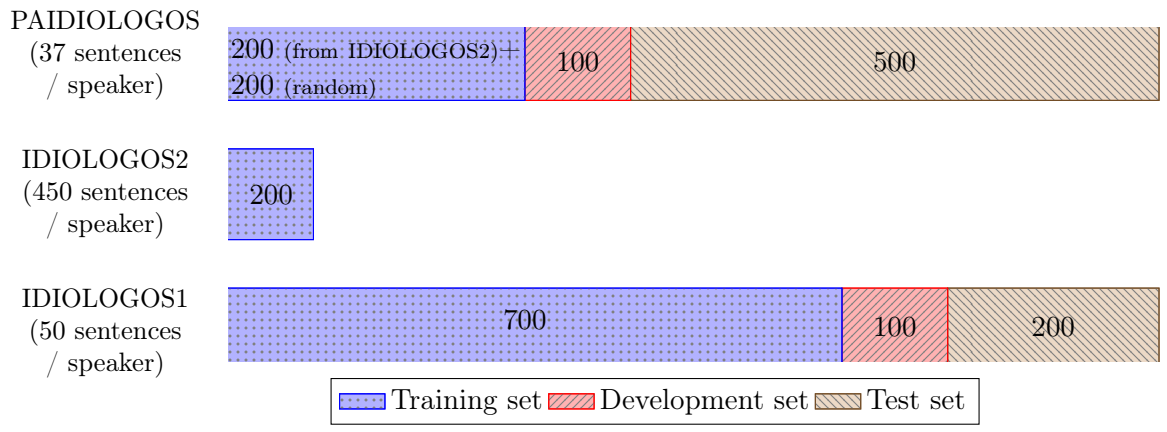

FIGURE A.1 – Distribution of speakers in NEOLOGOS training, development and test sets

Figure A.1 summarizes training, development and testing subsets used in this thesis. The training data includes 200 speakers that appear in both IDIOLOGOS1 and IDIOLOGOS2 (adult), plus random 200 speakers from IDIOLOGOS1 (adult) and 700 speakers from PAIDIOLOGOS (child). Development set includes 3 random utterances from 100 speakers of IDIOLOGOS1 and

from 100 speakers of PAIDIOLOGOS. The remaining 500 speakers of IDIOLOGOS1 and 200 speakers of PAIDIOLOGOS are used for the test set.

After removing sentences that contain non-intelligible words, the training set consists of about 5M running phones (4.4M phones for adult and 0.6M for child speech); the development set contains 13594 phones (10708 phones for adult and 2886 for child). The test set contains 781011 phones (712773 phones for adult and 68238 for child).

B

Baseline ASR systems

In this thesis ASR experiments are conducted on different datasets described in Appendix A. This section describes the configurations of the baseline experiments. When other configuration of the recognizer is used, this is explicitly indicated in the text of the document.

B.1 Automatic news transcription system

LVCSR experiments are based on the French radio broadcast transcription system of the LORIA laboratory [Jouvet and Fohr, 2013b]. The baseline decoder includes unsupervised speech detection (to distinguish the speech from non-speech events), diarization (to split audio stream on short segments that are likely to be produced by the same speaker) and classification of the recording quality (studio/telephone) and of the gender (male/female). In all experiments the front-end consists of standard 39 MFCC features (log energy + 12 cepstra with first and second derivatives) as described in Section 2.2. The back-end is based on the Sphinx ASR toolkit.

Depending on the training data used, two baselines are considered for the LVCSR experiments reported in this thesis. Each baseline has two versions: one assumes no knowledge about speaker gender and uses only separate models for studio and telephone quality data (the name ends by ".StTel") and the other is adapted with gender-dependent data using MLLR+MAP (the name ends by ".StTel.GD"). Table B.1 summarizes these baselines and the corresponding WER on non-African development and test sets of ESTER2 evaluation campaign.

Name	Training data	Senones	Gau	Lex	WER on ESTER data	
Studio-Telephone baselines (*.StTel)					Dev.11f	Test.17f
LVCSR.4500s	190h	4500	64	64k words	26.09	25.56
LVCSR.7500s	300h	7500	64	95k words	25.73	24.83
Studio-Telephone + Gender-dependent baselines (*.StTel.GD)					Dev.11f	Test.17f
LVCSR.4500s	190h	4500	64	64k words	25.23	24.46
LVCSR.7500s	300h	7500	64	95k words	24.47	23.57

Table B.1 – LVCSR baselines and the corresponding WERs

mdl.LVCSR.4500s

This model is trained on the ESTER2 training data (*ESTER2.Train*); i.e., about 190 hours of broadcast news. The model relies on 4500 tied states (senones). Each state is associated with GMM with 64 Gaussian components. The corresponding lexicon contains about 64000 words (125000 entries, including word pronunciation variants) The 3-gram language model is built from the textual annotations provided for the training data of the ESTER2 evaluation campaign and additional text materials.

mdl.LVCSR.7500s

The second baseline is built from the extended training dataset (*EEE.Train*) corresponding to 300 hours of radio broadcast and TV data , which allows to increase the number of senones up to 7500. 64 Gaussian components are used in each pdf. The lexicon and the language models are different for this baseline and the associated experiments. The lexicon contains 96000 words (145000 if pronunciation variants are counted) selected using a neural network-based approach [Jouvet and Langlois, 2013]. Part of the pronunciation lexicon is extracted from BDLEX and other pronunciations are obtained using a CRF-based (Conditional Random Field) and a JMM-based (Joint-Multigram Model) grapheme-to-phoneme converter [Bisani and Ney, 2008; Illina *et al.*, 2011; Jouvet *et al.*, 2012a]. The 3-gram language model is trained from text data extracted from newspapers, radio broadcast shows transcripts, French Gigaword corpus and recent web data (approximately 1.5G words in total)

B.2 Connected digits recognition system

Several baselines are considered for connected digits recognition. The experiments are based on *TIDIGITS* data (the dataset is described in Appendix A.2.2). The front-end and back-end of these baselines are identical. Each word-dependent phone is modeled by a 3-state HMM without skips. Each state density is modeled by 32 Gaussian components. The front-end consists of 13 standard MFCC (12 cepstral + log energy) with the first and second derivatives. Similar to other work with *TIDIGITS* [Burnett and Fanty, 1996], the signal is down sampled to 8 kHz and filtered below 200 Hz and above 3700 Hz in order to roughly simulate the telephone quality.

Table B.2 summarizes the baseline results. First, two SI models are trained from either adult subset only (*mdl.TIDIGITS.Adult*) or from the full training set that contains both adult and child speakers (*mdl.TIDIGITS.Full*). In addition, each model is used to produce the age- and gender-age dependent models. To do this, the corresponding SI models are adapted with MLLR+MAP. To assign the corresponding model to the segments of the test set, GMM classifier is learned from the training data. If class label is used in decoding, the results are similar.

Name	Model description	Adult	Child
mdl.TIDIGITS.Adult	**Training on adult data**	**0.64**	**9.92**
mdl.TIDIGITS.Adult.Age	+Age adaptation	0.64	1.22
mdl.TIDIGITS.Adult.GenAge	+Gender-Age adaptation	0.54	1.08
mdl.TIDIGITS.Full	**Training on adult+child data**	**1.66**	**1.88**
mdl.TIDIGITS.Full.Age	+Age adaptation	1.48	1.62
mdl.TIDIGITS.Full.GenAge	+Gender-Age adaptation	1.34	1.45

Table B.2 – TIDIGITS baseline WERs for SI, Age and Gender-Age adapted models

B.3 Noise-robust ASR system

In this thesis, the *CHiME* noisy speech data (described in Appendix A.2.1) are only used for verification of the Stranded GMM (StGMM) performance in Section 6.4.3. Although the dictionary is limited and the grammar is simple, the ASR with this dataset is challenging due to significant distortion of the signal. This appendix presents two baselines based on two different types of acoustic features. In both systems, each phone is modeled by an HMM with 3 states. However, the phones are not shared across different words (hence, word-dependent phones). Overall, the model has 128 context-independent 3-state HMMs without skips. Each state is modeled by 32 Gaussian mixtures.

For the first baseline experiments, standard 39 MFCC features (12 cepstra + log-energy, plus first and second order derivatives) with Cepstral Mean Normalization (CMN) are derived from noisy speech. The corresponding keyword accuracy for different SNR levels is shown in the row "mdl.CHIME.noisy" of Table B.3.

For the second baseline experiments the same features are extracted after speech enhancement using the *Flexible Audio Source Separation Toolkit* (FASST) [Ozerov and Vincent, 2011] with uncertainty information, included in the feature computation; the approach is described in [Tran *et al.*, 2014] and schematically shown in Figure B.1. Speech enhancement allows to significantly reduce the error rates compared to standard multi-condition training from noisy data. The keyword accuracy for different SNR levels achieved with enhanced features is shown in row "mdl.CHIME.enhanced" of Table B.3.

