

ENSEMBLE ACOUSTIC MODELING IN AUTOMATIC SPEECH RECOGNITION

A Dissertation

presented to

the Faculty of the Graduate School
at the University of Missouri-Columbia

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

by

XIN CHEN

Dr. Yunxin Zhao, Dissertation Supervisor

DECEMBER 2011

The undersigned, appointed by the dean of the Graduate School, have examined the dissertation entitled

Ensemble Acoustic Modeling in Automatic Speech Recognition

Presented by Xin Chen,

a candidate for the degree of PhD of computer science,

and hereby certify that, in their opinion, it is worthy of acceptance.

Professor Yunxin Zhao

Professor Dong Xu

Professor Xinhua Zhuang

Professor Dominic Ho

Acknowledgements

First of all, I would like to thank my advisor, Dr. Yunxin Zhao. It is a great honor to be her student. Her inspirations and help enlighten me on the ASR field. I feel so lucky to learn from her.

Additionally, many thanks to Dr. Dominic Ho, Dr. Dong Xu, and Dr. Xinhua Zhuang for being as my committee members and giving me great support. I also want to thank Dr. Wenjun Zeng for being as my committee member for my master degree. I really appreciate their help and providing me with invaluable guidance.

Moreover, I would like to thank Dr. Bryan and Dr. Kadri who were my mentors in Rosetta Stone Corporation, where I did my first summer internship in Boulder in 2009. Also, I want to thank Dr. Xiaodong Cui and Dr. Jian Xue who served as my mentors when I did my second summer internship in IBM T.J Watson Research center in 2010. In addition, I want to thank Dr. Dong Yu and Dr. Deng Li who served as my mentors when I did my third summer internship in Microsoft Research in 2011. Overall, I had great summers and learned many things.

Furthermore, I would like to thank my grandmother and my parents. Their support from the other side of the earth gives me courage and passion. Also, I would like to thank my wife Yue for her enduring encouragement.

Last but not least, much respect to my current officemates Yi Zhang, Xie Sun, Tuo Zhao and Xiuzhen Huang, as well as previous officemates, Rong Hu, Jian Xue, Lily Che. It was my pleasure to work with them and get a lot of help from them.

TABLE OF CONTENTS

Acknowledgements	ii
TABLE OF CONTENTS	iii
List of Figures	vii
List of Tables	ix
Abstract	xi
Chapter 1	1
Introduction	1
1.1 Motivation and proposed methods	1
1.2 Outline of the dissertation	5
Chapter 2	7
Statistical Speech Recognition	7
2.1 General idea of statistical speech recognition	7
2.2 Speech pre-processing	9
2.3 Language modeling	11
2.4 Acoustic modeling	13
2.4.1 Hidden Markov Model (HMM) in speech recognition	13
2.5 Pronunciation dictionary and lexical tree	16
2.6 Viterbi algorithm	18
2.7 Improve base model quality using enhanced training and features	19
2.7.1 Discriminative training	19
2.7.2 MLP feature	20
2.7.3 Cross Validation Expectation Maximization (CVEM)	22

2.8 Summary	23
Chapter 3	24
Explicit Phonetic Decision Tree Tying	24
3.1 Phonetic decision tree background	27
3.2 Explicit PDT tying	30
3.3 Explicit PDT tying for multiple models.....	33
3.4 Hierarchical ensemble model based on different mixture size	34
Chapter 4.....	36
Data Sampling Based Ensemble Acoustic Modeling	36
4.1 Ensemble model for acoustic score combination.....	36
4.2 Sampling training data to generate EAM.....	39
4.3 Cross-validation data sampling.....	41
4.4 Random data sampling.....	42
4.5 Speaker clustering based data sampling	43
4.6 Combiner design	47
Chapter 5	49
Ensemble Acoustic Model Clustering.....	49
5.1 Distance measures for Gaussian density clustering	49
5.1.1 KL divergence.....	49
5.1.2 Entropy change	50
5.1.3 Bayes error, Bhattacharyya and Chernoff distances	50
5.1.4 Weighted distances	51
5.2 Gaussian component clustering algorithm.....	51

5.2.1 Entropy based N-Best distance Refinement (NBR).....	52
5.2.2 K-step LookAhead (KLA)	53
5.2.3 Breadth-First Search (BFS) global optimization	54
5.2.4 Two-pass global optimization.....	54
Chapter 6.....	56
Experiments on Speech Recognition Tasks	56
6.1 ASR tasks.....	56
6.1.1 Telehealth task	56
6.1.2 TIMIT task.....	57
6.1.3 Pashto task.....	58
6.1.4 Broadcast news task.....	58
6.2 Experimental results for the telehealth automatic captioning task	59
6.2.1 Experimental results for explicit PDT tying	59
6.2.2 Experimental results for multiple acoustic models based on explicit PDT	63
6.2.3 Experimental results for cross validation data sampling	64
6.2.4 Experimental results for random data sampling	68
6.2.5 Experimental results for enhanced training	69
6.3 Experimental results for the TIMIT phone recognition task.....	72
6.3.1 CV data sampling based ensemble acoustic models.....	72
6.3.2 Speaker clustering based ensemble acoustic models.....	73
6.3.3 Integration of discriminative training and MLP feature in ensemble acoustic models.....	77

6.3.4 Data sampling vs. question sampling.....	81
6.4 Experiments on the Pashto ASR task.....	83
6.4.1 Experimental results of NBR.....	83
6.4.2 Experimental results of global optimization.....	85
6.4.3 Experimental results on two-pass model structure refinement.....	86
6.5 Experiment results on the broadcast news ASR task.....	87
Chapter 7.....	88
Base Model Quality, Inter-Model Diversity, and EAM quality.....	88
7.1 Explicit measures of inter-model diversity.....	88
7.2 Experimental results on base-model quality, inter-model quality, and EAM quality.....	90
Chapter 8.....	97
Conclusion and Future Extension.....	97
References.....	99
VITA.....	106

List of Figures

Figure 2.1 An example of HMM for a phoneme unit using Gaussian pdf in each emitting state [25]	14
Figure 2.2 Part of a sample pronunciation dictionary	17
Figure 2.3 Fraction of a lexical tree	18
Figure 2.4 Example of MLP with two hidden layers.....	20
Figure 2.5 Illustration of MLP tandem feature generation	21
Figure 2.6 Illustration of CVEM training	22
Figure 2.7 Diagram of an automatic speech recognition system	23
Figure 3.1 The three levels of speech recognition	24
Figure 3.2 An example of decision tree construction	28
Figure 3.3 Explicit decision tree triphone tying.....	31
Figure 3.4 An example of triphone tying in ensemble acoustic model	33
Figure 4.1 An example of ROVER framework	37
Figure 4.2 Ensemble model framework in the telehealth captioning system	38
Figure 4.3 Generating an ensemble model by data sampling	39
Figure 4.4 Illustration of 5-fold cross validation	41
Figure 4.5 Illustration of CV based EAM.....	42
Figure 5.1 Illustration of using (a) K-step LA and (b) search for the best path.....	53
Figure 6.1 Block diagram of the automatic captioning system for telemedicine[55]	56
Figure 6.2 Effects of different mixture sizes on word accuracy	68
Figure 6.3 Effects of EAMs on the telehealth task (mixture size=16 per GMD)	70

Figure 6.4 Effects of DT training on word accuracy	71
Figure 6.5 Effects of EAM with EM or CVEM training on the TIMIT task.....	73
Figure 6.6 Effects of LSC versus RS in training ensemble acoustic models.....	74
Figure 6.7 Effects of LSC and DSC in training ensemble acoustic models	76
Figure 6.8 Effects of MPE and MLP features on ensemble acoustic models.....	78
Figure 6.9 Phone accuracies of LSC and CV data sampled ensemble models.....	81
Figure 6.10 Effects of EAM clustering with different compression rates	82
Figure 6.11 Effects of NBR method on WER	84
Figure 6.12 Effects of weighted Bhattacharyya criterion and NBR on WER	85
Figure 6.13 Accuracy performances of single model and EAM on BN task	87
Figure 7.1 Illustration of computing KL distance evaluation as the diversity measure for EAM	89
Figure 7.2 Correct and competing scores from one test sentence (a) baseline model (b) 10-fold CV ensemble model	92
Figure 7.3 Effects of DT on base model and EAM quality with different fractions of training data	96

List of Tables

Table 3.1 An example of a state cluster in EPDT in the telehealth task.....	32
Table 6.1 Datasets used in the telehealth task: speech (min.)/text (no. of words)....	57
Table 6.2 Word accuracy obtained from EPDT tying 1	60
Table 6.3 Word accuracy obtained from EPDT tying 2	60
Table 6.4 Word accuracy obtained from the extreme case of EPDT tying	62
Table 6.5 Word accuracy obtained from combining the baseline model and the 6- Tree models.....	63
Table 6.6 Word accuracy obtained from combining the baseline, the 3-Tree, and the 6-Tree models	64
Table 6.7 Word accuracy obtained from 10-fold cross-validation based ensemble acoustic model	65
Table 6.8 Effects of base classifiers on word accuracy	66
Table 6.9 Effects of fold size on word accuracy (averaged over 5 doctors).....	67
Table 6.10 Effects of different mixture sizes on word accuracy	67
Table 6.11 Word accuracy of the ensemble models generated by random sampling without replacement.....	69
Table 6.12 Effects of MLP and ensemble MLP features in ensemble acoustic models	79
Table 6.13 Comparison on TIMIT phone recognition accuracies of question sampling and data sampling based EAMs.	81
Table 6.14 Evaluation on speed improvement using NBR.....	84

Table 6.15 WERs (%) from using the proposed global clustering algorithm.....	85
Table 6.16 WER results (%) on two-pass model structure refinement.....	86
Table 7.1 Acoustic model quality measured on Dr.2's data set.....	91
Table 7.2 Ensemble model quality, base model quality and explicit measures on inter-model diversity.....	93
Table 7.3 TIMIT phone recognition accuracy of the 10-fold CV data sampling EAM (mix16) with the enhanced training methods and features	94

Abstract

Combining multiple acoustic models to improve the overall acoustic model quality is a young and promising direction in Automatic Speech Recognition (ASR). Previous works on acoustic modeling of speech signals such as Random Forests (RFs) or Phonetic Decision Trees (PDTs) has produced significant improvements in recognition accuracy. In this dissertation, several new approaches of using data sampling to construct an Ensemble of Acoustic Models (EAM) for speech recognition are proposed. A straightforward method of data sampling is Cross Validation (CV) data partition. In the direction of improving inter-model diversity within an EAM for speaker independent speech recognition, we propose Speaker Clustering (SC) based data sampling and develop two algorithms, including the Likelihood based Speaker Clustering (LSC) and speaker model Distance based Speaker Clustering (DSC). In the direction of improving base model quality as well as inter-model diversity, we further investigate the effects of several successful techniques of single model training in speech recognition on the proposed ensemble acoustic models, including Cross Validation Expectation Maximization (CVEM), Discriminative Training (DT), and Multiple Layer Perceptron (MLP) features. We also propose using an ensemble of Multiple models with Different Mixture Sizes (MDMS) to improve EAM quality. We have evaluated the proposed methods on TIMIT speaker-independent phoneme recognition task as well as on a telemedicine automatic captioning task of speaker-dependent continuous speech recognition. The proposed EAMs have led to significant improvements in recognition accuracy over conventional Hidden Markov Model (HMM) baseline systems, and the

integration of ensemble acoustic models with CVEM, DT and MLP has also significantly improved the accuracy performances of CVEM, DT, and MLP based single model systems. We further investigated the largely unstudied factor of inter-model diversity, and proposed several methods to explicit measure inter-model diversity. We demonstrate a positive relation between enlarging inter-model diversity and increasing EAM quality.

HMM-based acoustic models built from data sampling EAM are generally very large, especially when a large number of models or full covariance matrices are used for Gaussian densities. Therefore, compacting the acoustic model to a reasonable size for practical applications while maintaining a reasonable performance is needed. Toward this goal, in this dissertation, we discuss and investigate several distance measures and algorithms for clustering methods. The distance measures include Entropy, KL, Bhattacharyya, Chernoff and their weighted versions. For clustering algorithms, besides the conventional greedy agglomerative clustering, algorithms such as N-Best distance Refinement (NBR), K-step LookAhead (KLA), Breadth-First Search (BFS) are proposed. Experiments on the TIMIT task have shown that in comparison with the original EAM model, the compacted models using the clustering methods can maintain the model accuracy, while the size of the compacted model is largely decreased. Experiments in compacting EAM on a Pashto ASR task have shown that the proposed clustering methods can lead to better quality than the conventional clustering methods.

Unlike the implicit PDT based states tying that has been used in most ASR systems as well as in the recent RF based PDTs, explicit PDT (EPDT) state tying that allows Phoneme data Sharing (PS) is considered for its potential capability in capturing pronunciation variations. The ensemble approach of combining multiple acoustic models

is applied to the EPDT, where a combination of explicit PDT and implicit PDT models has been investigated to reduce phone confusions.

Chapter 1

Introduction

Automatic Speech Recognition (ASR) is a very promising technology and it has a potentially wide range of applications. However, current recognition accuracy performance still needs improvement for the technology to be effectively deployed in the tasks of everyday life. Towards improving the accuracy performance of speech recognition, new techniques of modeling speech signals that are motivated by Ensemble Classifiers (EC) in machine learning are highly desirable.

1.1 Motivation and proposed methods

Current ASR systems normally use a single acoustic model. Although these systems work very well for native talkers in certain tasks, mismatch between feature and model often occurs, especially in noisy situations, where the accuracy performances of these systems can be severely degraded. Therefore, improving the performance of state-of-the-art speech recognition systems by modeling features with better acoustic model remains a challenging task.

Combining multiple speech recognition systems through ROVER-like word hypotheses integration has been established as an effective approach to improve the accuracy performance of automatic speech recognition [1][2][3]. In ROVER, the individual speech recognition systems work independently and the decoding word hypotheses of the multiple systems are combined. In general, the recognition accuracy performance of a combined system is improved by the diversity among the different

speech recognition systems. In machine learning, many methods have been proposed for designing ensemble classifiers [4], where a very successful method is the random forests of decision trees constructed from random samplings on features and data [5], and the latter is also referred to as bagging. Recently, a novel technique of multiple acoustic model combination has been proposed [6], where a random sampling on phonetic questions was used to generate random forests of phonetic decision trees to construct an ensemble of acoustic models. This technique combines the acoustic scores of each speech analysis frame from multiple acoustic models during decoding search while using only one decoding engine as in a conventional single recognition system. As was shown in [6], this model-level combination has a significantly lower computation complexity in comparison with the system-level combination.

Many techniques have been developed over the years to improve the quality of conventional acoustic models, and some of the techniques, such as discriminative training, MLP feature, and CVEM, hold good potentials in integrating with the approach of ensemble acoustic modeling. Discriminative training differs from maximum likelihood training in that it targets at minimizing classification errors instead of maximizing model-data fit. The commonly used DT methods include minimum classification error (MCE) [7], maximum mutual information (MMI) [8], and minimum phone/word error (MPE/MWE) [9], etc. DT has been shown successful in large systems as well as in TIMIT phoneme recognition task. MLP-based features are commonly derived from the phone posterior probability outputs of a MLP neural network. Using MLP based features in a HMM system was first proposed in [10] as a TANDEM approach. Concatenating MLP features with the traditional MFCC or PLP features has subsequently been proven

very effective on several tasks [11]. CVEM is a maximum likelihood training algorithm that attempts to compensate for the overfitting problem in the conventional EM algorithm [12].

In this dissertation, we propose several data sampling methods to construct Ensembles of Acoustic Models (EAM). In the data sampling approach, the sensitivity of the acoustic models to the training data is exploited to produce diversity in EAM. Using data sampling to build EAM has several potential advantages: data sampling is straightforward in implementation, data clustering with respect to different variation factors can be performed to improve the inter-model diversity of EAM, and using data sampling to build EAM is a promising approach for large datasets, where a single model may fail to utilize all the information from data.

For data sampling, we first propose Cross Validation (CV) data partition to build EAM, and towards improving inter-model diversity for EAM, we next propose speaker clustering based data sampling, including Likelihood based Speaker Clustering (LSC) and speaker model Distance based Speaker Clustering (DSC). We use speaker clustering to produce multiple datasets for building an EAM, where speaker characteristics are similar within each set and dissimilar between sets, and from such different datasets inter-model diversity may be built into an EAM. We demonstrate that although data sampling produced base models in an EAM are weaker than the conventional single model because of the reduced amounts of training data, the inter-model diversity in EAM increases faster than the decrease of base model quality, and therefore an effective EAM can be obtained as the result.

Since both base model quality and inter-model diversity contribute to the quality of an EAM, in order to better understand this, we investigate the effects of using the enhanced training methods of CVEM and DT and the enhanced feature of MLP on these two aspects of EAM quality. We demonstrate that through EAM the positive impacts of the enhanced trainings and feature on recognition accuracy are amplified due to the increase in inter-model diversity which is not present in conventional acoustic models. We used several measures to determine inter-model diversity and confirmed its positive relationship with the recognition accuracy performance of EAM. We have conducted experimental evaluations on a TIMIT phone recognition task, a telemedicine automatic speech captioning task, and North America broadcast news LVCSR task [13]. We have obtained significant improvements on phone and word accuracy performances on these tasks using the proposed methods.

Although EAM models show better quality than single acoustic models, in some cases where decoding speed is critical or computational resource is limited, a EAM may not be a good choice since it is normally larger than a single model and therefore requires a lot of computational power. So it is of interest to compact the EAM by performing clustering at the Gaussian Mixture Model (GMM) level while still maintain superior performance over the single acoustic model. In this dissertation, several similarity measurements are investigated and several global optimization methods are proposed for clustering the GMMs, and these methods are evaluated in a Pashto ASR task. We also evaluated conventional methods in compacting EAM on TIMIT task, and we show that a compacted EAM can achieve a practical model size while maintaining high quality over single acoustic model.

1.2 Outline of the dissertation

This dissertation is organized into eight chapters:

In Chapter 1, an overview of this dissertation is given.

In Chapter 2, a brief introduction is given to the background of the state-of-the-art Hidden Markov Model (HMM) based speech recognition. The commonly used techniques of language modeling, acoustic modeling and phonetic decision tree based state tying are introduced in this chapter.

In Chapter 3, the method of explicit phonetic decision tree tying is proposed and discussed.

In Chapter 4, data sampling based ensemble methods are proposed and discussed.

In Chapter 5, a comprehensive study of compacting EAM is discussed, including the distance measures and the clustering algorithms.

In Chapter 6, the proposed methods for EAM are evaluated on the TIMIT task, the telehealth automatic captioning task, and the broadcast news task. A series of experiments have shown that significant improvements in recognition accuracy are achieved by using the proposed methods. The proposed global optimization for Gaussian clustering is evaluated on the Pashto ASR task, and better recognition results are obtained with the proposed methods over the conventional methods.

In Chapter 7, a comprehensive investigation on EAM quality is discussed. Assuming that the overall EAM quality is dependent on base model quality and inter-model diversity, we push up base model quality and explicitly measure the change in inter-model diversity. We discuss the methods for measuring the inter-model diversity, and present observations and analyses.

Finally, a conclusion and directions for future extensions are given in Chapter 8.

Chapter 2

Statistical Speech Recognition

2.1 General idea of statistical speech recognition

Speech is the most convenient and natural everyday communication method among humans, and it is a very promising interface between computer and human. After a half century of evolution [14], Automatic Speech Recognition (ASR) systems nowadays are finding applications in everyday's life. For example, automatic customer service system allows people to use voice to select a menu when calling a bank. Another example is the telemedicine automatic captioning project in our lab, where ASR can be used to help people who have hearing loss to read a captioned message that a doctor's speech conveys over a long distance. ASR has very meaningful applications, and we are devoting passionate efforts to enhance the technology's recognition accuracy, the decoding speed, as well as the system functionality.

Generally, we can describe speech recognition as a time series classification problem. It attempts to find an optimized word sequence that best matches a speech utterance. Currently, the most successful method of speech recognition is based on Bayesian decision theory [15]. Given a data sample x , the posterior probability of the class C_j is computed from the prior probabilities of C_1, C_2, \dots, C_k , and the conditional probabilities of x given $C_i, i = 1, 2, \dots, k$:

$$P(C_j | x) = \frac{P(x \cap C_j)}{P(x)} = \frac{P(C_j)P(x | C_j)}{\sum_{i=1}^k P(C_i)P(x | C_i)}, \quad (2.1)$$

when we apply Bayes rule to the speech recognition problem, we can rewrite the decision problem as:

$$\hat{W} = \arg \max_W p(W | O) \quad (2.2)$$

where $W = w_1, w_2, \dots, w_n$ is the sequence of words (with unknown length n) in an utterance produced by the speaker which generates the acoustic feature vector sequence $O = o_1, o_2, \dots, o_T$; $p(W)$, usually called the language model, is the a priori probability of the word sequence W , which is independent of the observation O ; $p(O)$ is the a priori probability of the observed speech utterance O , which is independent of all word sequence hypotheses, and so it can be ignored in the last line of formula (2.2); $p(O/W)$ is the probability that the speaker produces the acoustic feature vector sequence O assuming W is the underlying word sequence.

Statistical modeling for estimating $p(W)$ is called language modeling. The most commonly used language model is N -gram, which will be discussed in Section 2.3.

Statistical modeling for estimating $p(O/W)$ is called acoustic modeling. Here words are usually decomposed into sub words such as phonemes or syllables since they are more trainable from a finite amount of speech data. We use lexical trees to represent words by sub-words, usually phonemes. The most commonly used acoustic model is Hidden Markov Model (HMM) of Context-Dependent (CD) phones. We discuss the details of acoustic modeling in Section 2.4.

In practice, the word sequence hypothesis is determined by:

$$\hat{W} = \arg \max_w \frac{p(O|W)p(W)^\alpha e^\beta}{p(O)} = \arg \max_w p(O|W)p(W)^\alpha e^\beta \quad (2.3)$$

The parameter α in $p(W)^\alpha$ is referred to as a language model scale factor, which is used to balance the scores of acoustic model and language model. The parameter e^β is called word insertion penalty, which is used to control the length of the word hypothesis sequence. These parameters are extremely important in controlling the performance of an ASR system and are often fine-tuned before a speech recognition system is deployed in real applications.

Speech recognition engine commonly works by using Viterbi algorithm [16] to search over a large hypothesis space, determining the best word sequence that has the highest probability of generating the speech utterance. This part will be discussed in Section 2.6.

2.2 Speech pre-processing

Digital speech signals, which are waveform signals sampled at a certain clock rate, are not suitable to be directly used in training acoustic models. Pre-processing is such a procedure that converts the original waveform of speech into the type of presentation that only contains necessary information for speech recognition. Typically, the speech sound waves are captured by a microphone and converted to electrical signals. Then Analog-to-digital conversion samples speech signal at discrete time intervals (e.g. sampling rate=16KHZ), which becomes the input to an ASR system. The sampled data is used to generate feature vectors. This process is called feature analysis. Generally, a feature

vector is computed per 10ms time, from an overlapped sliding window of 20 to 25 ms. Commonly used features are as follows:

1. Linear Predictive Coefficient (LPC) – a speech sample at time t is approximated as a linear combination of the immediate past p speech samples, and the combination coefficients are assumed constant over each speech frame [17].
2. Perceptual Linear Prediction (PLP) - a variation of linear prediction taking into account of human auditory perception model [18].
3. Mel Frequency Cepstral Coefficients (MFCC) - cepstrum is computed by first warping the energy spectrum according to the Mel frequency scale and then taking the cosine transform on the log energies in subbands predefined for the Mel frequency scale[19].

The above mentioned features are all considered to be short-term stationary features and they cannot cover the temporal dynamics in speech. It is a common practice to use the first-order and second-order time-derivatives of such static features to capture the time dynamic information [20].

The extracted features can be further transformed to improve ASR system performance. Such transformation algorithms include linear discriminant analysis (LDA or HLDA [21]), vocal tract length normalization (VTLN), independent component analysis (ICA) [22], principal components analysis (PCA) [15], etc. The goal of speech pre-processing is to produce discriminative and robust features to close the gap between the performance of human listeners and that of ASR systems.

