Unsupervised Methods for Speaker Diarization

by

Stephen Shum

B.S., University of California, Berkeley (2009)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2011

Author .....
Department of Electrical Engineering and Computer Science
May 6, 2011

Certified by .....
James R. Glass
Principal Research Scientist
Thesis Supervisor

Certified by .....
Najim Dehak
Research Scientist
Thesis Supervisor

Accepted by .....
Professor Leslie A. Kolodziejski
Chair, Department Committee on Graduate Students

Unsupervised Methods for Speaker Diarization

by

Stephen Shum

Submitted to the Department of Electrical Engineering and Computer Science
on May 6, 2011, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

Given a stream of unlabeled audio data, speaker diarization is the process of determining "who spoke when." We propose a novel approach to solving this problem by taking advantage of the effectiveness of factor analysis as a front-end for extracting speaker-specific features and exploiting the inherent variabilities in the data through the use of unsupervised methods. Upon initial evaluation, our system achieves state-of-the art results of 0.9% Diarization Error Rate in the diarization of two-speaker telephone conversations.

The approach is then generalized to the problem of K-speaker diarization, for which we take measures to address issues of data sparsity and experiment with the use of the von Mises-Fisher distribution for clustering on a unit hypersphere. Our extended system performs competitively on the diarization of conversations involving two or more speakers. Finally, we present promising initial results obtained from applying variational inference on our front-end speaker representation to estimate the unknown number of speakers in a given utterance.

Thesis Supervisor: James R. Glass
Title: Principal Research Scientist

Thesis Supervisor: Najim Dehak
Title: Research Scientist

Acknowledgments

When it comes to research advisors, I do not think I could have found a better combination than Jim and Najim. The ever-contrasting viewpoints within our trio never made for a dull moment, and yet the balance between Jim's insightful patience coupled with Najim's impulsive brilliance created an environment teeming with intellectual potential. I certainly had my hands full trying to make the most of this opportunity. Surrounding Jim is always the steady sense of calm and purpose; during every interaction, he always knows exactly what questions to ask to guide me down the right path. Things with Najim were more volatile, but only in the sense that his abundant energy and infectious passion for the field often rubbed off, and now we are both slightly insane (but in a good way, I think). I cannot speak enough of my appreciation for both Jim and Najim; in truth, I really could not have done this without them.

I am indebted to everyone in the Spoken Language Systems Group, a place which I have called home these past two years. And it truly pains me that listing the number of ways in which each of you makes my day brighter might in fact double the length of this thesis. An exclusive shoutout, however, must go to Marcia, who has quite possibly taught me more about life in the last two years than I ever planned on knowing. Thank you for finding so many ways to keep things interesting every time I journey to the 4th floor.

Furthermore, to my partner in crime, Ekapol, upon whom I unload the struggles of my work on a daily basis and who, in return, lets me finish his lunch while telling me how to make things better. By the time we both graduate, it is fair to say that he will have completed two Master's degrees and two Ph.D's. To be honest, I wouldn't have it any other way.

I would like to thank Doug Reynolds for his valuable advice in helping me shape the general direction of this work. Thank you also, to Reda Dehak for all his help in software development, without which I think my experiments might still be running.

Sitting on the third floor has given me the privilege of sharing an office (G350) with some truly incredible people. To Katrina, Greg, Mason, Michael, Adam, Timo, Mitch, and Ian: I cannot imagine myself in a better work environment than the one you have provided. Thanks for putting up with me.

One cannot possibly make it through graduate school alone. In these last two years, I have met some incredible people that I am lucky to call my friends. Listing them all would be a little nuts, but it would be unfair not to mention at least a few of those who keep me sane on a regular basis: John C, Michael A, Samantha G, Michael P, Nick H, Phil N, and Alicia N. You guys rock.

Roommates belong in their own special category, and so do close friends. So when Nick L told me back in April 2010 that he was moving to Boston, I knew he would end up in both categories. I was also pretty sure that come September 2010, I would be eating better home-cooked food and drinking a lot more beer. While I should not comment on the latter, I was definitely correct about the former. The truth is, Nick simply amazes me. There is not another person I know that possesses the same amount of outgoing energy and positive attitude, nor is there a person who is as willing to listen or help out as Nick. Our time spent chatting, watching basketball, skiing/snowboarding, or going out on epic bike rides together has been absolutely instrumental to the success of my endeavors. Thanks for making this year awesome.

My final thanks goes to Mom and Dad, whose endless support and encouragement I could not have survived without. From the burden of worry they have had to bear on my behalf to their sigh of relief that I have finally completed this stage of my studies, my parents have always been there for me. Any time of day or night, through thick and thin, triumphs and tribulations, I always knew that I could count on them to provide the most fitting words of comfort or advice. For that and so much more, no one could possibly give them the credit they truly deserve. Thank you for everything.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Imagine an educational setting where lectures and discussions are not simply recorded, but also automatically segmented and annotated to produce a complete transcript of the audio file. Brilliant, "heat-of-the-moment" insights and explanations would be preserved without the need for post-session manual labor, allowing for efficient search and retrieval of the material in the future. Further imagine that those same recordings are made publicly available on some indexed database accessed via spoken inquiry. These educational resources can change the way people learn: students with disabilities can enhance their educational experience, professionals can keep up with advancements in their field, and people of all ages can satisfy their thirst for knowledge [8].

The continually decreasing cost of and increasing access to processing power, storage capacity, and network bandwidth is facilitating the amassing of large volumes of audio, including podcasts, lectures, meetings, and other "spoken documents". There is a growing need to apply automatic human language technologies to allow efficient and effective searching, indexing, and accessing of these information sources. Using speech recognition to extract the words being spoken in the audio is a good place to start, but raw transcripts are difficult to read and often do not capture all the information contained within the audio. As such, other technologies are needed to extract the relevant meta-data which can make the transcripts more readable and provide context and information beyond the simple word sequence of automatic speech recognition. Speaker turns and sentence boundaries are examples of such meta-data, both of which help provide a richer transcription of the audio, making transcripts more readable and potentially helping with other tasks such as summarization, parsing, or machine translation [9].

In general, a spoken document is a single channel recording that consists of multiple audio sources, which can include different speakers, music segments, types of

noise, et cetera. Audio diarization is defined as the task of marking and categorizing the different audio sources within a spoken document. The types and details of the audio sources are application specific. For our purposes in particular, the purpose of a speaker diarization system is, given an unlabeled audio file, to mark where speaker changes occur (*segmentation*), and then associate the different segments of speech belonging to the same speaker (*clustering*) [9].

This thesis explores a new set of approaches to speaker diarization based on the recent successes of factor analysis-based methods. We adapt the methods from speaker recognition to exploit the notion of intra-conversation variability for the diarization of two-speaker telephone conversations. That approach is then generalized to the problem of K-speaker diarization, in which the number of speakers K is known *a priori*. Along the way, we address issues of data sparsity and experiment with the probabilistic modeling of data on a unit hypersphere. Finally, we extend upon previous work on variational methods to tackle the general diarization task where the number of participating speakers and their respective identities are unknown.

The rest of this paper is organized as follows: Chapter 2 will review some of the previous approaches taken in speaker diarization, after which Chapter 3 will introduce the theory behind factor analysis in speaker verification, which serves as the backbone to our work. We describe the development of our diarization system for two-speaker telephone conversations in Chapter 4, and then extend those methods in Chapter 5 to conversations involving K speakers, where K is given. In Chapter 6 we outline the theory behind Variational Bayesian methods for statistical inference and apply these ideas to the task of estimating an unknown number of speakers (K) in an audio stream. Finally, Chapter 7 concludes this thesis with a summary of its contributions and a look at possible directions for future work.

Chapter 2

Background and Related Work

There exists a large amount of previous work on the problem of speaker diarization, or "who spoke when." In [9], Tranter and Reynolds put together a comprehensive and thorough review of different approaches to diarization. This chapter will briefly summarize the key ideas behind those various systems in an attempt to motivate the rest of the work described in this thesis.

2.1 The Bayesian Information Criterion

Because of its relative simplicity, the Bayesian Information Criterion (BIC) has served as a backbone to the development of many approaches in diarization. Derived directly from the Laplace approximation, whose details canbe found in [10], BIC can be seen as a penalized maximum likelihood criterion, as it introduces a penalty for any given model based on the number of free parameters that need to be estimated in that model [11]. To be precise, let Y be the dataset we are modeling, and let $M_i \in \mathcal{M}$ be some model. Denote $\#(M_i)$ as the number of free parameters in model M_i and assume that we are to maximize some likelihood function $\mathcal{L}(Y, M_i)$ separately for each model. The BIC score is then defined as

$$\mathrm{BIC}(M_i) = \log \mathcal{L}(Y, M_i) - \lambda \frac{1}{2} \#(M_i) \times \log L \tag{2.1}$$

where L denotes the number of data points (i.e. cardinality) of our set Y, and λ is the BIC weight (typically set to 1.0) [1]. We can see that BIC penalizes the models with more parameters to be estimated, and thus the model that maximizes the BIC score ought to correspond to some form of optimality between model simplicity and modeling the data.

2.2 A Baseline System

Many systems have been designed and tuned using BIC. One such system, developed by MIT Lincoln Laboratory, serves as a baseline for our work as well as the work in [12]. Figure 2-1 provides a system diagram of this approach.

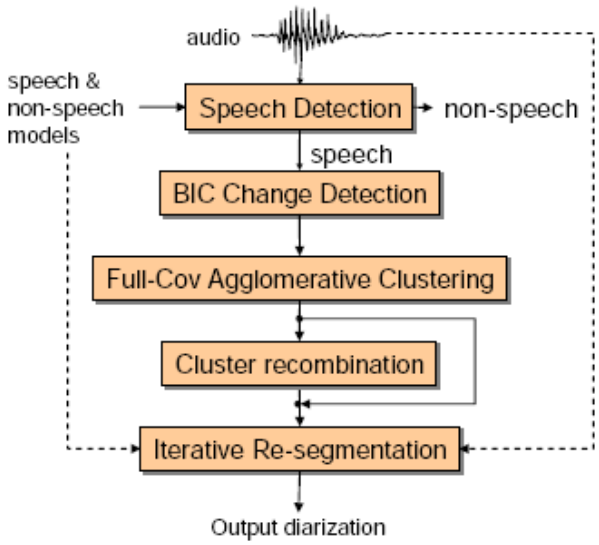

Figure 2-1: *System diagram of the MIT Lincoln Laboratory RT-04 BIC-based baseline diarization system. Figure taken from [1].*

In general, most systems begin with *speech detection*, a step whose aim is to find regions of speech in the audio stream. Depending on the domain data being used, the non-speech regions that are discarded can consist of many different forms of acoustic phenomena such as silence, music, room noise, background noise, or even cross-talk [9]. The general approach used for speech detection is maximum likelihood classification with Gaussian Mixture Models (GMMs) trained on labeled training data. Other approaches can also be used, including multi-state Hidden Markov Models (HMMs) or even speech (phoneme- or word-level) recognizers.

The aim of the next step, *change detection*, is to find points in the audio stream that are likely to be change points between audio sources. The approach in [1] is a variation on the BIC-based technique introduced in [11]. As depicted in Figure 2-2, this method searches for change points within a window using a penalized likelihood ratio test of whether the data in the window is better modeled by a single distribution (Hypothesis 0: no change point) or by two different distributions (Hypothesis 1: change point) [1]. If a change is found, the window is reset to the change point and the search is restarted. If no change point is found, the window is increased and the

search is re-done.

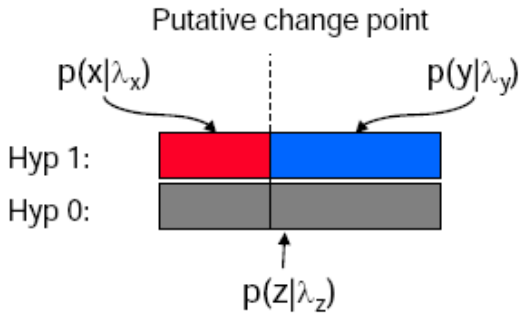

Figure 2-2: *Illustration of BIC-based speaker change point detection for a given point in the window. Figure taken from [1].*

The completion of both speech and speaker change detection gives us an initial segmentation from which to begin our *clustering*. The purpose of this stage is to associate or cluster segments from the same speaker together. Ideally, this procedure produces one cluster for each speaker in the audio with all segments from a given speaker in a single cluster. Our baseline system uses an agglomerative hierarchical clustering approach which consists of the following steps [12]:

1. Initialize leaf clusters of tree with segments from the change detection stage.
2. Compute pairwise distances between each cluster.
3. Merge closest clusters.
4. Update distances of the remaining clusters to new cluster.
5. Iterate steps 3-4 until some stopping criterion is met.

Depending on its application, each system might have a different distance metric (e.g. generalized likelihood ratio distance function, cosine distance, et cetera), merging process, and stopping criterion. The work in [1] uses a BIC-based stopping criterion with the BIC weight λ tuned on development data suited towards the specific diarization task.

The *cluster recombination* step takes advantage of state-of-the-art speaker recognition techniques to refine the clusters generated previously [9]. There exists a wide repertoire of techniques for speaker modeling and matching; Chapter 3 will provide an overview of these methods and introduce the theory behind the state-of-the-art.

The last stage found in many diarization systems is an *iterative re-segmentation* of the audio via Viterbi decoding using the final cluster models and non-speech models.

The purpose of this stage is to refine the original segment boundaries and to fill in short segments that may have been removed for more robust processing in the clustering stage [9].

2.3 Variations

It should be noted that the steps involved in building the baseline diarization system described previously are easily modified. For instance, after the iterative re-segmentation step, the approach in [12] runs its clustering step one more time before outputting the final diarization results.

2.3.1 HMM-based Approaches

Alternatively, another approach has found success with an integrated scheme for segmentation and clustering. Based on an evolutive-HMM (E-HMM), detected speakers help influence both the detection of other speakers and the speaker boundaries [13]. The recording is represented by an ergodic HMM in which each state represents a speaker and the transitions model the changes between speakers. The initial HMM contains only one state and represents all of the data. For each iteration, a short speech segment assumed to come from a non-detected speaker is selected and used to build a new speaker model. A state is then added to the HMM to reflect this new speaker and the transition probabilities are modified accordingly [9]. A new segmentation is then generated from a Viterbi decode of the data with the new HMM, and each model is adapted using the new segmentation. This phase is repeated until the speaker labels no longer change, and the process of adding new speakers is repeated until there is no gain in terms of comparable likelihood or there is no data left to form a new speaker.