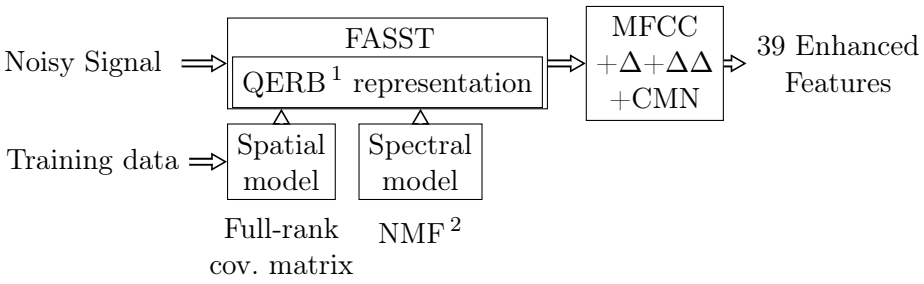

FIGURE B.1 – Schematic algorithm of the enhanced feature extraction using the *Flexible Audio Source Separation Toolkit* (FASST)

Model	-6dB	-3dB	0dB	3dB	6dB	9dB	AVG
mdl.CHiME.noisy	55.75	60.08	69.58	77.67	80.08	84.25	**71.24**
mdl.CHiME.enhanced	73.00	78.00	82.92	85.83	89.17	90.33	**83.21**

TABLE B.3 – Keyword recognition accuracy (%) on the development set of 1st track of CHiME 2013 task. SI HMM baselines with noisy and enhanced MFCC features

2. 1. Quadratic Equivalent Rectangular Bandwidth - 160 bands
2. 2. Non-negative Matrix Factorization

C

Phone selection for KL-based clustering

Two different unsupervised speech data clustering algorithms are used in this thesis: *ML-based* and *KL-based*. An advantage of *KL-based* clustering over *ML-based* clustering is that it uses phone-dependent GMMs instead of a single GMM per class. For *KL-based* clustering, the divergences $D(p_j^u||p_j^c)$ estimated for each phone j belonging to a segment u and a class c are simply averaged to obtain the segment level divergence $D_{Tot}(p^u||p^c)$ (as defined by the Equation 3.5). A general approach consists in introducing different weights for different phones so that the final segment distance is defined as a weighted combination of the phone-level distances. In this appendix, we consider a simplified approach, which aims to find and discard phones that are meaningless (or harmful) for the classification.

C.1 Computing phone scores

For selecting relevant phones for *KL-based* classification, the following naive criterion is proposed. First, for each segment u of the training set the corresponding class $C(u)$ is estimated based on minimum of the average KL divergence measure (as in the conventional classification algorithm)

$$C(u) = \arg\min_c D_{Tot}(p^u||p^c) \tag{C.1}$$

Then, for each phone j, the two following values are computed using phone-dependent distances. First, the average intra-class divergence for the phone j is computed as an average KL measure over all segments of the training data with respect to the classes assigned by the KL divergence criterion defined in Equation C.1 (i.e., using all phones)

$$DIntra_j = \frac{1}{N_{uj}} \sum_u D(p_j^u||p_j^{C(u)}) \tag{C.2}$$

where N_{uj} denotes the number of segments, in which the phone j appears at least once and the summation over u means summation over all such segments in the training set.

Similarly, inter-class divergence can be considered for each particular phone j. It is computed by averaging the divergences between each segment and each class except the one that is selected by using all phones:

$$DInter_j = \frac{1}{N_{uj} \cdot (Z-1)} \sum_u \sum_{c \neq C(u)} D(p_j^u||p_j^c) \tag{C.3}$$

where Z denotes the total number of classes and the summation is done over all classes except the class $C(u)$, which is assigned by Equation C.1.

The values $DIntra_j$ and $DInter_j$ denote the intra-class and inter-class distances computed using only phone j. They can be interpreted as contribution of the phone j in computation of the resulting divergences. Consequently, they can give some information about disagreement between the average divergences and the phone-level divergences. The "usefulness" of the phone might be determined by the following score:

$$PhScore_j = \frac{DInter_j}{DIntra_j} \tag{C.4}$$

Using the measure defined by Equation C.4, the phones can be ranked. The selected phones are then used for new clustering of the training data and for classification of the test data for class-based ASR.

C.2 Ranking phones according to score

An experimental study on ranking the phones according to the $PhScore_j$ is described in this section. For this study large set of continuous speech data (*EEE.Train*) is first clustered using *KL-based* algorithm. The classifier consists of 1 state per phone model with the GMM density modeled by 64 Gaussian components. Separate classifiers are trained for studio and telephone quality data. After clustering with conventional segment-level divergence measure, the value of $PhScore_j$ is computed and the phones are ranked (ranking is done separately for studio and telephone data). Table C.1 shows the full phone set in Sphinx notation all together with the filler units (SIL, pau, +parole+, +bouche+, +resp+, +micro+, +rire+, +divers+) sorted by decreasing $PhScore$.

Model	Phones
Telephone data	pau, j, euf, e, ng, swa, ai, eh, eu, y, +parole+, a, h, in, gn, au, i, w, g, oh, an, o, m, +divers+, z, u, +bouche+, on, +resp+, l, v, d, ge, b, SIL, r, n, t, +micro+, ch, p, s, k, +musique+, f, +rire+
Studio data	s, pau, j, ch, z, ng, +micro+, euf, ge, t, +resp+, e, y, swa, +divers+, d, eh, eu, f, ai, p, b, h, a, w, au, g, k, in, i, +parole+, SIL, gn, o, oh, an, +bouche+, on, u, n, m, r, l, +rire+, v, +musique+

TABLE C.1 – List of acoustic units sorted by decreasing $PhScore_j$ value (Equation C.4)

Ranked phone sets do not really correspond to the initial expectation that fillers and consonants are not relevant for the classification. While music (+musique+) and laugh (+rire+) filler units are at the bottom of the list, short pause (pau) is useful according to the selected score. Another observation is that some consonants and fricatives seem to be irrelevant for the classification of the telephone quality data, though some of them are useful according to the selected criterion (for example, the phone "s" is at the top of the list for studio, but almost at the bottom for telephone data)

C.3 ASR performance evaluation with phone selection

After scoring and sorting the phones by *PhScore* value, KL-based clustering of whole training data (*EEE.Train*) is done using different subsets of phones, and new class-based models are constructed by adapting the studio and telephone models with 7500 senones and 64 Gaussian components per density (*mdl.LVCSR.7500s.StTel*) with MLLR. Table C.2 summarizes the performance of class-based ASR achieved with the classification based on full phone set and with the classification based on the reduced sets (30 and 15 best phones from the lists shown in Table C.1) with 16 and 32 class-based models on development non-African subset of ESTER2 data (*ESTER2.Dev.11f*).

# classes	Phone set used in classification		
	Full (baseline)	**30 best**	**15 best**
16 classes	23.84	23.93	24.06
32 classes	23.60	23.62	23.92

TABLE C.2 – WER of class-based ASR on development set with *KL-based* classification using full and partial sets of units

Discarding 1/3 phones (30 best) leads to similar ASR performance as if all phones are used in classification. When 2/3 of the phones are discarded, the performance slightly degrades.

C.4 Conclusion

KL-based clustering with phone-dependent GMMs has an additional flexibility compared to a simple ML-based clustering with phone-independent GMM. This appendix presented an intuitive approach for discarding the phones, which are less relevant for the classification.

Note that this measure is not pretending to be the best choice and the topic was not investigated in detail. For example, this approach does not take into account the phone errors. As a result, some of the phones that are useful for the training data clustering might fail in test data because of errors in the first decoding pass.

D

Evaluation on TIDIGITS with initial model trained on adult data only

D.1 CS-CDW-GMM

This appendix describes the experiment on *TIDIGITS* data similar to the experiments described in Section 5.3.2 using different initial models. The objective is still to compare SI, gender-dependent and gender-age-dependent models with *Class-Based modeling* (CB-HMM) and *Class-Structured with Class-Dependent mixture Weights GMM* (CS-CDW-GMM). While in Section 5.3.2 it is assumed that the initial SI model is trained from mixed adult and child data, in this appendix the initial SI model is trained using only adult data. The baselines for this experiment are described in Table B.2 and consist of the following:

— *SI model* (SI HMM) trained on adult data (*mdl.TIDIGITS.Adult*). In all models, each state is modeled by 32 Gaussian components and standard 39 cepstral coefficients (MFCC+Δ+ΔΔ);

— *Gender-age-dependent* model (Gen+Age HMM) achieved by adapting the parameters of the SI model with MLLR+MAP using 4 speaker classes (man, woman, boy, girl). Also, a GMM classifier with 256 Gaussian components is trained for classifying the test data.

The same *ML-based* classes as in Section 5.3.2 are used for constructing CB-HMM and CS-CDW-GMM. Table D.1 summarizes the best performances achieved with two baselines and the proposed approaches.

Model	Details	Classes	Parameters/density	Adult	Child
SI HMM	1 pass	1	$78{\cdot}32{+}32{=}2528$	0.64	9.92
Gen+Age HMM	2 pass	4	$4{\cdot}(78{\cdot}32{+}32){=}10112$	0.54	1.08
CB-HMM	2 pass	4	$4{\cdot}(78{\cdot}32{+}32){=}10112$	0.63	0.96
CB-HMM	2 pass	32	$32{\cdot}(78{\cdot}32{+}32){=}80896$	2.23	2.68
CS-CDW-GMM	2 pass	4	$78{\cdot}32{+}32{\cdot}4{=}2624$	**0.57**	**0.94**
CS-CDW-GMM	2 pass	32	$78{\cdot}32{+}32{\cdot}32{=}3520$	**0.77**	**1.31**

TABLE D.1 – WERs and the number of model parameters per density on the *TIDIGITS* task achieved with SI (SI HMM) and Gender-Age-Dependent (Gen+Age HMM) baselines, Class-Based ASR (CB-HMM) and Class-Structured with Class-Dependent Weights GMM (CS-CDW-GMM)

Compared to the experiments described in Section 5.2, the WER achieved with gender-age-dependent models (Gen+Age HMM) is very low. CS-CDW-GMM provides similar results as CB-HMM. Also, the models that rely on unsupervised clustering yield similar performances as the supervised Gen+Age HMM. Increasing the number of classes leads to performance degradation. This may be due to classification errors that become crucial if the models are too specific to the data associated with corresponding classes.

D.2 CS-StGMM

This appendix completes the experiments described in Appendix D.1 by using CS-CDW-GMM to initialize the *Class-Structured Stranded GMM* (CS-StGMM). The experiment is similar to the one described in Section 7.2.1, except that the initial models are trained on the adult data only.

As detailed in Chapter 6, the difference of CS-StGMM from the conventional StGMM consists in the initialization of the model parameters. While in StGMM the parameters are initialized from the conventional HMM-GMM, the CS-StGMM is initialized either from *Class-Structured GMM* (CS-GMM), or from *Class-Structured with Class-Dependent Weights GMM* (CS-CDW-GMM). It is shown in Section 7.2.1 that the CS-StGMM performance is sensitive to the initialization. The best way to initialize the model parameters is to use re-estimated CS-CDW-GMM and then replace the class-dependent mixture weights by *Mixture Transition Matrices* (MTMs). Table D.2 summarizes the performances of the described models, when the initial models are trained on adult data only. However, further adaptation and re-estimation steps rely on the full *TIDIGITS* training data (adult and child subsets).

