Training Discriminative HMM by Optimal Allocation of Gaussian Kernels

5 authors, including:

Peng Liu
King Abdullah University of Science and Technology
164 PUBLICATIONS **3,117** CITATIONS
SEE PROFILE

Jun Du
Sun Yat-Sen University
132 PUBLICATIONS **2,565** CITATIONS
SEE PROFILE

Frank K. Soong
Microsoft
307 PUBLICATIONS **5,975** CITATIONS
SEE PROFILE

Training Discriminative HMM by Optimal Allocation of Gaussian Kernels

Zhijie Yan[1], Peng Liu[2], Jun Du[1], Frank Soong[2], and Renhua Wang[1]

[1] iFlytek Speech Lab, University of Science and Technology of China, Hefei
[2] Microsoft Research Asia, Beijing
yanzhijie@ustc.edu pengliu@microsoft.com unuedjwj@ustc.edu

Abstract. We propose to train Hidden Markov Model (HMM) by allocating Gaussian kernels non-uniformly across states so as to optimize a selected discriminative training criterion. The optimal kernel allocation problem is first formulated based upon a non-discriminative, Maximum Likelihood (ML) criterion and then generalized to incorporate discriminative ones. An effective kernel exchange algorithm is derived and tested on TIDIGITS, a speaker-independent (man, woman, boy and girl), connected digit recognition database. Relative 46–51% word error rate reductions are obtained comparing to the conventional uniformly allocated ML baseline. The recognition performance of discriminative kernel allocation is also consistently better than the non-discriminative ML based, nonuniform kernel allocation.

1 Introduction

Hidden Markov Models (HMMs) formulated as mixtures of Gaussian densities have been well developed and successfully deployed to the state-of-the-art Automatic Speech Recognition (ASR) systems. It is known that Maximum Likelihood Estimation (MLE) is a consistent estimate of the underlying distribution under certain assumptions, and Baum-Welch MLE trained Gaussian Mixture Models (GMMs) are also viewed as the best way to approximate the “true” distributions of the speech data.

In experimental or academic ASR systems, the number of Gaussian kernels for every model unit (e.g., states in phonemic HMMs) is usually fixed at a constant value [1]. Given a total number of Gaussian kernels of the whole system, we allocate them to every model in a uniform manner regardless of each model’s underlying distribution. In this kind of kernel allocation, some model units are over-allocated. The kernels they get are more than necessary, or even can not be reliably estimated due to the lack of training data. At the same time, however, some of other model units are under-allocated. More kernels are needed to increase the acoustic resolution of the models so that the fine structure of the distributions in training data can be better modeled.

[1] This work has been done when the first author was a visiting student with Speech Group, Microsoft Research Asia.

In order to alleviate this problem, many criteria have been exploited for building statistical models with adequate topology and number of parameters (e.g., Akaike Information Criterion (AIC) [2], Bayesian Information Criterion (BIC) [3], and Minimum Description Length (MDL) [4]). Realistic acoustic models have also been built by allocating parameters non-uniformly across model units [5–7]. In [8], we come up with the concept of *Parsimonious HMM Modeling*, which aims at adequate number of Gaussian kernels for each state in an HMM system when the total number of kernels is given. An effective algorithm is proposed for Maximum Likelihood (ML) based kernel allocation, and performance improvements are reported for both small and large vocabulary ASR tasks. However, because we are usually more concerned about the classification ability of an ASR system, it is intuitive to introduce some discriminative criteria, e.g., Maximum Mutual Information (MMI) [9], Minimum Classification Error (MCE) [10], for this nonuniform kernel allocation problem.

For ML based Gaussian kernel allocation, the solution is quite straightforward because maximizing the likelihood of one state is independent of any other state. However, for discriminative kernel allocation, the situation is different. As we know, in discriminative training, we are essentially maximizing some measure of the posterior probability of the reference. Because posterior probability is related with not only the likelihood of the reference itself, but also the likelihoods of all other competitors in the hypothesis space, it is then cumbersome to obtain a straightforward solution for kernel allocation as that in ML sense. Therefore, some heuristic metric is needed in discriminative sense, to reflect the relationship between kernel allocation and discrimination.