The main advantage of such an integrated approach is that the system uses all of the provided information in the audio at each step [9]. A similar sort of approach was developed based on the so-called "Infinite Hidden Markov Model," where a Hierarchical Dirichlet Process (HDP) was introduced on top of an HMM (hence, an HDP-HMM), which allows for up to a countably infinite number of HMM states (i.e. speakers) [14]. From this, a "sticky" extension was proposed to allow for more robust learning of smoothly varying dynamics, which provided an elegant and effective way to estimate the number of participate speakers in an audio stream [15]. Further work was done in [16] to relax the Markovian constraints of a standard HMM in order to capture state durations that are not geometrically distributed; this resulted in an

HDP-HSMM that took into account explicit-duration semi-Markovianity.

2.3.2 Variational Bayes Approaches

In one sense, Hierarchical Dirichlet Processes have become the cornerstone of non-parametric Bayesian statistics, and the development of Markov Chain Monte Carlo (MCMC) sampling methods have enabled the practical application of these methods to a variety of problems, including diarization [17]. However, there are other methods that have been explored and developed as well. One such class of techniques is provided by *variational inference* [18], whose application to diarization was pioneered by Valente through the use of Variational Bayesian Gaussian Mixture Models [10] and further extended to incorporate factor analysis priors by Kenny [12]. The nature of these methods serves as the initial inspiration for the work detailed in Chapter 6.

2.3.3 Factor Analysis-based Approaches

Factor analysis, whose theory will be described in Chapter 3, has proven to be very effective in tackling the problem of speaker modeling [5]; it is therefore natural to try to bring these methods to bear on the problem of diarization [12]. Speaker diarization using factor analysis was first introduced in [19] using a stream-based approach. This technique performs an on-line diarization (with a fixed time lag) where a conversation is seen as a stream of fixed duration time slices ($\sim$60sec). The system operates in a causal fashion by producing segmentation and clustering for a given slice without requiring the following slices [20]. Speakers detected in the current slice are compared with previously detected speakers to determine if a new speaker has been detected or previous models should be updated. This approach attained state-of-the-art results on the CallHome telephone diarization task in which each conversation involved an unknown number of speakers [19]. We use these results as a benchmark for the development of our system in Chapters 5 and 6.

Finally, we briefly describe a system inspired by both factor analysis and variational Bayesian methods. The work was motivated by a desire to build on the success of factor analysis methods as well as to capitalize on the advantages that a Bayesian approach may bring to the diarization problem (e.g. convergence guarantees, using probability distributions to avoid unnecessarily premature hard decisions, automatic regularization, et cetera) [20]. This approach to speaker clustering takes advantage, in the Bayesian sense, of a highly informative factor analysis-based prior and iteratively estimates the posterior distribution of the speaker model as well as that of each segment (i.e. to which speaker each segment is assigned). In the NIST 2008 SRE

two-speaker telephone diarization task, this algorithm peformed significantly better than both the baseline and stream-based approaches, thus attaining state-of-the-art results. Not only do we use these results as a benchmark for the development of our initial system in Chapter 4, they also serve as an initial starting point for our subsequent explorations in Chapter 6.

2.4 Data Sets

In this section, we provide an overview of the data that we use to build and evaluate our diarization system. To train our models - namely the Universal Background Model and Total Variability matrix to be discussed in Chapter 3 - we use the Switchboard (English) and Mixer (multilingual) Corpora. The Mixer Corpus was used during the NIST Speaker Recognition Evaluations (SREs) in 2004, 2005, and 2006. All in all, these data include over 1000 hours of speech from a variety of languages and, for the most part, match the data used to train the models in [12].

In Chapter 4, we evaluate our system on the summed-channel telephone data from the NIST 2008 SRE consisting of 2215 two-speaker telephone conversations, each approximately five minutes in length ($\approx$ 200 total hours). This, once again, is the same evaluation set that was used to obtain our benchmark results in [12].

When we extend our initial approach to handle an arbitrary number of speakers in Chapter 5, we evaluate our system on the multilingual CallHome data, a corpus of multi-speaker telephone conversations. For proper comparison with the current state-of-the-art results presented in [19], we use a corresponding subset of data from the NIST 2000 Speaker Recognition Evaluation [21]. This subset consists of 500 recordings, each 2-5 minutes in length, containing between two and seven participants. Table 2.1 provides a summary of the CallHome corpus broken down by number of speakers and language spoken.

	NUMBER OF SPEAKERS						
LANGUAGE	**2**	**3**	**4**	**5**	**6**	**7**	**TOTAL**
Arabic	50	28	10	3	4		**95**
English	49	7					**56**
German	52	12	3				**67**
Japanese	53	10	2	3			**68**
Mandarin	47	50	18	2	1		**118**
Spanish	52	29	10	2	1	2	**96**
TOTAL	**303**	**136**	**43**	**10**	**6**	**2**	**500**

Table 2.1: *Summary of CallHome corpus broken down by number of participating speakers and language spoken.*

Chapter 3

Speaker Recognition Using Factor Analysis

At the heart of speaker diarization lies the problem of speaker modeling; logically, successful techniques in speaker modeling should also be capable of producing good results in diarization [12]. In recent years, methods in *factor analysis*, where a low-dimensional space of "factors" is used to statistically model a higher dimensional "feature space," have proven to be very effective in speaker recognition, the task of verifying whether two utterances are spoken by the same speaker [5]. To lay the theoretical groundwork for the rest of this thesis, we provide some intuition on how factor analysis was developed to serve as a front-end to extract speaker-specific information from an audio sequence.

3.1 The GMM-UBM Approach

Given an utterance u and a hypothesized speaker S, the task of speaker verification is to determine if u was spoken by S. To keep things simple, we usually make the assumption that u contains speech from only one speaker. This single-speaker detection task can be restated as a basic hypothesis test between H_1, where u is from the hypothesized speaker S, and H_0, where u is *not* from the hypothesized speaker [2]. To make this decision, we apply the likelihood ratio test given by

$$\frac{p(u|H_1)}{p(u|H_0)} \begin{cases} \geq \theta & \text{accept } H_1 \\ < \theta & \text{reject } H_1 \end{cases} \tag{3.1}$$

The whole point of speaker verification is to determine techniques to compute values for the two likelihoods, $p(u|H_0)$ and $p(u|H_1)$. For notational purposes, we can let

H_1 be represented by a probabilistic model λ_S that characterizes the hypothesized speaker S, and we can use $\lambda_{\bar{S}}$ to represent the probabilistic model of the alternative hypothesis H_0. While the model in λ_S is well defined and can usually be estimated via some enrollment speech from S, the model for H_0 is less well defined since it needs to represent the entire space of possible alternatives to the hypothesized speaker [2].

The approach that is used to tackle the problem of alternative hypothesis modeling is to pool speech from many speakers and train a single model known as the Universal Background Model (UBM). The advantage of this approach is that a single speaker-independent model can be trained once for a particular task and then used for all hypothesized speakers in the task [2]. We need to be careful to ensure that the speakers used in the training of the UBM will not, upon deployment of the system, be used or enrolled as hypothesis speakers.

3.1.1 Gaussian Mixture Models

The Gaussian Mixture Model (GMM) is a generative model used widely in speaker verification. It can be seen as a semi-parametric probabilistic method that, given appropriate front-end features, adequately represents a speech signal and its variabilities [3]. Given a GMM θ consisting of C components and F-dimensional feature vectors, the likelihood of observing a given feature vector y is computed as

$$p(y|\theta) = \sum_{c=1}^{C} \pi_c \mathcal{N}_c(y|\mu_c, \Sigma_c) \tag{3.2}$$

where the sum of the mixture weights $\sum_c \pi_c = 1$, and $\mathcal{N}_c(y|\mu_c, \Sigma_c)$ is a multivariate Gaussian with F-dimensional mean vector μ_c, $F \times F$ covariance matrix Σ_c, and probability distribution function

$$\mathcal{N}_c(y|\mu_c, \Sigma_c) = \frac{1}{(2\pi)^{F/2}|\Sigma_c|^{1/2}} \exp\{-\frac{1}{2}(y-\mu_c)'\Sigma_c^{-1}(y-\mu_c)\}. \tag{3.3}$$

Though it is possible to use full covariances in the implementation, it is a bit more computationally efficient to use only diagonal covariances. The density modeling of a C-th order full covariance GMM can equally well be achieved using a diagonal covariance GMM of higher order (i.e. $C' > C$) [2]. Finally, to preserve our previously defined notation, we can denote $\theta = \{\theta_1, ..., \theta_C\}$, where $\theta_c = \{\pi_c, \mu_c, \Sigma_c\}$.

Given a sequence of feature vectors $u = \{y_1, y_2, ..., y_L\}$, we make the assumption that each observation vector is independent of the other observations [3]. As such, the likelihood of a given utterance u given the model θ is simply the product of the

likelihood of each of the L frames. Because the multiplication of many probabilities on a machine can potentially lead to computational underflow, we instead use the log-likelihood in our computation as follows:

$$\log p(u|\theta) = \sum_{t=1}^{L} \log p(y_t|\theta) \tag{3.4}$$

3.1.2 The EM Algorithm

Given a collection of training vectors, the maximum likelihood (ML) parameters of a model θ can be estimated via the Expectation-Maximization (EM) algorithm [22]. The EM algorithm iteratively refines the GMM parameters to monotonically increase the likelihood of the estimated model for the observed feature vectors. That is, if u were an utterance represented by a set of observed feature vectors $u = \{y_1, y_2, ..., y_L\}$ as given in the previous subsection, then for iterations k and $k+1$, $p(u|\theta^{(k+1)}) > p(u|\theta^{(k)})$ [2]. As such, given an initial model $\theta^{(0)}$, we hold the initial parameters fixed and then estimate an updated model $\theta^{(1)}$. Then we hold $\theta^{(1)}$ fixed to estimate $\theta^{(2)}$ and so on until some convergence threshold is reached (usually determined by some sort of stability in the data likelihood). The equations for these ML parameter updates for each Gaussian c, where $c \in \{1, ..., C\}$ can be found in [3].

3.1.3 Universal Background Model

In the GMM-UBM Approach to speaker verification, we use a single, speaker- independent background model to represent $p(\cdot|\theta_{\bar{S}})$. The UBM is a large GMM trained to represent the speaker-independent distribution of features; in particular, we are looking for speech that is reflective of the expected alternative speech to be encountered during recognition. This applies to the type of speech (i.e. casual conversation or business meeting or telephone service), the quality and channel (i.e. close-talking microphone? microphone array? shouting outdoors? telephone?), and the speaker population (i.e. males, females and/or children).

3.1.4 Speaker Enrollment via MAP Adaptation

We have now discussed how a UBM is trained to represent the alternative hypothesis; what still remains is a speaker-specific model for our speaker S. The initial inclination may be to apply the EM algorithm on the given speaker's training data; however, the amount of speaker-specific data that is present would be much too sparse to give a good representation of the speaker. We may even end up modeling the channel

characteristics or other aspects of the data instead. By contrast, the larger abundance of speech data used to estimate the UBM might be a better starting point for modeling a specific speaker [3]. Known as *maximum a posteriori* (MAP) adaptation, we derive the speaker's model by updating the well-trained parameters in the UBM. This provides a tighter coupling between the speaker's model and the UBM which not only produces better performance than using separate (decoupled) models, but also allows for a fast-scoring technique.

The process of MAP adaptation is very similar to that of the EM algorithm and, along with the fast log-likelihood ratio scoring technique, is further detailed in [2]. Here, we provide a brief overview of the procedure. Given a UBM parameterized by θ_{UBM} and training feature vectors from the hypothesized speaker $u_S = \{y_1, y_2, ..., y_L\}$, we first determine the probabilistic alignment between the training vectors and the UBM mixture components. That is, for UBM mixture c, we compute

$$\gamma_t(c) = P(c|y_t, \theta_{\text{UBM}}) = \frac{\pi_c \mathcal{N}_c(y_t|\mu_c, \Sigma_c)}{\sum_{i=1}^{C} \pi_i \mathcal{N}_i(y_t|\mu_i, \Sigma_i)} \tag{3.5}$$

and then use $\gamma_t(c)$ and y_t, $t = 1, ..., L$ to compute the relevant Baum-Welch statistics for the weight, mean, and covariance parameters of our UBM:

$$N_c\left(u_S\right) = \sum_{t=1}^{L} P\left(c|y_t, \theta_{\text{UBM}}\right) = \sum_t \gamma_t(c) \tag{3.6}$$

$$\bar{F}_c\left(u_S\right) = \frac{1}{N_c(u_S)} \sum_t \gamma_t(c) \cdot y_t \tag{3.7}$$

$$\bar{S}_c\left(u_S\right) = \frac{1}{N_c(u_S)} \sum_t \gamma_t(c) \cdot y_t y_t^* \tag{3.8}$$

These sufficient statistics from the training data are used to update the old UBM sufficient statistics for mixture c to generate adapted parameters according to the following update equations:

$$\hat{\pi}_c = \beta \left(\alpha_c \frac{N_c(u_S)}{L} + (1 - \alpha_c)\pi_c \right) \tag{3.9}$$

$$\hat{\mu}_c = \alpha_c \bar{F}_c(u_S) + (1 - \alpha_c)\mu_c \tag{3.10}$$

$$\hat{\Sigma}_c = \alpha_c \bar{S}_c(u_S) + (1 - \alpha_c)(\Sigma_c + \mu_c \mu_c^*) - \hat{\mu}_c \hat{\mu}_c^* \tag{3.11}$$

The scale factor β is computed over all adapted mixture weights to ensure that $\sum_c \hat{\pi}_c = 1$, and α_c, where $c = 1, ..., C$, are the data-dependent adaptation coefficients controlling the balance between old and new estimates of the GMM parameters.

These coefficients are defined as

$$\alpha_c = \frac{N_c(u_S)}{N_c(u_S) + r} \tag{3.12}$$

where r is a constant relevance factor. The update equations (3.10)-(3.12) can be derived from the general MAP estimation equations for a GMM using constraints on the prior distribution as discussed in [2]. However, the update equation (3.9) for the weight parameter does not follow from the general MAP estimation equations. The actual MAP-based equation is

$$\hat{\pi}_c = \frac{r + N_c(u_S)}{L + Cr} \tag{3.13}$$

It was found experimentally that using this estimate reduced performance compared to using the current weighted average in (3.9). Figure 3-1 displays an example of MAP adapting the mean and covariance parameters of the observed Gaussians. In practice, however, only the mean vectors μ_c, $c = 1, ..., C$ are adapted; updated weights and covariance matrices do not significantly affect system performance [3].