Model	Details	Classes	Parameters/density	Adult	Child
SI HMM	1 pass	1	$78{\cdot}32{+}32{=}2528$	0.64	9.92
StGMM	1 pass	1	$78{*}32{+}2{*}32{*}32{=}4544$	**0.51**	**9.90**
CS-CDW-GMM	2 pass	4	$78{\cdot}32{+}32{\cdot}4{=}2624$	0.57	0.94
CS-StGMM	1 pass	4	$78{*}32{+}2{*}32{*}32{=}4544$	**0.48**	**0.82**
CS-CDW-GMM	2 pass	32	$78{\cdot}32{+}32{\cdot}32{=}3520$	0.77	1.31
CS-StGMM	1 pass	32	$78{*}32{+}2{*}32{*}32{=}4544$	**0.54**	**0.79**

TABLE D.2 – WERs and the number of model parameters per density on the TIDIGITS data achieved with SI (SI HMM), Class-Based ASR (CB-HMM), Class-Structured with Class-Dependent Weights GMM (CS-CDW-GMM) baselines, conventional *Stranded GMM* (StGMM) and *Class-Structured Stranded GMM* (CS-StGMM)

When the conventional StGMM is initialized from HMM-GMM parameters, the resulting model accuracy is improved on adult data and unchanged on child data. Initializing the CS-StGMM from CS-CDW-GMM parameters always leads to better performance. For CS-StGMM the results with 32 classes are similar to those obtained with 4 classes, while a degradation with 32 classes is observed for CS-CDW-GMM. MTMs allow to move between components during decoding, which makes CS-StGMM more robust than CS-CDW-GMM, which relies on classification of the segments before decoding.

E

Stranded GMM with speech enhancement

This appendix describes experiments with StGMM similar to those presented in Section 6.4.3, but using here a more advanced feature extraction that includes speech enhancement. The enhanced features are calculated with FASST [Ozerov and Vincent, 2011] as described in Appendix B.3. The SI HMM-GMM baseline (*mdl.CHiME.enhanced*) is used for both comparison and initialization of the StGMM model parameters. The training is done in the same way, as for the StGMM trained from noisy data in Section 6.4.3. The performances of the baseline GMM and StGMM with and without intra-state *Mixture Transition Matrices* (MTMs) re-estimation and of the StGMM with means and variances re-estimated are summarized in Table E.1.

Model		-6dB	-3dB	0dB	3dB	6dB	9dB	AVG
baseline mdl.CHiME.enhanced		73.00	78.00	82.92	85.83	89.17	90.33	**83.21**
SGMM training	Loop MTM							
MTM	trained	72.67	76.83	81.00	86.33	88.33	90.33	82.58
MTM	**fixed**	73.17	78.33	82.58	86.50	89.25	90.67	**83.42**
MTM+μ+σ	**fixed**	73.50	79.00	82.83	86.58	89.67	90.92	**83.75**

TABLE E.1 – Keyword recognition accuracy (%) for development set of CHiME 2013 task. The comparison is done for different approaches of StGMM training. The initialization is done from *mdl.CHiME.enhanced* baseline. The standard 39 MFCC features are extracted after speech enhancement for both train and development data

Similarly to the experiments with noisy features reported in Section 6.4.3, McNemar statistical significance test was applied [Gillick and Cox, 1989]. Comparing StGMM and HMM-GMM in the experiments with enhanced features leads to the probability $P = 0.040$ of how likely the improvement is achieved by chance. This means that the results are statistically significant (with respect to 95% confidence interval).

F

Stranded GMM parameters estimation. Derivation from Q-function

This appendix describes the detailed derivation of EM algorithm for Stranded GMM formulated in Section 6.2.

The model is parameterized by the observation sequence $\boldsymbol{\mathcal{O}} = (\mathbf{o}_1, \ldots, \mathbf{o}_T)$, state sequence $\boldsymbol{\mathcal{Q}} = (q_1, \ldots, q_T)$ and component sequence $\boldsymbol{\mathcal{M}} = (m_1, \ldots, m_T)$, where every $\mathbf{o}_t$ is an observation feature vector, $m_t \in \{1, \ldots, M\}$ is the index of the GMM density component associated with a state $q_t \in \{1, \ldots, N\}$ at time t, M denotes the number of such components in the mixture and N is the total number of HMM states. The joint likelihood is defined as follows:

$$\begin{aligned} P(\boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\lambda}) &= P(\boldsymbol{\mathcal{O}}|\boldsymbol{\mathcal{M}}, \boldsymbol{\mathcal{Q}}, \boldsymbol{\lambda}) P(\boldsymbol{\mathcal{M}}|\boldsymbol{\mathcal{Q}}, \boldsymbol{\lambda}) P(\boldsymbol{\mathcal{Q}}|\boldsymbol{\lambda}) \\ &= \prod_{t=1}^{T} P(\mathbf{o}_t|m_t, q_t) P(m_t|q_{t-1}, q_t, m_{t-1}) P(q_t|q_{t-1}) \end{aligned} \tag{F.1}$$

The goal is to derive the model parameters $\boldsymbol{\lambda}^*$ that maximize the likelihood of the observation data given initial model parameters $\boldsymbol{\lambda}'$, which is equivalent to maximizing the auxiliary function $Q(\boldsymbol{\lambda}, \boldsymbol{\lambda}')$:

$$\boldsymbol{\lambda}^* = \arg\max_{\boldsymbol{\lambda}} Q(\boldsymbol{\lambda}, \boldsymbol{\lambda}') \tag{F.2}$$

$$Q(\boldsymbol{\lambda}, \boldsymbol{\lambda}') = E_{\boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}'}[\log P(\boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}})] = \sum_{\boldsymbol{\mathcal{Q}}} \sum_{\boldsymbol{\mathcal{M}}} P(\boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \log P(\boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\lambda}) \tag{F.3}$$

where summation over $\boldsymbol{\mathcal{Q}}$ and $\boldsymbol{\mathcal{M}}$ means summation over all possible sequences of states and density components.

Substituting the joint likelihood defined by Equation 6.1 and the model parameters into Equation F.3 and replacing the logarithm of product by the sum of logarithms, the Q-function

is rewritten with respect to the model parameters as follows:

$$\begin{aligned}
Q(\boldsymbol{\lambda}, \boldsymbol{\lambda}') &= \sum_{j=1}^{N}\sum_{l=1}^{M}\sum_{t=1}^{T} P(q_t = j, m_t = l|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \log b_{jl}(\mathbf{o}_t) \\
&+ \sum_{i,j=1}^{N}\sum_{k,l=1}^{M}\sum_{t=1}^{T} P(q_{t-1} = i, q_t = j, m_{t-1} = k, m_t = l|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \log c_{kl}^{(ij)} \\
&+ \sum_{i,j=1}^{N}\sum_{t=1}^{T} P(q_{t-1} = i, q_t = j|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \log a_{ij} \\
&= Q_B(\boldsymbol{\lambda}, \boldsymbol{\lambda}') + Q_C(\boldsymbol{\lambda}, \boldsymbol{\lambda}') + Q_A(\boldsymbol{\lambda}, \boldsymbol{\lambda}')
\end{aligned}$$

Then, by independently maximizing $Q_A(\boldsymbol{\lambda}, \boldsymbol{\lambda}')$, $Q_C(\boldsymbol{\lambda}, \boldsymbol{\lambda}')$ and $Q_B(\boldsymbol{\lambda}, \boldsymbol{\lambda}')$ re-estimation equations for all model parameters can be derived using the method of Lagrange multipliers.

F.1 Estimation of the state transition probabilities

Let us solve the optimization problem for the state transition probabilities a_{ij} by considering the corresponding part $Q_A(\boldsymbol{\lambda}, \boldsymbol{\lambda}')$ of the Q-function. Note that for StGMM state transition probabilities this equation is the same as for conventional HMM:

$$Q_A(\boldsymbol{\lambda}, \boldsymbol{\lambda}') = \sum_{i,j=1}^{N}\sum_{t=1}^{T} P(q_{t-1} = i, q_t = j|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \log a_{ij} \tag{F.4}$$

subject to the following constraint:

$$\sum_{j=1}^{N} a_{ij} = 1$$

The Lagrangian for $Q_A(\boldsymbol{\lambda}, \boldsymbol{\lambda}')$ has the following form:

$$\mathcal{L}(A, \varphi) = \sum_{t=1}^{T}\sum_{i,j=1}^{N} P(q_{t-1} = i, q_t = j|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \log a_{ij} + \sum_{i=1}^{N} \varphi_i (1 - \sum_{j=1}^{N} a_{ij})$$

$$\begin{cases}
\dfrac{\partial \mathcal{L}(A, \varphi)}{\partial a_{ij}} = \dfrac{1}{a_{ij}} \displaystyle\sum_{t=1}^{T} P(q_{t-1} = i, q_t = j|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') - \varphi_i \equiv 0; \\
\dfrac{\partial \mathcal{L}(A, \varphi)}{\partial \varphi_i} = 1 - \displaystyle\sum_{j=1}^{N} a_{ij} \equiv 0.
\end{cases}$$

Substituting a_{ij} from the first equation into the second one the following system is achieved:

$$\begin{cases}
a_{ij} = \dfrac{1}{\varphi_i} \displaystyle\sum_{t=1}^{T} P(q_{t-1} = i, q_t = j|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}'); \\
\varphi_i = \displaystyle\sum_{j=1}^{N}\sum_{t=1}^{T} P(q_{t-1} = i, q_t = j|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}').
\end{cases}$$

Finally, the resulting equation for updating the state transition probabilities a_{ij} is obtained:

$$a_{ij} = \frac{\sum\limits_{t=1}^{T} P(q_{t-1} = i, q_t = j | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}')}{\sum\limits_{t=1}^{T} P(q_{t-1} = i | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}')} \tag{F.5}$$