In [5], a discrimination metric related with kernel weight count is defined, and a successive kernel splitting algorithm is proposed. Because the metric is only heuristic, the greedy kernel splitting may lead to unstable results especially when the target total number of kernels is large. As an alternative, we calculate the metric upon state level, and propose an algorithm starting from a well-trained flat HMM system, which tries to *exchange* kernels among over-allocated and under-allocated states. The well-trained flat system serves as a good reference in calculating reliable posterior probabilities, which can lead to more stable and effective kernel allocation behavior.

To optimize kernel allocation in discriminative sense, we aim at improving classification performance using same amount of parameters than in the original flat system. From another point of view, we can also achieve comparable performance with fewer parameters and reduce computation payload. Our algorithm was tested in a connected digit recognition experiment on the TIDIGITS database. Experimental results show that by using discriminative parsimonious HMM modeling, word error rate can be significantly reduced when compared with the flat, ML trained or discriminative trained baseline. It also outperforms our ML based, non-uniformly allocated model, too. Furthermore, we also build a model using a hybrid version of kernel allocation scheme using both ML and discriminative criteria, and the hybrid model obtains the best recognition performance.

The rest of this paper is organized as follows: In Section 2, the kernel allocation schemes are represented as a unified optimization problem; In Section 3, we provide a brief review of our previous work on ML based kernel allocation; In section 4, the kernel allocation problem is discussed under the discriminative criterion of MMI, and a kernel exchange algorithm is proposed. Experimental results and discussions are listed in Section 5. Finally, we draw our conclusions and future work in Section 6.

2 Representations

In an HMM system with J states where each state output *pdf*, b_j, is characterized by a GMM as:

$$b_j(\boldsymbol{o}) = \sum_{k=1}^{m_j} w_{jk}\mathcal{N}(\boldsymbol{o};\boldsymbol{\mu}_{jk},\boldsymbol{\Sigma}_{jk}) \tag{1}$$

in which m_j is the number of Gaussian kernels in state j, and w_{jk}, $\boldsymbol{\mu}_{jk}$ and $\boldsymbol{\Sigma}_{jk}$ are the weight, mean and covariance matrix of the k^{th} kernel in that state, respectively.

In parsimonious HMM modeling, we first define a kernel allocation configuration vector $\boldsymbol{m} = (m_1, \ldots, m_J)^\top$. The parsimonious solution is formulated as the following optimization:

$$\hat{\boldsymbol{m}} = \text{argmax}_{\boldsymbol{m}}\mathcal{F}(\boldsymbol{m}) \quad \text{s.t.} \sum_{j=1}^{J} m_j = JM_{\text{T}}, M_{\text{L}} \le m_j \le M_{\text{U}} \tag{2}$$

where M_{T} is the target average number of kernels per state, and $\mathcal{F}$ is the selected optimization criterion. In general, states with too many or too few kernels should be avoided to prevent skew probability distributions. So M_{U} and M_{L} are set as the upper and lower-bounds of the number of kernels for each state.

In the following sections, we denote $\gamma_j^{\text{r}}(t)$ and $\gamma_j^{\text{g}}(t)$ as the posterior probability or state occupancy of the j^{th} state at time t, given the reference and the word graph based hypotheses, respectively. L^{r} and L^{g} are the log-likelihoods of the training data, given the reference and the word graph.

3 Maximum Likelihood Based Kernel Allocation

Maximum Likelihood is the most commonly used criterion in statistical training. Before comparing it with discriminative criteria, we first briefly review our previous work on ML based nonuniform kernel allocation [8]. Under ML criterion, $\mathcal{F}$ in Eq.(2) is the total likelihood of the training data:

$$\mathcal{F}(\boldsymbol{m}) = L^{\text{r}}(\boldsymbol{m}) = \sum_{j=1}^{J}\sum_{t=1}^{T} \gamma_j^{\text{r}}(t) \log b_j(\boldsymbol{o}_t) \tag{3}$$

Table 1. Step-back algorithm for Maximum Likelihood based kernel allocation

```
Initialization:
    Do Baum-Welch training to get a uniformly allocated model with all $m_j = M_{\mathrm{U}}$, record all state likelihood functions during the iterative mixture splitting process
Pruning:
    While total number of kernels $> J \times M_{\mathrm{T}}$
        Find the state $i$ with minimal likelihood reduction:
            $i = \mathrm{argmin}_{j, m_j > M_{\mathrm{L}}} [L_j(m_j) - L_j(m_j - 1)]$
        Decrease $m_i$ and total number of kernels by 1
Kernel grouping and refinement:
    For each state $1 \le j \le J$
        Group the $M_{\mathrm{U}}$ kernels to $m_j$ kernels
    Do Baum-Welch retraining
```