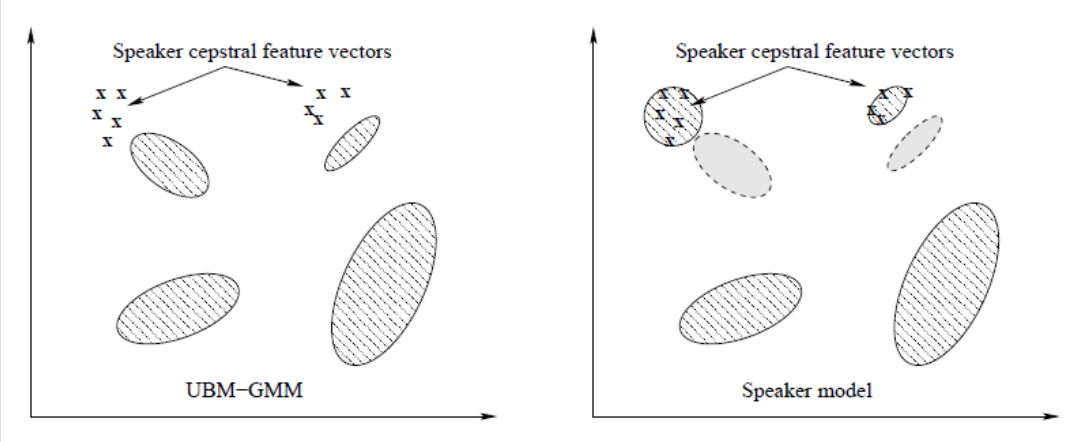

Figure 3-1: *Depiction of maximum a posteriori (MAP) adaptation, taken from [2, 3].*

3.2 Joint Factor Analysis

The GMM-UBM-MAP approach achieved great success, but suffered from issues of data sparsity in the enrollment phase. That is, in the presence of limited speaker training data, MAP adaptation sometimes failed to capture a thorough and complete representation of the speaker's true model. In particular, the lack of data prevented some components of the UBM from being adapted. What was needed was some

way to correlate, or link together, the different Gaussian components of the UBM. If found, these inter-component relationships could then help us obtain a more complete representation of an enrolled speaker despite lack of training data.

Such notions motivated the exploration of the *eigenvoices* and *eigenchannels* as applied to the previously described GMM-UBM-MAP approach for speaker modeling [3]. Known formally as Joint Factor Analysis [23], this theory begins with the idea that a speaker model obtained by adapting from a UBM (parametrized with C mixture components in a feature space of dimension F) can also be viewed as a single *supervector* of dimension $C \cdot F$ along with a diagonal "super"-covariance matrix of dimension $CF \times CF$. This mean supervector is generated by concatenating all the Gaussian component means, while the covariance matrix is generated by respectively concatenating (along its diagonal) all the diagonal covariances of each mixture.

The point of factor analysis in general is that a measured data vector (i.e. speaker supervector) can be high-dimensional, but we may have reason to believe that the data lie near a lower-dimensional subspace [24]. The key assumption in Joint Factor Analysis is that the GMM supervector of the speaker- and channel-dependent M for a given utterance can be broken down into the sum of two supervectors

$$M = s + c \tag{3.14}$$

where supervector s depends on the speaker and supervector c depends on the channel. Moreover, we can write

$$s = m + Vy + Dz \tag{3.15}$$

$$c = Ux \tag{3.16}$$

where each of V and U are low-rank matrices that represent the lower dimensional subspaces in which the speaker and channel lie. Respectively, these are known as the eigenvoices and the eigenchannels. Lastly, m is the speaker- and channel-independent supervector that can also be interpreted as the initial UBM supervector, while D is a diagonal $CF \times CF$ matrix that serves a purpose similar to that of MAP adaptation in the original GMM-UBM framework. In particular, it models the residual variabilities of a speaker that are not captured by V. The exact details of this theory are beyond the scope of this thesis, but the basic idea is depicted in Figure 3-2 and a thorough explanation can be found in [3]. The terminology *Joint* Factor Analysis comes from the fact that there are three latent variables to be estimated (x, y, and z) jointly. "Traditional" Factor Analysis usually involves only one latent variable.

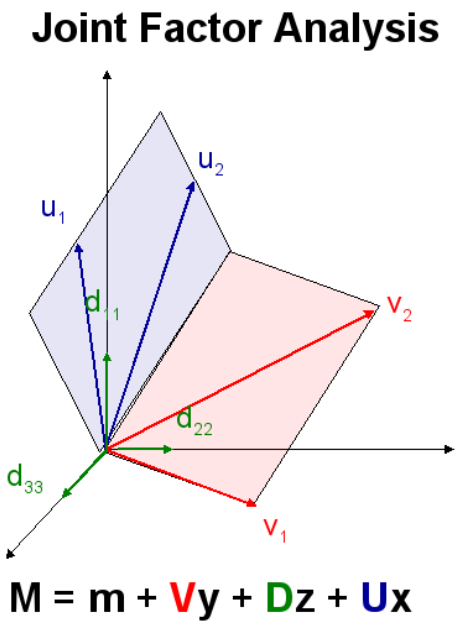

Figure 3-2: *A cartoon depicting the essentials of Joint Factor Analysis [4].*

3.3 The Total Variability Approach

The Joint Factor Analysis approach to speaker recognition presents powerful tools to better represent speaker variabilities and compensate for channel or, more generally, session inconsistencies [25]. In practice, however, its relative complexity in both theory and implementation motivates a more simplified solution. A more recent approach represents all the factors in a (single) total variability space with no distinction made between speaker and channel subspaces [3]. We can then represent the speaker- and session-dependent supervector M as

$$M = m + Tw \tag{3.17}$$

where m is still the speaker- and session-independent supervector taken from a UBM, T is a rectangular matrix of low rank that defines the new total variability space and w is a low-dimensional random vector with a normally distributed prior $\mathcal{N}(0, I)$. The remaining variabilities not captured by T are accounted for in a diagonal covariance matrix, $\Sigma \in \mathbb{R}^{CF \times CF}$. The components of w are referred to as "total factors," while w itself can be referred to as a "total factor vector," or an *i-vector*, short for "Intermediate Vectors" for their intermediate representation between an acoustic feature vector and a supervector, or "Identity Vectors" for their compact representation of a speaker's identity. The following subsections will briefly summarize the theory behind this approach.

3.3.1 Parameter Training and Estimation

The main difference between training the total variability matrix T and learning the eigenvoice V from JFA is that in training the eigenvoice, all recordings of a given speaker are considered to belong to the same person, whereas in training T, each instance of a given speaker's set of utterances is regarded as having been produced by a different speaker. In the rest of this section, we outline the key details behind the estimation of T as well as the extraction of w.

Sufficient Statistics

The total factor vector w is a latent variable whose posterior distribution can be determined using Baum-Welch statistics from the UBM [5]. Suppose our given utterance u is represented as a sequence of L frames $u = \{y_1, y_2, ..., y_L\}$. Then similar to the previous equations (3.6)-(3.8), the relevant Baum-Welch statistics are

$$N_c\left(u\right) = \sum_{t=1}^{L} P\left(c|y_t, \theta_{\text{UBM}}\right) = \sum_{t} \gamma_t(c) \tag{3.18}$$

$$F_c\left(u\right) = \sum_{t=1}^{L} P\left(c|y_t, \theta_{\text{UBM}}\right) y_t = \sum_{t} \gamma_t(c) \cdot y_t \tag{3.19}$$

$$S_c\left(u\right) = \text{diag}\left(\sum_{t} \gamma_t(c) \cdot y_t y_t^*\right) \tag{3.20}$$

where $c = 1, ..., C$ is the index of the corresponding Gaussian component. Exactly like in (3.5), $\gamma_t(c) = P\left(c|y_t, \theta_{\text{UBM}}\right)$ corresponds to the posterior probability of mixture component c (from our UBM parametrized by θ_{UBM}) generating the frame y_t at time t. These posteriors are easily calculated from the UBM.

To make our future notation a bit simpler, let us denote the centralized first and second order Baum-Welch statistics by $\tilde{F}_c(u)$ and $\tilde{S}_c(u)$ as follows:

$$\tilde{F}_c(u) = F_c\left(u\right) - N_c\left(u\right) m_c = \sum_{t} \gamma_t(c) \cdot (y_t - m_c) \tag{3.21}$$

$$\tilde{S}_c(u) = \text{diag}\left(\sum_{t} \gamma_t(c) \cdot (y_t - m_c)(y_t - m_c)^*\right) \tag{3.22}$$

where m_c is the subvector corresponding to mixture component c of our supervector m.

Lastly, let $N(u)$ be the $CF \times CF$ diagonal matrix whose diagonal blocks are $N_c(u) \cdot I$ $(c = 1, ..., C)$, then let $\tilde{F}(u)$ be the $CF \times 1$ supervector obtained by concatenating

$\tilde{F}_c(u)$ $(c = 1, ..., C)$, and similarly let $\tilde{S}(u)$ be the $CF \times CF$ diagonal matrix whose diagonal blocks are $\tilde{S}_c(u)$ $(c = 1, ..., C)$.

Another EM Algorithm

Given a set of utterances $U = \{u_1, ..., u_k\}$, we estimate our hyperparameters T and Σ through multiple iterations of the following EM algorithm:

1. Initialize m to be the supervector of means and, similarly, Σ to be the covariance matrices defined by our UBM θ_{UBM}. Pick a desired rank R for the Total Variability Matrix T[1], and initialize this $CF \times R$ matrix randomly.

2. The E-step: For each utterance u, calculate the posterior distribution of $w(u)$ using the current estimates of T and Σ.

3. The M-step: Update T and Σ by solving a set of linear equations in which the $w(u)$'s play the role of the explanatory variables.

4. Repeat steps 2 and 3 until the parameters converge, The convergence can be observed in the complete data log likelihood as given by

$$P_{T,\Sigma}(U) = \sum_{u \in U} \log P_{T,\Sigma}(u) \tag{3.23}$$

   The specifics of this function are a bit involved and have been omitted for clarity; more details can be found in [26].

The Posterior Distribution of w

For each utterance u, let $l(u)$ be the $R \times R$ matrix defined by

$$l(u) = I + T^* \Sigma^{-1} N(u) T \tag{3.24}$$

Then the posterior distribution of $w(u)$ conditioned on the acoustic observations of an utterance u is Gaussian with mean $l^{-1}(u) T^* \Sigma^{-1} \tilde{F}(u)$ and covariance matrix $l^{-1}(u)$.

This is the same as Proposition 1 of [26]; we have reproduced it here, along with the following proof for completeness.

[1]Depending on the amount of available training data and the desired application, this can range between 40 and 800. For NIST-based speaker recognition tasks, a rank of 600 achieved state-of-the-art results; for speaker diarization, best results were obtained using a rank of 400

Proof. In order to show that the posterior distribution of $w(u)$ is of the stated form, it is enough to show that

$$P_{T,\Sigma}(w|u) \propto \exp\left(-\frac{1}{2}(w-a(u))^* l(u)(w-a(u))\right)$$

where

$$a(u) = l^{-1}(u)T^*\Sigma^{-1}\tilde{F}(u)$$

Dropping the references to u, we can apply Bayes' Rule to obtain

$$P_{T,\Sigma}(w|u) \propto P_{T,\Sigma}(u|w)\cdot\mathcal{N}(w|0,I) = P_{T,\Sigma}(\{y_1,...,y_L\}|w)\cdot\mathcal{N}(w|0,I)$$

By letting $\bar{\mathbf{Y}}_t$ be a supervector of y_t repeated C times and $\bar{\gamma}_t$ be a $CF\times 1$ supervector where each UBM component $c\in\{1,...,C\}$ corresponds to $\gamma_t(c)$ repeated F times, we can write the first term as

$$\begin{aligned} P_{T,\Sigma}(\{y_1,...,y_L\}|w) &= \prod_{t=1}^{L} P_{T,\Sigma}(y_t|w) \\ &\propto \exp\left(-\frac{1}{2}\sum_t \bar{\gamma}_t(\bar{\mathbf{Y}}_t-(m+Tw))^*\Sigma^{-1}(\bar{\mathbf{Y}}_t-(m+Tw))\right) \end{aligned}$$

Then

$$\begin{aligned} P_{T,\Sigma}(w|u) &\propto P_{T,\Sigma}(\{y_1,...,y_L\}|w)\cdot\mathcal{N}(w|0,I) \\ &\propto \exp\left(-\frac{1}{2}\sum_t \bar{\gamma}_t(\bar{\mathbf{Y}}_t-(m+Tw))^*\Sigma^{-1}(\bar{\mathbf{Y}}_t-(m+Tw))\right)\cdot\exp\left(-\frac{1}{2}w^*w\right) \\ &= \exp\left(-\frac{1}{2}\sum_t \bar{\gamma}_t\left((\bar{\mathbf{Y}}_t-m)^*\Sigma^{-1}(\bar{\mathbf{Y}}_t-m) - 2w^*T^*\Sigma^{-1}(\bar{\mathbf{Y}}_t-m)\right.\right. \\ &\qquad \left.\left. +w^*T^*\Sigma^{-1}Tw\right) - \frac{1}{2}w^*w\right) \end{aligned}$$

Dropping irrelevant terms not dependent on w, we get

$$\begin{aligned} P_{T,\Sigma}(w|u) &\propto \exp\left(w^*T^*\Sigma^{-1}\sum_t \bar{\gamma}_t(\bar{\mathbf{Y}}_t - m) - \frac{1}{2}w^*T^*\Sigma^{-1}Tw\sum_t \bar{\gamma}_t - \frac{1}{2}w^*w \right) \\ &= \exp\left(w^*T^*\Sigma^{-1}\tilde{F}(u) - \frac{1}{2}w^*T^*\Sigma^{-1}N(u)Tw - \frac{1}{2}w^*w \right) \\ &= \exp\left(w^*T^*\Sigma^{-1}\tilde{F}(u) - \frac{1}{2}w^*(T^*\Sigma^{-1}N(u)T + I)w \right) \end{aligned}$$