F.2 Estimation of the mixture transition probabilities

The next part of the Q-function is used to find the re-estimation equations for the mixture transition probabilities $c_{kl}^{(ij)}$: h

$$Q_C(\boldsymbol{\lambda}, \boldsymbol{\lambda}') = \sum_{i,j=1}^{N} \sum_{k,l=1}^{M} \sum_{t=1}^{T} P(q_{t-1} = i, q_t = j, m_{t-1} = k, m_t = l | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \log c_{kl}^{(ij)} \tag{F.6}$$

subject to the constraint:

$$\sum_{l=1}^{M} c_{kl}^{(ij)} = 1, \qquad \forall i, j, k$$

The Lagrangian for $Q_C(\boldsymbol{\lambda}, \boldsymbol{\lambda}')$ has the following form:

$$\begin{aligned} \mathcal{L}(C, \varphi) = &\sum_{t=1}^{T} \sum_{i,j=1}^{N} \sum_{k,l=1}^{M} P(q_{t-1} = i, q_t = j, m_{t-1} = k, m_t = l | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \log c_{kl}^{(ij)} \\ &+ \sum_{i,j=1}^{N} \sum_{k=1}^{M} \varphi_k^{(ij)} (1 - \sum_{l=1}^{M} c_{kl}^{(ij)}) \end{aligned}$$

Further calculations are similar to the ones that were done for state transition probabilities:

$$\begin{cases} \dfrac{\partial \mathcal{L}(C, \varphi)}{\partial c_{kl}^{(ij)}} = \dfrac{1}{c_{kl}^{(ij)}} \sum\limits_{t=1}^{T} P(q_{t-1} = i, q_t = j, m_{t-1} = k, m_t = l | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') - \varphi_k^{(ij)} \equiv 0; \\ \dfrac{\partial \mathcal{L}(C, \varphi)}{\partial \varphi_k^{(ij)}} = 1 - \sum\limits_{l=1}^{N} c_{kl}^{(ij)} \equiv 0. \end{cases}$$

$$\begin{cases} c_{kl}^{(ij)} = \dfrac{1}{\varphi_k} \sum\limits_{t=1}^{T} P(q_{t-1} = i, q_t = j, m_{t-1} = k, m_t = l | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}'); \\ \varphi_k^{(ij)} = \sum\limits_{l=1}^{M} \sum\limits_{t=1}^{T} P(q_{t-1} = i, q_t = j, m_{t-1} = k, m_t = l | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}'). \end{cases}$$

$$c_{kl}^{(ij)} = \frac{\sum\limits_{t=1}^{T} P(q_{t-1} = i, q_t = j, m_{t-1} = k, m_t = l | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}')}{\sum\limits_{t=1}^{T} P(q_{t-1} = i, q_t = j, m_{t-1} = k | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}')} \tag{F.7}$$

F.3 Estimation of the density function parameters

In order to estimate the Gaussian means $\boldsymbol{\mu}_{jl}$ and variances $\boldsymbol{\Sigma}_{jl}$ associated with state j and density component l, the following part of the Q-function is maximized:

$$Q_B(\boldsymbol{\lambda}, \boldsymbol{\lambda}') = \sum_{j=1}^{N} \sum_{l=1}^{M} \sum_{t=1}^{T} P(q_t = j, m_t = l | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \log b_{jl}(\mathbf{o}_t) \tag{F.8}$$

By substituting the Equation 6.2 into Equation F.8, taking log and excluding constants (as they disappear after taking derivatives), the Q-function is rewritten with respect to Gaussian means and variances:

$$Q_B(\boldsymbol{\lambda}, \boldsymbol{\lambda}') = \sum_{j=1}^{N} \sum_{l=1}^{M} \sum_{t=1}^{T} P(q_t = j, m_t = l | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \left(-\frac{1}{2} \log |\boldsymbol{\Sigma}_{jl}| - \frac{1}{2} (\mathbf{o}_t - \boldsymbol{\mu}_{jl})^T \boldsymbol{\Sigma}_{jl}^{-1} (\mathbf{o}_t - \boldsymbol{\mu}_{jl}) \right)$$

Taking partial derivative with respect to the mean vector, and setting it to zero, the following equation is achieved:

$$\frac{\partial Q_B(\boldsymbol{\lambda}, \boldsymbol{\lambda}')}{\partial \boldsymbol{\mu}_{jl}} = \sum_{t=1}^{T} P(q_t = j, m_t = l | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \boldsymbol{\Sigma}_{jl}^{-1} (\mathbf{o}_t - \boldsymbol{\mu}_{jl}) \equiv 0$$

Solving this equation, the re-estimation formula for the Gaussian means is obtained:

$$\boldsymbol{\mu}_{jl} = \frac{\sum_{t=1}^{T} P(q_t = j, m_t = l | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \cdot \mathbf{o}_t}{\sum_{t=1}^{T} P(q_t = j, m_t = l | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}')} \tag{F.9}$$

The equation for updating the variances is derived in a similar way and finally computed as follows:

$$\boldsymbol{\Sigma}_{jl} = \frac{\sum_{t=1}^{T} P(q_t = j, m_t = l | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \cdot (\mathbf{o}_t - \boldsymbol{\mu}_{jl})(\mathbf{o}_t - \boldsymbol{\mu}_{jl})^T}{\sum_{t=1}^{T} P(q_t = j, m_t = l | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}')} \tag{F.10}$$

F.4 Proof of convergence of the Stranded GMM training algorithm

This section proves that maximizing the Q-function leads to the growth (or at least, non-degradation) of the likelihood of observed data in the Stranded Gaussian Mixture Model training.

$$Q(\boldsymbol{\lambda}, \boldsymbol{\lambda}') = E_{\boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}} | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}'} [\log P(\boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}})] = \sum_{\boldsymbol{\mathcal{Q}}} \sum_{\boldsymbol{\mathcal{M}}} P(\boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}} | \boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \log P(\boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}} | \boldsymbol{\lambda})$$

The likelihood of the observation given the model can be expressed as

$$
\begin{aligned}
\log P(\boldsymbol{\mathcal{O}}|\boldsymbol{\lambda}) &= \log \sum_{\boldsymbol{\mathcal{Q}}} \sum_{\boldsymbol{\mathcal{M}}} P(\boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\lambda}) \\
&= \log \sum_{\boldsymbol{\mathcal{Q}}} \sum_{\boldsymbol{\mathcal{M}}} P(\boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \frac{P(\boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\lambda})}{P(\boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}')} \\
&= \log E_{\boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}'} \left[\frac{P(\boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\lambda})}{P(\boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}')} \right] \\
&\geq E_{\boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}'} \left[\log \frac{P(\boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\lambda})}{P(\boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}')} \right] \\
&= E_{\boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}'} \left[\log \frac{P(\boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\lambda}) P(\boldsymbol{\mathcal{O}}|\boldsymbol{\lambda}')}{P(\boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\lambda}')} \right] \\
&= E_{\boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}'} \left[\log P(\boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\lambda}) - \log P(\boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\lambda}') + \log P(\boldsymbol{\mathcal{O}}|\boldsymbol{\lambda}') \right] \\
&= Q(\boldsymbol{\lambda}, \boldsymbol{\lambda}') - Q(\boldsymbol{\lambda}', \boldsymbol{\lambda}') + \sum_{\boldsymbol{\mathcal{Q}}} \sum_{\boldsymbol{\mathcal{M}}} P(\boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{M}}|\boldsymbol{\mathcal{O}}, \boldsymbol{\lambda}') \log P(\boldsymbol{\mathcal{O}}|\boldsymbol{\lambda}') \\
&= Q(\boldsymbol{\lambda}, \boldsymbol{\lambda}') - Q(\boldsymbol{\lambda}', \boldsymbol{\lambda}') + \log P(\boldsymbol{\mathcal{O}}|\boldsymbol{\lambda}')
\end{aligned}
$$

Rearranging the elements of the resulting equation, gives:

$$\log P(\boldsymbol{\mathcal{O}}|\boldsymbol{\lambda}) - \log P(\boldsymbol{\mathcal{O}}|\boldsymbol{\lambda}') \geq Q(\boldsymbol{\lambda}, \boldsymbol{\lambda}') - Q(\boldsymbol{\lambda}', \boldsymbol{\lambda}')$$

Therefore, if $Q(\boldsymbol{\lambda}, \boldsymbol{\lambda}') \geq Q(\boldsymbol{\lambda}', \boldsymbol{\lambda}')$, then $P(\boldsymbol{\mathcal{O}}|\boldsymbol{\lambda}) \geq P(\boldsymbol{\mathcal{O}}|\boldsymbol{\lambda}')$.
So, maximizing the Q function increases the likelihood of the observed data.