Eq. (3) can be optimized by testing the likelihoods of the models trained for every setup of $\boldsymbol{m}$. However, this strategy is too expensive for practical applications. In [8], we proposed a step-back algorithm instead of the global likelihood optimization. This algorithm can be carried out mainly based upon two basic assumptions. The first assumption is that in Baum-Welch training, state segmentations of the references are assumed to change only slightly for different setups of $\boldsymbol{m}$. Hence, we can regard the state likelihood $L_j^{\mathrm{r}}(m_j)$ to be state-specific, and decompose the total likelihood as:

$$L^{\mathrm{r}}(\boldsymbol{m}) = \sum_{j=1}^{J} L_j^{\mathrm{r}}(m_j) \tag{4}$$

Note that in Eq. (4), $L_j^{\mathrm{r}}(m_j) = \sum_{t=1}^{T} \gamma_j^{\mathrm{r}}(t) \log b_j(\boldsymbol{o}_t)$ is the expected state likelihood function which depends only on the parameters of state j. Therefore, we can approximate the global likelihood optimization as a combinatorial optimization of the state likelihood functions. Actually, the first assumption is based on a helpful nature of likelihood: the likelihood of the observation given a certain model is independent of all other competing models. Also, as we can see, all these state likelihood functions are by-products of a conventional iterative mixture splitting training process. They can easily be obtained as the guidance for kernel allocation.

Even under the first assumption, the combinatorial optimization is still an NP-hard problem. Another assumption is made that all state likelihood functions $L_j^{\mathrm{r}}(m_j)$ are convex, so that the problem can be solved in a step-back manner. The physical meaning of the second assumption is that the marginal contribution of each new Gaussian kernel to the state likelihood does not increase with increasing number of kernels. This assumption is quite reasonable in most of the cases in speech modeling.

Based on the two assumptions made above, we came up with an efficient step-back algorithm instead of the global optimization for ML based kernel allocation [8]. The algorithm can be roughly described by the pseudo-code in Table 1.

4 Discriminative Kernel Allocation

In contrast to maximum likelihood estimation, discriminative training optimizes HMM parameters to improve classification performance rather than the likelihood of the training data. Therefore, discriminative training is often seen to be more close to the evaluation criteria (e.g., Word Error Rate), and it is then quite intuitive to introduce some discriminative criteria for our nonuniform Gaussian kernel allocation problem. In this paper, one of the most widely used discriminative training criteria, Maximum Mutual Information, is employed for optimal kernel allocation. We try to allocate kernels non-uniformly across states based upon their influence on the discrimination of the recognition system.

In MMI sense, optimization criterion $\mathcal{F}$ in Eq. (2) becomes the mutual information defined as:

$$\mathcal{F}(\boldsymbol{m}) = L^{\mathrm{r}}(\boldsymbol{m}) - L^{\mathrm{g}}(\boldsymbol{m}) = \sum_{j=1}^{J}\sum_{t=1}^{T}\left[\gamma_j^{\mathrm{r}}(t) - \gamma_j^{\mathrm{g}}(t)\right]\log b_j(\boldsymbol{o}_t) \tag{5}$$

The main difference between ML based and MMI based optimization of Eq. (2) is that for the latter, the posterior probability of the reference state given the word graph can be affected by the parameters of any other competing state. Therefore, $\gamma_j^{\mathrm{g}}(t)$ is related not only to the number of kernels in state j, but also to the numbers of kernels in all its competing states. As a result, L^{g} can no longer be treated as state-specific and decomposed into state level. So there is no straightforward solution to optimize $\mathcal{F}$ like the one in ML sense.

Alternatively, we propose a kernel exchange algorithm to solve the optimization problem. This algorithm is performed in two steps: First, a state-level metric related to the marginal contribution to discrimination is computed. Consequently, based on this metric, those over-allocated states release some kernels to the under-allocated states in the second step. During this procedure, the total number of kernels in the whole system is kept constant because of the "exchange" manner.