And finally, a little bit of algebraic manipulation gives us

$$\begin{aligned} P_{T,\Sigma}(w|u) &\propto \exp\left(w^*T^*\Sigma^{-1}\tilde{F}(u) - \frac{1}{2}w^*(T^*\Sigma^{-1}N(u)T + I)w \right) \\ &= \exp\left(-\frac{1}{2}(w^*lw - 2w^*(l\cdot l^{-1})T^*\Sigma^{-1}\tilde{F} \right) \\ &\propto \exp\left(-\frac{1}{2}(w - l^{-1}T^*\Sigma^{-1}\tilde{F})^*l(w - l^{-1}T^*\Sigma^{-1}\tilde{F}) \right) \\ &= \exp\left(-\frac{1}{2}(w - a(u))^*l(u)(w - a(u)) \right) \end{aligned}$$

as desired. □

Maximum Likelihood Re-estimation

To reiterate the previous section, let $\mathbb{E}[\cdot]$ denote a posterior expectation. Then

$$\mathbb{E}[w(u)] = l^{-1}(u)T^*\Sigma^{-1}\tilde{F}(u) \tag{3.25}$$

$$\mathbb{E}[w(u)w^*(u)] = \mathbb{E}[w(u)]\cdot\mathbb{E}[w^*(u)] + l^{-1}(u) \tag{3.26}$$

where the posterior correlation matrix $\mathbb{E}[w(u)w^*(u)]$ follows from a well known fact in probability that, for two random variables A and B,

$$\mathrm{cov}(A,B) = \mathbb{E}[AB] - \mathbb{E}[A]\cdot\mathbb{E}[B]$$

The M-step of our EM algorithm entails accumulating the following statistics over the training set, where the posterior expectations and correlation matrices are calculated using the current estimates of m, T, Σ, and u ranges over all the given training utterances:

$$N_c = \sum_u N_c(u) \ (c = 1, ..., C) \tag{3.27}$$

$$\mathfrak{A}_c = \sum_u N_c(u) \cdot \mathbb{E}[w(u)w^*(u)] \ (c = 1, ..., C) \tag{3.28}$$

$$\mathfrak{B} = \sum_u \tilde{F}(u)\mathbb{E}[w^*(u)] \tag{3.29}$$

$$N = \sum_u N(u) \tag{3.30}$$

Now, to provide the algorithm: For each mixture component $c = 1, ..., C$ and for each feature dimension $f = 1, ..., F$, set $i = (c-1)F + f$; let τ_i denote the ith row of T and let $\mathfrak{b}_i$ denote the ith row of $\mathfrak{B}$. The T can be updated by solving the equations

$$\tau_i \mathfrak{A}_c = \mathfrak{b}_i \ (i = 1, ..., CF). \tag{3.31}$$

And lastly, the update formula for Σ is

$$\Sigma = N^{-1} \left(\sum_u \tilde{S}(u) - \text{diag}\left(\mathfrak{B}T^*\right) \right) \tag{3.32}$$

where we remove from Σ the variabilities that are already accounted for by T [25].

Minimum Divergence Re-estimation

Historically, the columns of T are analogous to eigenvoices, which correspond to high-dimensional representations of (somewhat) actual voices. On its own, maximum likelihood estimation produces a T whose eigenvalue-eigenvector structure is difficult to interpret. As a way to preserve the nature of this correspondence, another algorithm can be derived by using a divergence minimization approach to estimate the hyperparameters. This seems to converge much more rapidly than maximum likelihood estimation, since the only freedom it has is to rotate the eigenvectors in the total variability space and scale the eigenvalues. However, it has the property that it keeps the orientation of the total variability space fixed; thus it can only be used after maximum likelihood estimation has already been carried out [25].

Given initial estimates m_0 and T_0, the minimum divergence update formulas for

m and T are

$$m = m_0 + T_0\mu_w \tag{3.33}$$

$$T = T_0 R^*_{ww} \tag{3.34}$$

where

$$\mu_w = \frac{1}{\|U\|}\sum_u \mathbb{E}[w(u)] \tag{3.35}$$

and R_{ww} is an upper triangular matrix such that

$$R^*_{ww}R_{ww} = \frac{1}{\|U\|}\sum_u \mathbb{E}[w(u)w^*(u)] - \mu_w\mu^*_w \tag{3.36}$$

which can also be seen as the Cholesky decomposition of the expression above. $\|U\|$ is the number of training utterances, and the sums extend over all utterances in the training set. This update formula leaves the range of the covariance matrix TT^* unchanged, thus allowing only the freedom to rotate the eigenvectors of T and scale its corresponding eigenvalues (hence the minimum divergence) [25].

We have now gone over the details of the EM Algorithm used to train our parameters T and Σ, as well as the method that is used to estimate the posterior distribution of w. From here, we will use the posterior mean of w as our i-vector, a low-dimensional representation of the speaker in a given utterance.

3.3.2 Inter-session Compensation

One marked difference between the total variability representation and JFA is that total variability does not explicitly compensate for inter-session variability [27]. Once the data has been projected into the lower dimensional space, however, standard compensation techniques can still be applied in a straightforward and computationally efficient manner. Upon experimentation with a variety of different methods, it was found that the best performance in speaker recognition can be achieved with a combination of Linear Discriminant Analysis (LDA) followed by Within-Class Covariance Normalization (WCCN). The following paragraphs will briefly summarize these ideas; a more complete treatment can be found in [5].

In order to better discriminate between classes, LDA looks to define a new orthogonal basis (rotation) within the feature space. In this case, different speakers correspond to different classes, and a new basis is sought to simultaneously maximize the between-class variance (inter-speaker discrimination) and minimize the within-class variance (intra-speaker variability). We define these axes using a projection

matrix A composed of the eigenvectors corresponding to the highest eigenvalues of the general equation

$$\Sigma_b \nu = \lambda \Sigma_w \nu \tag{3.37}$$

where λ is the diagonal matrix of eigenvalues. The matrices Σ_b and Σ_w correspond to the between-class and within-class covariance matrices, respectively, and are calculated as follows:

$$\Sigma_b = \sum_{s=1}^{S} (w_s - \bar{w})(w_s - \bar{w})^t \tag{3.38}$$

$$\Sigma_w = \sum_{s=1}^{S} \frac{1}{n_s} \sum_{i=1}^{n_s} \left(w_s^{(i)} - \bar{w}_s\right)\left(w_s^{(i)} - \bar{w}_s\right)^t \tag{3.39}$$

where $\bar{w}_s = \frac{1}{n_s}\sum_{i=1}^{n_s} w_s^{(i)}$ is the mean of the total factor vectors $w_s^{(\cdot)}$ for each speaker s with n_s corresponding to the number of utterances for that speaker, and S is the total number of speakers.

The idea behind WCCN [28] is to scale the total variability space by a factor that is inversely proportional to an estimate of the within-class covariance matrix. This has the effect of de-emphasizing directions of high intra-speaker variability and thus makes for a more robust scoring operation. The within-class covariance matrix is estimated using the total factor vectors from a set of development speakers as

$$W = \frac{1}{S} \sum_{s=1}^{S} \frac{1}{n_s} \sum_{i=1}^{n_s} \left(A^t w_s^{(i)} - \tilde{w}_s\right)\left(A^t w_s^{(i)} - \tilde{w}_s\right)^t \tag{3.40}$$

where A is the LDA projection matrix as described previously, $\tilde{w}_s = \frac{1}{n_s}\sum_{i=1}^{n_s} A^t w_s^{(i)}$ is the mean of the LDA-projected total factor vectors $w_s^{(\cdot)}$ for each development speaker s, n_s corresponds to the number of utterances for the respective speaker, and S is the total number of development speakers. We use the Cholesky decomposition of the inverted matrix, $W^{-1} = BB^t$, whose application can be viewed as scaling the total variability space by B. The result of applying both LDA and WCCN is a new vector w', denoted

$$w' = B^t A^t w \tag{3.41}$$

where w is extracted from (3.25).

To reiterate this entire process at a higher level: we have discussed how to take an incoming stream of acoustic features $u = \{y_1, y_2, ..., y_L\}$ (e.g. MFCCs) and extract an i-vector w for the utterance using (3.25), which requires only the Baum-Welch statistics of zeroth (3.18) and first (3.21) order, as well as a previously estimated Total Variability matrix T and residual covariance Σ. We then perform inter-session

compensation on w based on (3.41) to obtain w'. The LDA projection matrix A and WCCN scaling matrix B are estimated using the i-vectors extracted from utterances corresponding to a set of development speakers that will not be seen during testing. Once w' has been obtained, we move on to the scoring process, which is detailed in the following section.

The following figures illustrate the effect of these inter-session compensation techniques. Figure 3-3 shows a plot of using only LDA, then adding WCCN, and finally including length normalization on a set of two-dimensional i-vectors. Notice how the incorporation of WCCN (middle plot) shrinks the space noticeably and really de-emphasizes the directions of high intra-speaker variability. We can also see how length normalization projects each speaker onto its own region on the unit circle, which motivates the notion of cosine similarity scoring to be discussed.

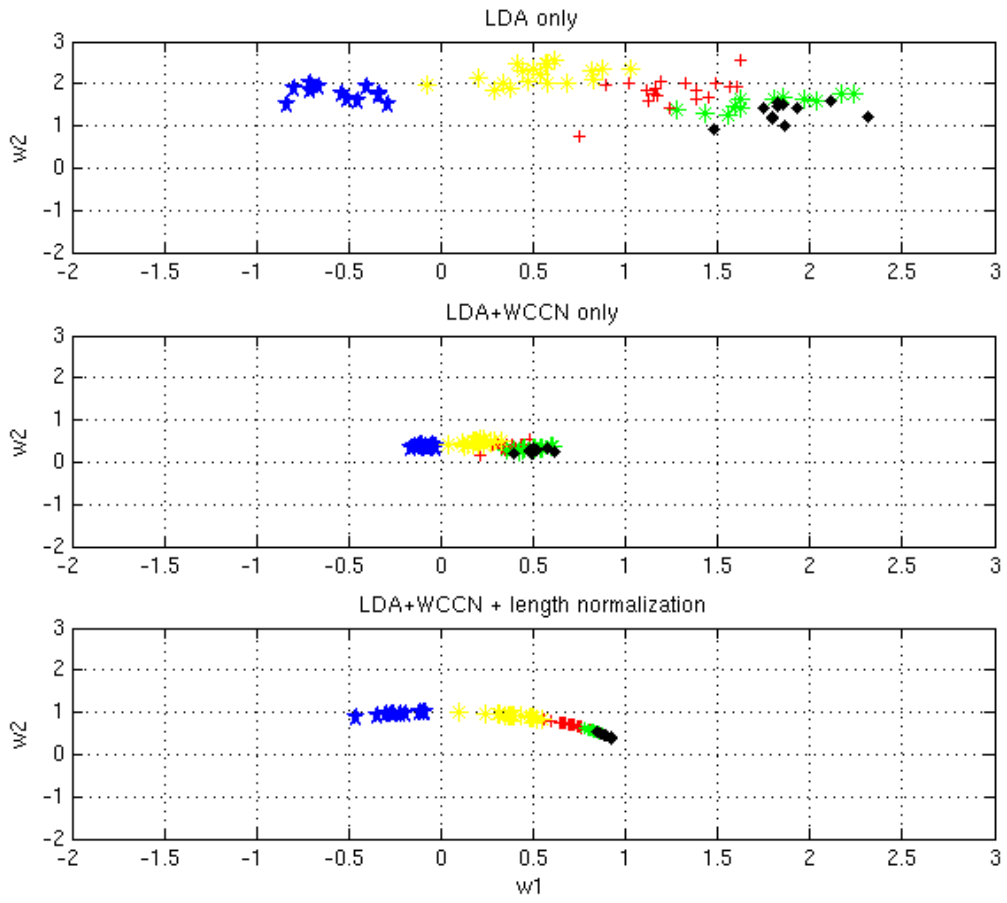

Figure 3-3: *Plots displaying the effect of LDA, WCCN, and length normalization as applied to various i-vectors. Different colors/markers correspond to different female speakers. Figure taken and adapted from [5].*

Figures 3-4 and 3-5 show, using graph embedding and visualization [7], the contrast between applying (Figure 3-5) and forgoing inter-session compensation (Figure 3-4). In both figures, different colors represent different speakers, and points that belong in the same group together will either appear next to each other or be connected by an edge.

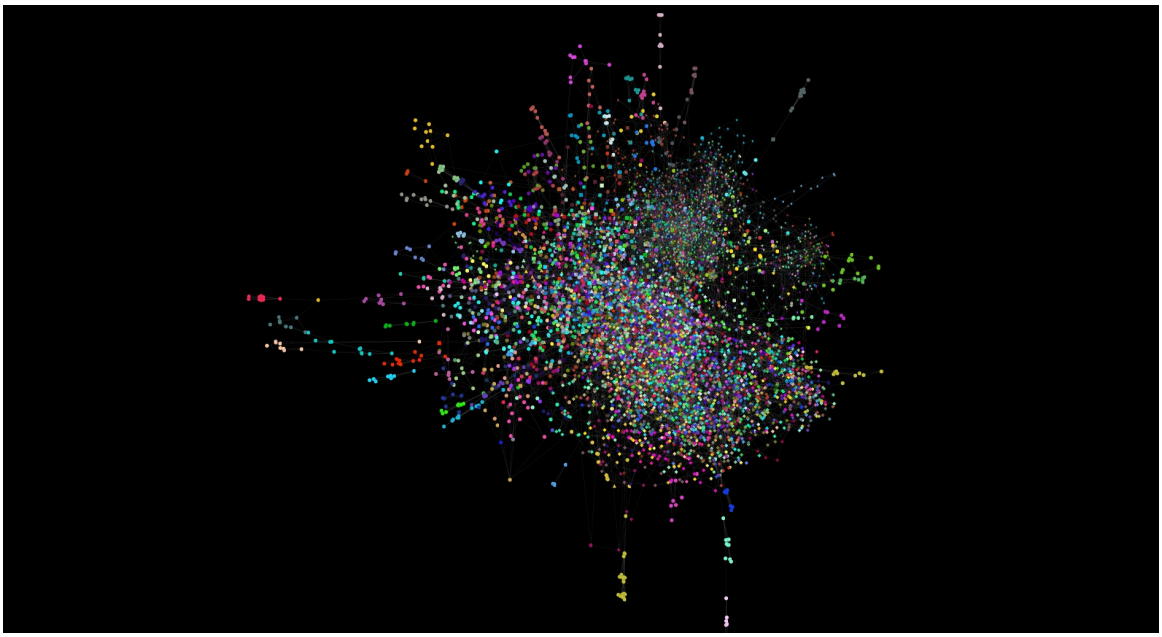

Figure 3-4: *Graph embedding-based visualization of male speaker i-vectors without inter-session compensation. Figure taken from work done on [6, 7]; reproduced with permission from the authors.*