Glossary

ANN : Artificial Neural Network
ASR : Automatic Speech Recognition

BHMM : Buried Hidden Markov Model

CAT : Cluster Adaptive Training
CB-HMM : Class-Based Hidden Markov Model
CD : Context-Dependent
CD-DNN : Context-Dependent Deep Neural Network
CDW-GMM : Gaussian Mixture Model with Class-Dependent Weights
CI : Context-Independent
CLR : Cross Likelihood Ratio
CMN : Cepstral Mean Normalization
CNN : Convolution Neural Network
CS-CDW-GMM : Class-Structured with Class-Dependent Weights Gaussian Mixture Model
CS-GMM : Class-Structured Gaussian Mixture Model
CS-StGMM : Class-Structured Stranded Gaussian Mixture Model

DBN : Dynamic Bayesian Network
DFT : Discrete Fourier Transform
DP : Dynamic Programming

EM : Expectation-Maximization

FASST : Flexible Audio Source Separation Toolkit
FFT : Fast Fourier Transform

GMM : Gaussian Mixture Model

HMM : Hidden Markov Model

i-vector : Identity Vector

JFA : Joint Factor Analysis

KL : Kullback-Leibler

LM : Language Model
LPC : Linear Predictive Coding
LSTM : Long Short-Term Memory
LVCSR : Large Vocabulary Continuous Speech Recognition

MAP : Maximum A Posteriori
MFCC : Mel Frequency Cepstral Coefficients
ML : Maximum Likelihood
MLE : Maximum Likelihood Estimation
MLLR : Maximum Likelihood Linear Regression
MSDM : Multilevel Speech Dynamics Model
MTM : Mixture Transition Matrix

NMF : Non-negative Matrix Factorization

PCA : Principal Component Analysis
pdf : Probability Density Function

RBM : Resticted Boltzmann Machine
RMI : Reference Speaker Interpolation
RNN : Recurrent Neural Network
ROVER : Recognizer Output Voting Error Reduction

SCDHMM : Semi-Sontinuous Density HMM
SD : Speaker-Dependent
SGMM : Subspace Gaussian Mixture Model
SI : Speaker-Independent
SM : Segmental Model
SMM : Segmental Mixture Model
SNR : Signal to Noise Ratio
StGMM : Stranded Gaussian Mixture Models
STT : Speech-To-Text

UBM : Universal Background Model

Bibliography

[Abdel-Hamid *et al.*, 2013] Ossama Abdel-Hamid, Li Deng, and Dong Yu. Exploring convolutional neural network structures and optimization techniques for speech recognition. In *Proc. INTERSPEECH*, pages 3366–3370, Lyon, France, 2013. ISCA.

[Atal and Hanauer, 1971] Bishnu S. Atal and Suzanne L. Hanauer. Speech analysis and synthesis by linear prediction of the speech wave. *JASA*, 50:637–655, 1971.

[Bahl *et al.*, 1986] Lalit R. Bahl, Peter F. Brown, Peter V. De Souza, and Robert L. Mercer. Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In *Proc. ICASSP*, pages 49–52, Tokyo, Japan, 1986. IEEE.

[Baker, 1975] James Baker. The DRAGON system–An overview. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 23(1):24–29, 1975.

[Baum and Eagon, 1967] Leonard E. Baum and John Alonzo Eagon. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bull. Amer. Math. Soc*, 73(3):360–363, 1967.

[Baum and Petrie, 1966] Leonard E. Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *The annals of mathematical statistics*, 37(6):1554–1563, 1966.

[Beaufays *et al.*, 2010] Françoise Beaufays, Vincent Vanhoucke, and Brian Strope. Unsupervised discovery and training of maximally dissimilar cluster models. In *Proc. INTERSPEECH*, pages 66–69, Makuhari, Japan, 2010. ISCA.

[Bellegarda, 2004] Jerome R. Bellegarda. Statistical language model adaptation: review and perspectives. *Speech communication*, 42(1):93–108, 2004.

[Benzeghiba *et al.*, 2007] Mohamed Benzeghiba, Renato De Mori, Olivier Deroo, Stephane Dupont, Theodora Erbes, Denis Jouvet, Luciano Fissore, Pietro Laface, Alfred Mertins, Christophe Ris, Richard Rose, Vivek Tyagi, and Christian Wellekens. Automatic speech recognition and speech variability: A review. *Speech Communication*, 49:763–786, 2007.

[Bilmes, 1998] Jeff A. Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. *International Computer Science Institute*, 4(510), 1998.

[Bilmes, 1999] Jeff A. Bilmes. Buried Markov models for speech recognition. *Acoustics, Speech, and Signal Processing*, 2:713–716, 1999.

[Bilmes, 2004] Jeffrey A Bilmes. Graphical models and automatic speech recognition. *Mathematical foundations of speech and language processing*, 138:191–245, 2004.

[Bisani and Ney, 2008] Maximilian Bisani and Hermann Ney. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451, May 2008.

[Bourlard and Morgan., 1994] Hervé Bourlard and Nelson Morgan. Connectionist speech recognition: a hybrid approach. *Springer*, 247, 1994.

[Bridle, 2004] John S. Bridle. Towards better understanding of the model implied by the use of dynamic features in HMMs. In *Proc. ICSLP*, pages 725–728, Jeju Island, Korea, 2004. ISCA.

[Burnett and Fanty, 1996] Daniel C. Burnett and Mark Fanty. Rapid unsupervised adaptation to children's speech on a connected-digit task. In *Proc. ICSLP*, volume 2, pages 1145–1148, Philadelphia, USA, 1996. IEEE, ISCA.

[Charlet *et al.*, 2005] Delphine Charlet, Sacha Krstulovic, Frédéric Bimbot, Olivier Boëffard, Dominique Fohr, Odile Mella, Filip Korkmazsky, Djamel Mostefa, Khalid Choukri, and Arnaud Vallée. Neologos: an optimized database for the development of new speech processing algorithms. In *Proc. INTERSPEECH*, pages 1549–1552, Lisbon, Portugal, 2005. ISCA.

[Cieri *et al.*, 2004] Christopher Cieri, David Miller, and Kevin Walker. The Fisher corpus: a resource for the next generations of speech-to-text. In *Proc. LREC*, volume 4, pages 69–71, Lisbon, Portugal, 2004. ELRA.

[Cohen *et al.*, 1995] Jordan Cohen, Terri Kamm, and Andreas G. Andreou. Vocal tract normalization in speech recognition: Compensating for systematic speaker variability. *JASA*, 97(5):3246–3247, 1995.

[Dahl *et al.*, 2010] George E. Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large vocabulary speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(1):30–42, 2010.

[Davis and Mermelstein, 1980] Steven Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 28(4):357–366, 1980.

[Davis *et al.*, 1952] Michael K. Davis, R. Biddulph, and Stephen Balashek. Automatic recognition of spoken digits. *JASA*, 24(6):637, November 1952.

[Dehak *et al.*, 2011] Najim Dehak, Patrick Kenny, Réda Dehak, and Pierre Ouellet. Front-end factor analysis for speaker verification. *Audio, Speech, and Language Processing, IEEE Transactions on*, 19(4):788–798, 2011.

[Deléglise *et al.*, 2009] Paul Deléglise, Yannick Esteve, Sylvain Meignier, and Teva Merlin. Improvements to the LIUM French ASR system based on CMU sphinx: what helps to significantly reduce the word error rate? In *Proc. INTERSPEECH*, pages 2123–2126, Brighton, United Kingdom, 2009. ISCA.

[Dempster *et al.*, 1977] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[Deng and Aksmanovic, 1994] Li Deng and Mike Aksmanovic. Speech recognition using hidden Markov models with polynomial regression functions as nonstationary states. *Speech and Audio Processing, IEEE Transactions on*, 2(4):507–520, 1994.

[Deng and Li, 2013] Li Deng and Xiao Li. Machine learning paradigms for speech recognition: An overview. *Audio, Speech, and Language Processing, IEEE Transactions on*, 21(5):1–30, 2013.

[Deng, 2006] Li Deng. *Dynamic speech models: theory, algorithms, and applications*, volume 2. Synthesis Lectures on Speech and Audio Processing, 2(1) edition, January 2006.

[Dennis H. Klatt, 1977] Dennis H. Klatt. Review of the DARPA speech understanding project. *JASA*, 62:1345–1366, 1977.

[Digalakis *et al.*, 1991] Vassilios V. Digalakis, J. Robin Rohlicek, and Mari Ostendorf. A dynamical system approach to continuous speech recognition. In *Proc. ICASSP*, pages 289–292, Toronto, Canada, 1991. IEEE.

[Digalakis *et al.*, 1995] Vassilios V. Digalakis, Dimitry Rtischev, and Leonardo G. Neumeyer. Speaker adaptation using constrained estimation of Gaussian mixtures. *Speech and Audio Processing, IEEE Transactions on*, 3(5):357–366, 1995.

[Dolmazon *et al.*, 1997] Jean-Marc Dolmazon, Frédéric Bimbot, Gilles Adda, Marc El-Beze, Jean-Claude Caërou, Jérôme Zeiliger, and Martine Adda-Decker. Organisation de la première campagne AUPELF pour l'évaluation des systèmes de dictée vocale. In *Proc. l'AUPELF-UREF*, pages 13–18, Avignon, France, 1997.

[Erman *et al.*, 1980] Lee D. Erman, Frederick Hayes-Roth, Victor R. Lesser, and D. Raj Reddy. The Hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty. *ACM Computing Surveys (CSUR)*, 12(2):213–253, 1980.

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

[Fukuda *et al.*, 2012] Takashi Fukuda, Ryuki Tachibana, Upendra Chaudhari, Bhuvana Ramabhadran, and Puming Zhan. Constructing ensembles of dissimilar acoustic models using hidden attributes of training data. In *Proc. ICASSP*, pages 4141–4144, Kyoto, Japan, 2012. IEEE.

[Furui, 1986] Sadaoki Furui. Speaker-independent isolated word recognition using dynamic features of speech spectrum. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 34(1):52–59, 1986.

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

[Gales, 2000] Mark Gales. Cluster adaptive training of hidden Markov models. *Speech and Audio Processing, IEEE Transactions on*, 8(4):417–428, 2000.

[Galliano *et al.*, 2005] Sylvain Galliano, Edouard Geoffrois, Djamel Mostefa, Khalid Choukri, Jean-François Bonastre, and Guillaume Gravier. The ESTER phase II evaluation campaign for the rich transcription of French broadcast news. In *Proc. INTERSPEECH*, pages 1149–1152, Lisbon, Portugal, 2005. ISCA.

[Galliano *et al.*, 2009] Sylvain Galliano, Guillaume Gravier, and Laura Chaubard. The ESTER 2 evaluation campaign for the rich transcription of French radio broadcasts. In *Proc. INTERSPEECH*, pages 2583–2586, Brighton, United Kingdom, 2009. ISCA.

[Garofolo *et al.*, 1993] John S. Garofolo, Lori F. Lamel, William M. Fisher, Jonathon G. Fiscus, and David S. Pallett. DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1. Technical report, NASA STI/Recon Technical Report N, 93, 27403, 1993.