The key issue here is how to define the discrimination metric. Intuitively, we use derivatives of the objective function with respect to the kernel weights as an indicator of kernel exchange. Based on the physical meaning of derivatives, a positive derivative value indicates that there is a need of incremental kernel allocation, while a negative value indicates a decreasing demand. Formally, the discrimination metric on state j can be calculated as the weighted sum of the derivatives on all its kernel weights:

$$H_j = \sum_{k=1}^{m_j} w_{jk}\frac{\partial\mathcal{F}}{\partial w_{jk}} \tag{6}$$

By substituting Eq. (5) into Eq. (6), we obtain:

$$H_j = \sum_{t=1}^{T} \left[\gamma_j^{\mathrm{r}}(t) - \gamma_j^{\mathrm{g}}(t)\right] \tag{7}$$

As we can see from Eq. (7), the discrimination metric of a state is measured by considering its count in the reference model against its count in the competing word graph of decoded hypotheses. As the summation term is made up of two parts, both the sign and magnitude of H_j can be analyzed in the following three cases:

A) $H_j \approx 0$ *indicates a well-modeled state with adequate number of kernels:*
This case indicates that $\gamma_j^{\mathrm{g}}(t) \approx \gamma_j^{\mathrm{r}}(t)$, or in another word, the reference state j dominates the decoded word graph. In this situation, because j is already in favor in decoding, no kernel allocation adjustment is needed;

B) $H_j > 0$ *indicates a poorly-modeled state where more kernels are needed:*
This case happens when the reference state j is not in favor in decoding against its competing states. In this situation, because $\gamma_j^{\mathrm{r}}(t) > \gamma_j^{\mathrm{g}}(t)$, a positive value of H_j will result. Therefore, $H_j > 0$ indicates that the state j is poorly-modeled so that more kernels are needed to improve its discrimination. Furthermore, the magnitude of H_j indicates how pool the state has been modeled;

C) $H_j < 0$ *indicates an interference state where the kernels are over-allocated:*
This case happens when state j is not the state in reference, but it is favored in decoding with respect to the reference. In this situation, because $\gamma_j^{\mathrm{r}}(t)$ is zero and $\gamma_j^{\mathrm{g}}(t)$ is positive, a negative value of H_j will result. Because state j interferes with the correct state decoding, its kernels need to be reduced to suppress the interference. Here, the magnitude of H_j represents how worse the state interferes other states.

To summarize, both the sign and magnitude of H_j form the basis of our kernel exchange algorithm: 1) $H_j \approx 0$ indicates that no kernel exchange for state j is needed; 2) Positive H_j indicates that the correct reference state is not favored against competing states in decoding, acoustic discrimination needs to be improved by assigning an extra kernel to state j; 3) Negative H_j indicates that the state interferes with correct hypothesis decoding, less kernels are more appropriate for state j to reduce the interference it causes.

Based on these properties of the discrimination matric H, we can first sort all the states by their H_j, and exchange certain number of kernels from the states with small and negative H to those with large and positive H. This kernel exchange algorithm can be shown as the pseudo-code in Table 2. Our discrimination metric H is defined at state level, as a result, kernel allocation is performed upon the state instead of the kernel level [5]. Besides, as the well-trained flat

Table 2. Kernel exchange algorithm for discriminative kernel allocation

```
Initialization:
    Do Baum-Welch training to get an ML model with fixed, uniform kernel allocation ($M_\mathrm{T}$ kernels per state)
Kernel allocation and discriminative training (MMI):
    Collect statistics to compute discrimination metric $H_j$
    Sort the states by $H_j$
    Kernel exchange:
        Set $n = 0$, define $N$ kernels to be exchanged
        While $n < N$
            Find the state $i$ with most negative $H_i$:
                $i = \mathrm{argmin}_{j, m_j > M_\mathrm{L}, H_j < 0} H_j$
            Decrease $m_i$ by 1, remove the state from the sort list
            Find the state $i$ with most positive $H_i$:
                $i = \mathrm{argmax}_{j, m_j < M_\mathrm{U}, H_j > 0} H_j$
            Increase $m_i$ by 1, remove the state from the sort list
            Increase $n$ by 1
    Do MMI retraining
```

ML model can provide us with relatively reliable estimate of the posterior probabilities, the kernel exchange method can be more stable and effective than a top-down greedy splitting.

5 Experiments

5.1 Experimental Setup

The proposed kernel exchange algorithm was tested in a connected digit, TIDIGITS database. The vocabulary is made up of the digits of ‘one’ to ‘nine’, plus ‘zero’ and ‘oh’. All four categories of speakers, i.e., men, women, boys and girls, were used for both training and testing.