3.3.3 Cosine Similarity Scoring

The benefit of needing to estimate only one latent random variable is that this vector w is of low dimension and is, in theory, the full and final representation of a speaker's identity as ascertained from the utterance. As such, factor analysis can be used as a method for front end feature extraction, potentially eclipsing the need to continue using the log-likelihood ratio (LLR) scoring function discussed for the GMM-UBM and JFA approaches. Instead, the simple cosine similarity has recently been applied successfully in the total variability space to compare two i-vectors for making a speaker detection decision [5]. Given two i-vectors (which may or may not have undergone inter-session compensation) - one from a previously enrolled target speaker w'_{target} and

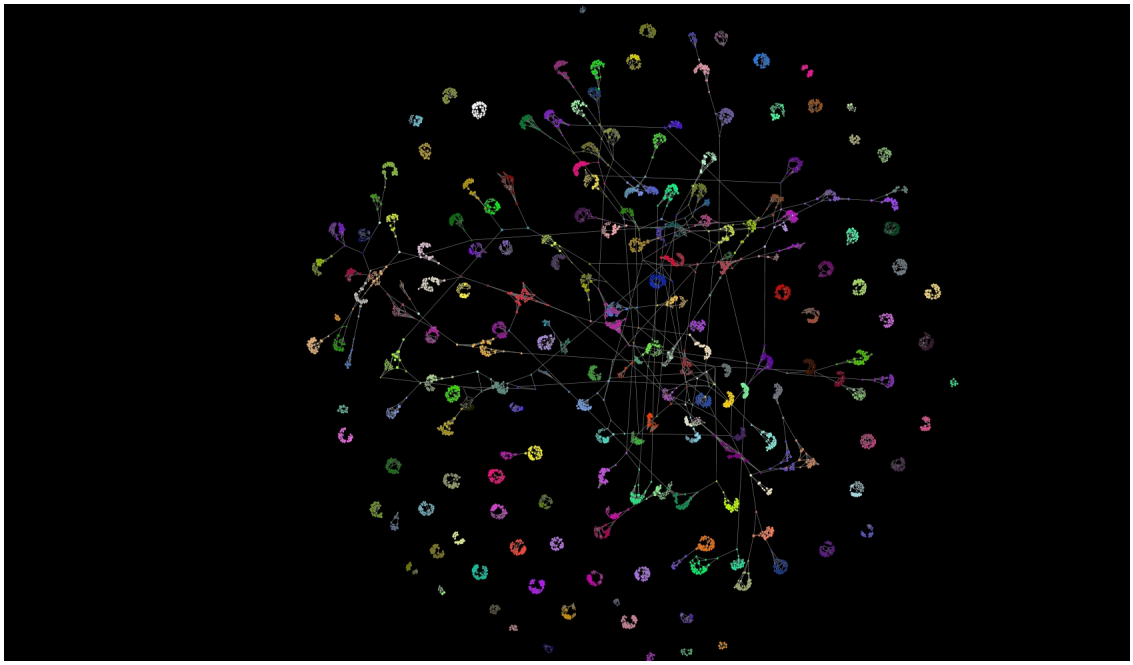

Figure 3-5: *Graph embedding-based visualization of male speaker i-vectors after inter-session compensation. Points grouped close together represent the same speaker, while the visible edges between clouds also represent close proximity between two points (via the cosine distance metric). Figure taken from work done on [6, 7]; reproduced with permission from the authors.*

another from an unknown test utterance w'_{test} - the cosine similarity score is given as

$$\text{score}\left(w'_{\text{target}}, w'_{\text{test}}\right) = \frac{\left(w'_{\text{target}}\right)^t \left(w'_{\text{test}}\right)}{\|w'_{\text{target}}\| \|w'_{\text{test}}\|} \gtrless \theta \tag{3.42}$$

where θ is the hypothesis accept/reject decision threshold. This scoring function is considerably less complex than the LLR scoring operations used in JFA [29].

3.4 Discussion

To provide some perspective, a typical UBM for the speaker recognition task will have on the order of $C \approx 1000$ mixture components, while the dimensionality of our features F will range from 13 to 60. Taking $F \approx 20$ makes M a 20,000-dimensional vector. However, we typically constrain the rank of T to be on the order of hundreds (i.e. $\texttt{rank}(T) \approx 400$). Thanks to this significant reduction in dimensionality, it is much easier computationally to work in the "Total Variability Space" defined by T.

In this chapter, we have provided a historical outline of the theory behind factor

analysis for speaker recognition. All math aside, however, what is most important to take away is an intuitive understanding of how i-vectors came to be and what they represent: given an utterance of any length, we can extract a single low-dimensional Total Factors vector that contains all the neccessary information regarding the identity of the speaker in that utterance. It is also worth noting that the given utterance should contain speech from only one speaker.

The Total Variability approach has achieved state-of-the art performance in speaker recognition [5]; in particular, it has demonstrated the ability to model speakers given rather short segments of speech ($\sim$ 10sec) [27]. As such, the rest of this thesis will describe our attempts to adapt this approach to speaker diarization.

Chapter 4

Exploiting Intra-Conversation Variability

The Total Variability approach has achieved state of the art performance in the task of speaker verification [5]; it is therefore natural to try to adapt these methods for the problem of speaker diarization. The extraction of i-vectors, however, assumes that a given utterance contains speech from only one speaker, whereas the point of diarization is to mark where different speakers spoke in a conversation. Thus, as was mentioned previously, the problem of speaker diarization consists of two parts: segmentation and clustering. For the latter, we will take advantage of Total Variability and its effectiveness in speaker modeling. This chapter will describe the components of our initial diarization system built for the two-speaker summed-channel telephone data of the NIST 2008 SRE.

4.1 Shortcomings

We begin by recognizing the shortcomings of standard (speaker verification-based) inter-session compensation techniques when applied to speaker diarization. Assuming we have some initial segmentation in place, we can extract an i-vector for each segment. Following the standard recipe for speaker recognition tasks as described in the previous chapter, we can apply LDA, WCCN, and length-normalization as in (3.41). Upon doing so, however, the diarization system produced results that were far worse than expected. Interestingly enough, this outcome was consistent with previous work that applied joint factor analysis to speaker diarization; the use of eigenchannels was ineffective in [12].

Figure 4-1 plots three histograms of cosine similarity scores for i-vectors corre-

sponding to the same speaker (within-speaker scores) and those corresponding to different speakers (between-speaker scores). What we expected to see in the "All Scores" histogram was a distinctly bimodal distribution, with the within-speaker scores having a mean/mode closer to 1 while the between-speaker scores have a mean/mode somewhere around 0. However, what we see instead are two distributions that are almost indistinguishable from each other.

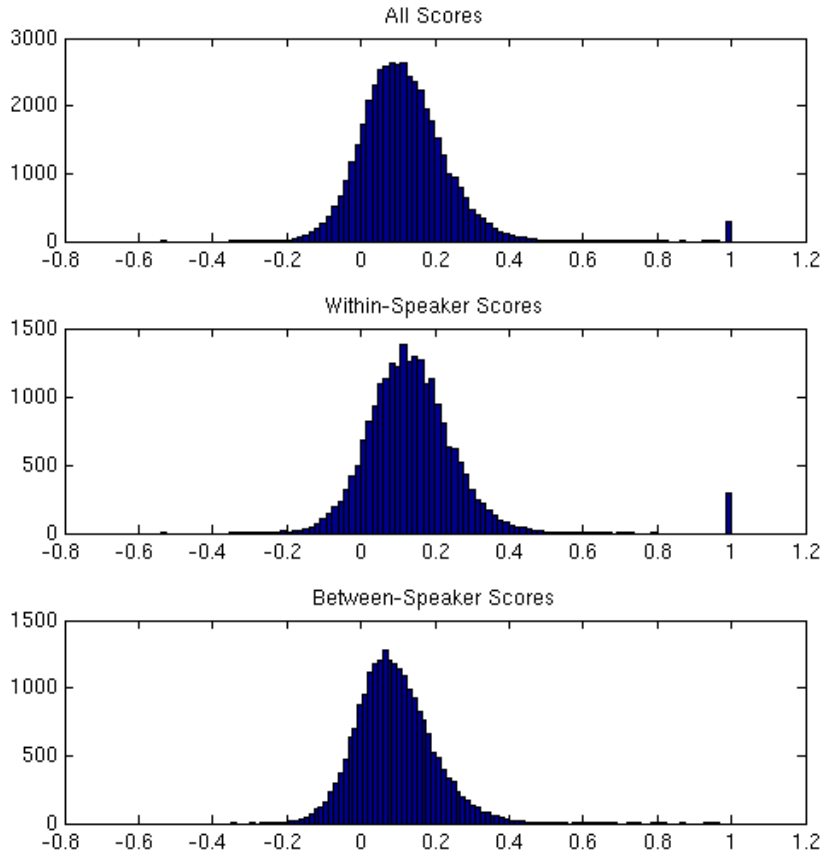

Figure 4-1: *Histograms of the cosine similarity scores between i-vectors from a randomly selected two-speaker telephone conversation. "Within-Speaker Scores" correspond to the scores obtained between two i-vectors of the same speaker, "Between-Speaker Scores" correspond to scores obtained between two i-vectors of different speakers, while "All Scores" correspond to the union of the two.*

This observation gave way to the realization that compensating for inter-session variability may actually be unnecessary in the problem of telephone diarization. Because we are working on summed-channel telephone conversations, there is really no *inter*-session. In fact, the ability to detect some change in the channel characteristic

of a summed-channel telephone recording could actually help us to determine speaker changes. Rather, what we really care about are *intra*-session (or intra-conversation) variabilities within each audio file. Such insight paved the way for the rest of this work.

4.2 The Basic System

Having observed the ineffectiveness of standard inter-session compensation techniques on short-segment i-vectors, the most straightforward baseline approach is to cluster our set of i-vectors without any sort of pre-processing (i.e. LDA, WCCN, etc.). As such, we build an initial system consisting of the following components.

4.2.1 Segmentation

We obtain an initial segmentation on the summed-channel telephone data using a Harmonicity and Modulation Frequency-based Voice Activity Detector (VAD) described in [30]. Its output gives us the start/stop times for segments that are classified as speech. Note also that it is possible for these segments to contain speech from more than one speaker, such as from overlapped speech. Over the entire test set, the average length of these segments is 1.09s with a standard deviation of 0.648s. Though the segment lengths range widely between 0.03s and 11.31s, we chose to use this VAD without any additional refinements.

4.2.2 First Pass Clustering

To perform clustering on the set of i-vectors extracted for each segment, we use K-means ($K = 2$) clustering based on the cosine distance. The iterative nature of this algorithm allows it to self-correct poor initializations, whereas other methods such as the bottom-up approach of agglomerative hierarchical clustering used in the baseline system of [12] uses only one iteration to make hard decisions.

4.2.3 Re-segmentation

After an initial clustering, we refine our initial segmentation boundaries using a Viterbi re-segmentation and Baum-Welch soft speaker clustering algorithm detailed in [12]. At the acoustic feature level, this stage initializes a 32-mixture GMM for each of the clusters (Speaker A, Speaker B, and non-speech N) defined by the First Pass Clustering. Posterior probabilities for each cluster are then calculated given

each feature vector x_t (i.e. $P(A|x_t), P(B|x_t), P(N|x_t)$) and pooled across the entire conversation, providing a set of Baum-Welch statistics from which we can re-estimate each respective speaker's GMM. As a way to stabilize this unsupervised procedure, the non-speech GMM is never retrained. In the Viterbi stage, each frame is assigned to the speaker/non-speech model with the highest posterior probability. This algorithm runs until convergence but is capped at 20 Viterbi iterations, each of which involves 5 iterations of Baum-Welch re-estimation [12].

4.2.4 Second Pass Refinements

We further refine the diarization results of the Re-segmentation stage by extracting a single i-vector for each respective speaker using the (newly-defined) re-segmentation assignments. That is, for all of the segments attributed to speaker A, we pool the respective Baum-Welch statistics together to extract w_{A}, and follow a similar process for speaker B to obtain w_{B}. Each segment i-vector w_i, which is also newly extracted from the Re-segmentation assignments, is then reassigned to the speaker whose i-vector is closer in cosine similarity. In the next iteration, we use the most recent reassignments to pool the Baum-Welch statistics together to re-extract w_{A} and w_{B} once again, then reassign each w_i accordingly. This process continues until convergence - when the segment assignments no longer change. We can view this as another pass of K-means clustering, where the "means" are computed according to the process of i-vector estimation detailed in [5].

4.3 Initial Experiments

We used a gender-independent UBM of 1024 Gaussians built solely on 20-dimensional MFCC feature vectors without derivatives to train a gender-independent Total Variability matrix of rank 400. This configuration was chosen to be somewhat consistent with that of the Variational Bayesian (VB) system described in [12], though we will also report later on the results of using Total Variability matrices of different rank. As mentioned in Section 2.4, the training of all our model parameters was done using data from the Switchboard and Mixer corpora.

4.3.1 Evaluation Protocol

Set up by NIST, the Diarization Error Rate (DER) is the primary performance measure for the evaluation of diarization systems and is given as the time-weighted sum of the following three error types: *Miss* (M) - classifying speech as non-speech, *False*

Alarm (FA) - classifying non-speech as speech, and *Confusion* (C) - confusing one speaker's speech as from another [31]. Figure 4-2 depicts each type of error. In evaluating DER's, we first obtain a reference by applying a speech activity detector to each separate channel of the telephone conversation. Then, following the conventions for evaluating diarization performance, the evaluation code ignores intervals containing overlapped speech as well as errors of less than 250ms in the locations of segment boundaries. Although overlapped speech intervals do not count in evaluating DER's, the diarization systems do have to contend with overlapped speech in performing the speaker segmentation and clustering.

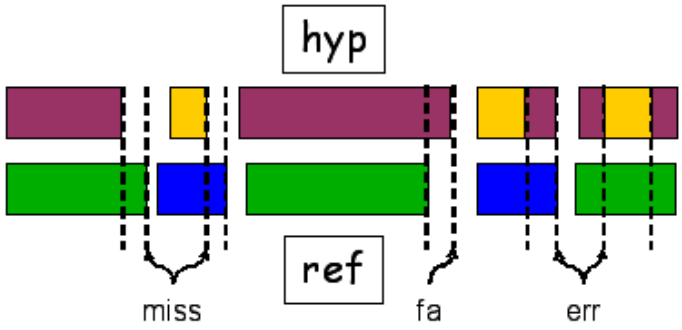

Figure 4-2: *Depiction of each type of Diarization Error. "Hyp" represents the hypothesized diarization output of some system; "Ref" represents the reference segmentation against which the hypothesis is evaluated.*

It is well known that diarization systems typically exhibit wide variations in performance across test files. Thus, in reporting our results for a given system, we will also show the standard deviation of the DER - calculated over all test files - in addition to the mean DER.