[Gauvain and Lee, 1992] Jean-Luc Gauvain and Chin-Hui Lee. MAP estimation of continuous density HMM: theory and applications. In *Proc. DARPA Workshop Speech Natural Language*, pages 185–190, San Mateo, USA, 1992.

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

[Gibson and Hain, 2010] Matt Gibson and Thomas Hain. Error approximation and minimum phone error acoustic model estimation. *Audio, Speech, and Language Processing, IEEE Transactions on*, 18(6):1269–1279, 2010.

[Gillick and Cox, 1989] Laurence Gillick and Stephen J. Cox. Some statistical issues in the comparison of speech recognition algorithms. In *Proc. ICASSP*, pages 532–535, Glasgow, Scotland, 1989. IEEE, IEEE.

[Gish and Ng, 1993] Herbert Gish and Kenny Ng. A segmental speech model with applications to word spotting. In *Proc. ICASSP*, pages 447–450, Minneapolis, USA, 1993. IEEE.

[Godfrey *et al.*, 1992] John J. Godfrey, Edward C. Holliman, and Jane McDaniel. SWITCHBOARD: Telephone speech corpus for research and development. In *Proc. ICASSP*, pages 517–520, San Francisco, USA, 1992. IEEE.

[Goldenthal, 1994] William Goldenthal. *Statistical trajectory models for phonetic recognition.* Phd thesis, Massachusetts Institute of Technology, 1994.

[Gong., 1997] Yifan Gong. Stochastic trajectory modeling and sentence searching for continuous speech recognition. *Speech and Audio Processing, IEEE Transactions on*, 5(1):33–44, 1997.

[Gorin and Jouvet, 2012] Arseniy Gorin and Denis Jouvet. Class-based speech recognition using a maximum dissimilarity criterion and a tolerance classification margin. In *Proc. SLT workshop*, pages 91–96, Miami, USA, 2012. IEEE.

[Gorin and Jouvet, 2013] Arseniy Gorin and Denis Jouvet. Efficient constrained parametrization of GMM with class-based mixture weights for automatic speech recognition. In *Proc. LTC-6th Language & Technologies Conference*, pages 550–554, 2013.

[Gorin and Jouvet, 2014a] Arseniy Gorin and Denis Jouvet. Component structuring and trajectory modeling for speech recognition. In *Proc. INTERSPEECH*, Singapore, 2014. ISCA.

[Gorin and Jouvet, 2014b] Arseniy Gorin and Denis Jouvet. Modélisation de trajectoires et de classes de locuteurs pour la reconnaissance de voix d'enfants et d'adultes. In *Proc. JEP*, Le Mans, France, 2014.

[Gorin and Jouvet, 2014c] Arseniy Gorin and Denis Jouvet. Structured GMM based on unsupervised clustering for recognizing adult and child speech. In *SLSP*, Grenoble, France, 2014. LNAI/LNCS.

[Gorin *et al.*, 2014] Arseniy Gorin, Denis Jouvet, Emmanuel Vincent, and Dung Tran. Investigating stranded GMM for improving automatic speech recognition. In *HSCMA workshop*, pages 192–196, Nancy, France, 2014. IEEE.

[Graff *et al.*, 1997] David Graff, Zhibiao Wu, Robert MacIntyre, and Mark Liberman. The 1996 broadcast news speech and language-model corpus. In *Proc. DARPA SLT*, pages 11–14, Boston, USA, 1997.

[Gravier *et al.*, 2012] Guillaume Gravier, Gilles Adda, Niklas Paulson, Matthieu Carré, Aude Giraudel, and Olivier Galibert. The ETAPE corpus for the evaluation of speech-based TV content processing in the French language. In *Proc. LREC*, pages 114–118, Istanbul, Turkey, 2012. ELRA.

[Grézl *et al.*, 2007] Frantisek Grézl, Martin Karafiát, Stanislav Kontár, and Jan Cernocký. Probabilistic and bottle-neck features for LVCSR of meetings. In *Proc. ICASSP*, volume 4, pages 757–760, Honolulu, USA, 2007. IEEE.

[Hain, 2002] Thomas Hain. Implicit pronunciation modelling in ASR. In *Proc. PMLA workshop*, pages 129–134, Aspen Lodge, USA, 2002. ISCA.

[Han *et al.*, 2007] Yan Han, Johan De Veth, and Lou Boves. Trajectory clustering for solving the trajectory folding problem in automatic speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 15(4):1425–1434, 2007.

[Hazen and Glass, 1997] Timothy J. Hazen and James R. Glass. A comparison of novel techniques for instantaneous speaker adaptation. In *Proc. EUROSPEECH*, pages 2047–2050, Rhodes, Greece, 1997. ISCA.

[Hemphill *et al.*, 1990] Charles T. Hemphill, John J. Godfrey, and George R. Doddington. The ATIS spoken language systems pilot corpus. In *Proc. DARPA speech and natural language workshop*, pages 96–101, 1990.

[Hermansky, 2000] Hynek Hermansky. Tandem connectionist feature extraction for conventional HMM systems. In *Proc. ICASSP*, pages 1635–1638, Istanbul, Turkey, 2000. IEEE.

[Hinton *et al.*, 2006] Geoffrey Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[Hinton *et al.*, 2012] Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, and Andrew Senior. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97, 2012.

[Holmes and Russell, 1999] Wendy J. Holmes and Martin J. Russell. Probabilistic-trajectory segmental HMMs. *Computer Speech & Language*, 13(1):3–37, 1999.

[Huang *et al.*, 2001] Chao Huang, Tao Chen, Stan Z. Li, Eric Chang, and Jian-Lai Zhou. Analysis of speaker variability. In *Proc. INTERSPEECH*, pages 1377–1380, Aalborg, Denmark, 2001. ISCA.

[Hwang and Huang, 1993] Mei-Yuh Hwang and Xuedong Huang. Shared-distribution hidden Markov models for speech recognition. *Speech and Audio Processing, IEEE Transactions on*, 1(4):414–420, 1993.

[Hwang, 1993] Mei-Yuh Hwang. *Subphonetic acoustic modeling for speaker-independent continuous speech recognition*. Phd thesis, Carnegie Mellon University, 1993.

[Illina and Gong., 1997] Irina Illina and Yifan Gong. Elimination of trajectory folding phenomenon: HMM, trajectory mixture HMM and mixture stochastic trajectory model. In *Proc. ICASSP*, pages 1395–1398, Munich, Germany, 1997. IEEE.

[Illina *et al.*, 2004] Irina Illina, Dominique Fohr, Odile Mella, and Christophe Cerisara. The automatic news transcription system: ANTS. In *Proc. INTERSPEECH*, pages 377–380, Jeju Island, Korea, 2004. ISCA.

[Illina *et al.*, 2011] Irina Illina, Dominique Fohr, and Denis Jouvet. Grapheme-to-phoneme conversion using conditional random fields. In *Proc. INTERSPEECH*, pages 2313–2316, Florence, Italy, 2011. ISCA.

[Jelinek *et al.*, 1975] Frederick Jelinek, Lalit R. Bahl, and Robert L. Mercer. Design of a linguistic statistical decoder for the recognition of continuous speech. *Information Theory, IEEE Transactions on*, 21(3):250–256, 1975.

[Jelinek, 1969] Frederick Jelinek. Fast sequential decoding algorithm using a stack. *IBM Journal of Research and Development*, 13(6):675–685, November 1969.

[Jelinek, 1990] Frederick Jelinek. Self-organized language modeling for speech recognition. In *Readings in speech recognition*, pages 450–506. 1990.

[Jelinek, 1997] Frederick Jelinek. *Statistical methods for speech recognition*. MIT press, 1997.

[Jouvet and Fohr, 2013a] Denis Jouvet and Dominique Fohr. Analysis and Combination of Forward and Backward based Decoders for Improved Speech Transcription. In *Proc. TSD*, pages 84–91, Pilsen, Czech Republi, 2013. Springer Verlag.

[Jouvet and Fohr, 2013b] Denis Jouvet and Dominique Fohr. Combining forward-based and backward-based decoders for improved speech recognition performance. In *Proc. INTERSPEECH*, pages 652–656, Lyon, France, 2013. ISCA.

[Jouvet and Langlois, 2013] Denis Jouvet and David Langlois. A machine learning based approach for vocabulary selection for speech transcription. In *Proc. TSD*, volume 8082, pages 60–67. Springer Verlag, September 2013.

[Jouvet and Vinuesa, 2012] Denis Jouvet and Nicolas Vinuesa. Classification margin for improved class-based speech recognition performance. In *Proc. ICASSP*, pages 4285–4288, Kyoto, Japan, 2012. IEEE.

[Jouvet *et al.*, 2012a] Denis Jouvet, Dominique Fohr, and Irina Illina. Evaluating grapheme-to-phoneme converters in automatic speech recognition context. In *ICASSP*, pages 4821–4824, Kyoto, Japan, March 2012. IEEE.

[Jouvet *et al.*, 2012b] Denis Jouvet, Arseniy Gorin, and Nicolas Vinuesa. Exploitation d'une marge de tolérance de classification pour améliorer l'apprentissage de modèles acoustiques de classes en reconnaissance de la parole. In *Proc. JEP-TALN-RECITAL*, pages 763–770, Grenoble, France, 2012.

[Juang and Rabiner, 1985] Biing-Hwang Fred Juang and Lawrence R. Rabiner. Mixture autoregressive hidden Markov models for speech signals. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 33(6):1404–1413, 1985.

[Juang and Rabiner, 2005] Biing-Hwang Fred Juang and Lawrence R. Rabiner. Automatic speech recognition–A brief history of the technology development. *Encyclopedia of Language and Linguistics*, 2005.

[Juang *et al.*, 1997] Biing-Hwang Fred Juang, Wu Hou, and Chin-Hui Lee. Minimum classification error rate methods for speech recognition. *Speech and Audio Processing, IEEE Transactions on*, 5(3):257–265, 1997.