2.3 Language modeling

Given a sequence of previously spoken words, what is the probability of the word that will be spoken next? Language Model (LM) is used to answer such a question. With LM we can reduce decoding search space by predicating a word sequence as well as improve recognition performance by providing syntax information. There are different proposals for LM, including Context-Free-Grammar (CFG) [23] which uses a set of knowledge based rules to predict words in sentences, and the widely used n-gram model [24] which is much more successful in real tasks because of its simplicity and effectiveness.

The probability of a certain word sequence W is denoted as $p(W)$, which can be calculated in the following way:

$$\begin{aligned} p(W) &= p(w_1, w_2, \dots, w_N) \\ &= p(w_1)p(w_2 | w_1)p(w_3 | w_1, w_2) \cdots p(w_N | w_1, w_2, \dots, w_{N-1}) \\ &= \prod_{i=1}^N p(w_i | w_1, w_2, \dots, w_{i-1}) \end{aligned} \quad (2.4)$$

where $p(w_i | w_1, w_2, \dots, w_{i-1})$ is the probability that word w_i will follow the previously presented word sub sequence w_1, w_2, \dots, w_{i-1} . To reduce the model complexity, we assume the occurrence of a word only depends on $n-1$ previous words. If we define a language model under the assumption that the occurrence of a word depends only on its previous two words or one word, we will get trigram language model or bi-gram language model, respectively.

The most commonly used n -gram language model in speech recognition system is n equals to 3, or trigram. When n equals to 4, the model complexity is largely increased compared with trigram and therefore it consumes a lot of computation as well as storage space. A trigram language model estimates word sequence probability in the following way:

$$\begin{aligned}
 p(W) &= p(w_1, w_2, \dots, w_N) \\
 &= p(w_1)p(w_2 | w_1)p(w_3 | w_1, w_2) \cdots p(w_N | w_{N-2}w_{N-1}) \\
 &= p(w_1)p(w_2 | w_1) \prod_{i=3}^N p(w_i | w_{i-2}w_{i-1})
 \end{aligned} \tag{2.5}$$

We use the maximum likelihood estimation (MLE) method to estimate the LM parameters. For trigrams, the parameters can be obtained as the following:

$$p(w_i | w_{i-2}w_{i-1}) = \frac{C(w_{i-2}w_{i-1}w_i)}{C(w_{i-2}w_{i-1})} \tag{2.6}$$

In the above equation, C is the count on the number of appearances of the word n -gram in a training corpus.

Due to the sparseness of training data, smoothing techniques are needed to make language model more robust because some trigrams do not appear frequently enough to train a language model. The core issue of smoothing is to assign a nonzero probability to unobserved word strings. Backing-off model is one of the most commonly used smoothing techniques. The idea is to use low-order n -gram to approximate the probabilities of those uncommon words, for example:

$$\hat{p}(w_i | w_{i-n+1}, \dots, w_{i-1}) = \begin{cases} p(w_i | w_{i-n+1}, \dots, w_{i-1}), & \text{if } c(w_{i-n+1}, \dots, w_i) > 0 \\ \alpha(w_{i-n+1}, \dots, w_{i-1}) p'(w_i | w_{i-n+2}, \dots, w_{i-1}), & \text{if } c(w_{i-n+1}, \dots, w_i) = 0 \end{cases} \quad (2.7)$$

In this way, if the n -gram is seen in the training data, then the maximum likelihood estimated probability will be used (normally discounted). Otherwise, we back off to the smoothed lower-order model.

2.4 Acoustic modeling

Acoustic model is used to characterize the acoustic-phonetic characteristics of speech signals. Hidden Markov Model is able to capture the time dynamics of speech signals and therefore is widely used in acoustic modeling.

2.4.1 Hidden Markov Model (HMM) in speech recognition

HMM is used to model speech signals by characterizing speech with a sequence of states and transitions between the states, and from which the acoustic score $p(O/W)$ can be computed.

In HMM, speech signal is generated by a Markov chain of hidden states, and each state is associated with a stationary distribution which is usually a Gaussian mixture density referred to as Gaussian Mixture Model (GMM). The transitions between states represent the non-stationary time-evolution in a speech signal.

Figure 2.1 shows an HMM with 5 states and fixed transitions, which is what we used in acoustic modeling of phoneme units for speech recognition [25]. This HMM includes 3 emitting states and 2 non-emitting states. The three emitting states (S_1, S_2, S_3)

generate speech observations with Gaussian mixture densities. The transition from state i to state j is specified by the transition probability a_{ij} . The two non-emitting states (S_0 and S_4) are an entry state and an exit state, respectively. These two states do not generate any observation but are used as the entry and exit interface for each HMM (both states are reached only once). The left-to-right topology of HMM is used to describe the temporal characteristics of speech signal, that is, the current state is only dependent on itself and its previous states, but not on future states.

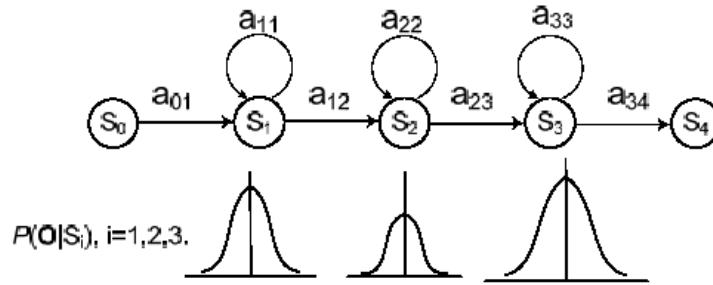


Figure 2.1 An example of HMM for a phoneme unit using Gaussian pdf in each emitting state [25]

In a hidden Markov model, the transition probability a_{ij} is defined by the following:

$$a_{ij} = P_r(s(t) = j | s(t-1) = i) \quad (2.8)$$

where $s(t)$ is the state index at time t . For a N -state HMM, we have $a_{ij} \geq 0$ and $\sum_{j=1}^N a_{ij} = 1$

for every i, j . For speech modeling, the output probability distribution of a HMM state can be modeled by a Gaussian Mixture Density (GMD) as below:

$$p(o | s) = \sum_{m=1}^M \frac{C_m}{(2\pi)^{d/2} |\Sigma_m|^{1/2}} e^{\left\{-\frac{1}{2}(o-\mu_m)^T \Sigma_m^{-1} (o-\mu_m)\right\}} \quad (2.9)$$

This is a mixture of multivariate Gaussian Densities, where M is the number of Gaussians, μ_m and Σ_m are the mean vector and covariance matrix for the m -th Gaussian component, d is the dimension of the feature vector, C_m is the weight of the m -th Gaussian component with the constraints $C_m \geq 0$ and $\sum_{m=1}^M C_m = 1$.

As we can see, each emission distribution symbolizes a sound event such as a phone state. The distribution must be discriminating enough to give the largest probability to the correct phone as well as robust enough to account for the variabilities in natural speech. Several methods have been used to train acoustic model parameters including state transition probabilities and the parameters of the emission probability densities at each state. Given $\{a_{ij}\}$ and $b(o/s_i)$, $i = 1 \sim N$, $j = 1 \sim N$, the likelihood of an observation sequence O given word sequence W is calculated as:

$$p(O | W) = \sum_S p(O, S | W) \quad (2.10)$$

where $S = s_1, s_2, \dots, s_T$ is the hidden Markov model state sequence that generates the observation vector sequence $O = o_1, o_2, \dots, o_T$. The joint probability of O and the state sequence S given W is a product of the transition probabilities and the emitting probabilities

$$p(O, S | W) = \prod_{t=1}^T b_{s_t}(o_t) a_{s_t, s_{t+1}} \quad (2.11)$$

Practically, for the emission pdf of Eq. (2.8), formula (2.9) can be approximately

calculated as the joint probability of the observation vector sequence O with the most possible state sequence, i.e.,

$$p(O|W) \approx \max_s p(O, S|W). \quad (2.12)$$

In Large Vocabulary Continuous Speech Recognition (LVCSR) systems, it is more accurate to build a HMM for each word or syllable. However, this is a very expensive implementation. In our system and most LVCSR systems in the world, Context-Dependent (CD) phonemes are used as the basic recognition units. HMMs are built for CD phone units and the model of a word string is concatenated from the CD phone units according to a dictionary lexical tree and LM.

2.5 Pronunciation dictionary and lexical tree

A pronunciation dictionary defines the phoneme constituents for each word in the vocabulary. Figure 2.2 gives some entries of a dictionary used in our Telehealth system. Here multiple pronunciations will be regarded as having an equal a priori probability.

.
 .
 .
 OVERSEEING ow v er s iy ih nx sil
 OVERSEEN ow v er s iy n sil
 OVERSEEN(2) ow v er s iy nx sil
 OVERSEER ow v er s iy er sil
 OVERSEES ow v er s iy z sil
 OVERSELL ow v er s eh l sil
 OVERSENSITIVE ow v er s eh n s ih t ih v sil
 OVERSENSITIVITY ow v er s eh n s ah t ih v ih t iy sil
 OVERSHADOW ow v er sh ae d ow sil
 OVERSHADOWED ow v er sh ae d ow d sil
 OVERSHADOWING ow v er sh ae d ow w ih nx sil
 OVERSHOOT ow v er sh uw t sil
 .
 .
 .

Figure 2.2 Part of a sample pronunciation dictionary

Lexical tree is a type of prefix tree that organizes a large dictionary in a speech recognition system in an efficient way. A fraction of a lexical tree corresponding to Figure 2.2 is shown below in Fig 2.3:

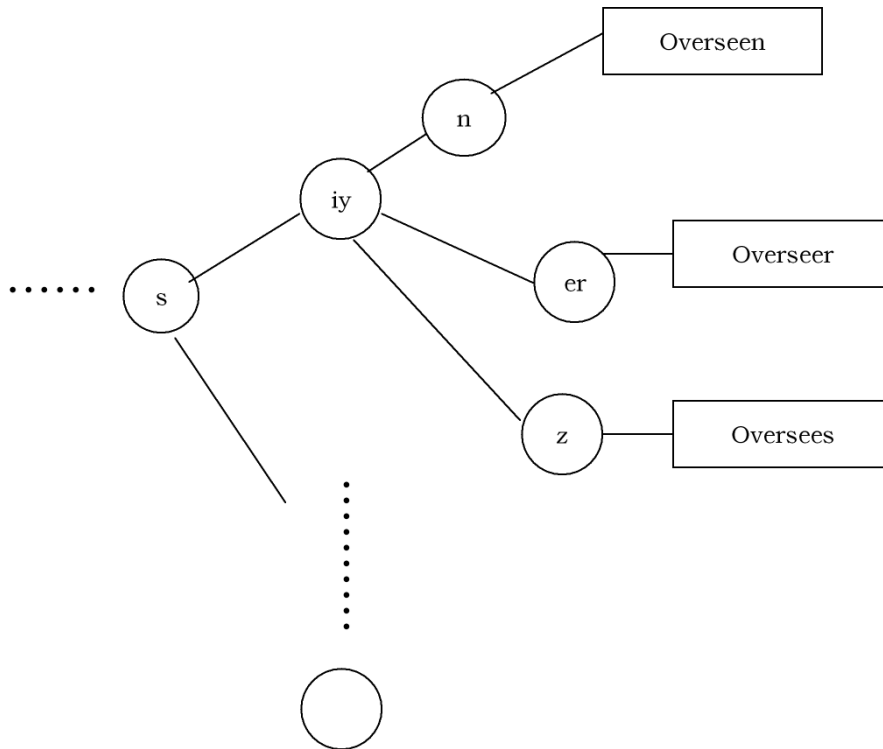


Figure 2.3 Fraction of a lexical tree

2.6 Viterbi algorithm

Viterbi algorithm [16], which is based on Dynamic Programming (DP) [27], is a very successful time-synchronous decoding algorithm. DP is widely used as an optimization method to decompose a big problem into small sub problems.

Speech decoding search is consisted of chiefly two parts. The first part is Forward-extension. All possible paths are extended from time 0 to time T-1 where T is the number of acoustic feature vectors in a sentence. During the extension, path scores are accumulated by combining the acoustic score and the language score for all acoustic vectors up to the current frame, and at each time each path will record its best previous word. Heuristic approaches as well as look-ahead methods can be used to prune the

search paths to increase the decoding speed. In a real time recognition task, we assume that if the silence length in a search path is longer than a fixed threshold or a filled pause appears, the search algorithm will backtrack to find the best partial path.

2.7 Improve base model quality using enhanced training and features

2.7.1 Discriminative training

MLE training is aimed at optimization the model parameters to maximize the likelihood of the training data observations. However, the practical criterion of AM quality is recognition error rate, and the mismatch between the objectives is the reason we seek for discriminative training which aims at reducing the error rate directly.

One discriminative training criterion is Maximum Mutual Information (MMI), with its objective function defined as follows [8]:

$$\begin{aligned}
 F_{mmi}(\lambda) &= \frac{1}{R} \sum_{r=1}^R \log(P(H_{ref}^r | O^r, \lambda)) \\
 &= \frac{1}{R} \sum_{r=1}^R \log\left(\frac{P(O^r | H_{ref}^r, \lambda) P(H_{ref}^r)}{\sum_H P(O^r | H, \lambda) P(H)}\right)
 \end{aligned} \tag{2.13}$$

where H_{ref}^r is the correct hypothesis of utterance r , H is all possible hypothesis, and R is the number of training utterances. The MMI criterion attempts to make the correct hypothesis more probable, while at the same time making incorrect hypotheses less probable.

There are other objective functions for discriminative training, for example, Minimum Phone Error (MPE), Minimum Word Error (MWE), and Large Margin (LM). Details of the criteria are discussed in [9][26].

2.7.2 MLP feature

A Multi-Layer Perceptron (MLP) is a feed-forward Artificial Neural Network (ANN) consisting of multiple layers of nodes in a directed graph [28]. Figure 2.4 illustrates a simple example of an ANN with two hidden layers.

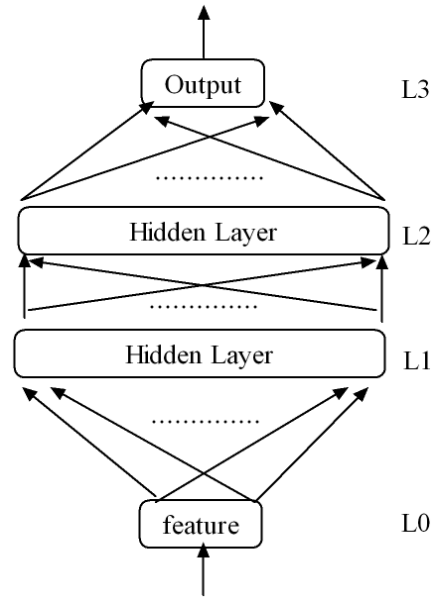


Figure 2.4 Example of MLP with two hidden layers

We can view the ANN as a classifier that maps input data onto appropriate output classes, and we also can view ANN as a feature extractor. In the hidden layers (L1, L2), the mappings follow the formulas 2.12 and 2.13:

$$x^L = \sigma(W^T x^{L-1} + a) \quad (2.14)$$

where $\sigma(t)$ is the sigmoid function: $\sigma(t) = \frac{1}{1+\exp(-t)}$

For the top layer (L3), soft max can be used to derive the posterior probability $p(y|x)$

$$p(y|x) = \frac{\exp(w_y^T x^L + a_y)}{\sum_{y'} \exp(w_{y'}^T x^L + a_{y'})} \quad (2.15)$$

where y is the output class vector given the input vector x from the last hidden layer, and the input x can be viewed as the ANN extracted features from the original input feature data. ANN is a powerful classifier, and it is a discriminative model in comparison with the generative model of HMM/GMM. MLP can be trained with a supervised learning technique called back propagation [29].

One way to use ANN in speech recognition is building a HMM/ANN hybrid, where we need to convert the class posterior probabilities from ANN into likelihood scores. With the recently proposed Deep Neural Network(DNN), which use a large number of hidden layers, this approach works very well in some large tasks [30]. Another way of using ANN that maintains traditional HMM/GMM framework is referred as tandem feature [10], which is illustrated in Figure 2.5.

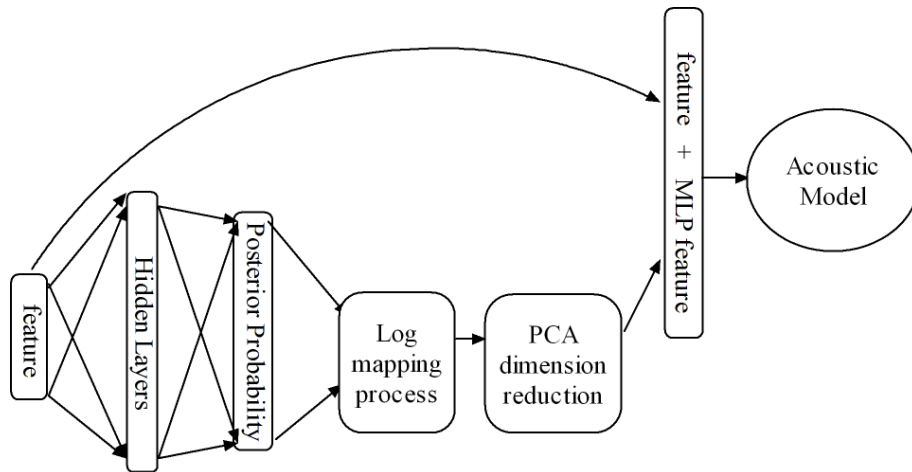


Figure 2.5 Illustration of MLP tandem feature generation

The original speech feature (for example 39 Dimension MFCC feature) is used to train a MLP. We therefore use the model to generate the posterior probabilities of the speech features for each phoneme class. After that, we first take log of the posterior probabilities to make the distribution more spread and then perform Principle Component Analysis

(PCA) for feature dimension reduction. We concatenate the dimension reduced MLP feature and the original MFCC feature for acoustic modeling.

2.7.3 Cross Validation Expectation Maximization (CVEM)

CVEM is a maximum likelihood training algorithm. Unlike the traditional Expectation Maximization (EM) training, it attempts to compensate for the overfitting problem in the conventional EM algorithm by using the data sampling technique of cross validation [12].

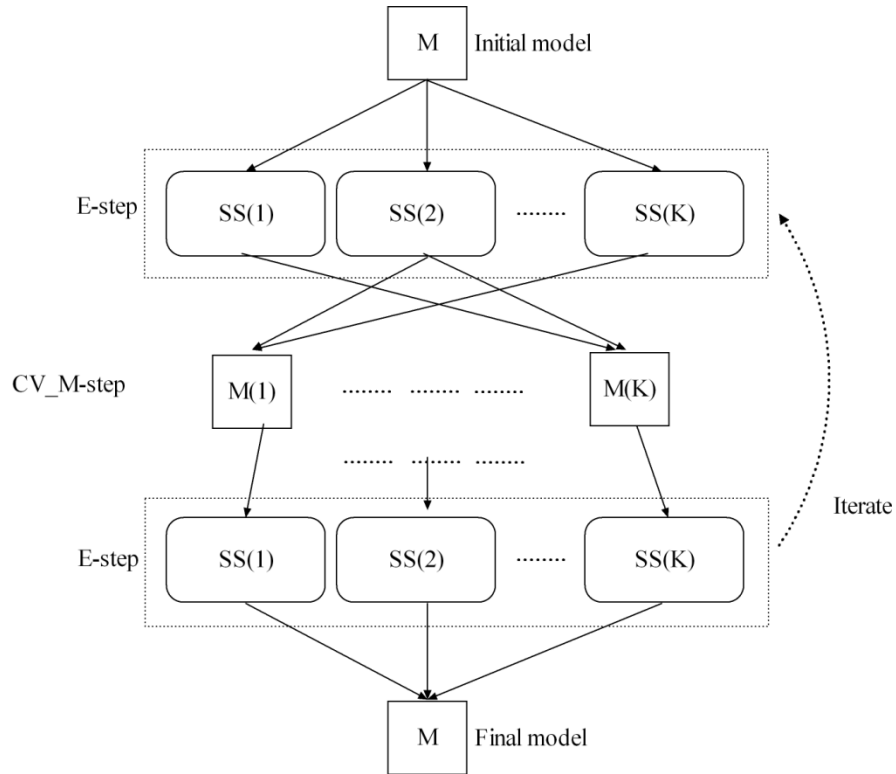


Figure 2.6 Illustration of CVEM training

For a K -fold CVEM, the training data is partitioned into K subsets. Within the loop of EM, K temporary models are used. In the E-step, sufficient statistics of the k th subset data are computed by using the k th model, and in the M-step, the k th model is updated by the sufficient statistics computed from all data excluding the k th subset. At the end of the EM iteration, the sufficient statistics of all data are pooled to estimate a single model.

2.8 Summary

A typical speech recognition system is usually organized as shown in the block diagram in Fig 2.6. The basic idea is training the models we discussed above with a labeled speech corpus, and using the trained models to find the best word sequences for the test speech inputs.

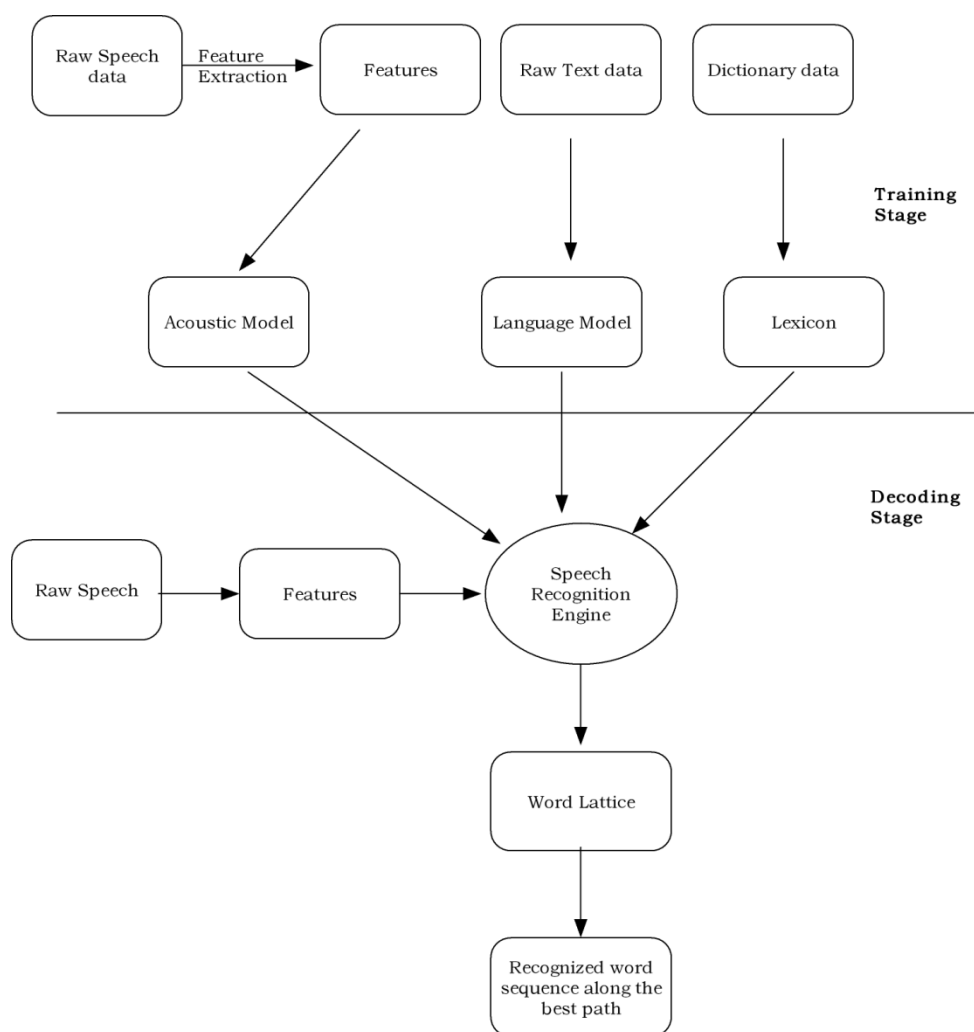


Figure 2.7 Diagram of an automatic speech recognition system

Chapter 3

Explicit Phonetic Decision Tree Tying

Speech recognition tasks can be categorized by different levels of difficulties. Conversational speech, which is characterized by wide variations in word pronunciations, is very hard for machine recognition among all the other tasks such as dictation or reading speech recognition. Especially, speaker-independent conversational speech recognition tasks need to handle more pronunciation variations than speaker-dependent ones since different people use different ways to pronounce words. To successfully model conversational speech, effectively handling pronunciation variations plays the key role. The following figure reveals 3 processing levels in typical ASR tasks.