We should realize that the Miss and False Alarm errors are solely caused by a mismatch between the reference speech activity detector and the diarization system's VAD and Re-segmentation output. A more straightforward metric for the effectiveness of our speaker modeling and clustering methods is in the measurement of Confusion error. In order to focus solely on this type of error, the results reported in [12] were based on the use of reference boundaries as the initial speech/non-speech segmentation, thus driving both miss and false alarm error rates to zero. On our end, we will first report on the detailed results achieved using our own VAD to provide an initial segmentation. Then, for proper comparison, we will also report on a final experiment done using the reference boundaries as the initial speech/non-speech segmentation.

4.3.2 Results

Following the work in [12], we evaluate the performance of our diarization system on the summed-channel telephone data from the NIST 2008 SRE. As covered in Section 2.4, this consists of 2215 two-speaker telephone conversations, each approximately five minutes in length ($\approx$ 200 total hours). Table 4.1 shows the results obtained from our initial system at each stage described in Section 4.2.

		Error Breakdown			
	DER (%)	M	FA	C	σ (%)
First Pass	11.4	7.7	2.0	1.7	8.0
Re-segmentation	5.6	0.8	2.4	2.4	8.6
Second Pass	4.7	0.8	2.4	1.5	7.8

Table 4.1: *Results obtained after each stage of the diarization procedure described so far. The configuration for the First Pass Clustering uses 400-dimensional i-vectors.*

From its low Speaker Confusion error, we can immediately see that the First Pass Clustering, done on raw i-vectors without any sort of pre-processing, is already very effective in discriminating between speakers. Furthermore, the helpfulness of the Re-segmentation step in cleaning up the missed speech from the initial segmentation is readily apparent. At the same time, the observed increase in Speaker Confusion error and its standard deviation after Re-segmentation demonstrates the limitations in the ability of acoustic features to model speaker characteristics. Nevertheless, the Second Pass Refinement stage cleans up the Speaker Confusion error, leaving us with a very competitive initial result.

Table 4.2 compares the performance of our initial system to that of the systems described in [12]. The BIC-based system served as a baseline for both the Stream-based and the VB-based FA work. All of those systems were initialized using the reference speech detection boundaries; thus, they incurred no Miss (M) or False Alarm (FA) error, and all of their error is attributed to Speaker Confusion (C). For a valid comparison, we report the results of our initial system using the reference boundaries as an initial segmentation, denoted "Ref VAD." The use of these reference speech/nonspeech boundaries, however, does not necessarily imply that a speaker change has occurred between two consecutive i-vectors, nor does it preclude the possibility of a segment containing speech from more than one speaker. Rather, these segments are defined purely by the speech/nonspeech boundaries as indicated by the reference segmentation. Lastly, we report the results obtained using our "Own VAD" as described in 4.2.1.

We can see that our "Ref VAD" system performs slightly better on average than

	Speaker Confusion (%)	σ_C (%)
BIC-based Baseline	3.5	8.0
Stream-based FA	4.6	8.8
VB-based FA	1.0	3.5
Ref VAD + TV400	1.4	5.2
Own VAD + TV400	1.5	4.8

Table 4.2: *Comparison of diarization results on the NIST SRE 2008 Summed-Channel Telephone Data.* (BIC - Bayesian Information Criterion; FA - Factor Analysis; VB - Variational Bayes; VAD - Voice Activity Detector; TV - Total Variability)

our "Own VAD" system as a result of a mismatched initial segmentation. At the same time, the error standard deviation of the "Ref VAD" system is a little higher than that of our "Own VAD," which suggests that the experimental performance of the "Ref VAD" system may be a bit less consistent than its counterpart. At the end of the day, however, the difference in performance between our two systems is mostly invariant to our choice of initial segmentation. That said, we would like to come closer to attaining the same state-of-the-art result of 1.0% Speaker Confusion as in [12]. To do so, we go back to the idea that i-vectors were designed to contain primarily the information necessary to discriminate between speakers.

4.4 Directions of Maximum Variability

As before, we assume that we are given some initial segmentation and thus can extract an i-vector for each segment without doing any inter-session compensation. We further assume that there are exactly two speakers in the given conversation. Of course, it is not known *a priori* where our two respective speakers lie in the Total Variability space, but because i-vectors were designed to contain primarily speaker-specific information, the most prominent source of variability between these i-vectors ought to be attributed to differences between the speakers' voices.

We can find the directions of maximum variability within our set of i-vectors in the Total Variability space by using Principal Component Analysis (PCA). Though it can be derived in a number of different ways [32], PCA can be seen as a method of rotating the coordinate axes of our Total Variability space such that our data (i-vectors) can be expressed in the fewest dimensions possible. As a result of PCA, the mean of the data is centered at the origin, the first principal component will lie along the direction of maximal variance in the dataset, the second principal component will lie in the direction of maximal variance that is orthogonal to the first component, and so on. Figure 4-3 shows the first two principal components of a set of Total Factors

extracted from a male/female conversation[1]. The red triangles correspond to i-vectors extracted from male speaker segments, while the blue circles correspond to i-vectors from female speaker segments. The plot also includes, in black `x`'s, the i-vectors corresponding to overlapped speech segments. To be sure, the PCA projection was calculated on all i-vectors including these overlapped speech segments, as we have not yet explored ways to distinguish between overlapped and non-overlapped speech. However, this would be a fruitful topic for future work.

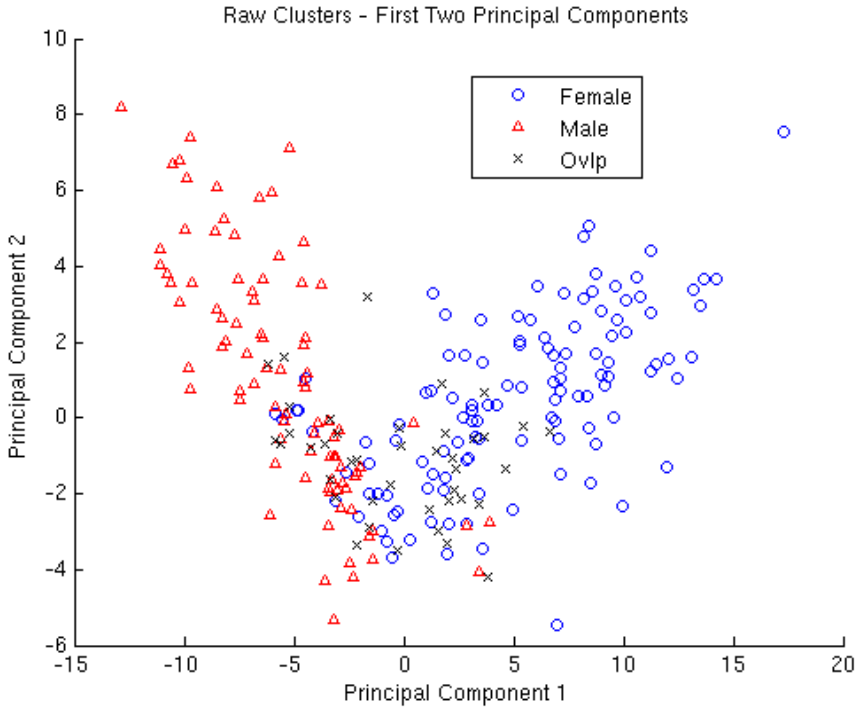

Figure 4-3: *Plot of the first two dimensions (principal components) of PCA-projected speaker i-vectors. The triangles in red represent i-vectors of a male speaker, while the blue circles represent i-vectors of a female speaker in the same conversation. The black* `x`*'s correspond to i-vectors representing overlapped speech.*

Though this is a visualization of only the first two principal components from an initial i-vector dimension of 400, we can already see a distinct separation between the sets of total factor vectors corresponding to different speakers. Furthermore, it can be observed that the separation between the two clusters is primarily directional. This is because each i-vector has a standard normal prior distribution, and it is apparent from Equation (3.25) that the magnitude of an i-vector is proportional to the number of acoustic frames used to calculate its estimate. Thus, the information relevant for our purposes may be contained not in the magnitude of the i-vector, but in its relative

[1]Plots for same-gender (i.e. male/male or female/female) conversations are similar.

orientation. Figure 4-4 shows a length-normalized version of the first two principal components for the same two speakers seen in Figure 4-3. Notice how the majority of each cluster can be found in distinctly different regions of the unit circle. This further motivates the use of the cosine similarity as a metric for comparing i-vectors.

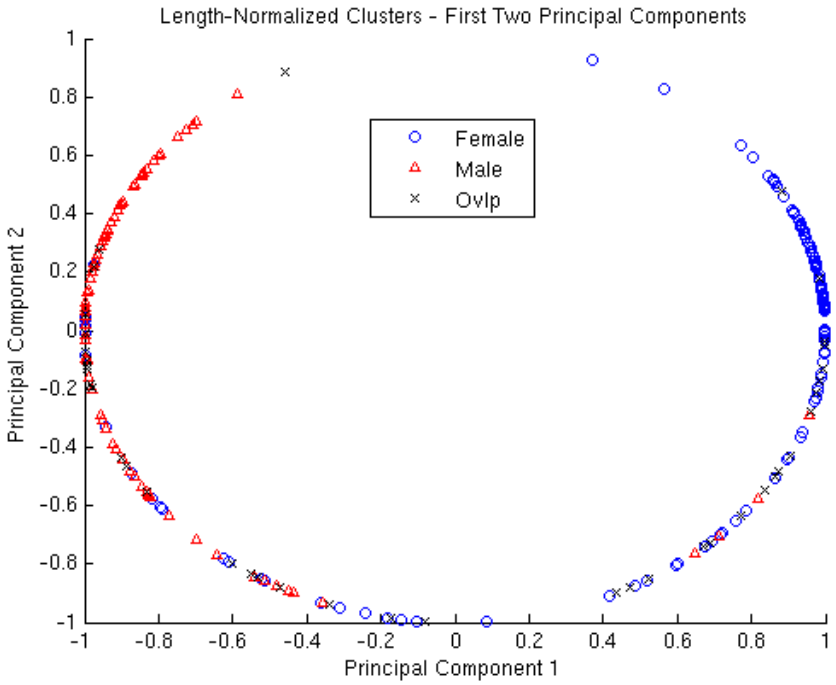

Figure 4-4: *Plot of the length-normalized speaker i-vectors after applying a two dimensional PCA-projection across the entire conversation. Notice also the random scatter of the black $\boldsymbol{x}$'s corresponding to overlapped speech segments.*

4.4.1 PCA-based Dimensionality Reduction

The amount of speaker separation that is apparent in just the first two principal dimensions raises the question of whether or not all 400 i-vector dimensions are actually necessary to perform our initial clustering task. It may be the case that the higher dimensions (i.e. less principal components) simply add noise to our discrimination. To that end, we ran experiments to see the effect of dimensionality reduction on diarization performance in the First Pass Clustering stage. Figure 4-5 details these results, and it is evident that we can maintain a high level of performance despite keeping only 5 of the original 400 i-vector dimensions.

It is important to realize, however, that 5 is not some magical dimension that is optimal for diarization performance. Rather, the act of reducing to 5 dimensions merely optimizes, on the average, the case of two-speaker diarization in which we

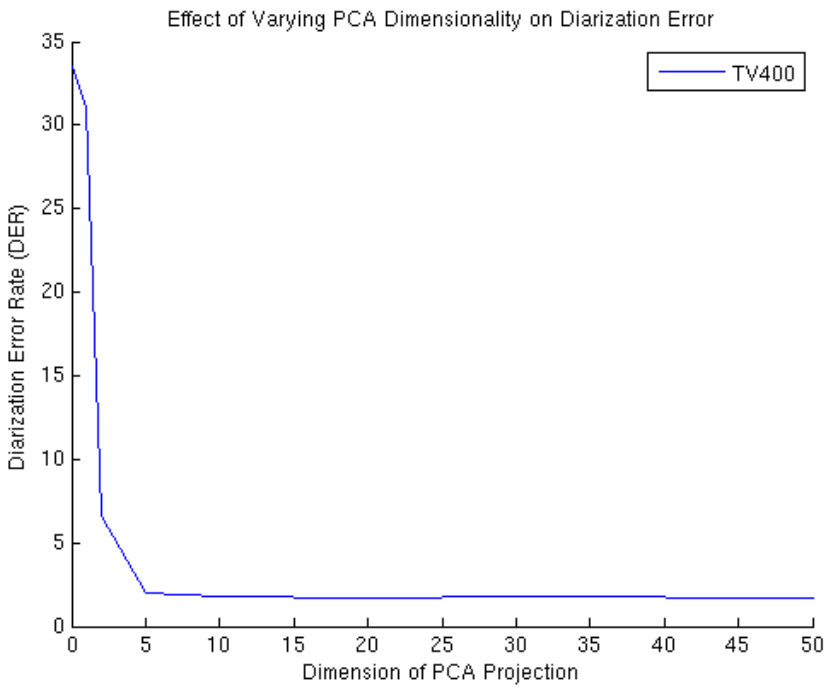

Figure 4-5: *Plot of First Pass Speaker Confusion error as a function of PCA-projected i-vector dimension. The original i-vector dimension is 400. Note that a 0-dimensional projection is the base case in which everything is assigned to a single speaker.*

begin with 400 i-vector dimensions. Indeed, because PCA is performed on the set of i-vectors on a *per utterance* basis, it is likely that the number of PCA dimensions that is optimal for discriminating between (possibly more than 2) speakers in a given utterance is different from the number of dimensions that is optimal for another utterance. Furthermore, the best performing dimension is not invariant to choice of i-vector dimension (i.e. rank of the Total Variability space). Thus, for a more standardized comparison across all fronts, it makes more sense to let the data decide the number of PCA dimensions to keep. We can instead specify some sort of parameter that regulates the amount of information that we would like for our dimensionality reduction to preserve. In particular, let us specify a certain percentage of eigenvalue mass and use the principal directions corresponding to the n largest eigenvalues such that

$$\min_n \frac{\sum_{i=1}^{n} \lambda_i}{\sum_{j=1}^{D} \lambda_j} \geq p \tag{4.1}$$

where we assume that our set of eigenvalues $\{\lambda_i\}$ is indexed in decreasing order and D is the initial i-vector dimension. For some additional insight, Table 4.3 provides some statistics regarding the number of dimensions used for different values of p given

an initial i-vector dimension of $D = 400$.