The digits were modeled using 10-state, left-to-right, no-skip, whole-word HMMs. Different training strategies and kernel allocation criteria were evaluated, and Fig. 1 illustrates the models we constructed and compared:

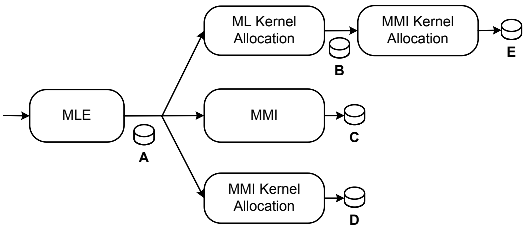

Fig. 1. Experimental setup

Model A: uniformly allocated model using MLE training;
Model B: non-uniformly allocated model using ML criterion;
Model C: uniformly allocated model using MMI training;
Model D: non-uniformly allocated model using MMI criterion;
Model E: non-uniformly allocated model using both ML and MMI criteria.

We used Extended Baum-Welch (EBW) algorithm [11] to re-estimate parameters in MMI training. For the models using MMI based kernel allocation, 10% of the states were chosen to exchange their kernels in each iteration, and we performed the exchange process iteratively for five times. After each exchange process, two MMI training iterations without kernel exchange was also performed to refine the kernel *pdf* parameters.

5.2 Experimental Results

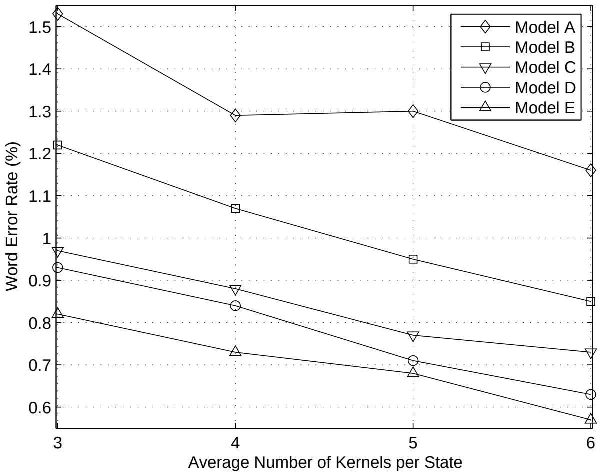

Fig. 2. Word error rate under different training strategies and kernel allocation criteria

The recognition performances from models A to E in average Word Error Rate (WER) are shown in Fig. 2. The WER of a digit is defined as the sum of its deletion, substitution and insertion errors, normalizing by its count in the testing set transcription. Consistent improvements can be observed for non-uniformly allocated models, in comparing with uniformly allocated ones.

We took the models with average 6 kernels per state for analysis, and the kernel allocation results and recognition performances under different training strategies and kernel allocation criteria are given in Table 3. For ML based kernel allocation, a relative error reduction of 26.72% compared to the baseline Model

Table 3. Average state kernel number and word error rate under different training strategies and kernel allocation criteria

Kernel Allocation and Recognition Performance Summaries													
Model		One	Two	Three	Four	Five	Six	Seven	Eight	Nine	Zero	Oh	Overall
A	#kernels	6.0											6.0
	WER(%)	0.58	0.96	0.56	0.86	0.21	0.21	0.21	2.83	0.59	0.37	5.38	1.16
B	#kernels	4.4	5.9	6.5	6.1	5.6	7.0	6.3	5.6	6.3	6.9	5.4	6.0
	WER(%)	0.08	0.88	0.35	0.86	0.16	0.21	0.21	1.44	0.62	0.43	4.19	0.85
C	#kernels	6.0											6.0
	WER(%)	0.32	0.72	0.35	0.64	0.27	0.27	0.27	1.58	0.56	0.27	2.77	0.73
D	#kernels	5.9	6.0	6.0	6.0	6.0	6.3	6.5	5.4	6.1	6.8	5.0	6.0
	WER(%)	0.24	1.22	0.32	0.56	0.29	0.16	0.21	0.94	0.62	0.24	2.12	0.63
E	#kernels	4.1	6.2	6.5	6.4	5.7	7.0	6.7	5.3	6.4	7.6	4.2	6.0
	WER(%)	0.11	0.82	0.37	0.56	0.21	0.13	0.13	0.72	0.78	0.24	2.20	0.57

A was obtained. For MMI based kernel exchange, the reduction was 45.69%. The best performance was obtained by the hybrid Model E, with a relative error reduction of 50.86%.