[Jurafsky and Martin, 2009] Dan Jurafsky and James H. Martin. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Pearson Education, London, second edition, 2009.

[Kajarekar *et al.*, 1999] Sachin S. Kajarekar, Narendranath Malayath, and Hynek Hermansky. Analysis of sources of variability in speech. In *Proc. EUROSPEECH*, pages 343–346, Budapest, Hungary, 1999. ISCA.

[Kim *et al.*, 2004] Do Yeong Kim, Srinivasan Umesh, Mark Gales, Thomas Hain, and Phil C. Woodland. Using VTLN for broadcast news transcription. In *Proc. ICSLP*, pages 1953–1956, Jeju Island, Korea, 2004. ISCA.

[Koller and Friedman, 2009] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[Korkmazskiy *et al.*, 1997] Filipp Korkmazskiy, Biing-Hwang Juang, and Frank Soong. Generalized mixture of HMMs for continuous speech recognition. In *Proc. ICASSP*, volume 2, pages 1443–1446, Munich, Germany, 1997. IEEE.

[Kuhn *et al.*, 2000] Roland Kuhn, Jean-Claude Junqua, Patrick Nguyen, and Nancy Niedzielski. Rapid speaker adaptation in eigenvoice space. *IEEE Transactions on Speech and Audio Processing*, 8(6):695–707, 2000.

[Lammert *et al.*, 2013] Adam Lammert, Louis Goldstein, Shrikanth Narayanan, and Khalil Iskarous. Statistical methods for estimation of direct and differential kinematics of the vocal tract. *Speech Communication*, 55(1):147–161, January 2013.

[Lawson *et al.*, 2003] Aaron D. Lawson, David M. Harris, and John J. Grieco. Effect of foreign accent on speech recognition in the NATO n-4 corpus. In *Proc. EUROSPEECH*, pages 1505–1508, Geneva, Switzerland, 2003. ISCA.

[Le *et al.*, 2007] Viet-Bac Le, Odile Mella, and Dominique Fohr. Speaker diarization using normalized cross likelihood ratio. In *Proc. DARPA*, pages 1869–1872, Antwerp, Belgium, 2007. ISCA.

[Lee *et al.*, 1990] KF Lee, HW Hon, and Raj Reddy. An overview of the SPHINX speech recognition system. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 38(1):35–45, 1990.

[Lee *et al.*, 2001] Akinobu Lee, Tatsuya Kawahara, and Kiyohiro Shikano. Julius—an open source real-time large vocabulary recognition engine. In *Proc. EUROSPEECH*, pages 1691–1694, Aalborg, Denmark, 2001. ISCA.

[Lee, 1990] Kay-Fu Lee. Context-dependent phonetic hidden Markov models for speaker-independent continuous speech recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 38(4):599–609, 1990.

[Leggetter and Woodland, 1994] Christopher John Leggetter and Philip C. Woodland. *Speaker adaptation of HMMs using linear regression.* University of Cambridge, Department of Engineering, 1994.

[Leonard and Doddington, 1993] Gary R. Leonard and George Doddington. TIDIGITS speech corpus. *Texas Instruments, Inc*, 1993.

[Liu and Sim, 2012] Shilin Liu and Khe Chai Sim. Implicit trajectory modelling using temporally varying weight regression for automatic speech recognition. In *Proc. ICASSP*, pages 4761–4764, Kyoto, Japan, March 2012. IEEE.

[Liu and Sim, 2013] Shilin Liu and Khe Chai Sim. An investigation of temporally varying weight regression for noise robust speech recognition. In *Proc. INTERSPEECH*, pages 2963–2967, Lyon, France, 2013. ISCA.

[Liu *et al.*, 1993] Fu-Hua Liu, Richard M. Stern, Xuedong Huang, and Alejandro Acero. Efficient cepstral normalization for robust speech recognition. In *Proc. HLT workshop*, pages 69–74, Plainsboro, USA, 1993.

[Liu *et al.*, 2011] Cong Liu, Yu Hu, Li-Rong Dai, and Hui Jiang. Trust region-based optimization for maximum mutual information estimation of HMMs in speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 19(8):2474–2485, 2011.

[Lowerre, 1976] Bruce T. Lowerre. *The harpy speech recognition system.* Phd thesis, Carnegie Mellon University, January 1976.

[Mak *et al.*, 2006] Brian Mak, Tsz-Chung Lai, and Roger Hsiao. Improving reference speaker weighting adaptation by the use of maximum-likelihood reference speakers. In *Proc. ICASSP*, Toulouse, France, 2006. IEEE.

[Mao *et al.*, 2005] Mark Z. Mao, Vincent Vanhoucke, and Brian Strope. Automatic training set segmentation for multi-pass speech recognition. In *Proc. ICASSP*, pages 685–688, Philadelphia, USA, 2005. IEEE.

[McDermott *et al.*, 2007] Erik McDermott, Timothy J. Hazen, Jonathan Le Roux, Atsushi Nakamura, and Shigeru Katagiri. Discriminative training for large-vocabulary speech recognition using minimum classification error. *Audio, Speech, and Language Processing, IEEE Transactions on*, 15(1):203–223, 2007.

[Morgan *et al.*, 2005] Nelson Morgan, Qifeng Zhu, Andreas Stolcke, Kemal Sonmez, Sunil Sivadas, Takahiro Shinozaki, and Mari Ostendorf. Pushing the envelope-aside. *Signal Processing Magazine, IEEE*, 22(5):81–88, 2005.

[O'Shaughnessy, 2013] Douglas O'Shaughnessy. Acoustic Analysis for Automatic Speech Recognition. *Proc. IEEE*, 101(5):1038–1053, 2013.

[Ostendorf *et al.*, 1992] Mari Ostendorf, Ashvin Kannan, Owen Kimball, and J. Robin Rohlicek. Continuous word recognition based on the stochastic segment model. In *Proc. DARPA Workshop CSR*, pages 53—-58, Stanford, USA, 1992.

[Ostendorf *et al.*, 1996] Mari Ostendorf, Digalakis Vassilios, and Kimball Owen. From HMM's to segment models: A unified view of stochastic modeling for speech recognition. *Speech and Audio Processing, IEEE Transactions on*, 4(5):360–378, 1996.

[Ozerov and Vincent, 2011] Alexey Ozerov and Emmanuel Vincent. Using the FASST source separation toolbox for noise robust speech recognition. In *International Workshop on Machine Listening in Multisource Environments*, 2011.

[Panchapagesan and Alwan, 2009] Sankaran Panchapagesan and Abeer Alwan. Frequency warping for VTLN and speaker adaptation by linear transformation of standard MFCC. *Computer speech and language*, 23(1):42–64, 2009.

[Pitrelli *et al.*, 1995] John F. Pitrelli, Cynthia Fong, Suk H. Wong, Judith R. Spitz, and Hong C. Leung. PhoneBook: A phonetically-rich isolated-word telephone-speech database. In *Proc. ICASSP*, pages 101–104, Detroit, USA, 1995. IEEE.

[Poritz, 1982] Alan B. Poritz. Linear predictive hidden Markov models and the speech signal. In *Proc. ICASSP*, pages 1291–1294, Paris, France, 1982. IEEE.

[Povey and Woodland, 2002] Daniel Povey and Philip C. Woodland. Minimum phone error and I-smoothing for improved discriminative training. In *Proc. ICASSP*, pages I–105, Orlando, USA, 2002. IEEE.

[Povey and Yao, 2012] Daniel Povey and Kaisheng Yao. A basis representation of constrained MLLR transforms for robust adaptation. *Computer Speech & Language*, 26(1):35–51, 2012.

[Povey *et al.*, 2008] Daniel Povey, Dimitri Kanevsky, Brian Kingsbury, Bhuvana Ramabhadran, George Saon, and Karthik Visweswariah. Boosted MMI for model and feature-space discriminative training. In *Proc. ICASSP*, pages 4057–4060, Las Vegas, USA, 2008.

[Povey *et al.*, 2011a] Daniel Povey, Lukáš Burget, Mohit Agarwal, Pinar Akyazi, Feng Kai, Arnab Ghoshal, Ondřej Glembek, Nagendra Goel, Martin Karafiát, Ariya Rastrow, Richard C. Rose, Petr Schwarz, and Samuel Thomas. The subspace Gaussian mixture model—A structured model for speech recognition. *Computer speech and language*, 25(2):404–439, 2011.

[Povey *et al.*, 2011b] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Karel Vesely, Georg Stemmer, Jan Silovsky, Petr Schwarz, Yanmin Qian, Nagendra Goel, Mirko Hannemann, and Petr Motlicek. The Kaldi speech recognition toolkit. In *Proc. ASRU workshop*, pages 1–4, Hawaii, US, 2011. IEEE.

[Price *et al.*, 1988] Patti Price, William M. Fisher, Jared Bernstein, and David S. Pallett. The DARPA 1000-word resource management database for continuous speech recognition. In *Proc. ICASSP*, pages 651–654, New York, USA, 1988. IEEE.

[Rabiner, 1989] Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77:257–286, 1989.

[Ravishankar, 1996] Mosur Ravishankar. *Efficient algorithms for speech recognition.* Phd thesis, Carnegie Mellon University, 1996.

[Reynolds *et al.*, 2000] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10(1):19–41, 2000.

[Riedhammer *et al.*, 2012] Korbinian Riedhammer, Tobias Bocklet, Arnab Ghoshal, and Daniel Povey. Revisiting semi-continuous hidden Markov models. In *Proc. ICASSP*, pages 4721–4724, Kyoto, Japan, 2012. IEEE.

[Rose *et al.*, 2011] Richard Rose, Shou-Chun Yin, and Yun Tang. An investigation of subspace modeling for phonetic and speaker variability in automatic speech recognition. In *Proc. ICASSP*, pages 4508–4511, Prague, Czech Republic, 2011. IEEE.

[Rybach *et al.*, 2009] David Rybach, Christian Gollan, Georg Heigold, Björn Hoffmeister, Jonas Lööf, Ralf Schlüter, and Hermann Ney. The RWTH aachen university open source speech recognition system. In *Proc. INTERSPEECH*, pages 2111–2114, Brighton, United Kingdom, 2009. ISCA.