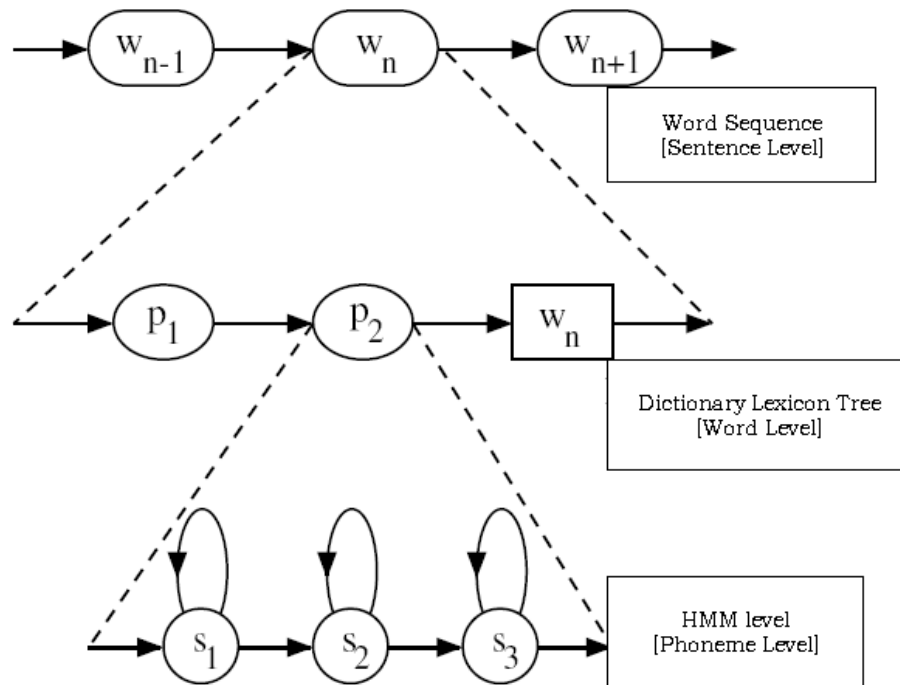


Figure 3.1 The three levels of speech recognition

According to this three-level speech recognition framework, we can apply different methods to solve the speech variation problem at different levels. At the sentence level, we can use linguistic features of words to model prosody induced variations [31]. At word level, the variations are normally modeled by multiple pronunciations in a word dictionary. The use of context-dependent acoustic models can be categorized to phoneme level [32][33][34]. The following is an example of multiple pronunciations for the word LETTER:

LETTER[a]:	L	EH	T	AXR
LETTER[b]:	L	EH	DX	AXR

Simply put, this method attempts to incorporate several most likely pronunciations for every word. “Letter” has different pronunciations in different circumstances, and so, the two pronunciations are both valid. We normally add both of them to the lexicon tree to make sure that no matter which pronunciation is observed, we will have a good chance of getting the correct word “letter”. We refer this kind of solution as “explicit approach” in modeling speech variation. However, this approach is expensive and error prone. Also it decreases the recognition speed, since a large lexicon tree means a large space in hypothesis search. Furthermore, introducing multiple pronunciations for each word will also add confusion, since the discrimination between acoustic features is not strong enough, and the confusion will affect both the training procedure and the decoding procedure. In many works only small improvements to word accuracy performance were observed [35] for the “explicit approach”.

At the HMM level, each state in a HMM can be modeled by a Gaussian mixture density which is robustly tied to the same state of several different CD-phonemes. The state tying is usually done by performing a data driven clustering or by combining knowledge and data in a Phonetic Decision Tree (PDT) based tying. [36] Therefore, each state can implicitly handle some speech variations while maintaining a compact model. Implicit methods are believed to be a better solution than explicitly adding multiple pronunciation entries for each word in a lexicon. First, it is more balanced between modeling speech variations and avoiding confusions. Second, its implementation for decoding search is easier.

Many efforts have been made to improve PDT state tying in acoustic modeling, For example, k-step look-ahead and stochastic full lookahead is one approach that attempt to build globally optimized trees instead of the traditional locally optimized decision trees [37]. Robust PDT is proposed with a two-level segmental clustering that includes the basic PDT and the agglomerative clustering of rare acoustic phonetic events [38]. Furthermore, instead of using phoneme level data to build PDT, acoustic model can also be trained based on the syllable structure of speech [39].

This chapter is organized as follows. First in Section 3.1 we discuss the background of PDT clustering. In Section 3.2 we talk about the proposed explicit PDT clustering that allows sharing data between different phones. Finally, we discuss how to enhance the performance of speech recognition by adopting multiple methods for acoustic modeling.

3.1 Phonetic decision tree background

As discussed above, Context-Dependent (CD) phone units are used in acoustic modeling because acoustic realization of a phoneme changes with the articulations of its neighboring phonemes. The most common CD HMM model is triphone, which has a good balance between complexity and efficiency. Researchers argue that long Context-Dependent phone units promise a better performance, but with a huge cost of increased model parameters. The consequence is compromising the training and decoding speed, the storage space, as well as the robustness of parameter estimation when training data is limited.

The target of PDT is to cluster triphone states. As we've just discussed, speech variations can be modeled by the clustered states, also called tied states. Each clustered state is shared by several similar triphones. In this way, each clustered state has more training data than individual triphones and is robust to handle pronunciation variations.

Unlike pure data driven clustering methods such as K-means, knowledge based PDT is much widely used in speech modeling due to its effectiveness for large data sets. The knowledge source we have is linguistic characteristic of the phonemes and their neighbors. For example:

"Nasal"	{ *+m,*+n,*+en,*+nx }
"IVowel"	{ *+ih,*+iy }
"OVowel"	{ *+ao,*+oy,*+aa }
"Front"	{ *+p,*+pd,*+b,*+m,*+f,*+v,*+w,*+wh,*+iy,*+ih,*+eh }

For each triphone, we have two contexts, the left phone and the right phone. Questions that are used to split nodes in a decision tree are generated accordingly. For example, we have two questions for Nasal clusters as follows:

“R_Nasal” { *+m,*+n,*+en,*+nx }

“L_Nasal” { m+*,n+*,en+*,nx+* }

where R_Nasal checks whether the right neighbor of the center phone is a nasal-type phone, and L_Nasal checks whether the left neighbor of the center phone is a nasal-type phone.

The Decision Tree construction procedure is described below in Fig 3.2:

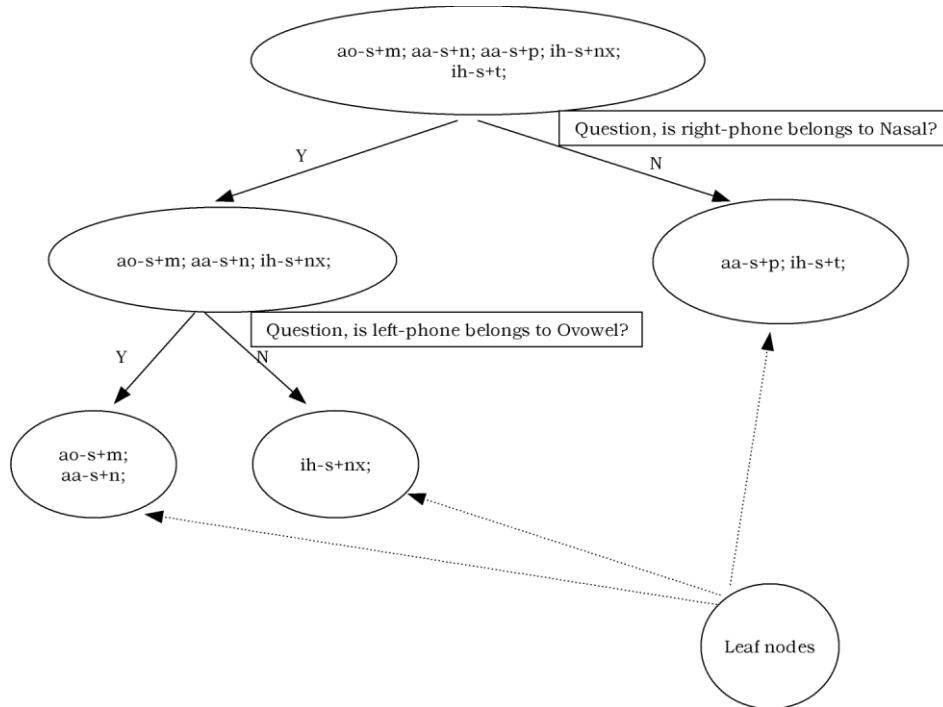


Figure 3.2 An example of decision tree construction

At the beginning, the root node contains all the triphone data with the center phone “s”. The nodes are split to leaf nodes by using the linguistic knowledge based questions we have just discussed. The broad categorizations of phones such as vowel, nasal, etc are used to form the questions. The questions ask if the triphones’ left context belongs to a specific category or if the triphones’ right context belongs to a specific category. The criterion for question selection is based on the likelihood gain. The question that produced the maximum likelihood gain locally will be used to split the node and two children nodes will be obtained. The likelihood gain is defined as:

$$\Delta L = L_{left} + L_{right} - L_{parent} \quad (3.1)$$

where the data distribution at each node is modeled by a Gaussian density. The same procedure is recursively applied to each node until it is stopped by some termination thresholds. Two thresholds are commonly used: one is minimum data count, and the other is minimum of the likelihood gain. The data count threshold is used because the leaf nodes should have enough data; otherwise it will not be possible to reliably estimate model parameters for each clustered state. The likelihood gain threshold is used to avoid unnecessary node splits.

This is the knowledge driven approach, because we cluster the triphones according to the linguistic contexts. However, data verification is also used to decide which question should be applied in each node. So the PDT approach is believed to have a better performance than pure data driven clustering such as K-means, and therefore it is widely used in ASR systems. Another advantage of PDT is that it can play the classification role. Many triphones may not appear in training data, but they still can be tied to a clustered state according to its linguistic properties.

3.2 Explicit PDT tying

Generally speaking, PDT is able to model pronunciation variations if we have enough training data [40]. However, training data are still very precious and expensive to obtain. What if we have a small amount of training data? Let's look at the following special case:

LETTER[a]:	L	EH	T	AXR
LETTER[b]:	L	EH	DX	AXR
LADDER[a]:	L	AA	D	AXR

When we observe the pronunciation pattern [b] for the word LETTER from a speech input, the triphone EH-T-AXR will have a likelihood score much lower than EH-DX-AXR if the pronunciation [b] for "LETTER" is not in the lexicon, then the correct hypothesis "LETTER" might not survive in the decoding search and an error word hypothesis, i.e. "LADDER", may be generated. This is a very common situation and is the key issue that we need to consider. It is believed that CD-phone modeling is able to model this kind of pronunciation variations, if training data are enough. In [40], the authors also argue that under the condition of very limited training data, the triphone acoustic model would not be robust enough to model pronunciation variations. It is a big challenge that with very limited data, how do we robustly model the pronunciation variations so as to increase the word recognition accuracy of ASR systems?

Since we have already used some linguistic knowledge in triphone clustering in decision trees, what if we use similar knowledge again to perform clustering on the center phone? This Phoneme Tying (PT) approach can force explicitly data sharing between center phonemes that have similar characteristics, and data sharing is expected to enhance the pronunciation variation modeling especially in limited training data. This idea is illustrated in the following example in Figure 3.3.

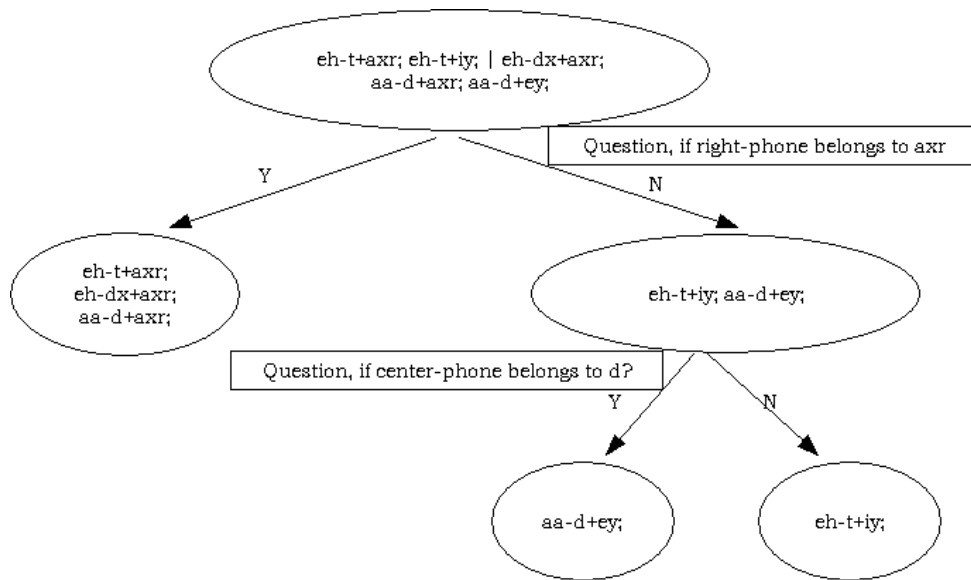


Figure 3.3 Explicit decision tree triphone tying

In this example we can see that the tri-phoneme eh-t+axr is supplemented by some training data that belongs to the center phoneme d. Therefore it may enhance the model to solve the insufficient data and the variation problems. Unfortunately this approach also introduces confusion between the phoneme t and the phoneme d. The consequences could be that the discrimination between the phoneme t and phoneme d is decreased.

In the traditional PDT clustering, we build a PDT for each state of each phoneme. Suppose we have k phonemes and n emitting states in HMM, then we will have k*n independent decision trees. This can be considered as the extreme case of the Explicit

PDT (EPDT). Due to center Phoneme data Sharing (PS), a minimum of n , and a maximum of $k*n$ trees can be built depending on the top-down clustering strategy.

We conducted experiments on several selected center phone clustering strategies in EPDT and found that in some types of clusterings, the EPDT will generate improved recognition results. Detailed experiments are presented in Chapter 6.

Table 3.1 An example of a state cluster in EPDT in the telehealth task

State ST_21_40
ae+z hh-ae+z r-ae+z r-ae+dh g-eh+dh w-eh+dh r-eh+z s-eh+z wh-eh+dh hh-eh+z

We also tested EPDT on another extreme case, which put all the phoneme data together and only built a Single Tree (SingleTree) for each state. This approach was originally proposed in [41]. In that research, a very positive gain in recognition accuracy was reported on the SwitchBoard task [42] in comparison with the baseline decision tree tying as discuss in Section 3.1.

Unfortunately, the performance gain of the single-tree method is marginal in our telehealth ASR task. Here is a possible explanation: by using the single-tree approach, we benefit from modeling pronunciation variations, but we also suffer from the confusions that are introduced by sharing phoneme data. Comparing with the speaker independent SwitchBoard task, our telehealth task is speaker dependent, and thus less pronunciation variations may be present. Therefore, the performance loss may be due to a larger confusion error than gains in pronunciation variation modeling.

How to address this problem? We adopt the approach of combining multiple models and discuss it in the next section.

3.3 Explicit PDT tying for multiple models

As we've just discussed, single tree explicit PDT tying is not suitable for a speaker dependent task since it introduces more confusion than benefiting from modeling pronunciation variations. How to decrease the confusion as well as to maintain the pronunciation variation modeling that we wish accomplish? Here we adopt the combining multiple models method that is potentially capable of maintaining the gain from pronunciation variation modeling but also decreasing the confusion.

Simply put, the combining multiple models approach allows each triphone to be tied not only to one state cluster, but also tied to multiple state clusters that are generated in different ways. Look at the following example:

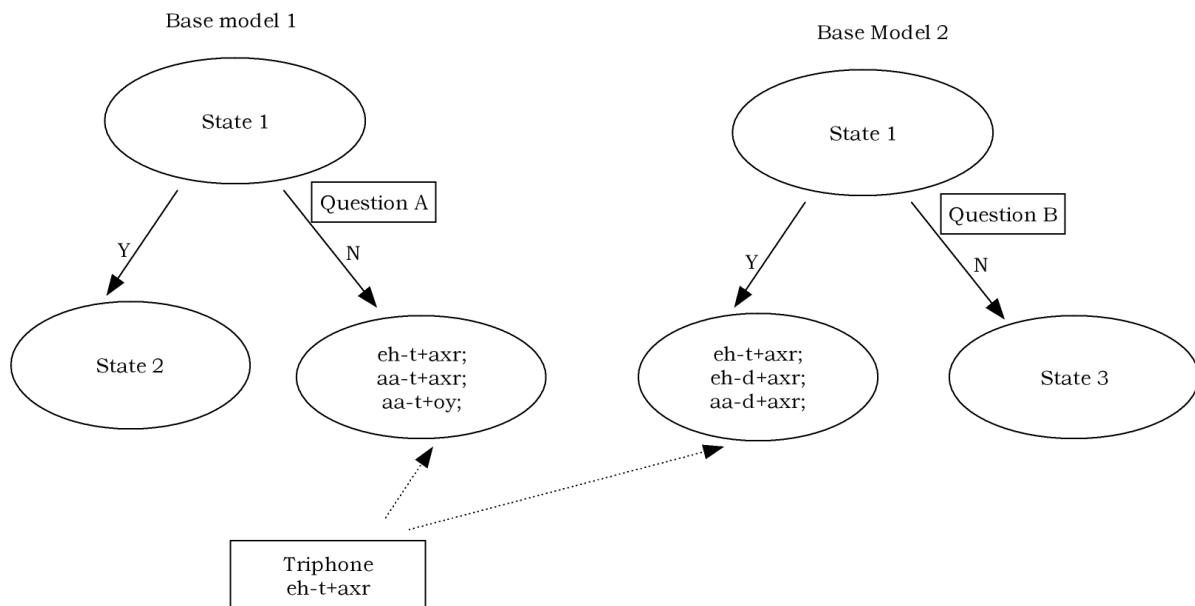


Figure 3.4 An example of triphone tying in ensemble acoustic model

In this example, Triphone eh-t+axr is now tied to two state clusters. Here we combine the baseline model that has $K*3$ trees (K monophones, 3 emitting states) with the 3-Tree model and a 6-Tree model where we use 3 trees for vowels and 3 trees for consonants. In the decoding stage, for each triphone state we compute the GMD likelihood score from each base model and combine the scores of the three base models using a simple average. Some of the other score combining methods will be discussed in Chapter 4.

It is noted that the combining method of tying triphone models across different trees follows the method of [6], where random forests were used to generate an ensemble of acoustic models. In the current work, different models are generated by applying explicit knowledge in EPDTs as well as by the baseline models, rather than by randomly sampling questions in phone-state specific PDTs.

By applying this idea, we can maintain the purity of the baseline model and also provide a partial solution to the problem of pronunciation variation across phonemes. As the result, model robustness is improved and performance gain is shown in our experiment. Detailed experiment results will be discussed in Chapter 6.

3.4 Hierarchical ensemble model based on different mixture size

The previously discussed method of combining EPDT model and baseline PDT models as an ensemble model can be viewed as a hierarchical ensemble modeling approach. The baseline PDT model has no sharing in center phones. The 6-tree EPDT model with 3 trees for vowels and 3 trees for consonants has some sharing, and the 3-tree

EPDT model has more sharing than both the baseline PDT model and the 6-tree EPDT model since it allows data sharing between any two center phones. Hierarchical ensemble model has the potential ability to improve classification performance, which has been shown in [43] on a handwriting recognition task.

Mixture size is an important parameter in GMD. A small mixture sized model requires small amount of training data and is normally inaccurate. A large mixture sized model is accurate but requires a lot of training data to be reliable estimated. Here mixture size is a very good parameter for generating a hierarchical ensemble model. Therefore Hierarchical GMM (HGMM) or Multiple models with Different Mixture Sizes (MDMS) is proposed. For each tied triphone state, we simply train GMD models with different mixture sizes and combine their likelihood scores together during decoding search for each speech frame. This method helped improve word accuracy performance in our telehealth task as well as other ASR tasks. Detailed experimental results are discussed in Chapter 6.

Chapter 4

Data Sampling Based Ensemble Acoustic Modeling

Although compromised in computation speed, combining multiple classifiers is widely observed to produce improved classification accuracy in many tasks.

In order to obtain an ensemble model, first, we need to decide the base classifier (Gaussian Mixture Density is a dominate approach for acoustic modeling of triphone units); second, we need to decide the methods for producing a model ensemble, such as feature sampling used in Random Forest [6] or data sampling; third, we need to decide how to combine the outputs from different classifiers.

In this chapter, we continue investigation on ensemble method for speech modeling. In Section 4.1 and 4.2 we discuss the background of ensemble approach used in speech recognition. In Section 4.3 we propose a Cross Validation (CV) based data sampling method that generates very good results. In Section 4.4 we discuss random data sampling method. In Section 4.5 we propose speak-clustering based data sampling in order to improve inter-model diversity as well as EAM quality.

4.1 Ensemble model for acoustic score combination

Ensemble method is a very promising direction that is under active investigation in many machine learning applications. In the speech recognition field, the system combining approach named ROVER is very successful in reducing word error rates [1]. Combining at the system output level, ROVER uses several speech recognition systems

to perform speech decoding simultaneously and combine their outputs through an alignment of word hypothesis. ROVER generates the best word sequence through a majority voting procedure. ROVER enhances word accuracy performance, but it also introduces system complexity and computation cost, thus compromised decoding speed, which is a key factor of system performance in online recognition tasks. A simple example of ROVER is described in Figure 4.1.

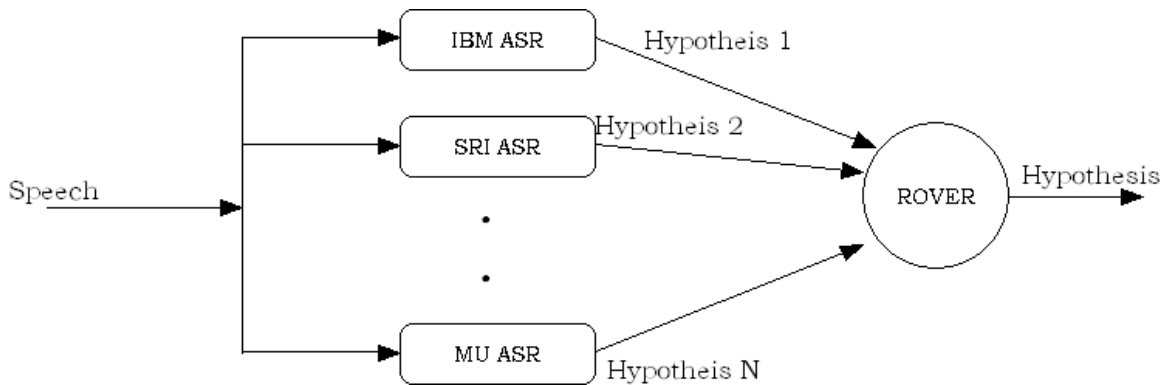


Figure 4.1 An example of ROVER framework

Unlike ROVER, our ensemble method combines a set of acoustic models. This idea is as the following: several acoustic models are used to compute the likelihood scores for the same speech utterance and the scores are combined for each speech frame (at the acoustic method level); the acoustic scores are integrated along with language model scores to generate the most possible word hypotheses. It is a simple and low cost implementation which, amazingly, gives very good results.