Pct. Eig. Mass (p)	Avg Dim (n)	Min n	Max n
30%	10.3	5	10
50%	25.5	16	33
80%	70.1	52	84

Table 4.3: *Comparison of the number of PCA-dimensions needed for different proportions of eigenvalue mass. These statistics were computed over 200 randomly selected test files from the NIST 2008 SRE.*

To demonstrate the robustness of this technique for dimensionality reduction across different initial i-vector dimensions, Figure 4-6 plots diarization performance as a function of the chosen proportion of eigenvalue mass for initial i-vector dimensions of 100 and 400. Ultimately, it was decided that $p = 0.5$ (i.e. 50% eigenvalue mass), which still gives good results despite a significant reduction in dimensionality, would be used in our subsequent experiments.

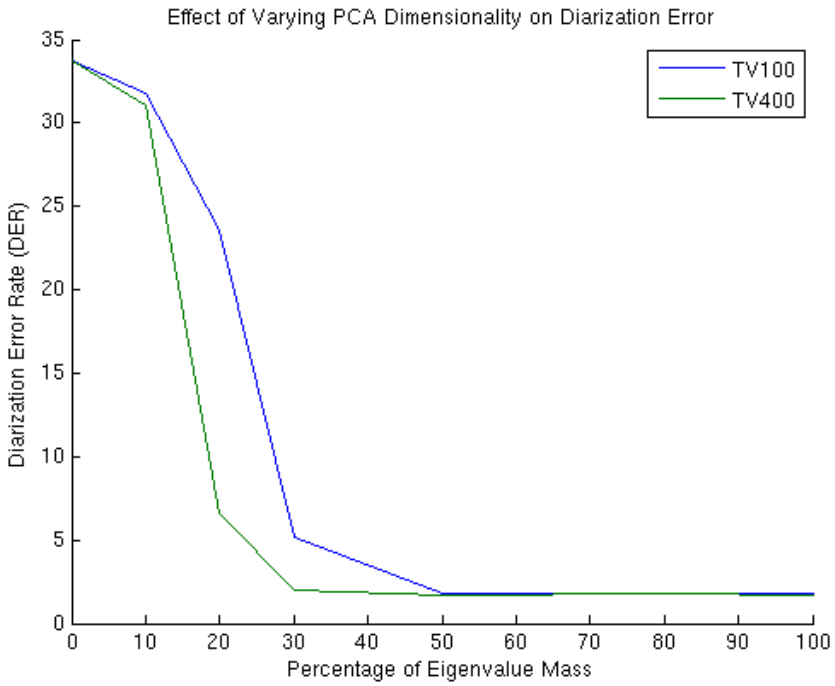

Figure 4-6: *Plot of Speaker Confusion error as a function of eigenvalue mass percentage used in PCA-based Dimensionality Reduction. Note that 0% eigenvalue mass also corresponds to the base case, in which everything is assigned to a single speaker.*

4.4.2 Eigenvalue-Weighted Scoring

We can further emphasize the importance of the PCA directions with the most variability (i.e. largest eigenvalues) by introducing the following weighted modification to our cosine similarity score

$$\text{score}\,(w_1', w_2') = \frac{(w_1')^t\,\Lambda\,(w_2')}{\|\Lambda^{\frac{1}{2}} w_1'\| \cdot \|\Lambda^{\frac{1}{2}} w_2'\|} \tag{4.2}$$

where w_i' is the PCA-projected i-vector and Λ is the corresponding diagonal matrix of the eigenvalues. Additionally scaling our PCA-projected i-vector components by the square root of the eigenvalues $\Lambda^{\frac{1}{2}}$ gives us added emphasis on the directions of higher variability (i.e. the "most" principal components). The histograms in Figure 4-7 show the stark contrast between using and not using eigenvalue-weighted scaling to compare speaker i-vectors. In particular, notice how incorporating eigenvalue-weighted scaling helps to better separate the distributions of within-speaker scores and between-speaker scores. In the case of a male/female conversation, for example, it is likely that the first principal component corresponds somewhat to the gender differentiation; thus, it makes sense to have this component carry the most weight in our scoring function. Though PCA naturally gives more scoring weight to the larger principal components, our experiments showed (along with Figure 4-7) that increasing this effect artificially had a positive impact on performance.

4.4.3 Experiment Results

The diagram in Figure 4-8 depicts a summary of our system described thus far. We evaluate this new configuration once again on the summed-channel telephone data from the NIST2008 SRE. Table 4.4 shows the results obtained from our system at each stage described in Section 4.2.

		Error Breakdown			
	DER (%)	M	FA	C	σ (%)
First Pass	13.8	7.7	2.0	4.0	9.6
Re-segmentation	6.2	0.8	2.4	2.9	8.6
Second Pass	4.7	0.8	2.4	1.5	7.0

Table 4.4: *Results obtained after each stage of the diarization procedure. The configuration for the First Pass Clustering uses 400-dimensional i-vectors as input to a data-centered PCA-projection involving 50% of the eigenvalue mass.*

In comparison to our initial system, the Miss and False Alarm errors are the same at each stage due to the nature of our initial segmentation and Re-segmentation.

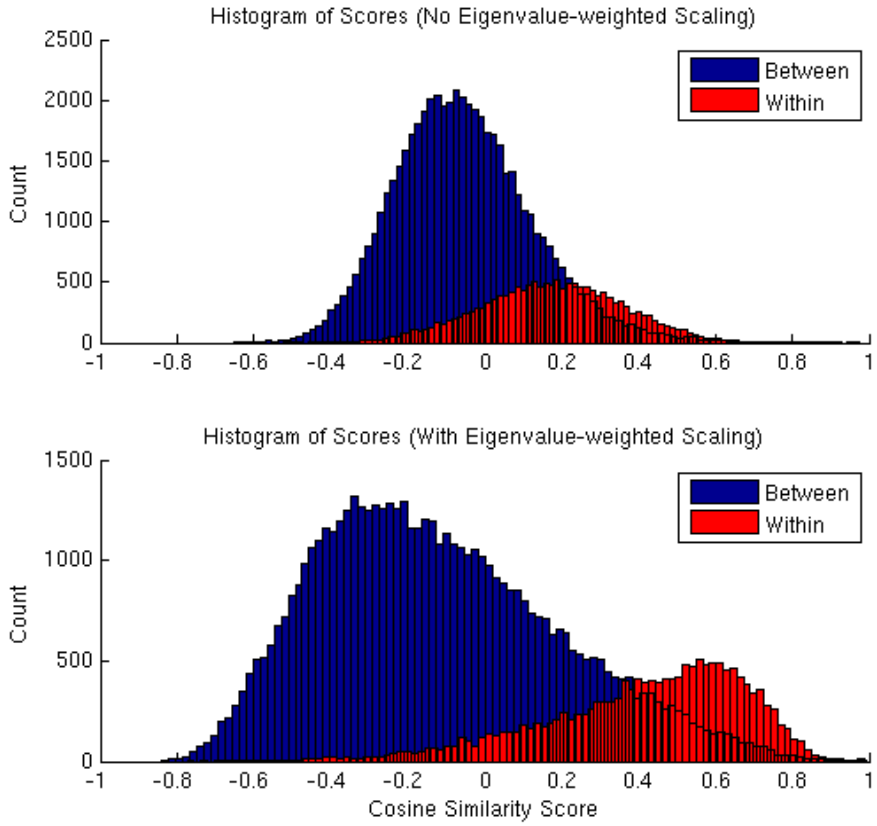

Figure 4-7: *Contrasting histograms between not using (top) and using (bottom) eigenvalue-weighted scaling/scoring to compare within- and between- speaker i-vectors.*

Thus, the differences to note are in Speaker Confusion. Ultimately, we arrive at the same Diarization Error Rate as in our initial system; how we got there is a different matter. Despite the promise of our preceding discussion, we can see that the initial approach actually does much better in the First Pass (1.7% Confusion) than our current approach (4.0%). As a result, a heavy burden was placed on the Second Pass Refinement stage to clean up the Speaker Confusion.

4.4.4 Explaining First-Pass Discrepancies

In an effort to explain the drastic discrepancy in First Pass Clustering results between our initial approach and the current approach involving PCA, we should take a closer look at the details. In particular, the standard procedure for PCA is to center the data such that its mean is at the origin. However, though Figure 4-4 provides a very convincing example of the potential effectiveness of directional clustering via the cosine distance, we should realize that applying the cosine similarity metric on

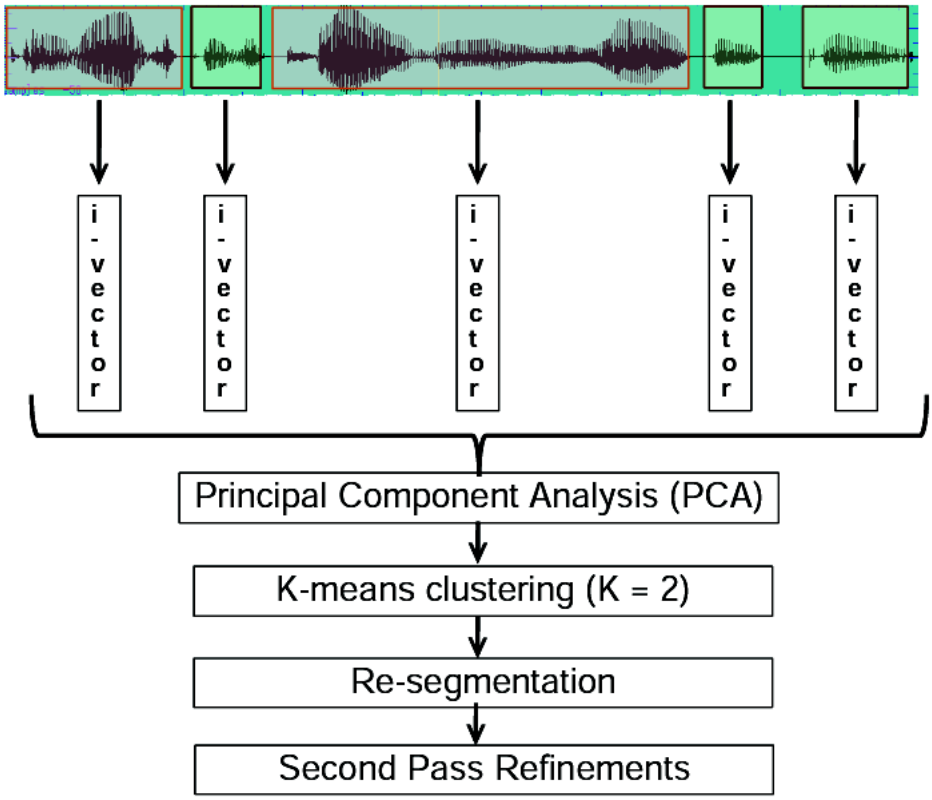

Figure 4-8: *Diagram of our proposed diarization system. Each step is explained in Section 4.2.*

data-centered PCA-projected i-vectors is no longer the same as the cosine similarity applied to i-vectors for speaker recognition.

We know from before that all i-vectors have a zero mean, identity covariance prior. Indeed, if we pooled together many i-vectors from a variety of different utterances, the global mean of this diversified dataset would indeed have zero mean. However, we perform PCA on a *per utterance* basis, and since i-vectors model both speaker *and* channel variabilities, we cannot assume that a set of i-vectors extracted from a given utterance has zero mean. Thus, in the process of centering our data, we introduce a translational bias which corrupts the original nature of our angular cosine similarity metric. This potentially explains the discrepancy we see in the First Pass Clustering results between our initial (no PCA) approach and the current data-centered PCA approach.

4.4.5 An Alternate Flavor of PCA

Alternatively, we could consider a flavor of PCA in which we do not center our data at the origin. That is, we can compute the eigenvectors and eigenvalues associated with the directions of maximum variance under the *a priori* assumption that the mean of our data is zero, but without regard to whether or not it actually is. This is not so outrageous; since i-vectors are extracted under a standard normal prior, the global mean of an utterance's i-vectors is unlikely to be far from the origin.

Calculated from the same conversation as Figures 4-3 and 4-4, Figure 4-9 shows the first two principal components that result from this "rotation-only" flavor of PCA. Subsequently, Figure 4-10 shows the length-normalized plot of the clusters. We can see that the the first direction of maximum variability in Figure 4-9 using rotation-only PCA does not quite discriminate between clusters as well as the first principal component does in Figure 4-3 using data-centered PCA. Furthermore, while rotation-only PCA still manages to separate the two clusters onto different regions of the unit circle (Figure 4-10), the effect is not quite as pronounced as in Figure 4-4, where data-centered PCA splits our respective speakers onto opposite regions of the unit circle, making for more obvious clustering decisions.

It should be noted that "rotation-only" PCA is, as its name suggests, purely a rotation; thus, unless dimensionality reduction is involved (i.e. keeping $< 100\%$ eigenvalue mass), this method will return the exact same results as the initial system did in Tables 4.1 and 4.2. It was found subsequently that neither dimensionality reduction using this flavor of PCA nor the use of eigenvalue-weighted scoring had any significant effect on diarization performance. Thus for the sake of simplicity, we continue by comparing only the initial system (no PCA, standard cosine scoring) and the data-centered PCA system with eigenvalue-weighted scoring.

4.5 The Final Setup

We report the steps that led to our best empirical results on the NIST 2008 SRE summed-channel telphone data. From those reported in Table 4.4, our results can be further improved by optimizing over different initial ranks of the Total Variability (TV) matrix. Table 4.5 shows the statistics obtained from the various i-vector dimensions attempted. Note that data-centered PCA (50% eigenvalue mass) is still applied to the set of i-vectors corresponding to each individual test file.[2]

[2] A similar experiment was performed on the initial system, but because the results were nearly identical to those of Table 4.5, they were omitted for simplicity.