[Sak *et al.*, 2014] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv preprint arXiv:1402.1128*, 2014.

[Saraclar *et al.*, 2000] Murat Saraclar, Harriet Nock, and Sanjeev Khudanpur. Pronunciation modeling by sharing Gaussian densities across phonetic models. *Computer Speech & Language*, 14(2):137–160, 2000.

[Schultz and Waibel, 2001] Tanja Schultz and Alex Waibel. Language-independent and language-adaptive acoustic modeling for speech recognition. *Speech Communication*, 35(1):31–51, 2001.

[Shannon, 1951] Claude E. Shannon. Prediction and entropy of printed English. *Bell system technical journal*, 30(1):50–64, 1951.

[Sim and Gales, 2007] Khe Chai Sim and Mark Gales. Discriminative semi-parametric trajectory model for speech recognition. *Computer Speech & Language*, 21(4):669–687, 2007.

[Su *et al.*, 1996] Jian Su, Haizhou Li, Jean-Paul Haton, and Kai-Tat Ng. Speaker time-drifting adaptation using trajectory mixture hidden Markov models. In *Proc. ICASSP*, pages 709–712, Atlanta, USA, 1996. IEEE.

[Sun and Deng., 1995] Don X. Sun and Li Deng. Analysis of acoustic-phonetic variations in fluent speech using TIMIT. In *Proc. ICASSP*, pages 201–204, Detroit, USA, 1995. IEEE.

[Teng *et al.*, 2009] Wen Xuan Teng, Guillaume Gravier, Frédéric Bimbot, and Frédéric Soufflet. Speaker adaptation by variable reference model subspace and application to large vocabulary speech recognition. In *Proc. ICASSP*, pages 4381–4384, Taipei, Taiwan, 2009. IEEE.

[Tran *et al.*, 2014] Dung Tran, Emmanuel Vincent, and Denis Jouvet. Extension of uncertainty propagation to dynamic MFCCs for noise robust ASR. In *Proc. ICASSP*, pages 5544–5548, Florence, Italy, 2014. IEEE.

[Vincent *et al.*, 2013] Emmanuel Vincent, Jon Barker, Shinji Watanabe, Jonathan Le Roux, Francesco Nesta, and Marco Matassoni. The second CHiME Speech Separation and Recognition Challenge: Datasets, tasks and baselines. In *ICASSP*, pages 69–74, Vancouver, Canada, 2013. IEEE.

[Vintsyuk, 1968] Taras K. Vintsyuk. Speech discrimination by dynamic programming. *Kibernetika*, 4(1):52–57, 1968.

[Viterbi, 1967] Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2):260–269, 1967.

[Vu *et al.*, 2014] Ngoc Thang Vu, David Imseng, Daniel Povey, Petr Motlicek, Tanja Schultz, and Hervé Bourlard. Multilingual Deep Neural Network based Acoustic Modeling For Rapid Language Adaptation. In *ICASSP*, 2014.

[Wakita, 1973] Hisashi Wakita. Direct estimation of the vocal tract shape by inverse filtering of acoustic speech waveforms. *Audio and Electroacoustics, IEEE Transactions on*, 21(5):417–427, 1973.

[Wang *et al.*, 2011] David Wang, Robert J. Vogt, Sridha Sridharan, and David Dean. Cross likelihood ratio based speaker clustering using eigenvoice models. In *Proc. INTERSPEECH*, pages 957–960, Florence, Italy, 2011. ISCA.

[Wegmann *et al.*, 1996] Steven Wegmann, Don McAllaster, Jeremy Orloff, and Barbara Peskin. Speaker normalization on conversational telephone speech. In *ICASSP*, pages 339–341, 1996.

[Wellekens, 1987] Christian Wellekens. Explicit time correlation in hidden Markov models for speech recognition. In *Proc. ICASSP*, pages 384–386, Dallas, USA, 1987. IEEE.

[Wenxuan *et al.*, 2007] Teng Wenxuan, Guillaume Gravier, Frédéric Bimbot, and Frédéric Soufflet. Rapid speaker adaptation by reference model interpolation. In *Proc. INTERSPEECH*, pages 258–261, Antwerp, Belgium, 2007. ISCA.

[Woodland and Povey, 2002] Philip C. Woodland and Daniel Povey. Large scale discriminative training of hidden Markov models for speech recognition. *Computer Speech & Language*, 16(1):25–47, 2002.

[Yannick Estève *et al.*, 2010] Yannick Estève, Thierry Bazillon, Jean-Yves Antoine, Frédéric Béchet, and Jérôme Farinas. The EPAC corpus: manual and automatic annotations of conversational speech in French broadcast news. In *Proc. LREC*, pages 1686–1689, Valletta, Malta, 2010. ELRA.

[Ye and Mak, 2012] Guoli Ye and Brian Mak. Speaker-ensemble hidden Markov modeling for automatic speech recognition. In *Proc. ISCSLP*, pages 6–10, Hong Kong, China, 2012. IEEE.

[Young *et al.*, 1994] Steve J. Young, Julian J. Odell, and Philip C. Woodland. Tree-based state tying for high accuracy acoustic modelling. In *Proc. HLT*, pages 307–312, Plainsboro, New Jerey, USA, 1994. IEEE.

[Young *et al.*, 2006] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying (Andrew) Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland. *The HTK book version 3.4*. 2006.

[Zhang *et al.*, 2011] Yu Zhang, Jian Xu, Zhi-Jie Yan, and Qiang Huo. An i-vector based approach to training data clustering for improved speech recognition. In *Proc. INTERSPEECH*, pages 789–792, Florence, Italy, 2011. ISCA.

[Zhao and Juang, 2012] Yong Zhao and Biing-Hwang Fred Juang. Stranded Gaussian mixture hidden Markov models for robust speech recognition. In *Proc. ICASSP*, pages 4301–4304, Kyoto, Japan, 2012. IEEE.

[Zhao and Juang, 2013] Yong Zhao and Biing-Hwang Fred Juang. Modeling heterogeneous data sources for speech recognition using synchronous hidden Markov models. In *Proc. ICASSP*, pages 7403–7407, Vancouver, Canada, 2013. IEEE.

Résumé

Cette thèse se concentre sur la structuration du modèle acoustique pour améliorer la reconnaissance de la parole par modèle de Markov. La structuration repose sur l'utilisation d'une classification non supervisée des phrases du corpus d'apprentissage pour tenir compte des variabilités dues aux locuteurs et aux canaux de transmission. L'idée est de regrouper automatiquement les phrases prononcées en classes correspondant à des données acoustiquement similaires. Pour la modélisation multiple, un modèle acoustique indépendant du locuteur est adapté aux données de chaque classe. Quand le nombre de classes augmente, la quantité de données disponibles pour l'apprentissage du modèle de chaque classe diminue, et cela peut rendre la modélisation moins fiable. Une façon de pallier ce problème est de modifier le critère de classification appliqué sur les données d'apprentissage pour permettre à une phrase d'être associée à plusieurs classes. Ceci est obtenu par l'introduction d'une marge de tolérance lors de la classification ; et cette approche est étudiée dans la première partie de la these.

L'essentiel de la thèse est consacré à une nouvelle approche qui utilise la classification automatique des données d'apprentissage pour structurer le modèle acoustique. Ainsi, au lieu d'adapter tous les paramètres du modèle HMM-GMM pour chaque classe de données, les informations de classe sont explicitement introduites dans la structure des GMM en associant chaque composante des densités multi-gaussiennes avec une classe. Pour exploiter efficacement cette structuration des composantes, deux types de modélisations sont proposés. Dans la première approche on propose de compléter cette structuration des densités par des pondérations des composantes gaussiennes dépendantes des classes de locuteurs. Pour cette modélisation, les composantes gaussiennes des mélanges GMM sont structurées en fonction des classes et partagées entre toutes les classes, tandis que les pondérations des composantes des densités sont dépendantes de la classe. Lors du décodage, le jeu de pondérations des gaussiennes est sélectionné en fonction de la classe estimée. Dans une deuxième approche, les pondérations des gaussiennes sont remplacées par des matrices de transition entre les composantes gaussiennes des densités. Les approches proposées dans cette thèse sont analysées et évaluées sur différents corpus de parole qui couvrent différentes sources de variabilité (âge, sexe, accent et bruit).

Mots-clés: Reconnaissance de la parole , classification non supervisée , modèles de classes de locuteurs , modèles stochastiques de trajectoire , variabilité de locuteur

Abstract

This thesis focuses on acoustic model structuring for improving HMM-based automatic speech recognition. The structuring relies on unsupervised clustering of speech segments of the training data in order to handle speaker and channel variability. The idea is to split the data into acoustically similar classes. In conventional multi-modeling (or class-based) approach, separate class-dependent models are built via adaptation of a speaker-independent model. When the number of classes increases, less data becomes available for the estimation of the class-based models, and the parameters are less reliable. One way to handle such problem is to modify the classification criterion applied on the training data, allowing a given segment to belong to more than one class. This is obtained by relaxing the classification decision through a soft margin. This is investigated in the first part of the thesis.

In the main part of the thesis, a novel approach is proposed that uses the clustered data more efficiently in a class-structured GMM. Instead of adapting all HMM-GMM parameters separately for each class of data, the class information is explicitly introduced into the GMM structure by associating a given density component with a given class. To efficiently exploit such structured HMM-GMM, two different approaches are proposed. The first approach combines class-structured GMM with class-dependent mixture weights. In this model the Gaussian components are shared across speaker classes, but they are class-structured, and the mixture weights are class-dependent. For decoding a segment, the set of mixture weights is selected according to the estimated class. In the second approach, the mixture weights are replaced by density component transition probabilities. The approaches proposed in the thesis are analyzed and evaluated on various speech data, which cover different types of variability sources (age, gender, accent and noise).

Keywords: Speech recognition, unsupervised clustering, speaker class modeling, stochastic trajectory modeling, speaker variability