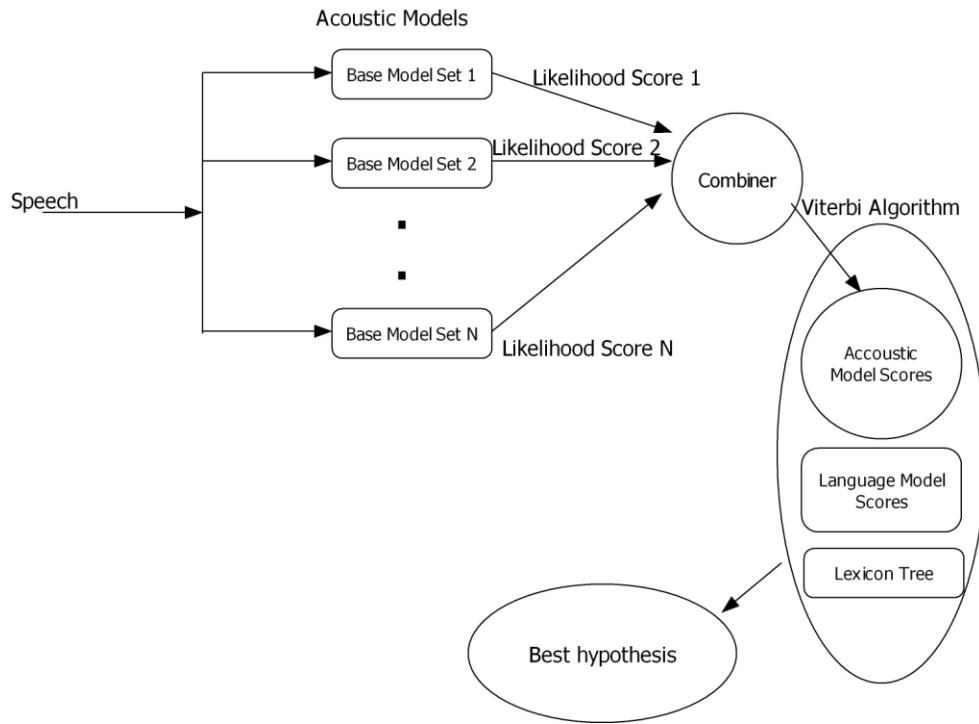


Figure 4.2 Ensemble model framework in the telehealth captioning system

This ensemble model framework was first introduced in our telehealth task as the Random Forest (RF) approach [6]. RF was used to train a set of PDTs for each speech unit and obtain multiple acoustic models accordingly by random sampling on decision tree questions, where the questions are also called features in the decision tree literature. Different combining methods such as arithmetic average, N-best average and weighted average were used to generate the combined score. The combining weights can also be obtained via maximum likelihood estimation or confidence measuring. The RF PDTs based ensemble classifier has been shown very successful in the telehealth task.

4.2 Sampling training data to generate EAM

We first choose a data sampling method to sample a training dataset D into K sampled date sets $D_k, k = 1, \dots, K$. Ensemble acoustic model training is then performed through a procedure of 4 steps, as shown below in Figure 4.3.

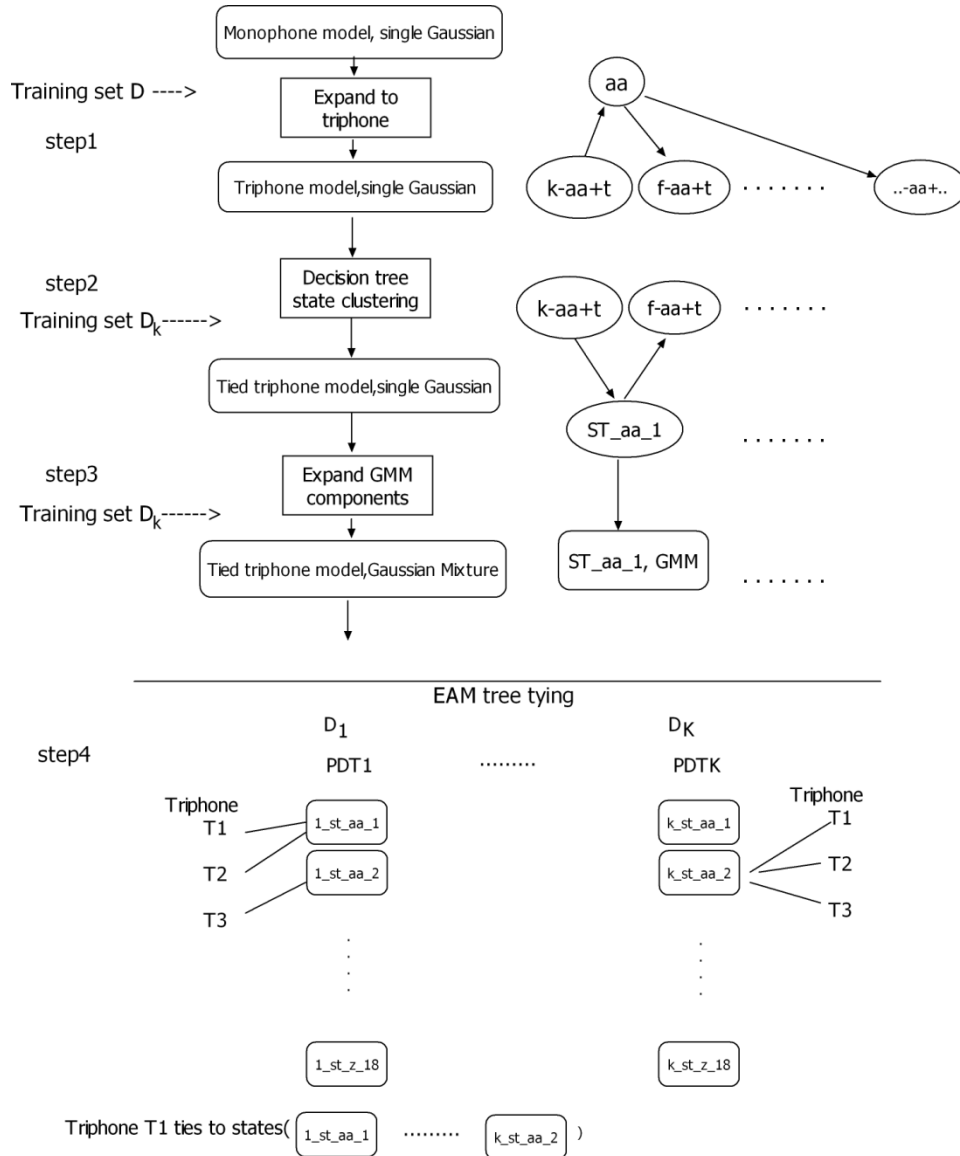


Figure 4.3 Generating an ensemble model by data sampling

In step1 we train a set of basic untied triphone models for every triphones by extending the monophone models. We apply all the training data in this step since it produces stable monophone models. In step2 we use PDT to do the state clustering where several triphones will be tied to one state cluster. Therefore we can decrease the parameters from the individual triphone models as well as increase the model robustness. In step3, we train Gaussian mixture density instead of single Gaussian density for each state cluster. In step 4 we tie each triphone to K state clusters that are generated by K PDTs trained from K sampled datasets.

Some of the triphones may not appear in the training data, which are called unseen triphones. In general, there will be more unseen triphones in a sampled dataset than the original full training set because a sampled dataset is a subset of the full training data set. However, due to the classification capability of the decision tree method, we are able to assign unseen triphones to the tied states in each tree, and therefore for each triphone state, no matter it is present or absent in a sampled dataset, we are able to tie it to K state clusters as described in the step 4 above.

When we use sampled training data in step2 and step3, both steps will generate variations in the models. Similar to sampling questions in RF, sampling data in step2 will generate different decision tree structures. It remains a question as to which method will produce a better performance. In Chapter 6, we will evaluate the difference between feature sampling and data sampling.

4.3 Cross-validation data sampling

In general cases of classifier design, we have a training data set to train models, we use a validation data set to tune some parameters in the models, and we use the test data to evaluate the performance of the models. However, in some cases, the amount of training data are small and therefore very precious. In such a case, we combine the validation data with the training data and use cross validation approach to tune the parameters.

Let D be a training set, and D_k be a subset for K -fold Cross-Validation (CV), that is,

$$D = \bigcup_{k=1}^K D_k \quad (4.1)$$

$$D_i \cap D_j = \phi \quad \text{for } i \neq j$$

For training the i th model, we use $D - D_i$ as the training data, and use D_i as the validation data. We do this K times for a K -fold CV and obtain the tuning parameters by averaging. Figure 4.4 shows an example for $K=5$ and $i=5$.

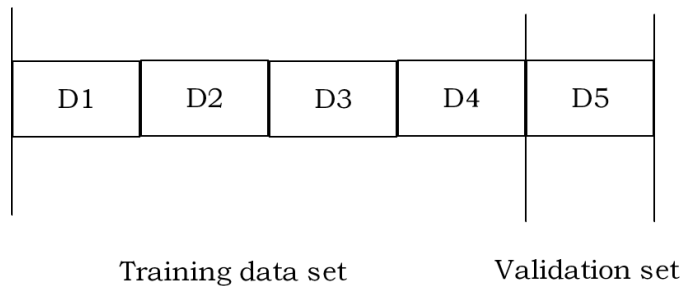


Figure 4.4 Illustration of 5-fold cross validation

CV based sampling is a special case of data sampling. The characteristic of CV based sampling is that, all the data samples will be used exactly $K-1$ times in model training. An example of CV based EAM is illustrated in Figure 4.5. It is believed here

that without prior knowledge, training data should be treated with equal importance and random sampling with replacement (bootstrap) or without replacement may produce bias.

Detailed experiments on CV based data sampling will be presented in Chapter 6.

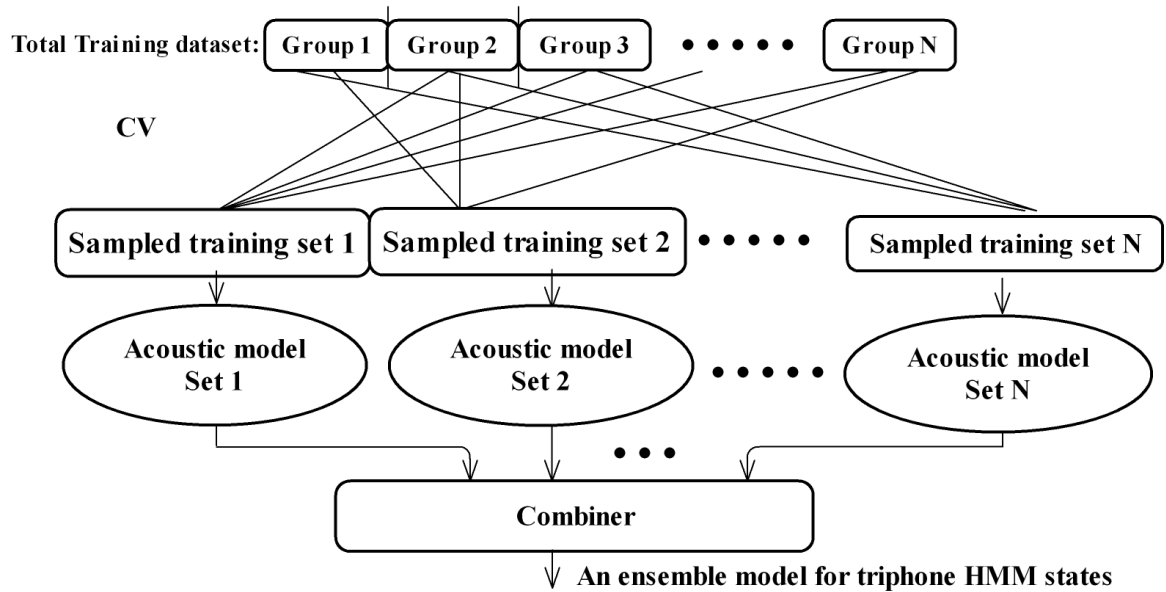


Figure 4.5 Illustration of CV based EAM

4.4 Random data sampling

Random sampling is a very common and simple method. We choose random sampling without replacement as our reference for comparison with the proposed CV based sampling. Here is the procedure:

Step 0. Clean subset X_i

Step 1. Randomly select a data sample from training data set X .

Step 2. Pull a data sample from the training data and place it into X_i .

Step 3. Repeat the steps 1 and 2 until data in X is less than $T\%$.

Return to Step 1 until we obtain K datasets $(X_1, X_2, X_3, \dots, X_k)$

Here, the $T\%$ in step 3 is a parameter that could be set to different values. We can choose 10% for comparison with 10-fold CV model. In the current task, the unit of data sampling is sentence. Details of the experiments will be presented in Chapter 6.

4.5 Speaker clustering based data sampling

For speaker independent speech recognition tasks, it is of interest to investigate using speaker clustering based data sampling to increase the diversity among the acoustic model sets, since the sampled training data sets thus produced may each reflect a type of speaker characteristics. The sampled datasets produced from the proposed speaker clustering are overlapped in general, and by controlling the amount of overlaps we may trade the accuracy of individual model sets for the diversity among them. It is noted that in [44], an utterance-clustering processing was performed on a training data set to generate non-overlapped data sets for training multiple recognition systems, which improved recognition accuracy over a single system trained from the full training dataset.

We propose to generate speaker clustering based overlapped data sets via a two-stage procedure. In the first stage, we train acoustic models referred to as core models which are used to generate overlapped speaker clusters for the second stage. We describe two speaker clustering methods for training the core models. The first method is to implicitly cluster the speakers based on their speech utterance likelihood scores (LSC). The second method is to explicitly cluster the speakers based on the distances between speaker

models by using the K-medoids algorithm [45]. In the second stage, the core models are used to compute the speech utterance likelihood scores to generate the overlapped, sampled training data sets with each having the size of a fixed fraction α of the total training data, where α implicitly controls the amount of overlap among datasets, i.e., the smaller the α , the less the overlap, and vice versa. By isolating the core model training from the overlapped speaker clustering into two stages, we obtain the flexibility of generating speaker clustered datasets with different fraction parameters without the need for retraining the core models. As in CV data sampling, from the overlapped, speaker clustered multiple training data sets, multiple sets of acoustic models are trained. It is noted that when the overlaps among the sampled datasets are small, adaptively estimating the combiner weights from test speech may be desirable to emphasize the model set that best fit the current test speaker. Here, however, only simple averaging is used for model combination. The two-stage procedure is detailed below for generating N overlapped speaker clusters which define N sampled data sets, and the two-stage procedure using the first method in stage-1 is illustrated in Figure 4.5.

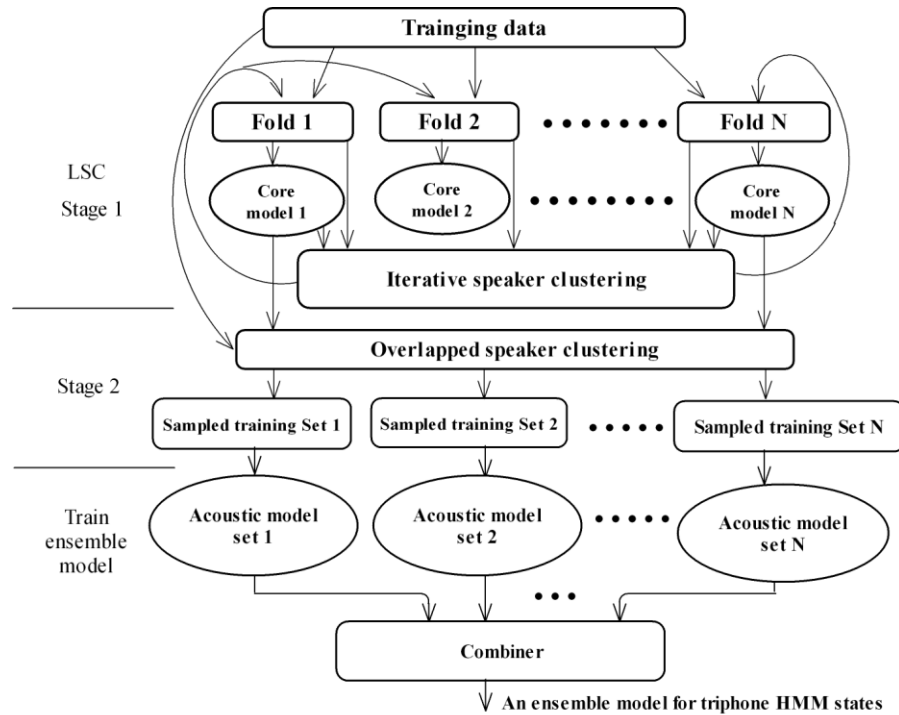


Figure 4.5 N-fold speaker clustering based data sampling for ensemble model training

Stage 1: Core model training

Method 1 Utterance-likelihood based speaker clustering

Step-1 Randomly select K speakers without replacement, with $K \leq \frac{1}{N}$ of total training speakers, and assign them to one of N folds; repeat N times to fill up the N folds.

Step-2 Train the core acoustic models for each fold by using the speech data of the selected K speakers in the fold.

Step-3 Calculate forced-alignment likelihood scores on the speech utterances of the selected speakers by using the core models. Reassign to each fold the K speakers whose averaged speech likelihood scores computed by the core model of the fold are ranked above the rest of the speakers. To avoid the problem that one speaker may be favored by many folds (which would decrease model diversity), a speaker selection probability, 1-

t/T , is applied in the speaker assignment step to limit the chance that a speaker is excessively used in many folds, where t is the number of folds that a speaker has been assigned in the current iteration, and T is the maximum allowed folds.

Step-4 Return to Step-2 until model convergence or a sufficient number of iterations has been reached.

Method 2 Speaker-model distance based speaker clustering

Step-1 Train a GMD speaker model for each speaker, and use the Monte Carlo sampling method of [46] to obtain the Kullback-Leibler (KL) distance between each pair of speaker models.

Step-2 Randomly sample N speaker models as the medoids, and use the N -medoids clustering algorithm with the KL distance to obtain N clusters.

Repeat this process of random initialization and N -medoids clustering to obtain multiple configurations of N clusters, and pick the configuration that satisfies the following two criteria:

The smallest cluster has at least L speakers (for example, $L=1/2N$ of total used speakers).

The average pairwise cluster distance is maximized (the distance between two clusters is defined as the maximum distance of two speakers in the two clusters).

Step-3 Train a core acoustic model for each speaker cluster.

Stage 2: Overlapped speaker clustering

For each sampled data set, fix its desired fraction of the full training dataset as $\alpha = M/N$ with $M < N$, which implicitly controls the data overlaps as discussed above. For each

speaker, make a rank order list of the N clusters or folds according to the likelihood scores of the speaker's data computed by the N core models, and assign each speaker to the top M folds. Like the CV method in Section 4.3, the folds are made to have approximately the same number of speakers, where if one or more folds are full, then the next fold or folds in the rank order list will be used for the speaker.

Note that for large training tasks, using the method 1 to train the core models requires less computation than the method 2, since in the method 1 it is suffice to use only a small percentage of training data. In contrast, the method 2 requires training the speaker models for all speakers, and computing the distances between every pair of speaker models is costly.

4.6 Combiner design

As we discussed at the beginning of this chapter, in the decoding stage of speech recognition we need to combine the acoustic scores from the multiple acoustic models. Linear combination or nonlinear combination such as Bayesian Belief Network can be used [47]. For simplicity, we just consider linear combination. Suppose we have a feature vector x_t , the likelihood of it belongs to a specific ensemble tied state H_l in a HMM is:

$$P(x_t | H_l) = \sum_{k=1}^K w_{lk} p(x_t | M_{l_k}) \quad (4.2)$$

where K is the number of models, $p(x_t | M_{l_k})$ is a Gaussian mixture density score from

the k th acoustic model. We need to estimate the weights w_{lk} that satisfy the constraint of

$$\sum_{k=1}^K w_{lk} = 1 \text{ and } w_{lk} > 0.$$

Therefore for a simple average the weights could be defined as $w_{lk} = \frac{1}{K}$.

By sorting the K likelihood scores $p(x_t | M_{l,k})$ in a decreasing order, we have several special cases of combining weights defined as the following:

MAX: $w_{lk} = (1, 0, 0, \dots, 0)_K$. We just choose the maximum score that the K models give.

m-best: $w_{lk} = (\frac{1}{m}, \frac{1}{m}, \frac{1}{m}, \dots, 0, 0)_K$. We select the m -best scores and average them.

m-Trimmed-Average: $w_{lk} = (0, 0, \dots, \frac{1}{m}, \frac{1}{m}, \frac{1}{m}, \dots, 0, 0)_K$. We throw away the best

and the worst few scores and average the rest. This is supposed to be more stable since it excludes the outliers.

Median: It is a special case of m-Trimmed -Average, when m is equal to 1.

There are other weight estimating methods, which were described in [6].

Chapter 5

Ensemble Acoustic Model Clustering

HMM-based acoustic models built from data sampling EAM are generally very large, especially when a large number of models or full covariance matrices are used for Gaussian densities. Therefore, compacting the ensemble acoustic model to a reasonable size for practical applications while maintaining its good performance is needed. Toward this goal, in this chapter, we discuss and investigate several distance measures and algorithms for clustering methods. The distance measures include Entropy, KL, Bhattacharyya, Chernoff and their weighted versions. For clustering algorithms, besides the conventional greedy agglomerative clustering, algorithms such as N-Best distance Refinement (NBR), K-step LookAhead (KLA), Breadth-First Search (BFS) are proposed.

5.1 Distance measures for Gaussian density clustering

In this section, we discuss several distances to measure the dissimilarity between a pair of Gaussian density functions. These distances include KL divergence, entropy change, Bhattacharyya distance, and Bayes error.

5.1.1 KL divergence

The KL divergence is commonly applied to measure the dissimilarity between two distributions [50]. Given the two distributions, $f_1(x)$ and $f_2(x)$, the KL divergence is defined as

$$D_{KL}(f_1 \parallel f_2) = \int f_1(x) \log \frac{f_1(x)}{f_2(x)} dx \quad (5.1)$$

Since this definition is asymmetric, a symmetric version defined below is used.

$$D_{KLS}(f_1 \parallel f_2) = \int [f_1(x) \log \frac{f_1(x)}{f_2(x)} + f_2(x) \log \frac{f_2(x)}{f_1(x)}] dx \quad (5.2)$$

5.1.2 Entropy change

The entropy criterion (ENT) measures the change of entropy when two distributions are merged. If the two distributions f_1 and f_2 are Gaussians, then the change in entropy [15] is computed as

$$D_{ent}(f_1 \parallel f_2) = (w_1 + w_2) \log |\Sigma| - w_1 \log |\Sigma_1| - w_2 \log |\Sigma_2| \quad (5.3)$$

where Σ is the covariance matrix after the merge, Σ_1 and Σ_2 are the covariance matrices the Gaussian combining candidates f_1, f_2 and w_1, w_2 are the weights for the Gaussians f_1 and f_2 in the Gaussian Mixture.

5.1.3 Bayes error, Bhattacharyya and Chernoff distances

The Bayes error measures the overlap of two distributions. Given the two distributions, $f_1(x)$ and $f_2(x)$, the Bayes error is defined as

$$D_{Bayes}(f_1 \parallel f_2) = \int \min(f_1(x), f_2(x)) dx \quad (5.4)$$

There is no closed-form expression for the Bayes error when $f_1(x)$ and $f_2(x)$ are multivariate Gaussians. However, it can be shown to be bounded from above by the Chernoff function:

$$C_s(f_1 \parallel f_2) = \int f_1(x)^s f_2(x)^{1-s} dx, \quad 0 \leq s \leq 1 \quad (5.5)$$

The Chernoff function in the above equation is a general function describing the distance between two distributions and it has close relationships with some other well-known distribution distances. Based on the Chernoff function, the Chernoff distance is defined to be the minimum of the above equation with respect to s :

$$D_{chern}(f_1 \parallel f_2) = \min_{0 \leq s \leq 1} \int f_1(x)^s f_2(x)^{1-s} dx \quad (5.6)$$

When both $f_1(x)$ and $f_2(x)$ are Gaussians, which is the case in this work, the Chernoff distance can be computed via Newton-Raphson algorithm. It is a convex function of s with a guaranteed convergence. A reasonable starting point is $s = 0.5$ which amounts to the Bhattacharyya distance. The details of formula derivation for the Newton-Raphson algorithm to obtain the Chernoff distance is elaborated in [51].