We also compared the kernel allocation behaviors of different criteria on a digit by digit basis:

For Model B using ML based kernel allocation, more kernels tend to be assigned to the digits which have richer phonetic contents. Longer digits like ‘six’ and ‘zero’ get more kernels than shorter digits like ‘oh’ and ‘one’. It is quite reasonable because of the nature of likelihood.

For Model D using discriminative kernel allocation, the kernel assignment is somewhat different. Longer digits do not necessarily get more kernels (e.g., ‘six’), and dramatic change happens to those troublesome digits like ‘oh’, ‘eight’ and ‘zero’. Because ‘oh’ and ‘eight’ are often observed to cause insertion errors and to be in favor in decoding against correct hypothesis, their kernels are reduced most in MMI based kernel allocation. These released kernels are then reassigned to the under-allocated digits (e.g., ‘zero’), so that their discriminations and recognition performances are improved.

Finally, as shown in Fig. 2 and Table 3, the best performance was obtained by Model E. This model uses ML based kernel allocation first as an initialization, and the likelihood-optimized model provides better estimate of posterior probabilities which are then used to guide the MMI based, discriminative, and sharper kernel refinement. This result suggests that different criteria can be combined together without conflict, to get an improved recognition performance.

Based on our experiments, we believe that a discriminative model can be trained by optimizing both its kernel *pdf* parameters and how these kernels are allocated. When compared with non-discriminative, ML based kernel allocation, better results can be obtained by discriminative kernel allocation. We can also combine these optimization criteria to achieve an improved performance.

6 Conclusions and Future Work

In this paper we propose a discriminative nonuniform Gaussian kernel allocation scheme for training HMM. The kernel allocation is formulated as a unified optimization problem, in which different criteria can be adopted. Two of the most widely used criteria, ML and MMI, are compared, and a kernel exchange algorithm for MMI based kernel allocation is devised. Experimental results show that better recognition performance can be obtained by optimizing kernel allocation discriminatively. And the best performance is obtained by first training an ML based model with nonuniform kernel allocation, and then refining it via the MMI kernel exchange. Kernel allocation behaviors under different criteria are also compared. We find these behaviors are quite reasonable, when the physical meaning of their corresponding criteria are considered. Adopting our discriminative kernel allocation scheme for large vocabulary tasks will be our future work.

References

1. S. J. Young, G. Evermann, et al., *The HTK Book*, Revised for HTK Version 3.3, 2005
2. H. Akaike, "Information Theory and an Extension of the Maximum Likelihood Principle," *International Symposium on Information Theory*, 2nd, Tsahkadsor, Armenian SSR, Hungary, pp. 267-281, 1973
3. G. Schwarz, "Estimating the Dimension of a Model," *Annals of Statistics*, Vol. 6(2), pp. 461-464, 1978
4. J. Rissanen, *Stochastic Complexity in Statistical Inquiry*, World Scientific Publishing Company, 1989.
5. Y. Normandin, "Optimal Splitting of HMM Gaussian Mixture Components with MMIE Training," in *Porc. ICASSP-95*, vol. 1, pp. 449-452, 1995.
6. M. Padmanabhan, L. R. Bahl, "Model Complexity Adaptation Using a Discriminant Measure" *IEEE Trans. Speech and Audio Processing*, vol. 8, no. 2, 2000
7. X. B. Li, F. K. Soong, T. A. Myrvoll, R. H. Wang, "Optimal Clustering and Non-Uniform Allocation of Gaussian Kernels in Scalar Dimension for HMM Compression," in *Proc. ICASSP-05*, vol.1, pp. 669-672, 2005
8. P. Liu, J. L. Zhou, F. K. Soong, "Parsimonious Modeling by Non-Uniform Kernel Allocation," *Technical Report*, Microsoft Research Asia, 2005.
9. L. R. Bahl, P. F. Brown, P. V. de Souza, R. L. Mercer, "Maximum Mutual Information Estimation of Hidden Markov Model Parameters for Speech Recognition," in *Proc. ICASSP-86*, pp. 49-52, 1986.
10. B. H. Juang, W. Chou, C. H. Lee, "Minimum Classification Error Rate Methods for Speech Recognition," *IEEE Trans. Speech and Audio Processing*, vol.5, no.3, pp. 257-265, 1997
11. Y. Normandin, *Hidden Markov Models, Maximum Mutual Information Estimation, and the Speech Recognition Problem*, Ph.D. thesis, Department of Electrical Engineering, McGill University, 1991