5.1.4 Weighted distances

It is noted that the KL, Bhattacharyya and Chernoff distances defined above do not include the weights of Gaussians in a Gaussian mixture densities. It is observed in [53] that weighted distances gave better performance than non-weighted distances. Therefore, it is of interest to evaluate weighted distance in clustering Gaussian components. The KL, Bhattacharyya, and Chernoff distances all have weighted forms. Here we take Bhattacharyya distance as an example. Given the two distributions, $f_1(x)$ and $f_2(x)$, suppose that the weight for $f_1(x)$ is w_1 and the weight for $f_2(x)$ is w_2 , the weighted Bhattacharyya distance is defined as

$$D_{bha}(f_1 \parallel f_2) = \int \sqrt{w_1 f_1(x) w_2 f_2(x)} dx \quad (5.7)$$

5.2 Gaussian component clustering algorithm

In this section, we discuss the proposed clustering algorithms. Suppose that a state in the EAM of random data sampling has a total of M Gaussians. Our task is to cluster the M Gaussian components into N Gaussian components. The most widely used method is to perform agglomerative clustering for each GMM state, i.e. calculate the distances between all pairs of the Gaussian component density in a mixture density and merge the

closest ones in a pair, so that the distance change between the GMMs before and after the merge are minimized. Repeat this process for M-N times to construct a binary tree of the Gaussian component density in the bottom-up direction. This process can be illustrated in the following formula.

$$D(f, g) = \sum_{i=1}^{M-N} \text{Distance}(f_a^i, f_b^i) \quad (5.8)$$

The conventional method is a greedy algorithm, where for the step i , we find the Gaussian density pair f_a^i and f_b^i with the smallest distance, combine them to a new Gaussian density component. We then repeat this process for the remaining Gaussian components until we reach the desired size. However, our optimization target is to find the minimum distance change $D(f, g)$ (For Entropy criterion then this criterion becomes minimum Entropy change), which is also called global optimization. In this case, for each step i , we are trying to look for the pair f_a^i and f_b^i such that their merge will lead to the minimum of $D(f, g)$. The exact minimization on $D(f, g)$ is computationally expensive and impractical. Here we discuss several methods to approximate the minimum $D(f, g)$.

5.2.1 Entropy based N-Best distance Refinement (NBR)

Both the Chernoff distance and the KL distance are computationally heavy, especially for full covariance Gaussians in a bootstrapped large GMM. It is noted that the computation of the Entropy distance is fast, and the clustering quality is also very good as evaluated in pilot experiments. The proposed NBR idea is using fast distance criterion like entropy to first select N-best combining candidates, and then use the slow but better distance criteria such as Chernoff or KL to refine the distance for the selected N candidate pairs. Since mixture weights are explicitly used in the entropy criterion, a potential advantage with the NBR approach is that the weights are implicitly used for non-weighted distances such as

KL, Bhattacharyya and Chernoff. In addition, NBR also provides more than one view of distance to potentially take advantage of complementary information from two different distances.

5.2.2 K-step LookAhead (KLA)

It is well known that the greedy approach of combining candidates based on the smallest distance in the current clustering stage only leads to a local optimization. Our target is global optimization that minimizes the total distance change $D(f,g)$ between the Gaussian mixture models f and g . In [37], KLA is applied to a global optimization of phonetic-decision-tree based triphone clustering. In this work, we hope global optimization can lead to better clustering than greedy local optimization, and KLA is therefore used to obtain a global optimized solution. Figure 5.1(a) illustrates an example where KLA no longer chooses the best combining candidate in the first step, and better solution will be obtained in the second step.

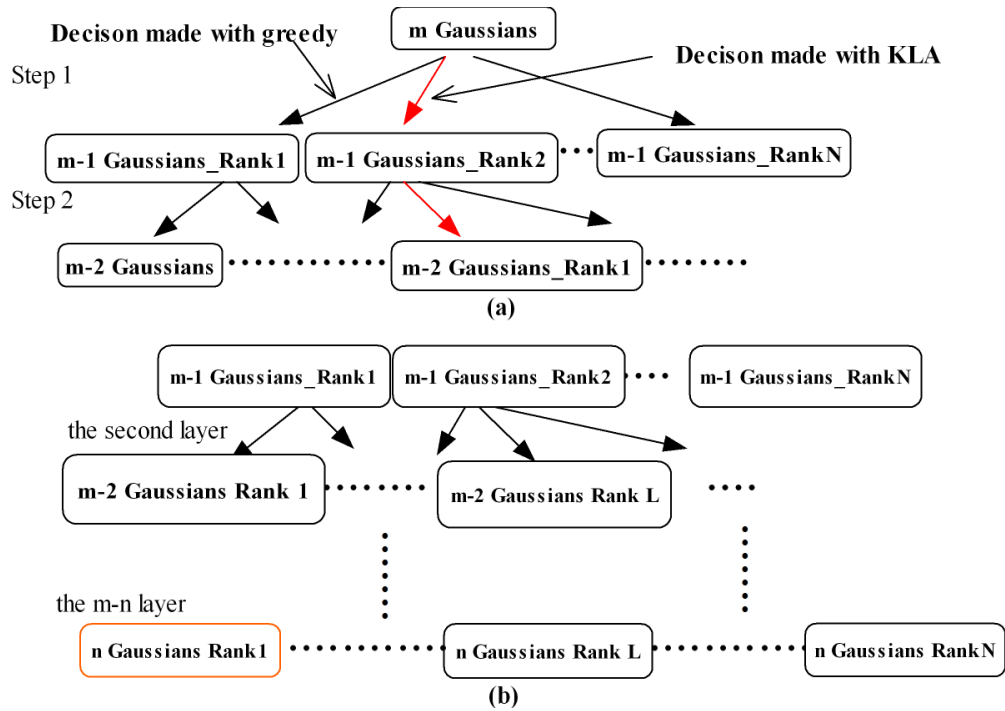


Figure 5.1 Illustration of using (a) K-step LA and (b) search for the best path

5.2.3 Breadth-First Search (BFS) global optimization

Searching the best combining path is another global optimization approach and it is computationally intensive. With a proper beam and pruning setup, the amount of computation can be reduced and a sub-optimal global solution can be obtained. Since acoustic model is trained off-line, and computational resource is becoming rich, it is worth a try to apply search methods in the task of Gaussian component clustering.

The clustering process using the Breadth First Search (BFS) algorithm is illustrated in Figure 5.1(b). At the first layer, we have one copy of the current mixture components. For each subsequent layer, we have the combining candidates from all copies of the previous layer. We extend the previous layer with all possible candidates, and keep top- N best copies in the subsequent layer as shown in Figure 5.1(b), where N is the beam width for this layer. Finally, we have N combining candidates at the last layer. We can therefore pick the best one among the N candidates.

Beam setup remains a question for this algorithm. A large beam will lead to slow speed, whereas a small beam will keep only limited candidates and the best path might be missed. Intuitively, at the beginning steps, we can keep the beam small, as the difference between the combining candidates is small. For the last few steps, the differences become bigger and we shall make the beam larger to avoid missing the potential best path. Based on this property, the beam width can be adaptively changed with the clustering process.

5.2.4 Two-pass global optimization

For a conventional one-pass clustering algorithm, each clustered state shall have N/M of its original Gaussian mixture components, so that the total number of Gaussians for the acoustic model is N . However, fixing the compression rate to N/M in each state is not the best option. Intuitively, some states may need more mixture components to represent a

more complex distribution while some states just need less due to their simpler distribution. To deal with this issue, we propose a two-pass approach to globally optimize the model structure. While fixing the total number of Gaussians to N , each tied state can have different compression rate. In the first pass, we can use Bayesian Information Criterion (BIC)[54] or other similar criteria to decide the specific number of Gaussian mixture components for each state. Following the information in the first pass, we conduct clustering in the second pass. In this way, we can still keep N Gaussian mixtures for the model, but the compression rate for each state is different.

Chapter 6

Experiments on Speech Recognition Tasks

6.1 ASR tasks

6.1.1 Telehealth task

Experiments were performed on the Telemedicine automatic captioning system developed in the Spoken Language and Information Processing Laboratory (SLIPL) at the university of Missouri-Columbia. Please refer to [55] for a detailed description of this task and system. The block diagram of this system is shown in Figure 6.1:

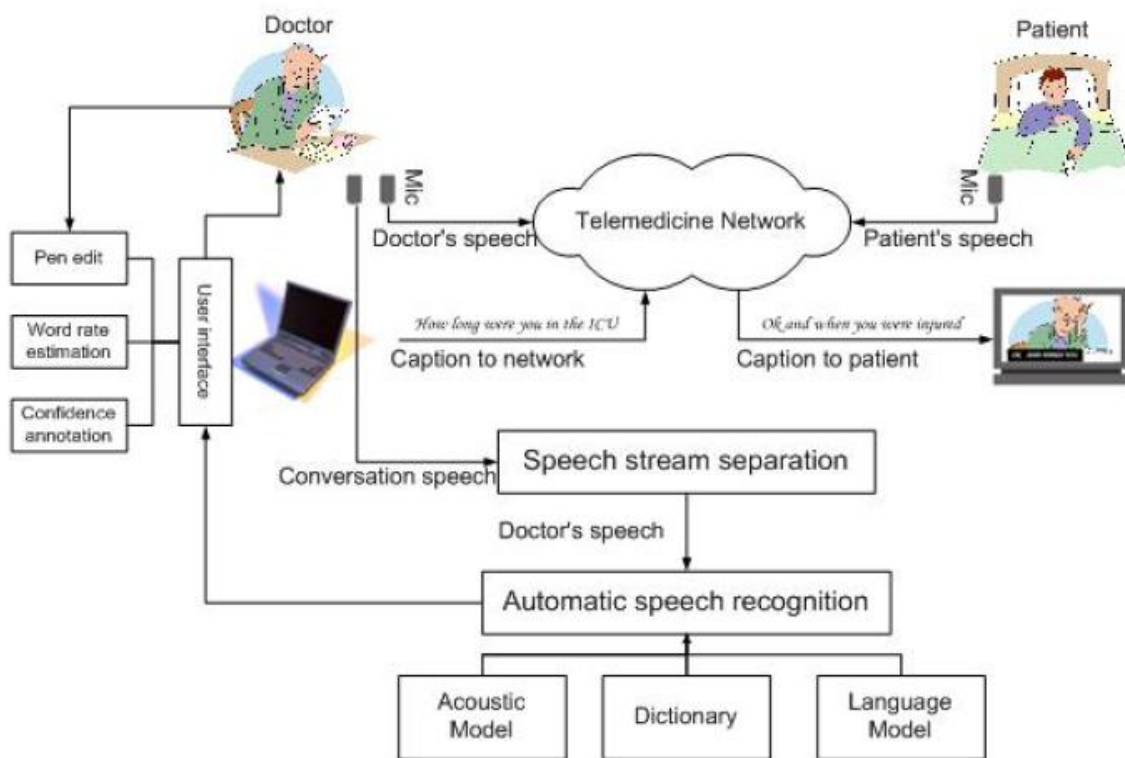


Figure 6.1 Block diagram of the automatic captioning system for telemedicine[55]

Speaker dependent acoustic models were trained for 5 speakers Dr. 1-Dr. 5. A summary of the data set is provided in Table 6.1. The training and test datasets were extracted speech data from healthcare providers' conversation with clients in mock

Telemedicine interviews. Original speech features consist of 39 components including 13 MFCCs and their first and second order time derivatives. Feature analysis was made at a 10 ms frame rate with 20 ms window size. Gaussian mixture density based Hidden Markov Models (GMD-HMM) were used for within-word triphone modeling, and the baseline GMM contained 16 Gaussian components. The task vocabulary was 46k, with 3.07% of vocabulary word being medical terms. Language models were word-class mixture trigram language models with Forward Weight Adjustment [56]. The decoding engine was based on TigerEngine 1.1 [57] that performed large vocabulary continuous speech recognition based on one-pass time synchronous Viterbi beam search, with novel Order-Preserving LM Context pre-computing that reduced LM look up time.

Table 6.1 Datasets used in the telehealth task: speech (min.)/text (no. of words).

	Training set	Test set
Dr. 1	210/35,348	29.8/5085
Dr. 2	200/39,398	14.3/2759
Dr. 3	145/28,700	19.3/3248
Dr. 4	180/39,148	27.8/6421
Dr. 5	250/44,967	12.1/3988

6.1.2 TIMIT task

The TIMIT phone recognition task was a popular ASR task, where the training set consisted of 3696 utterances from 462 speakers, and the standard test data set consisted of 1344 utterances from 168 speakers. The 39 phonemes as defined in [25] were used here. Phone bigram language model was trained from the transcriptions of all the 3696 utterances. Speech features consisted of 39 components including 13 MFCCs and their first and second order time derivatives, where feature analysis was made at a 10 ms frame

rate with a 20 ms window size. Again the phone HMMs each had three emitting states and each state had a GMD of size 16, and crossword triphone models were used. HTK [26] was used for training the individual model sets.

6.1.3 Pashto task

The data was collected and transcribed by DARPA under the Transtac project. [51] There were 135 hours of training data and 10 hours of test data. The feature space was constructed by splicing 9 frames of 24 dimensional PLP features and projecting down to a 40 dimensional space via LDA. Context-dependent quinphone states were tied by decision tree. A trigram language model with 1.2M of Tri- Bi- and Uni-grams was used for test, with a dictionary of 30K words. For acoustic modeling, full covariance Gaussian densities were used. The original bootstrapped acoustic model was constructed from 15 data-sampling produced base models with each subset covering 70% of original data. The bootstrapped model had 6K quinphone states with a total of 1.8M Gaussians. Two compression targets were investigated in this work: 100K Gaussians (1/18 of the original model) and 50K Gaussians (1/36 of the original model). The baseline method used a certain distance criterion with agglomerative clustering.

6.1.4 Broadcast news task

The North America broadcast news task was an important large vocabulary speaker-independent ASR task in the speech recognition community [52]. The training set consisted of 23.4 hours of HUB4 96 speech data (F0 condition) and 59.4 hours of HUB4 97 speech data (F0 and F2 conditions), and the standard 96 test data set was used. Decision tree tied CD-HMM with 40 phonemes were used. The vocabulary size was 65K. Trigram language model was trained in the Spoken Language and Information Processing Lab (SLIPL) from the training speech transcripts and the CSR Hub4 language model data

set (from LDC). The perplexity on the 96 test set was 182.8. The pronunciation dictionary was adopted from the CMU lexicon [58], and the CMU letter-to-sound toolkit [59] was used to obtain the pronunciation for the pronunciation unknown words. For acoustic modeling, speech features consisted of 39 components including 13 PLPs and their first and second order time derivatives, where analysis was made at a 10 ms frame rate with a 20 ms window size. Again the phone HMMs each had three emitting states and each state had a GMD of size 16, and crossword triphone models were used. HTK [26] was used for training the acoustic model sets. Evaluation was based on word accuracy.

6.2 Experimental results for the telehealth automatic captioning task

6.2.1 Experimental results for explicit PDT tying

Experiments were conducted on the Telemedicine automatic captioning data to evaluate the performance of the explicit PDT tying method described in Chapter 3. The acoustic models were obtained by implementing the explicit PDT tying together with the HTK toolkit [26].

We first carried out pilot experiments on clustering vowels and consonants, respectively, and the results for the two cases are showing in Table 6.2 and 6.3. The phoneme confusion candidates were adopted from the phonetic questions for building decision tree in HTK [26].

Table 6.2 Word accuracy obtained from EPDT tying 1

Dr.1 Data (2630 words) ¹	Accuracy
Baseline 50*3 trees	78.37%
Clustering (ae, eh, ey)	78.75%
Clustering (aw, ax)	78.67%
Clustering (ax, eh)	78.63%
Clustering (oh, om)	78.82%
Clustering (m, n)	78.48%
Clustering (t, k)	78.39%

Table 6.3 Word accuracy obtained from EPDT tying 2

Dr.1 Data (2630 words)	Accuracy
Clustering friction phones	
Baseline 50*3 trees	78.37%
Clustering (s , sh)	78.10%
Clustering (s , th)	77.72%
Clustering (b , p)	76.39%
Clustering (g , k)	78.37%
Clustering (sh , z)	78.37%
Clustering (f , v)	78.37%

¹ This dataset is a subset of the Dr.1's dataset, where the full set has 3248 words.

From Table 6.2 we can see that the best case of explicitly performing clustering increased word accuracy by up to 0.4% absolute comparing with the baseline. It is interesting that most of the improvements are on vowels and not on consonants. This is also consistent with the findings in [60] where phone substitution modeling was used in continuous speech recognition on TIMIT data. These results lead to the following conclusions:

1. Consonants are not suitable for explicit tying, (This may be due to the wide diversity that the different consonants have), and the confusions introduced by consonant clustering may be more than the benefits from pronunciation variations it solves.

2. Vowels are better choices for explicit tying because vowels are more stable.

3. We also observe that in some clustering cases, word accuracy did not change at all. That happens when the decision tree split the different phoneme data at the top levels, and therefore the training data from different center phones never mixed up and the resulting model is exactly the same as the baseline model. This indicates that although some phones are labeled alike based on the linguistic knowledge source, in conversational speech data they are still quite different.

In Table 6.4 we evaluated the extreme case of explicit PDT tying. We put the entire center phone data in the training set to generate a model with 3 Single-Tree PDTs model (3-tree model) according to 3 emitting states of triphone HMMs. We further separate the consonants and vowels to train a model with 6 single-tree PDTs to avoid data sharing between them.

Table 6.4 Word accuracy obtained from the extreme case of EPDT tying

Dr.2 Data (5085 words)		Dr.1 (3248)	Dr.2 (5085)	Dr.3 (3988)	Dr.4 (2759)	Dr.5 (6421)	Average ²
Baseline 50*3 trees	Accuracy	77.43%	81.26%	82.57%	74.01%	78.71%	79.23%
	Model size	1104	2076	1735	1479	1412	1591
3-Tree Model	Accuracy	75.55%	80.37%	83.95%	73.36%	78.20%	78.76%
	Model size	1077	2045	1717	1461	1386	1566
6-Tree Model	Accuracy	76.57%	81.71%	83.27%	74.63%	78.20%	79.26%
	Model size	1064	2035	1708	1436	1386	1556

It is obvious that in this task, the extreme case in explicit PDT tying did not generate good results in comparison with the 1.8% absolute word accuracy improvement in [41]. This may be due to the fact that our task is speaker dependent, and therefore less pronunciation variations appeared in the speech data.

We also include the baseline and the EPDT model sizes in terms of number of tied states in Table 6.4, where the EPDTs used the same decision tree construction thresholds of likelihood gain and data count as the baseline. The sizes for the two extreme cases of the EPDT models were smaller than the baseline model. This is due to the effect of the increased phoneme data sharing in EPDT. The 6-Tree model had a smaller size than the 3-tree model, although the 3-tree model was supposed to have more data sharing. This

² The average word accuracy was weighted by the word counts for each doctor's data set shown in the first row of Table 4.4.

might be explained by the greedy process of the decision tree construction. In the 3-tree model, we added 48 questions regarding center phones in the question set of HTK. Because the root node of each tree had a large data diversity due to the full set of phonemes, the phonetic questions concerning the center phone properties had better chances to be selected at the top levels of the 3 trees. The result is that some of the phoneme data sharing occurred in the 6-tree model did not happen in the 3-tree model because the phonemes were separated early.

6.2.2 Experimental results for multiple acoustic models based on explicit PDT

We combined the baseline model with the 6-tree model (from the 3 Single-Consonant-Trees plus 3 Single-Vowel-Trees), so that each triphone state was tied to two models. The results for the five doctors are shown in Table 6.5. Here average and max were two strategies of model score combination that are discussed in Chapter 4.

Table 6.5 Word accuracy obtained from combining the baseline model and the 6-Tree models

	Dr.1 (3248)	Dr.2 (5085)	Dr.3 (3988)	Dr.4 (2759)	Dr.5 (6421)	Average
Baseline	77.43%	81.26%	82.57%	74.01%	78.71%	79.23%
2 Models Average	77.56%	81.79%	83.63%	75.39%	79.66%	80.03%
2 Models Max	77.80%	81.95%	83.63%	75.50%	79.69%	80.13%

Comparing with solely using the 6-tree model, this approach gives a better result. The speed of the decoding engine is slightly decreased due to the increase in likelihood score computation time. We next added the 3-Tree model into the combination, and evaluated the performance as shown in Table 6.6,

Table 6.6 Word accuracy obtained from combining the baseline, the 3-Tree, and the 6-Tree models

	Dr.1 (3248)	Dr.2 (5085)	Dr.3 (3988)	Dr.4 (2759)	Dr.5 (6421)	Average
Baseline	77.43%	81.26%	82.57%	74.01%	78.71%	79.23%
3 Models Average	77.92%	82.22%	84.80%	75.75%	79.61%	80.44%
3 Models Max	78.02%	82.40%	84.83%	76.08%	79.69%	80.57%

This time the average accuracy gain is even higher than the 2 model combination results of Table 6.5, and we obtained an absolute word accuracy gains of approximately 1.3% with the MAX combination. It is believed that combining hierarchical tying models that introduce different scales of confusions among phones would benefit system performance, and this was also what we observed here in this experiment.

6.2.3 Experimental results for cross validation data sampling

In this experiment, we applied the Cross Validation (CV) based data sampling method for acoustic modeling, and used the models on the telehealth test set with the Tiger decoding engine.

Table 6.7 Word accuracy obtained from 10-fold cross-validation based ensemble acoustic model

10 CV Model	Dr.1 (3248)	Dr.2 (5085)	Dr.3 (3988)	Dr.4 (2759)	Dr.5 (6421)	Average
Baseline ³	76.69%	81.18%	83.05%	74.48%	78.74%	79.26%
Baseline	77.43%	81.26%	82.57%	74.01%	78.71%	79.23%
Average	79.37%	83.15%	85.26%	76.62%	81.11%	81.52%
Max	79.37%	82.93%	85.32%	76.15%	80.94%	79.67%
n-Best (n=5)	79.34%	83.17%	84.95%	76.44%	81.05%	81.42%

In this experiment we obtained 2.3% absolute word accuracy gain in using the average combining method. This was a significant improvement in the telehealth captioning task. For detailed accounts on the significance test on this task, please see [6].

Several issues should be addressed. First is the baseline model performance. The results for the individual 10 CV acoustic models are obtained from the test set of Dr.2, shown in Table 6.8.

³ This is the baseline used in [6]. The difference in the two baselines may be due to different parameter settings used in the decoding stage.

Table 6.8 Effects of base classifiers on word accuracy

Dr.2's data (5085 words)	Accuracy
Baseline	81.26%
Model 1	80.77%
Model 2	81.00%
Model 3	79.82%
Model 4	80.69%
Model 5	81.40%
Model 6	81.08%
Model 7	80.93%
Model 8	81.04%
Model 9	80.81%
Model 10	80.96%
10 Model Average	80.85%
Standard Deviation of word accuracies from the 10 models	0.004119

Here we observe that the performances of most of the base models were lower than the baseline, which indicated that the training data size and speech sound coverage were the key factors in recognition accuracy. However, ensemble model also benefited from the inter model diversity that sampling the training data has generated. More details regarding the inter model diversity and base model quality will be discussed in Chapter 7.

Table 6.9 Effects of fold size on word accuracy (averaged over 5 doctors).

	1 fold	5 folds	10 folds	20 folds
Word Accuracy	79.24%	80.88%	81.47%	81.36%

We further investigated the relationship between different fold sizes and word accuracy. It was believed that a larger fold size produces weaker diversity and better base model quality, while small fold size has the opposite effect. However, the performance of the base models produced by smaller fold size suffered because of a small amount of training data. This explained why the 5-folds CV ensemble model had the lowest word accuracy. We also investigated the effect of different mixture sizes on word accuracy for Dr.2's data set and the results are shown in Table 6.10.

Table 6.10 Effects of different mixture sizes on word accuracy

Dr.2 's Data (5085 words) combination method: Average	Single model Baseline	10 CV model
8 Mixture Models	79.80%	82.22%
16 Mixture Models	81.26%	83.15%
20 Mixture Models	81.20%	83.37%
24 Mixture Models	80.12%	83.64%

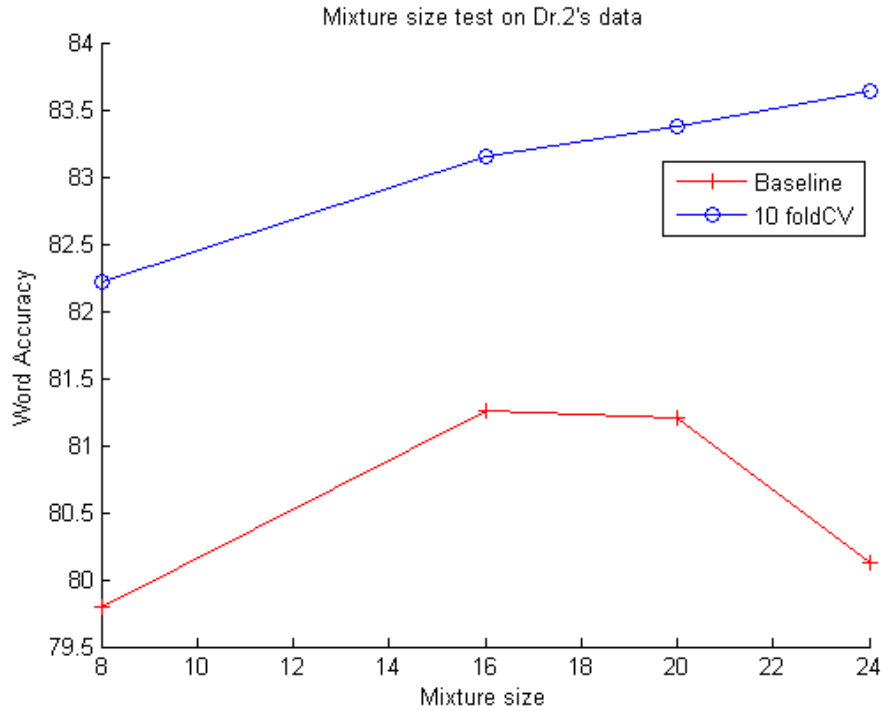


Figure 6.2 Effects of different mixture sizes on word accuracy

Here we can observe that mixture size affects word accuracy differently for the baseline and the ensemble models. For the baseline model, the accuracy reaches the highest level when the mixture size is equal to 16. It was because low mixture model was not accurate while high mixture model requires more data to train. The 10-fold CV ensemble model was observed to be superior to the best baseline model, and it had the property that larger mixture model yields better results. This could be explained by the variance reduction effect of ensemble models that avoids overfitting. In [6], a similar relation of word accuracy perform with mixture size of the GMDs was observed for the EAMs produced by question sampling.

6.2.4 Experimental results for random data sampling

In this experiment we generated multiple training data sets by randomly sampling the training data without replacement, which was described in Chapter 4. Here we obtained 4

ensemble models with different number of base models in a pilot experiment on a subset of Dr.1’s data, where 16 GMDs were used for each base model. The results are shown in Table 6.11.

Table 6.11 Word accuracy of the ensemble models generated by random sampling without replacement

Dr.1 Data (2630 words) ⁴	Average
Baseline	78.37%
10 models	79.06%
20 models	79.55%
30 models	80.08%
50 models	79.89%

We can observe that this method also produced a 1.7% absolute increase in word accuracy over the baseline, when the ensemble size was 30. However, the performance gain was inferior to the proposed CV based sampling.

6.2.5 Experimental results for enhanced training

In this experiment, we study the effect of combining acoustic models with different GMD mixture sizes as well as using EM or CVEM training on the telehealth task. In EM, 2 iterations were used, and in CVEM, the number of folds was set to 10, and 4 iterations of CVEM were used after 2 iterations of EM. The word accuracy results of these different cases are shown in Figure 6.4.

⁴ This dataset is a subset of the Dr.1’s dataset, where the full set has 3248 words.

We first combined three single model sets with mixture sizes of 16, 24 and 32 as the model 3M_MDMS and obtained a word accuracy of 79.7% under EM training. In contrast, the word accuracies for the single model sets with the GMD mixture sizes of 16, 24, and 32 were 79.2%, 77.8%, and 75.1%, respectively (only the baseline with mixture size 16 is shown in Fig.6). Although on the single model sets the accuracy performance degraded with the increase of mixture size, 3M_MDMS had an absolute accuracy gain of 0.5% over the best individual model sets. Under CVEM training, the 3M_MDMS produced an absolute accuracy gain of 1.2% over the baseline.

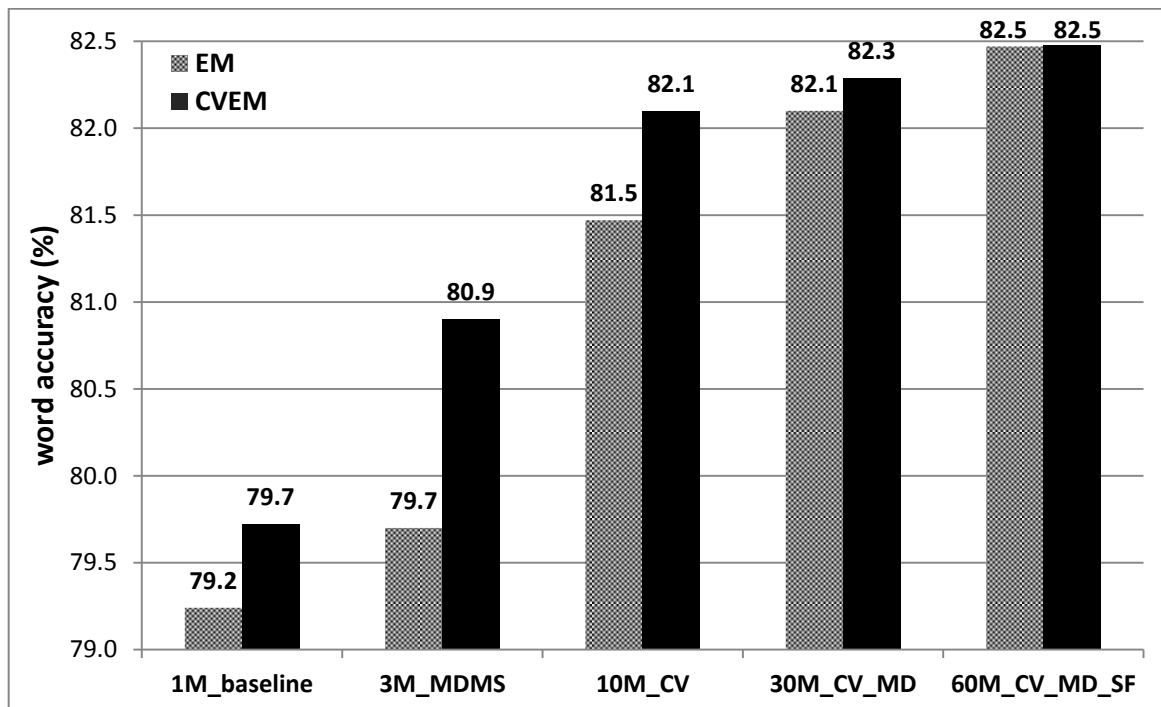


Figure 6.3 Effects of EAMs on the telehealth task (mixture size=16 per GMD)

We next combined three 10-fold CV models with the mixture sizes of 16, 24, and 32, respectively to form an EAM consisting of 30 model sets, referred to as 30M_CV_MD. In addition, we applied the $L=1/2$ shift on the 30M_CV_MD model to obtain an ensemble of 60 model sets consisting of 20 model sets from each of the three

mixture sizes, referred to as 60M_CV_MD_SF. The EM-based 10M_CV ensemble yielded an average word accuracy of 81.5% in contrast to the 79.7% given by CVEM based single model. The ensemble of the 60 model sets with EM training yielded an average word accuracy of 82.5%, which is a 3.2% absolute gain over the baseline. Although CVEM gave better performance than EM in general, on EAMs the superiority of CVEM reduced with the increase of the ensemble size. It is worth noting that the best result reported in [6] for the question-sampling based EAM using 20 model sets with the GMD mixture size of 24 gave an average accuracy of 82%, which is 0.5% lower than the best result we obtained with the data sampling EAM.

In addition a pilot experiment was conducted on discriminative training for Dr.2's data by using the MPE criterion, where the DT training used 1 iterations and the HTK suggested training parameters. The results are summarized in Figure 6.4.

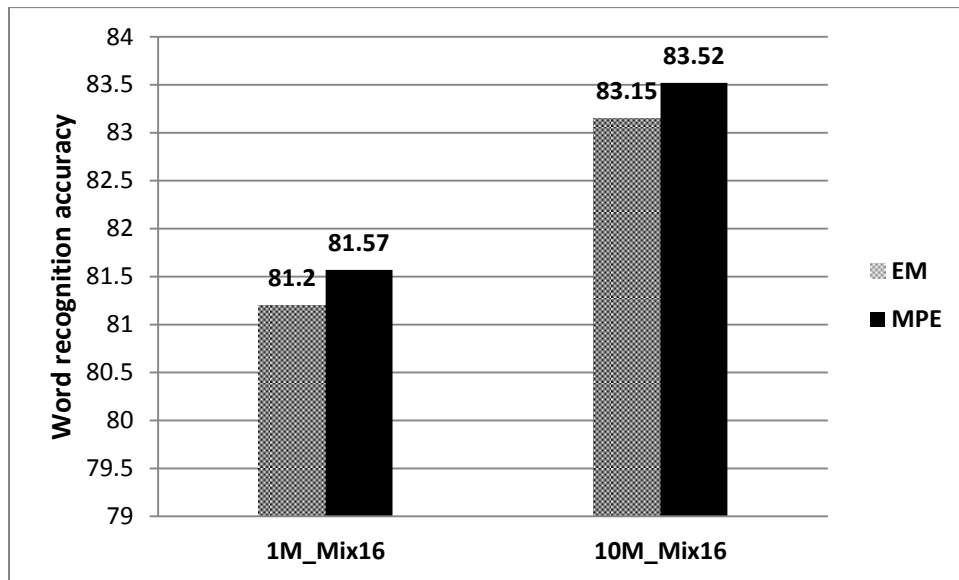


Figure 6.4 Effects of DT training on word accuracy

We can observe that MPE improved the EM trained single acoustic model and EAM.

However, the improvement was small, which may be due to the small amount of available data for discriminative training the relatively large model.

6.3 Experimental results for the TIMIT phone recognition task

For the TIMIT task, the data sampling methods of CV and speaker clustering were both applied. The effects of EM and CVEM were compared, and the MDMS was used. The phone labels provided in the TIMIT dataset were utilized to carry out the experimental evaluations of incorporating MPE training and MLP features into ensemble acoustic models.

6.3.1 CV data sampling based ensemble acoustic models

We first applied a 10-fold CV data sampling to produce an ensemble of 10 sets of acoustic models with the GMD mixture size equal to 16 (10M_EM, mix16), which gave a 1.3% absolute gain in phone accuracy over the baseline (1M_EM, mix16). We next increased the mixture size of the GMDs to 24 and 32, and trained their respective ensemble models (10M, mix24, mix32). Finally, for MDMS modeling, we combined the ensemble models with the two mixture sizes of 16 and 32 (10M, MDMS), and combined the single models with all three mixture sizes (1M, MDMS). For each of these model architectures, we also compared using EM versus using CVEM in model training. For EM, 2 iterations were used, and for CVEM, the number of folds was set to 10, and 8 iterations of CVEM were performed after 2 iterations of EM (This implementation was optimized in a pilot experiment). These results are summarized in Figure 6.5.

For CVEM, The ensemble of 10 model sets with mix32 yielded an average phoneme accuracy of 74.26%, a 2.54% absolute gain over the baseline. For EM, the ensemble of 20 model sets with MDMS yielded an average phoneme accuracy of 73.94%. It is worth noting that the ensemble acoustic models always performed better than the single models trained by either EM or CVEM. For the case of EM with mix24 and mix32, the single model quality decreased due to a lack of data for estimating the increased number of model parameters. However, the EAM model of mix24 and mix32 gave increased accuracy performance, indicating a larger gain from the increased inter-model diversity than the loss from the decreased base model quality.

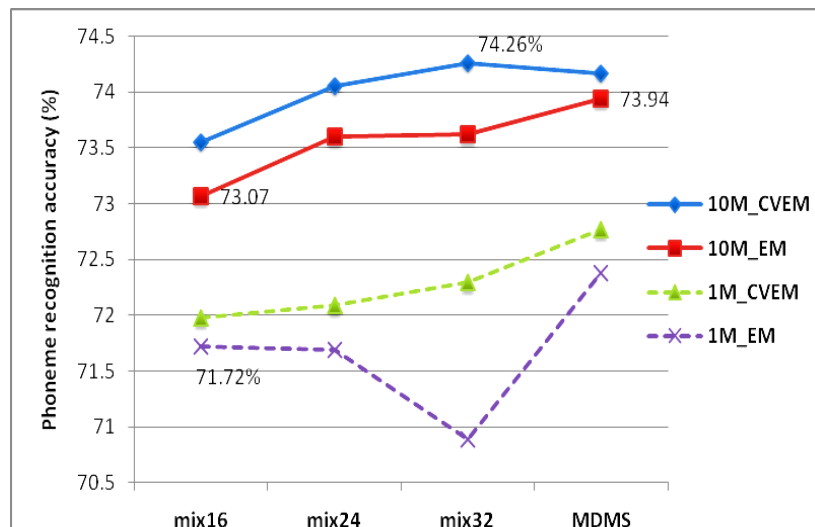


Figure 6.5 Effects of EAM with EM or CVEM training on the TIMIT task

6.3.2 Speaker clustering based ensemble acoustic models

We first evaluated the proposed data sampling method of Likelihood-based Speaker Clustering (LSC) in constructing ensemble acoustic models with the number of folds set to 10. In estimating the core models, 5% of the full set of training sentence utterances was allocated to each fold (in TIMIT this amounts to allocating 1/20 of total speakers to each fold since the speakers have equal number of sentences), the reuse time of each speaker

was empirically limited to 3, and 5 iterations was used in speaker clustering. The GMDs in the core models each had 16 mixture components. The overlapped data sets were generated by using different fractions α of the full training set, with α ranging from 90%, 70%, 50%, and down to 30%. For comparison, random sampling (RS) without replacement was also used for data sampling, where the overlapped sampled datasets were generated with the same fractions of the full training set as in LSC. The models were trained with EM. The phone recognition results are shown in Figure 6.6.

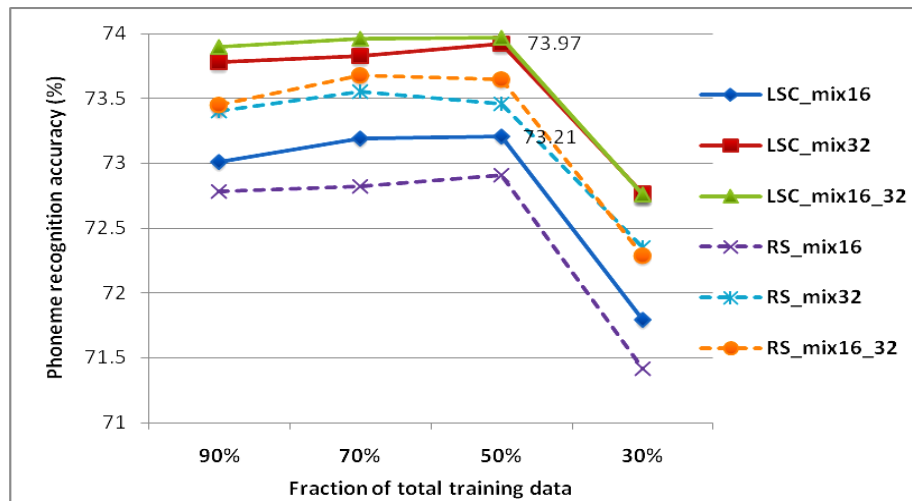


Figure 6.6 Effects of LSC versus RS in training ensemble acoustic models

As shown in Figure 6.6, the proposed LSC based ensemble model outperformed the RS based ensemble model for all four fraction values. Figure 6.6 suggests that the amount of overlap among the sampled training datasets was an important factor determining the quality of the ensemble acoustic models, and $\alpha=50\%$ appears to be the best choice for the TIMIT task. As the amount of overlap among the data sets decreased, each model set became weaker due to the reduced amounts of training data, while the diversity among the model sets increased. It is observed from comparing the LSC results in Figure 6.6 with the CV results in Figure 6.5 that at $\alpha=90\%$, LSC and CV based sampling gave

similar accuracy performance, since the heavy data overlap decreased the effect of speaker clustering on the diversity among the sampled datasets. At the GMD mixture size equal to 16, using LSC with $\alpha=50\%$ gave a 0.14% absolute accuracy gain over using the 10-fold CV, while the size of the models in the LSC ensemble is only about half the size of the models in the 10-fold CV ensemble, since in 10-fold CV, each model set was trained from 90% of total training data.

We next evaluated the proposed data sampling method of speaker-model Distance based Speaker Clustering (DSC). We used the two SA sentences from each speaker in the TIMIT training set to train a GMD speaker model with 16 mixture components, and used Monte Carlo sampling with 20K samples to compute the KL divergence between each pair of speaker models. The speaker clusters were selected from 10000 runs of random initialization and K-medoids clustering. The number of speaker clusters or folds was set to 10. For DSC, the overlapped data sets used the same fractions of the full training set as in LSC, ranging from 90%, 70%, 50%, and down to 30%. The ensemble acoustic models were trained by EM, and the phone recognition accuracy results are shown in Figure 6.7, where the LSC data sampling was included as a comparing case.

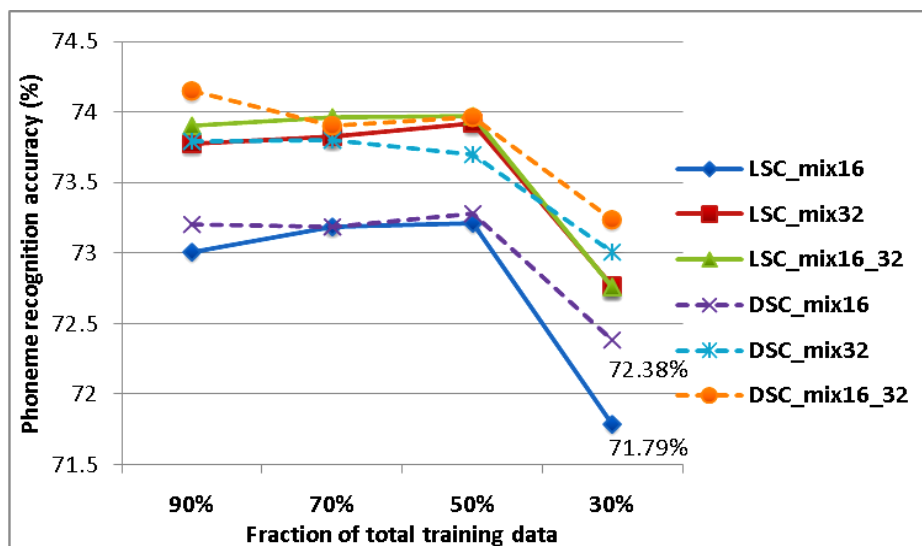


Figure 6.7 Effects of LSC and DSC in training ensemble acoustic models

For most cases shown in Figure 6.7, the DSC based ensemble models outperformed the LSC based ensemble models. At the fraction of 90%, 70%, and 50%, the performance difference between DSC and LSC ensembles was small, since the amounts of data overlap may have masked their differences. However, at the fraction of 30%, DSC outperformed LSC by 0.61% in phoneme recognition accuracy. This suggests that DSC worked better than LSC in the case of small data overlap, because in DSC, speaker models were explicitly used to generate the sampled data sets and the clustering results were selected from multiple configurations. For the limited training data of TIMIT, the acoustic models trained from the sampled datasets with small data-overlaps suffered from poor performance since each base model was weak due to the lack of training data. But having small data-overlaps among the sampled datasets improved the inter-model diversity which again produced a larger gain for the EAM quality than the loss caused by the decreased base model quality. If a very large dataset was used in acoustic model training, then the base models trained from sampled datasets with small overlaps may still be strong and more improvements to the EAM quality is expected. Since at the fraction of

$\alpha=50\%$ or larger the difference between the two speaker clustering data sampling methods was small, and the likelihood-based speaker clustering was faster in computation, our subsequent evaluation experiments used mainly CV and LSC data samplings.

6.3.3 Integration of discriminative training and MLP feature in ensemble acoustic models

Discriminative training of MPE on the base models was performed by using HTK [26], where the HTK suggested parameter values were used, and the total iterations was empirically set to 8. The MLPs were trained by using the NICO toolkit [61]. The input features to the MLP were the same 39-D MFCC based features used in the baseline HMM training. We used an MLP with 351 input nodes (8 symmetric context frames plus the current frame for 39 feature components per frame, i.e., $9*39$), 2 hidden layers having 200 nodes in each layer with time delayed recurrent connections, and 39 output nodes corresponding to 39 phonemes. The frame classification accuracy on the test set was 73.10%. For each speech frame, the posterior probabilities of 39 phonemes were obtained from the MLP output, and PCA was used to reduce the MLP feature dimension from 39 to 15 (empirically set). The PCA-reduced 15 dimensional MLP features were then concatenated with the original 39-D feature of MFCCs to form a 54 dimensional feature vector, referred to as 54D, for each speech frame. The phoneme recognition results with DT and MLP features are shown in Figure 6.8 for the cases of single model, the 10M_CV model, and the 10M_LSC model, all with mixture size of 16 for the GMDs, and the initial models of DT were trained by EM.

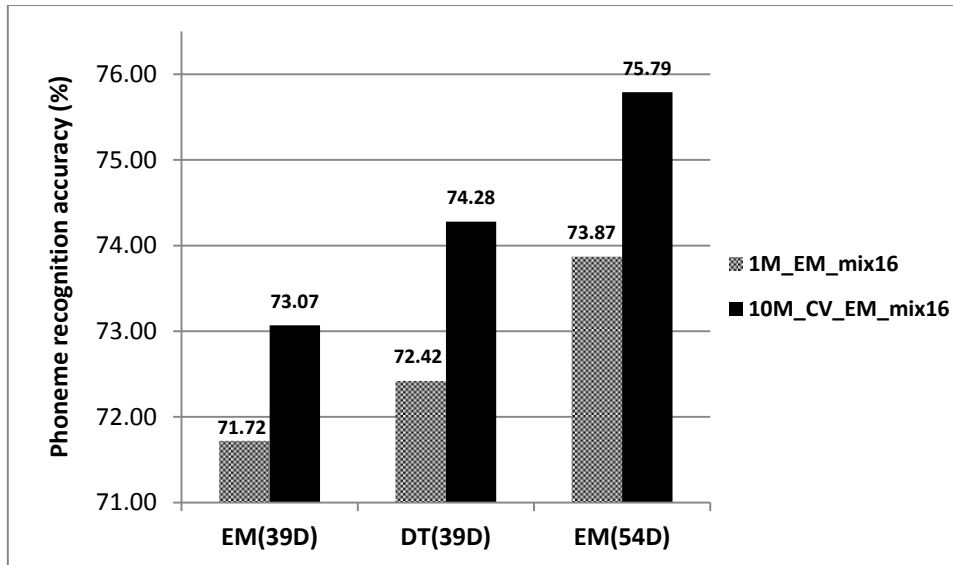


Figure 6.8 Effects of MPE and MLP features on ensemble acoustic models

Using MPE for single model training increased the phone accuracy by 0.70% over the baseline of EM training. For the case of CV data sampling with EM training, the accuracy gained by using the 10M ensemble over the single model was 1.35%, while using the DT trained 10M ensemble produced an accuracy gain of 1.86% over the DT single model. This result shows that our ensemble models have amplified the positive effect of DT training. Conversely, DT training has increased the contribution of model diversity to the EAM quality.

Using the 54D feature of MFCC+MLP in single model increased the phone accuracy by 2.15% over the baseline of MFCC feature. For the case of CV data sampling and the MFCC features, the accuracy gained by using the 10M ensemble over the single model was 1.35%, while using the MFCC+MLP features based 10M ensemble produced an accuracy gain of 1.92% over the MFCC+MLP single model, and the gain was 4.07% over the MFCC based single model. This result shows that our ensemble models has amplified the positive effect of MFCC+MLP features. Conversely, the MFCC+MLP

features have enabled the larger performance gain for the ensemble model over the single model by increasing the model diversity.

In integrating both DT and MLP features in the ensemble acoustic models, we further generated the proposed ensemble MLP features by using 10-fold CV data sampling to train an ensemble of 10 MLPs. PCA was applied to the posterior probabilities that were averaged over the base MLPs to reduce the MLP feature dimension from 39 to 15, which were again concatenated to the MFCCs to form a 54 dimensional feature vector, referred to as 54E.

In Table 6.12, we compared the effects of using the MFCC features (39D), the MFCC+MLP features from single MLP (54D), and MFCC+MLP features from ensemble MLP (54E), for the case of single model (1M), 10 model ensemble with EM training (10M_EM), 10 model ensemble with CVEM training (10M_CVEM), and 10-model ensemble with CVEM initialized MPE training (10M_CVEM_MPE), where the 10-fold CV data sampling was used to generate all the ensemble models.

Table 6.12 Effects of MLP and ensemble MLP features in ensemble acoustic models

	1M	10M_EM	10M_CVEM	10M_CVEM_MPE
54E_mix16	74.43%	76.16%	76.43%	76.95%
54E_mix16_32	74.58%	76.35%	76.67%	77.10%
54D_mix16	73.87%	75.79%	76.09%	76.58%
54D_mix16_32	74.24%	76.23%	76.28%	76.69%
39D_mix16	71.72%	73.07%	73.55%	74.27%
39D_mix16_32	72.38%	73.94%	74.17%	74.89%

The 54D features outperformed the 39D feature representation consistently, and the 54E features gave a 0.3%~0.6% absolute accuracy gain over the 54D features. The

ensemble of 20 model sets with the 54E features using mix16_32 that were obtained from CVEM initialized MPE training yielded a phoneme accuracy of 77.10%, which improved the baseline accuracy by 5.38% absolute, and in comparison with using the 54D features, the improvement was 0.41% absolute. The best result came from 50% LSC data sampling with 54E_mix16_32 and 10M_CVEM_MPE not shown in the table, which yielded a phoneme accuracy of 77.32%.

In Figure 6.9, we compare the accuracy performance between the ensemble acoustic models with 10 model sets obtained from CV data sampling and LSC data sampling, for the case of using 39D and 54D features and the model training methods of EM, CVEM, and CVEM+MPE. Each sampled data set in LSC data sampling had only 50% of the total training data, and that in CV data sampling had 90% of the total training data. Again, each model set trained from LSC was about half the size as that in CV. However, the stronger inter-model diversity due to the smaller amount of data overlap in LSC improved the performance of its ensemble acoustic models. It is seen in Figure 6.9 that out of the five cases, LSC outperformed CV in four cases, and only in the case of 54D_CVEM CV was slightly better than LSC.

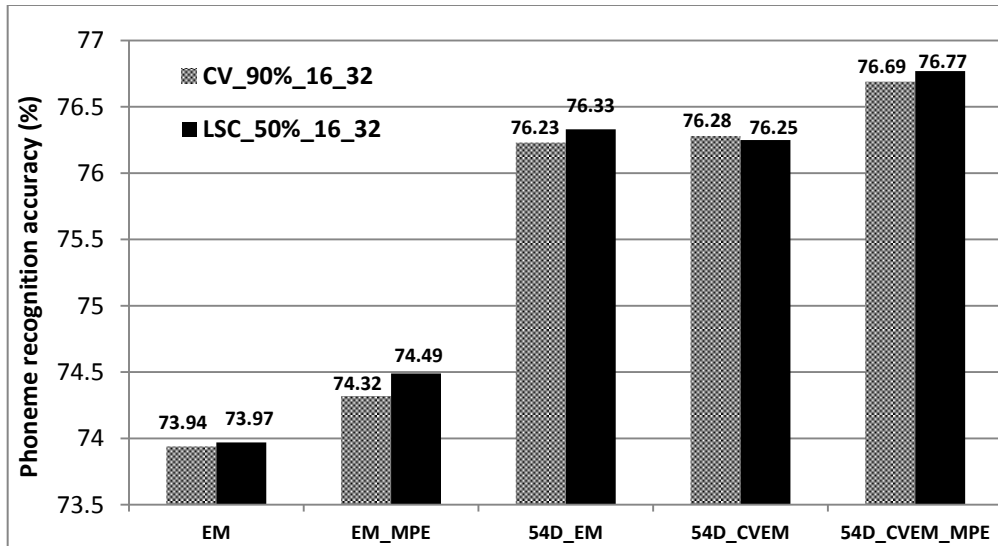


Figure 6.9 Phone accuracies of LSC and CV data sampled ensemble models

6.3.4 Data sampling vs. question sampling

It is of interest to evaluate the performance of the question sampling based EAM of [6] and the data sampling based EAM of this current work on a common task. We present the comparative experiment results in Table 6.13 for the TIMIT task. In Table 6.13, in addition to using the standard MFCC features of 39D_mix16, we included the ensemble MLP feature 54E and the MDMS mix16_32 in the comparison. The total number of phonetic questions was 264.

Table 6.13 Comparison on TIMIT phone recognition accuracies of question sampling and data sampling based EAMs.

	RS_QS_150 (57%)	DS_LSC_50%	RS_QS_90%	DS_CV_90%
39D_mix16	73.09%	73.21%	72.54%	73.07%
54E_mix16	76.24%	76.48%	75.63%	76.16%
39D_mix16_32	74.00%	73.97%	73.30%	73.94%
54E_mix16_32	76.46%	76.78%	75.87%	76.35%

When fixing the ensemble EAM to have 10 base models and the sampling rate to be 90% for question sampling (RS_QS_90%) and for data sampling (DS_CV_90%), the CV data sampling EAM gave better phone accuracy performance. We then randomly sampled 150 questions out of the 264 questions as suggested in [6] (RS_QS_57% sampling rate), which gave improved accuracy performance than the 90% question sampling. The data sampling method of 50% LSC gave slightly better result than the 57% question sampling except for the case of 39D_mix16_32, and the size of the models from the 50% LSC was only half the size of the models of the question sampling method.

6.3.5 EAM clustering

It is of interest to evaluate the accuracy performance of the clustered EAM and its comparison with the original or the source EAM. Here, we present the experiment results on the source EAM generated with 10-fold CV data sampling plus DT training and MLP feature. We set the target model size to 0.1, 0.2, 0.4, 0.8 times of its source EAM and used baseline agglomerative method with entropy criteria to perform the clustering. The results are shown in Figure 6.10.

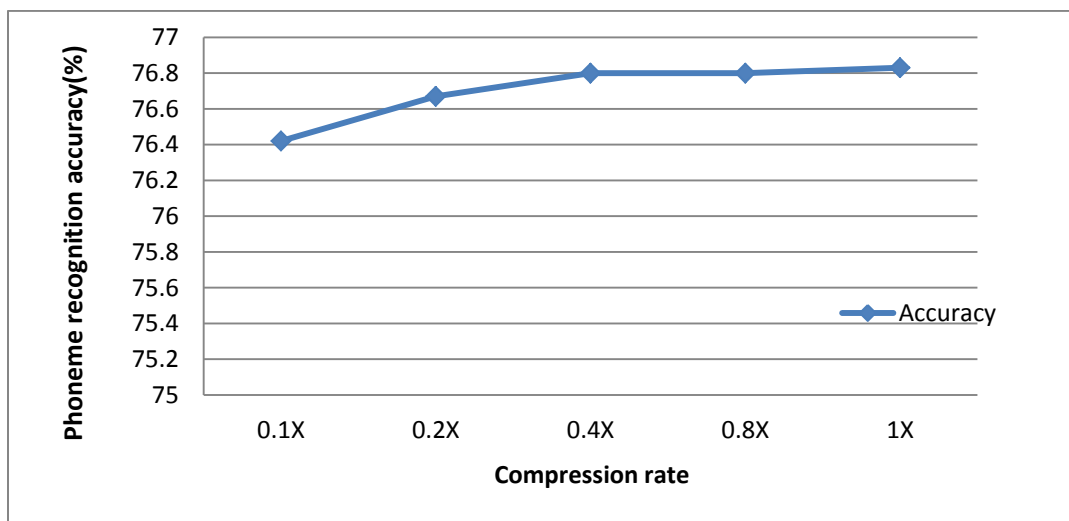


Figure 6.10 Effects of EAM clustering with different compression rates

From Figure 6.10, we see that reducing the model size decreased the accuracy performance. However, the accuracy drop was only 0.4% even if the model size was 10 times smaller than the source EAM. By increasing the model size, the accuracy performance gets better and eventually equal to that of the source EAM. This trend indicates that some information were lost when doing clustering. From this experiment, we show that clustering the data sampling based EAM allowed us to archive a similar decoding speed as in single models while giving a much higher accuracy performance than single models, and this method is therefore of practical value in limited resource environments such as embedded system. It is noted that in [6], clustering of random forest based EAM also showed a similar trend.

6.4 Experiments on the Pashto ASR task

Extensive EAM clustering experiments with several distance measures and several proposed clustering algorithms were carried out on the Pashto ASR task.

6.4.1 Experimental results of NBR

By normalizing the time cost for computing ENT as 1.0X, the time used for KL and Chernoff distance is presented in Table 6.14. It is noted that after using NBR the speed improvement is significant.

Table 6.14 Evaluation on speed improvement using NBR

	KL	Chernoff
Baseline	6.5X	24.4X
NBR	1.2X	1.9X

Figure 6.11 showed that the NBR has improved the baseline in terms of Word Error Rate (WER) too. It also outperformed the entropy distance produced compact model. Figure 6.12 provides results evaluated on the Bhattacharyya, and it showed that using weighted distance can improve the quality of clustering over non-weighted distance, especially when the compression rate is high. Moreover, implicitly imposing mixture weights from the NBR approach outperformed the weighted distance.

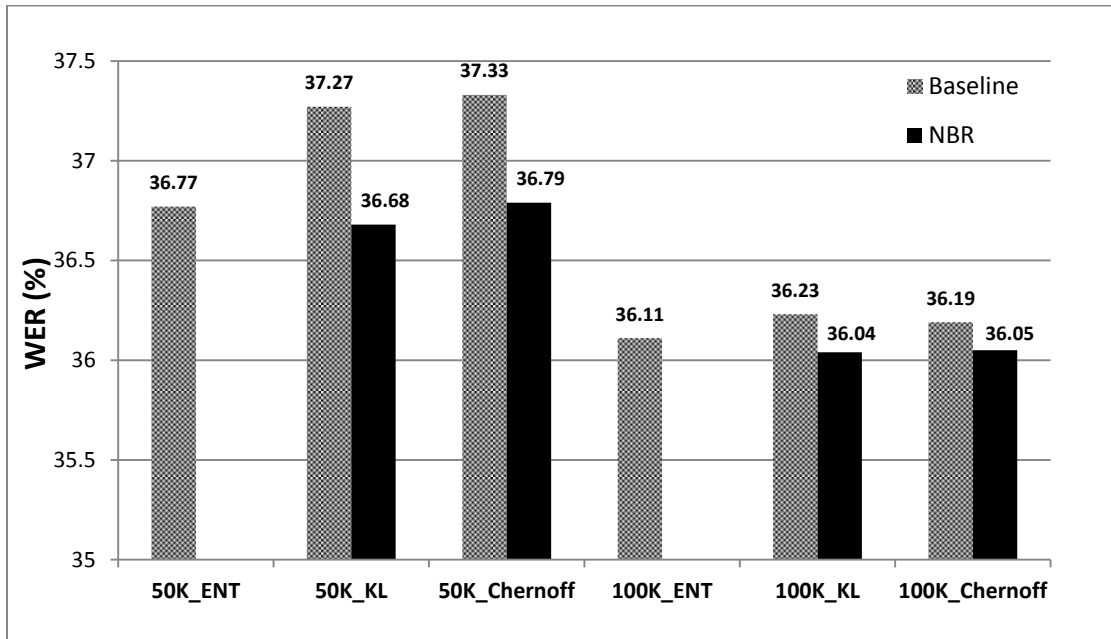


Figure 6.11 Effects of NBR method on WER

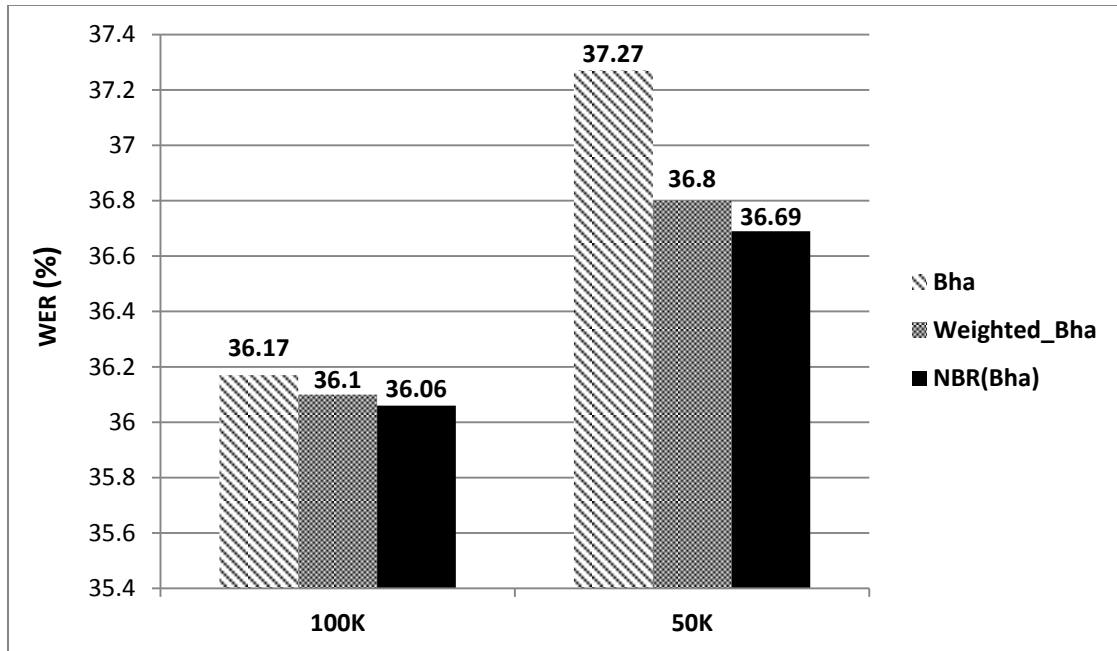


Figure 6.12 Effects of weighted Bhattacharyya criterion and NBR on WER

6.4.2 Experimental results of global optimization

We used Two-Step-Lookahead (2S-LA) in the global optimization experiment. The beam width was empirically set for BFS, where over the total M-N layers, the first 1/3 layers used beam width of 2, the middle 1/3 layers used beam width of 4, the last 1/3 layers used beam width of 8.

Table 6.15 WERs (%) from using the proposed global clustering algorithm

	Baseline	2S-LA	BFS
100K KL	36.23	36.11	36.14
100K ENT	36.11	36.08	36.08
50K ENT	36.77	36.81	36.60

As showed in Table 6.15, the global optimization solutions outperformed the local optimization except for the 50K ENT task with the 2S-LA method. We also measured

the global distance change to evaluate the performance of the proposed methods. The entropy change criterion was used. The greedy approach on state 0 had an overall entropy change of 3336.80. The 2S-LA had a very small improvement of 0.03 comparing with the greedy approach. The proposed BFS approach had an overall entropy change of 3299.04. This was a relatively large improvement over the 2S-LA approach as well as the local optimization methods, which showed that the proposed BFS algorithm was effective in approximating the globally optimized clustering solution.

6.4.3 Experimental results on two-pass model structure refinement

From Table 6.16, we can see that the case of 100K with two-pass model structure refinement produced improvement over the conventional one-pass 100K model, showing that the proposed methods were able to improve the model structure and had a positive effect on acoustic model quality. In the two-pass 100K experiment, The NBR method for the Chernoff distance generated model yielded 35.98% in WER, which was the best result obtained for the 1/18 compression rate performance.

Table 6.16 WER results (%) on two-pass model structure refinement

100K test	Baseline	NBR
Two-Pass KL	36.18	36.02
Two-Pass ENT	36.04	–
Two-Pass Che	–	35.98

6.5 Experiment results on the broadcast news ASR task

The Broadcast News ASR task (BN), is a much larger task than TIMIT and telehealth tasks, and therefore much longer time were taken for training and evaluation with limited computing resources. In Figure 6.13, we present some of the 96 test set's pilot results with acoustic models trained by 96 training data.

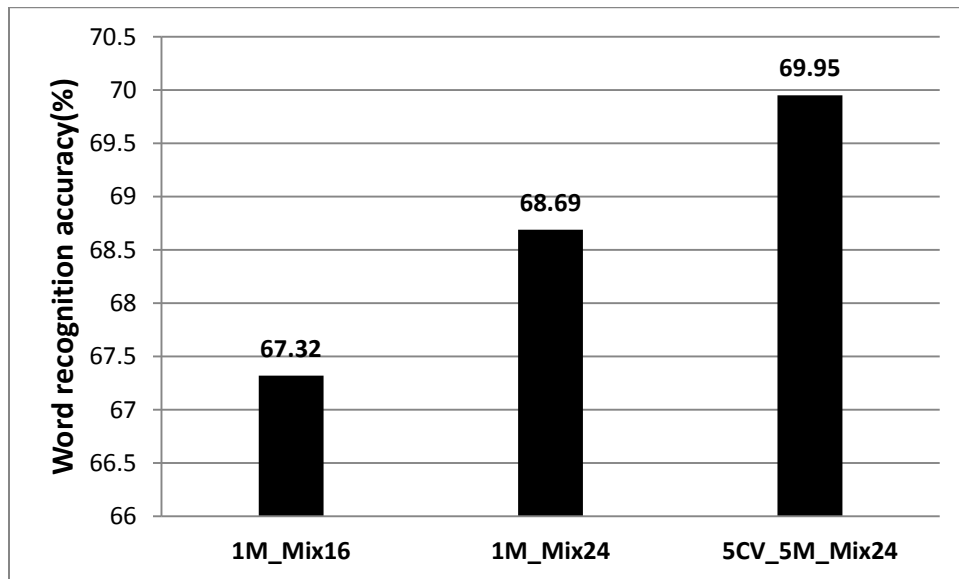


Figure 6.13 Accuracy performances of single model and EAM on BN task

The accuracy performance of the baseline model was 67.32%. Although we had a large amount of data, the data was unbalanced, with high appearance rates in some phones and low appearance rates in filled pauses. To better handle this problem of unbalanced data, we set different numbers of Gaussian densities in the Gaussian mixture models for different phoneme units based on the phone appearance rates in training data. Since most of the phone states used GMDs with 24 Gaussians, we refer to this model as Mix24. Mix24 had a 1.37% improvement over the Mix16 baseline. The EAM model using 5 fold CV data sampling improved another 1.26% over the Mix24 baseline.

Chapter 7

Base Model Quality, Inter-Model Diversity, and EAM quality

The quality of a EAM is contributed from both base model quality and inter-model diversity. For data sampling based EAM, since each acoustic model in an EAM is generated from a sampled data set and diversity can be generated by the difference in data, it is of interest to design some methods to explicitly measure inter-model diversity. [62] In this chapter, we propose several methods to explicitly measure inter-model diversity. It is also unknown how base model quality affects inter-model diversity and EAM quality. Therefore we will discuss some general ways for improving the base model quality by using discriminative training and MLP feature.

7.1 Explicit measures of inter-model diversity

We can implicitly measure the contribution of inter-model diversity to EAM quality by measuring recognition accuracy gain contributed by improved base model quality to EAM quality. To explicitly measure inter-model diversity, we propose three methods: standard deviation, classification agreement, and KL distance.

1. Standard deviation

The standard deviation measure is computed from the correct frame scores along the Viterbi decoded best paths for each tied triphone class to measure the score variations over different base models. The higher the variation is, the larger the diversity between the models.

2. Classification agreement

The classification agreement measure is derived from the base models' classification on each phoneme segment. For each phoneme segment obtained with the Viterbi forced alignment, if two base models produce an identical phoneme label, then the agreement count is incremented by 1. The classification agreement is the agreement count normalized by the total number of phoneme segments. The lower the classification agreement it is, the larger the diversity between the models.

3. KL distance.

As in the speaker distance based speaker clustering, the Monte Carlo sampling method [46] was used to approximate the KL distance between two Gaussian mixture densities within each tying group. Figure 7.1 shows an example:

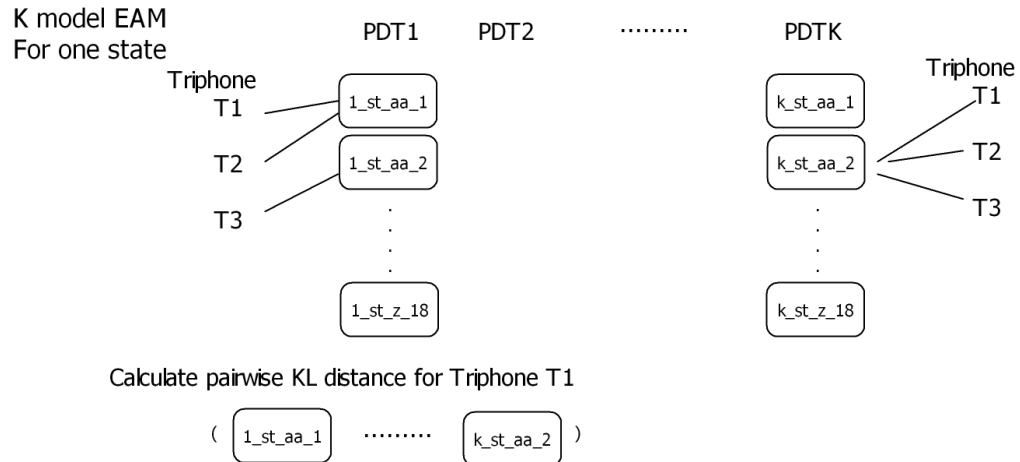


Figure 7.1 Illustration of computing KL distance evaluation as the diversity measure for
EAM

In order to reduce computation load, we selected 1000 most frequent triphones occurred in testing data, For each triphone in the list, we computed the KL distance between two GMDs in the corresponding HMM states of the triphone in each pair of acoustic model

sets, and the distances were averaged over the states and the 1000 triphones to represent the distance between the two model sets.

To measure the EAM quality in addition to recognition accuracy performance, we propose two methods: classification margin, and averaged correct score per frame.

4. Classification margin:

The classification margin is defined as the difference between the log likelihood score of the correct classification and that of its closest competitor, i.e.,

$$d(x_{i,k}^t) = \log p(x_{i,k}^t | \lambda_{C_{i,k}}) - \max_{C_j \in \Omega, C_j \neq \text{Phone}(C_i)} \log p(x_{i,k}^t | \lambda_{C_{j,k}})$$

where $x_{i,k}^t$ represents a speech frame at time t of the class $C_{i,k}$ corresponding to triphone i and state k , $\lambda_{C_{j,k}}$ denotes the HMM for the class of triphone j and state k , $\text{Phone}(C_j)$ defines the subset of triphone states that share the same center phone, and Ω defines all triphones. Since our focus is on the quality of the acoustic models, the prior $P(C_i)$'s are assumed uniform.

5. Averaged correct score per frame

While the margin measure characterizes model quality for class discrimination, we used the averaged correct score per frame to measure the model-data fit for EAM.

7.2 Experimental results on base-model quality, inter-model quality, and EAM quality

In this section, we analyze the effects of the proposed data sampling and the enhanced trainings and features on ensemble model quality and its two contributing factors of base model quality and inter-model diversity.

The measures of EAM quality of classification margin and averaged correct score per frame were computed separately on the train and test sets of the speaker Dr.2 from the telehealth task, and the results are given in Table 7.1. Note that here only the model 1M_16_CVEM used CVEM in parameter estimation, and the 5 other models all used EM. It is observed that our proposed ensemble acoustic models improved the model quality in all of the three aspects, i.e. increased classification margin, increased average correct score, and decreased variance of the correct scores.

Table 7.1 Acoustic model quality measured on Dr.2’s data set

Dr.2’s data	Margin		Average Score		Standard Deviation	
	train	test	train	test	train	Test
1M_32mix	-2.12	-5.34	-102.44	-105.20	21.02	21.10
1M_16_EM	-0.87	-4.59	-102.65	-103.64	20.97	20.84
1M_16_CVEM	-0.86	-4.58	-102.25	-103.24	21.16	20.69
3M_HIE	-1.77	-4.09	-100.90	-102.74	20.50	20.51
10M_CV	1.17	-3.28	-98.71	-100.40	20.06	19.99
30M_CV_HIE	1.14	-2.60	-96.05	-99.35	19.56	19.66

In Figure 7.2 we show the correct and the competing acoustic scores for one test sentence. It is clear that from the frame 220 to the frame 280 the baseline had large negative margins while the margins were small for the 10CV model. This difference explains the fact that the 10CV model ensemble produced the correct hypothesis “prevent”, while the baseline model gave the error hypothesis “prun vent”.

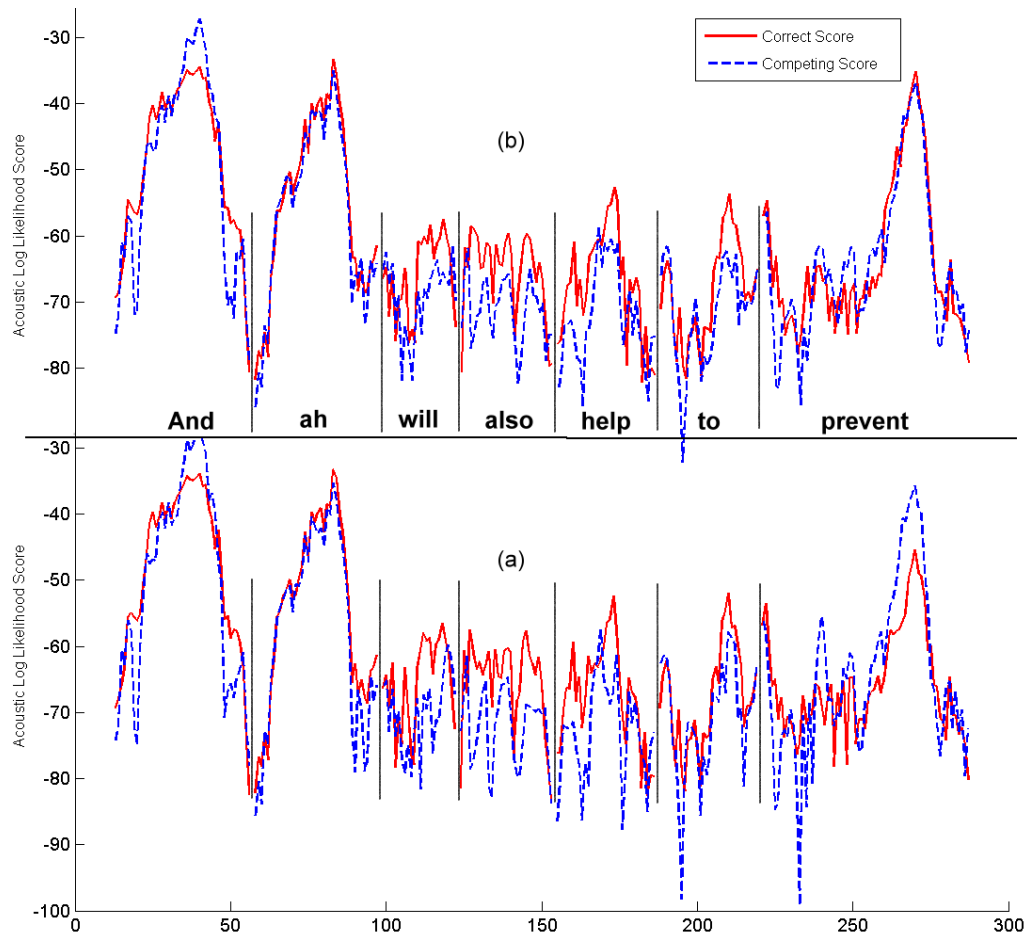


Figure 7.2 Correct and competing scores from one test sentence (a) baseline model
 (b) 10-fold CV ensemble model

We next used three methods to explicitly measure the inter-model diversity within an EAM, including standard deviation, classification agreement, and KL distance, as discussed in Section 7.1. In Table 7.2, we include the evaluation results for the TIMIT task the EAM model quality, the average base model quality, where the quality was measured by phone recognition accuracy and the three diversity measures for five ensemble acoustic models.

Table 7.2 Ensemble model quality, base model quality and explicit measures on inter-model diversity

	EAM	Base	Stddev	Agreement	KL_dist
39D_CV	73.07%	71.52%	1.56	0.846	18.035
54D_CV	75.79%	73.72%	2.07	0.864	33.939
54D_CV_CVEM_MPE	76.58%	74.24%	2.41	0.836	42.046
54D_LSC_50%	76.09%	72.99%	2.62	0.826	51.446
54D_LSC_50%_CVEM_MPE	76.77%	73.36%	2.85	0.805	55.191

In terms of the phone accuracy averaged over the 10 base models, the CV ensemble was better than the LSC ensemble, which makes sense as the LSC used 50% training data for each base model whereas the CV used 90%, and therefore the base model quality of LSC was inferior to CV. However, the phone accuracy or quality of LSC EAM quality was better than the CV EAM, which implies that the inter-model diversity increased faster than the decrease of base model quality in LSC. With the 54D feature, the enhanced trainings of MPE with CVEM initialization improved the quality of base models of CV and LSC_50% by 0.52% and 0.37%, respectively, while the corresponding improvements in EAM quality were 0.79% and 0.68%, respectively. The fact that with the enhanced training methods, the quality of the EAM improved faster than the base models suggests that the large improvements come from the increase in inter-model diversity, and the contribution of inter-model diversity can be viewed as the difference in the accuracy of EAM and the averaged accuracy of base models. Interestingly, for the 39D_CV case, the diversity contribution was 1.57%, and for the 54D_LSC_CVEM_MPE, the diversity

contribution increased to 3.41%. Our three diversity measures (columns 4-6) have shown clearly the same trend, i.e., the increase in EAM quality is correlated with the increase in standard deviation (Stddev) and KL distance as well as with the decrease in classification agreement.(The only exception is the Agreement measure when changing from 39D_CV to 54D_CV. We hypothesize the outlier might be caused by the hard labeling of speech segments.) This confirms our assumption of the important role of inter-model diversity on the quality of EAM and the diversity boosting effect of the enhanced training and feature methods of DT/CVEM/MLP.

Table 7.3 TIMIT phone recognition accuracy of the 10-fold CV data sampling EAM (mix16) with the enhanced training methods and features

	EM (Baseline)	CVEM	MPE	MLP (54D)	MLP (54E)	54D +CVEM+MPE
Single model	71.72	71.98 (0.27%)	72.42 (0.70%)	73.87 (2.15%)	74.43 (2.71%)	74.61 (2.89%)
Ensemble model	73.07 (1.35%)	73.55 (1.57%)	74.28 (1.86%)	75.79 (1.92%)	76.16 (1.73%)	76.69 (2.08%)
Overall gain	1.35%	1.84%	2.56%	4.07%	4.44%	4.97%

In Table 7.3 we summarize for the TIMIT task the accuracy performances of the single and the 10M_CV ensemble acoustic models with CVEM, DT, MFCC+MLP features (54D), MFCC+MLP ensemble features (54E), and the integration of CVEM+DT training with the 54D feature to illustrate the contributions of these methods to phoneme recognition accuracy, with the GMD mixture size fixed at 16. In the table, “Single model”

means using the six methods defined in the first row for training conventional single acoustic models, and “Ensemble model” means using the six methods in the first row for training 10M_CV models. The step by step accuracy gains are also shown in each cell, where in the second row the accuracy gain in each cell was measured against the EM baseline, in the third row the accuracy gain in a cell was measured against the cell in the second row of the same column, and in the last row the accuracy gain in a cell is the sum of the gains in the previous two rows of the same column, i.e., the total gain against the EM baseline. As indicated in the table. The accuracy gain of the EAM trained by EM over its single model was 1.35%, while for the other methods alone and combined, the gains over their respective single models were all larger, indicating that using CVEM, DT, and MLP features increased the inter-model diversity.

In Figure 7.3 we present pilot experimental results of DT on base model and EAM quality for the BN task. The same 5-fold CV data sampling based EAM was used (this model is discussed in Section 6.5). The DT training utterances were based on 97 dataset. The numerator and denominator lattices were generated with the baseline model trained with the 97 data (we did this pilot experiment in this way because the lattice generation was very computing intensive). The MMI criterion was used for DT training and 2 iterations of DT training were performed. The random sampled data sets for DT training ranged from 90%, 70%, down to 50% of total amount of data. The word accuracy was evaluated on the 96 test set. We can observe that with the decrease of the amount of data for training each base model, the base model quality slightly decreased, indicating that the quality of the base models might have suffered from the reduced amount of data, while inter-model diversity was increasing as indicated by the improved EAM quality. As

the result, the word accuracy performance of the EAM at 50% amount of training data for the base models was 0.74% higher than that of the EAM at 90% amount of training data for the base models.

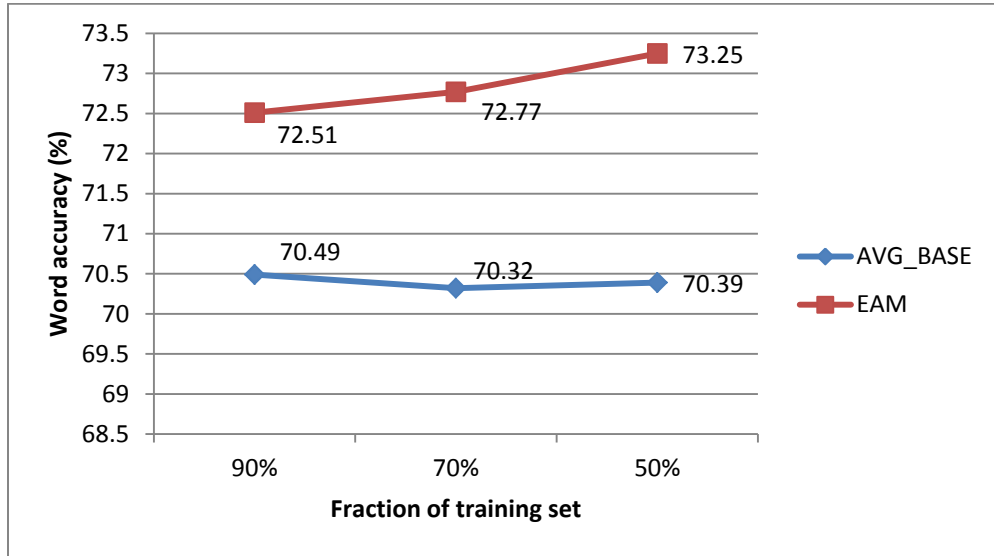


Figure 7.3 Effects of DT on base model and EAM quality with different fractions of training data

Chapter 8

Conclusion and Future Extension

In this dissertation, several ensemble acoustic modeling methods have been proposed and investigated for automatic speech recognition tasks. Inter-model diversity which plays a very important role in EAM quality is measured and studied. In addition, GMM clustering approach for model compaction is thoroughly studied, including several distance measures and several proposed clustering algorithms. In addition, an EPDT tying method is proposed to improve the PDT tying process in the single acoustic model framework. The main contributions of the current work include the following several aspects.

1. Data sampling methods for constructing ensemble acoustic models — a) a Cross Validation based data sampling method is proposed, which significantly improved the HMM baseline models on the tasks of telehealth automatic captioning and TIMIT phone recognition; b) Speaker clustering based data sampling methods are proposed and evaluated on the TIMIT phone recognition task, and these methods improved the EAM quality through increasing inter-model diversity.

2. DT and CVEM based enhanced training and MLP features are integrated with EAM in order to improve the base-model quality. We discovered that these traditional methods developed for single model training help improve inter-model diversity and therefore further improve the quality of EAM. We investigated the previously largely unstudied inter-model diversity and are able to show that inter-model diversity plays an important role in EAM quality.

3. Several similarity measures are investigated and several global clustering optimization algorithms are proposed in order to compact an EAM while maintain the EAM quality at the same time, these methods was evaluated in a Pashto ASR system and it showed improved accuracy performance over the traditional method of agglomerative clustering.

4. Explicit Decision Tree tying — by clustering center phone training data based on linguistic knowledge, we have obtained improved word accuracy in some cases. We further combined the extreme case of explicit decision tree models with the baseline model and the word accuracy has been improved notably.

Ensemble acoustic modeling is a very promising direction in ASR. Since inter-model diversity is a very important factor determining the EAM quality, it is of interest to improve it in two directions. The first direction is to improve inter-model diversity when building the EAM, for example the method we have already investigated – speaker clustering based data sampling, and boosting models at the decision tree construction step can be further investigated along this line. Another way is to build many models by random sampling on data or decision tree questions and using some criteria to select models that maximize the diversity. Along the model selection direction, it is of interest to perform model selection at model level, state level, and Gaussian density level.

References

- [1] J. G. Fiscus, “A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER)”, Proceeding of IEEE ASRU Workshop, pp. 347-352, 1997.
- [2] O. Siohan, B. Ramabhadran, and B. Kingsbury, “Constructing ensembles of ASR systems using randomized decision trees,” Proceeding of ICASSP, pp. I-197-I-200, 2005.
- [3] R. Zhang, Z. A. Bawab, A. Chan, A. Chotimongkol, D. Huggins-Daines, and A. I. Rudnicky, “Investigations on ensemble based semi-supervised acoustic model training,” Proceeding of Eurospeech, pp. 1677-1680, 2005.
- [4] T. G. Dietterich, “Ensemble methods in machine learning,” Proceeding of MCS, pp. 1-15, 2000.
- [5] L. Breiman, “Random forests,” Mach. Learn., vol. 45, pp.5-32, 2001.
- [6] J. Xue and Y. Zhao, “Random forests of phonetic decision trees for acoustic modeling in conversational speech recognition,” IEEE Trans. ASLP, vol.16, no. 3, pp. 519-528, 2008.
- [7] B.-H. Juang and S. Katagiri, “Discriminative learning for minimum error classification,” IEEE Trans. SP vol.40, 3043–3054, 1992.
- [8] L. R. Bahi, P.F. Brown, P.V. de Souza, and R.L. Mercer, “Maximum mutual information estimation of Hidden Markov Model parameters for speech recognition,” Proceeding of ICASSP, pp. 49–52, 1986.
- [9] D. Povey and P. C. Woodland, “Minimum phone error and I-smoothing for improved discriminative training,” Proceeding of ICASSP, vol. 1, pp. 105–108, 2002.

- [10] H. Hermansky, D. P. W. Ellis, and S. Sharma, "Tandem connectionist feature stream extraction for conventional HMM systems," *Proceeding of ICASSP* vol. III, pp. 1635–1638, 2000.
- [11] Q. Zhu, A. Stolcke, B.Y. Chen, and N. Morga, "Using MLP features in SRI's conversational speech recognition system," *Proceeding of ICSLP*, vol. 2, pp. 921–924, 2005.
- [12] T. Shinozaki and M. Ostendorf, "Cross-validation and aggregated EM training for robust parameter estimation," *Computer speech and language*, vol. 22, no. 2, pp. 185–195, 2008.
- [13] D. Graff, "An overview of Broadcast News corpora," *Speech Communication* vol. 37, issues 1-2, pp. 15-26.
- [14] L. R. Rabiner and B.-H. Juang, "Fundamentals of speech recognition," Prentice Hall Press, 1993.
- [15] R. V. Hogg and A. T. Craig, *Introduction to Mathematical Statistics*, Prentice Hall Press, 1995.
- [16] G. D. Forney Jr, "The viterbi algorithm," *Proceedings of IEEE*, pp. 268-278, 1973.
- [17] B. S. Atal, and M. R. Schroeder, "Predictive coding of speech signals," *Proceeding of AFCRL/IEEE conference on Speech Communication and processing*, pp. 360-361, 1967.
- [18] H. Hermansky, "Perceptual Linear Predictive (PLP) analysis of speech," *J. Acoustic Society of America*, 87, pp. 1738-1752, 1990.

- [19] P. Mermelstein, "Distance measures for speech recognition, psychological and instrumental," *Pattern Recognition and Artificial Intelligence*, C. H. Chen, Ed. New York: Academic, pp. 374-388, 1976.
- [20] S. Furui, "Speaker Independent Isolated Word Recognition Using Dynamic Features of Speech Spectrum," *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. 34, no. 1, pp. 52-59, 1986.
- [21] N. Kumar and A. G. Andreou, "Heteroscedastic Discriminant Analysis and Reduced Rank HMMs for Improved Speech Recognition," *Speech Communication*, vol. 26, pp. 283-297, 1998.
- [22] A. Hyvarinen and E. Oja, "Independent Component Analysis: a Tutorial," http://www.cis.hut.fi/aapo/papers/IJCNN99_tutorialweb/.
- [23] M. Tomita, "An efficient augmented-context-free parsing algorithm," *Computer Linguistics*, 13 (1-2), pp. 31-46, 1987.
- [24] F. Jelinek, "Up from trigrams! - the struggle for improved language models," *Proceeding of Eurospeech*, pp. 1037-1040, 1991.
- [25] K.-F. Lee and H.-W. Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(11), pp.1641-1648, 1989.
- [26] HTK Toolkit, <http://htk.eng.cam.ac.uk>.
- [27] R. Bellman, "Dynamic Programming", *Science*, vol. 153, pp 34-37, 1966.
- [28] S. Haykin, "Neural Networks: A Comprehensive Foundation", New York: Macmillan, 1994.

- [29] R. Hecht- Nielsen, “Theory of the back propagation neural network”, Proceedings of IJCNN, pp. 593 -605, 1989.
- [30] F. Seide, G. Li, and D. Yu, “Conversational Speech Transcription Using Context-Dependent Deep Neural Networks,” Interspeech 2011, pp. 437-440.
- [31] M. Ostendorf, I. Shafran, and R. Bates, “Prosody Models for Conversational Speech Recognition”, Proceeding of the 2nd Plenary Meeting and Symposium on Prosody and Speech Processing, pp. 147-154, 2003.
- [32] S. Seneff, “The use of subword linguistic modeling for multiple tasks in speech recognition,” Speech Communication, vol. 42, pp. 373–390, 2004.
- [33] R. Hu and Y. Zhao, “Knowledge-Based Adaptive Decision Tree State Tying for Conversational Speech Recognition,” Proceedings of ICASSP, pp. 2160 – 2168, 2007
- [34] S. Greenberg, “Speaking in shorthand-A syllable-centric perspective for understanding pronunciation variation,” Speech Communication, vol. 29, no. 2–4, pp. 159–176, 1999.
- [35] M. Riley, B. Byrne, M. Finke, S. Khudanpur, A. Ljolje, J. McDonough, H. Nock, M. Saraclar, C. Wooters, and G. Zavaliagkos, “Stochastic pronunciation modeling from hand-labeled phonetic corpora,” Speech Communication, vol. 29, pp. 209–224, 1999.
- [36] S. J. Young, J.J. Odell, and P.C. Woodland, “Tree-based state tying for high accuracy acoustic modeling,” Proceeding of HLT, 1994
- [37] J. Xue, Y. Zhao, “Novel Lookahead Decision Tree State Tying for Acoustic Modeling” Proceeding of ICASSP, pp 1133-1136, 2007
- [38] W. Reichl and W. Chou, “Robust decision tree state tying for continuous speech recognition,” IEEE Trans. Speech Audio Process., vol. 8, no. 5, pp. 555–566, Sep. 2000.

- [39] I. Shafran and M. Ostendorf, "Acoustic model clustering based on syllable structure," *Computer Speech Language*, vol. 17, no. 4, pp. 311–328, 2003.
- [40] D. Jurafsky, W. Ward, J. Zhang, K. Herold, X. Yu, and S. Zhang, "What kind of pronunciation variation is hard for triphones to model?" *Proceeding of ICASSP*, pp. 577–580, 2001.
- [41] H. Yu and T. Schultz, "Enhanced tree clustering with single pronunciation dictionary for conversational speech recognition," *Proceeding of Eurospeech*, pp. 1869–1872, 2003.
- [42] S. Greenberg, "The Switchboard transcription project," *LVCSR Summer Research Workshop*. Johns Hopkins University, 1996
- [43] A. Ko, R. Sabourin, and A. Britto. "A new HMM- Based ensemble generation method for numerical recognition," *MCS workshop*, pp. 52-61, 2007
- [44] T. Shinozaki and S. Furui, "Spontaneous speech recognition using a massively parallel decoder," *Proceeding of ICSLP*, pp. 1705–1708, 2004.
- [45] L. Kaufman and P. J. Rousseeuw, "Clustering by means of medoids," in *Statistical Data Analysis Based on the L1 Norm*, pp.405-416 , 1987.
- [46] J. Hershey and P. Olsen, "Approximating the Kullback Leibler divergence between gaussian mixture models," *Proceeding of ICASSP*, pp. 317–320, 2007.
- [47] S. Chindaro, K. Sirlantzis, and M. Fairhurst, "Modelling Multiple-Classifer Relationships using Bayesian Belief Networks," *MSC workshop proceedings*, 2007.
- [48] X. Chen, "Ensemble Methods in Large Vocabulary Continuous Speech Recognition", *Master thesis, Computer Science Department, University of Missouri*, 2008.

- [49] X. Cui, J. Xue, P. L. Dognin, U.V. Chaudhari, and B. Zhou, “Acoustic modeling with bootstrap and restructuring for low resourced languages,” Proceeding of Interspeech, pp. 291-294, 2010.
- [50] J. R. Hershey and P. A. Olsen, “Approximating the Kullback Leibler Divergence Between Gaussian Mixture Models,” Proceeding of ICASSP, pp.317-320, 2007.
- [51] X. Cui, X. Chen, J. Xue, P. Olsen, J. Hershey, and B. Zhou, “Acoustic modeling with bootstrap and restructuring based on full covariance”, Proceeding of Interspeech, pp. 4496-4499, 2011
- [52] Linguistic Data Consortium (LDC), [http:// www ldc.upenn.edu/](http://www ldc.upenn.edu/)
- [53] A. Ogawa and S. Takahashi, “Weighted distance measures for efficient reduction of Gaussian mixture components in HMM-based acoustic model,” Proceeding of ICASSP, pp.4173-4176, 2008.
- [54] S. Chen and P.S. Gopalakrishnan, “Clustering via the Bayesian information criterion with applications in speech recognition,” Proceeding of ICASSP, pp.645-648, 1998.
- [55] Y. Zhao, X. Zhang, R.-S. Hu, J. Xue, X. Li, L. Che, R. Hu, and L. Schopp, “An Automatic Captioning System for Telemedicine,” Proceeding of ICASSP, pp. I-957 – I-960, 2006.
- [56] X. Zhang, Y. Zhao, and L. Schopp, “A novel method of language modeling for automatic captioning in TC video conferencing,” IEEE Trans. Information Technology in Biomedicine, vol.11, pp. 332-337, 2007.
- [57] X. Li and Y. Zhao, “A fast and memory-efficient N-gram language model lookup method for large vocabulary continuous speech recognition,” Computer Speech & Language, vol. 21, iss. 1, pp. 1-25, 2007.

- [58] CMU pronunciation lexicon, <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>
- [59] CMU letter-to-sound toolkit, <http://www.speech.cs.cmu.edu/tools/lextool.html>
- [60] Y. Zhao, “Hierarchical mixture models and phonological rules in open-vocabulary speech recognition,” *Proceeding of Eurospeech*, pp. 1586-1590, 1995
- [61] N. Strom, “Phoneme probability estimation with dynamic sparsely connected artificial networks,” in the *Free Speech Journal*, no. 5, 1997.
- [62] C. Breslin and M. J. F. Gales, “Complementary System Generation using Directed Decision Trees”, *Proceeding of ICASSP*, pp. IV-337 – IV-340, 2006.

VITA

Xin Chen was born on Nov. 11, 1983, in Yancheng, Jiangsu, China. He received B.S. degree in Computer Science, from Tianjin University at Tianjin, China. He received M.S. degree in Computer Science from University of Missouri and he expects to receive Ph.D degree in Computer Science from University of Missouri at Columbia, Missouri, US, by the end of 2011.

His research interests include statistical acoustic modeling, automatic speech recognition, algorithm and machine learning.