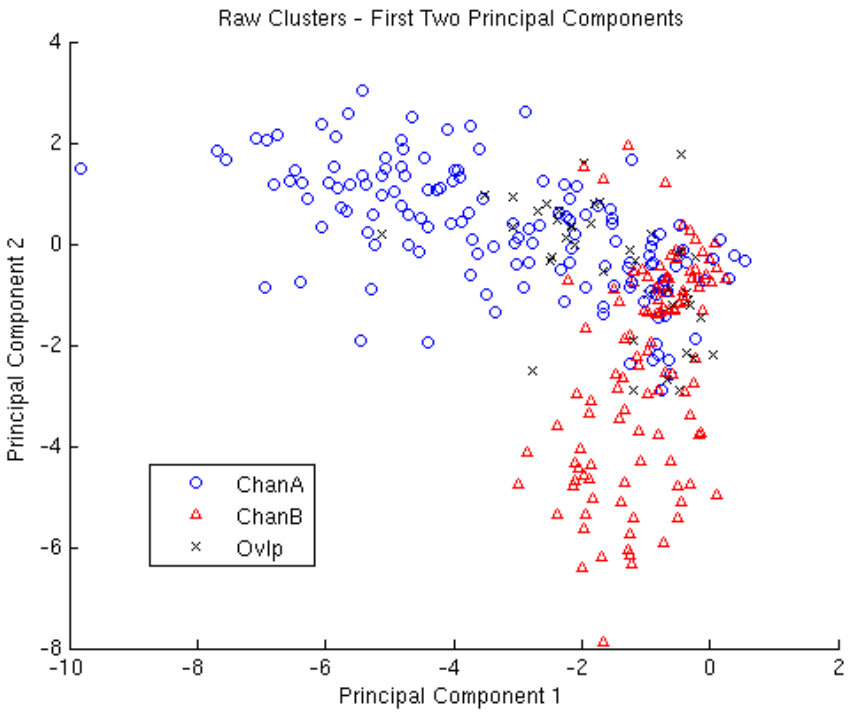

Figure 4-9: *Plot of the first two dimensions (principal components) of rotation-only PCA-projected speaker i-vectors. The triangles in red represent i-vectors of a male speaker, while the blue circles represent i-vectors of a female speaker in the same conversation. The black* $\boldsymbol{x}$*'s correspond to i-vectors representing overlapped speech.*

We settled on the TV100 configuration, which gave the best results despite a relatively low dimensionality, for our final experiment using the reference speech/non-speech boundaries to compare our system with those described in [12]. Table 4.6 summarizes our results obtained on the final configuration. To be sure, we report the results of our system (TV100, 50% PCA) using the reference boundaries as an initial segmentation, again denoted "Ref VAD."

We can see that our system, which follows the exact same evaluation protocol as the BIC, Stream- and VB-based systems, slightly outperforms the VB-based system. Despite its initial shortcomings in the First Pass, the best diarization performance was ultimately obtained by the TV100 configuration using the original, data-centered PCA. Table 4.7 breaks down the performance of the data-centered PCA system alongside that of the initial system without PCA and showcases once again the reliability of our Second Pass Refinement stage. At the end of the day, the difference between these results is minimal; at the level of 1% error, these systems are very finely tuned and susceptible to noise. Moreover, there exists a mismatch between our respective reference segmentations - we used a phone recognizer provided by Brno University of

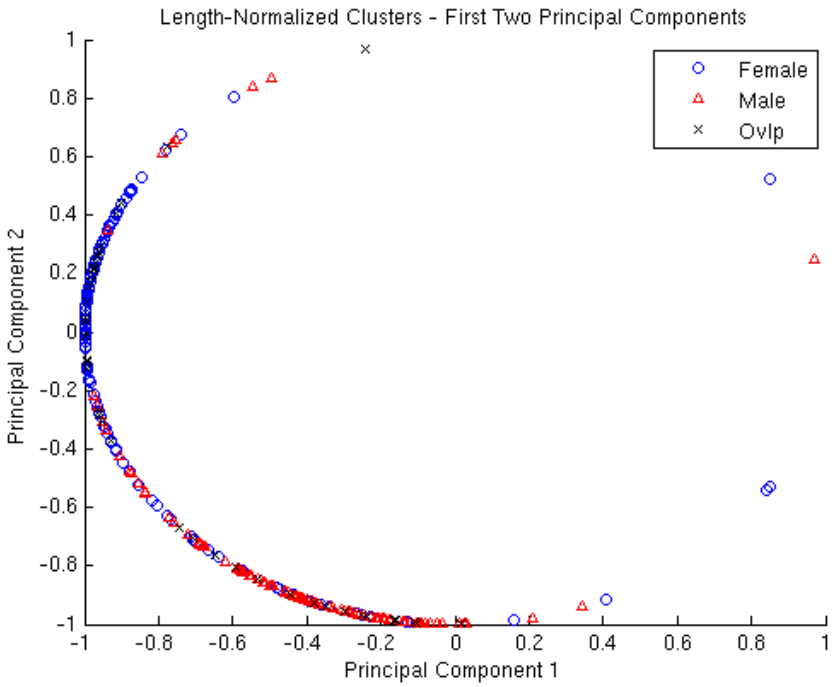

Figure 4-10: *Plot of the length-normalized speaker i-vectors after applying a two dimensional rotation-only PCA-projection across the entire conversation.*

	TV40	**TV100**	TV200	TV400	TV600
Avg Dim	7	**14**	20	26	28
DER (%)	4.5	**4.3**	4.4	4.7	4.6
σ (%)	6.6	**6.4**	6.4	7.0	6.9

Table 4.5: *Overall diarization performance of Total Variability matrices of varying rank. The second row lists the average number of dimenisions that resulted after the PCA projection (50%) was estimated.*

Technology, while the work in [12] used ASR transcripts provided by NIST - and it is unknown how such difference, however slight, might affect the results. Nevertheless, what is clear is that these approaches are both very successful in the two-speaker telephone diarization task at hand.

4.6 Discussion

Inspired by the success of factor analysis and Total Variability for the speaker modeling, we have developed a system that achieves state-of-the-art results on the two-speaker telephone diarization task. Our previous benchmark, the VB-based FA system described in [12], elegantly integrates the factor analysis paradigm with the prior

	Speaker Confusion (%)	σ_C (%)
BIC-based Baseline	3.5	8.0
Stream-based FA	4.6	8.8
VB-based FA	1.0	3.5
Ref VAD + TV100	**0.9**	**3.2**

Table 4.6: *Comparison of diarization results on the NIST SRE 2008 Summed-Channel Telephone Data.* (BIC - Bayesian Information Criterion; FA - Factor Analysis; VB - Variational Bayes; VAD - Voice Activity Detector; TV - Total Variability)

	No PCA		Data-Cntrd PCA	
	Confusion (%)	σ_C (%)	Confusion (%)	σ_C (%)
First Pass	2.6	5.0	2.9	4.2
Re-segmentation	2.3	6.0	2.4	5.6
Second Pass	1.3	4.8	**0.9**	**3.2**

Table 4.7: *Results obtained after each stage of the diarization procedure while using the reference segmentation. "No PCA" refers to the initial approach that involves no pre-processing of the i-vectors prior to K-means clustering, while "Data-Cntrd PCA" refers to the use of PCA dimensionality reduction (50% eigenvalue mass) on the TV100 configuration.*

work on Variational Bayesian methods for speaker diarization described in [10]. In a search for added simplicity, we utilized the effectiveness of the cosine similarity metric in the Total Variability subspace.

There are still many ways in which we can improve and refine this initial approach. For one, there is a need to address the problem of overlapped speech detection. Finding a good way to robustly detect and remove corrupted segments - whether in the cepstral domain or in the Total Variability space - would be helpful for our PCA initialization and subsequent clustering [33]. Additionally, our reported results have been restricted to two-speaker telephone conversations; we have yet to address the issue of applying our system to a conversation setting involving more than two or even an unknown number of speakers. To that end, we see potential in directly extending our approach to conversations with more than two speakers, as well as in applying Variational Bayesian methods for model selection (i.e. determining the number of speakers) and clustering in the Total Variability space [10].

Chapter 5

Towards K-speaker Diarization

Though we were able to achieve good results with the configuration described in the previous chapter, the task was limited to the problem in which we know there are exactly two speakers in the given telephone conversation. To solve the speaker diarization problem in general, we need to be able to handle the case in which there are more than two speakers, as well as the case in which we do not know how many speakers are present. We approach these issues in incremental fashion and begin by tackling the former before discussing methods to address the latter in Chapter 6.

5.1 CallHome Data

In order to continue using the same Total Variability framework as in our previous experiments, we evaluate our system on the multilingual CallHome data, a corpus of multi-speaker telephone conversations. A summary of this corpus broken down by number of speakers and language spoken was provided in Table 2.1 of Section 2.4. We evaluate on a subset of the data from the NIST 2000 Speaker Recognition Evaluation, which amounts to 500 recordings, each 2-5 minutes in length, containing between two and seven participants [21].

To provide some further context as to the nature of these data, the conversations are between family members, where one member may be calling home from overseas. As a result, one or both sides of the conversation may have multiple participants (e.g. father, mother, brother, etc.). Furthermore, the existing familiarity between callers results in a significant increase in the amount of overlapped speech, which makes our diarization task more difficult.

5.2 Extending the Previous Approach

We directly extend the speaker diarization system detailed in Chapter 4 to the setting in which we are given the number of speakers ($K \geq 2$) present in a given audio stream. This is easily done at the First Pass K-means clustering stage by setting K to be the given number of speakers. Every other step in the system is the same as before, as both the Re-segmentation and Second Pass Refinement algorithms were designed to handle an arbitrary number of clusters.

In order to focus solely on the problem of speaker clustering, we use the reference speech/non-speech segmentation and extract i-vectors using a Total Variability matrix of rank 100 (TV100). Forgoing any form of dimensionality reduction, the i-vectors are clustered using K-means via the cosine distance as before. One refinement we do make is to initialize our cluster means using a bottom-up hierarchical clustering method similar to the algorithm described in Section 2.2. Otherwise, K-means initializes its cluster means randomly; this works fine in the two-speaker case, but increasing the number of speakers makes our data less predictable and more prone to inconsistencies.

5.2.1 Initial Results

We use the state-of-the-art results reported in [19] as our evaluation benchmark; an overview of the system was provided in Section 2.3.3. Figure 5-1 provides a plot of the results, broken down by number of speakers per conversation, at each stage of the diarization process and compares our performance to that of our benchmark. The plot also details, along the x-axis, the number of conversations that contained x speakers. These statistics show that there are very few conversations featuring more than four speakers and thus explains the correspondingly high variance in those results.

In the cases of two- and even three-speaker conversations, our system outperforms our benchmark. And as we increase the number of speakers in the conversation, our diarization error also increases steadily and begins to perform consistently worse than our benchmark. One possible explanation for some of these discrepancies is that our respective background and factor analysis models were trained on different datasets. The work in [19] emphasized exposure to a variety of different languages; thus they built their models using the multilingual Callfriend database as well as the Italian, Swedish, and Brasilian Portuguese SpeechDat corpora, whereas we trained our models using the English Switchboard and multilingual Mixer corpora as discussed in Section 2.4. Nevertheless, these differences should only account for a fraction of our performance mismatch; we seek a more profound explanation by taking a closer look at the evaluation data.

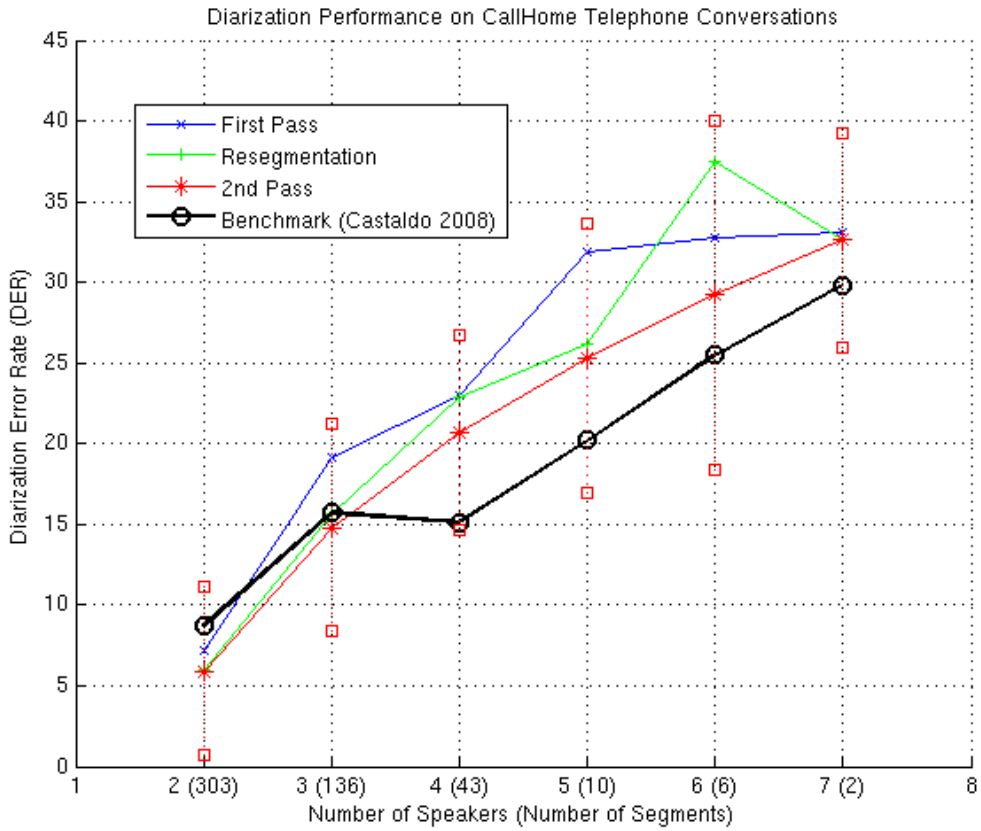

Figure 5-1: *Plot of diarization error as a function of the number of speakers in the conversation. First Pass results are denoted in blue, Re-segmentation results in green, and Second Pass in red along with an error interval* $\pm\frac{1}{2}$ *a standard deviation. The results of our benchmark are shown in black. Also provided, in parentheses, along the x-axis are the number of conversations that contained x number of speakers.*

5.3 The Issue of Data Sparsity

Figure 5-2 shows a histogram of the proportion of i-vector segments attributed to each of the speakers in every three speaker CallHome conversation. What it shows are the relative proportions of i-vectors attributed to the most, second, and least talkative speakers in a given three speaker conversation. As it turns out, the least talkative speaker accounts for an average of less than 15% of the i-vectors extracted from a given conversation, while the most talkative speaker accounts for over 50%. This adds a significant degree of difficulty to our clustering procedure, as those 15% of i-vectors attributed to the least talkative speaker can potentially be written off as noise.

Our benchmark system processes the incoming audio in slices of 60 seconds in

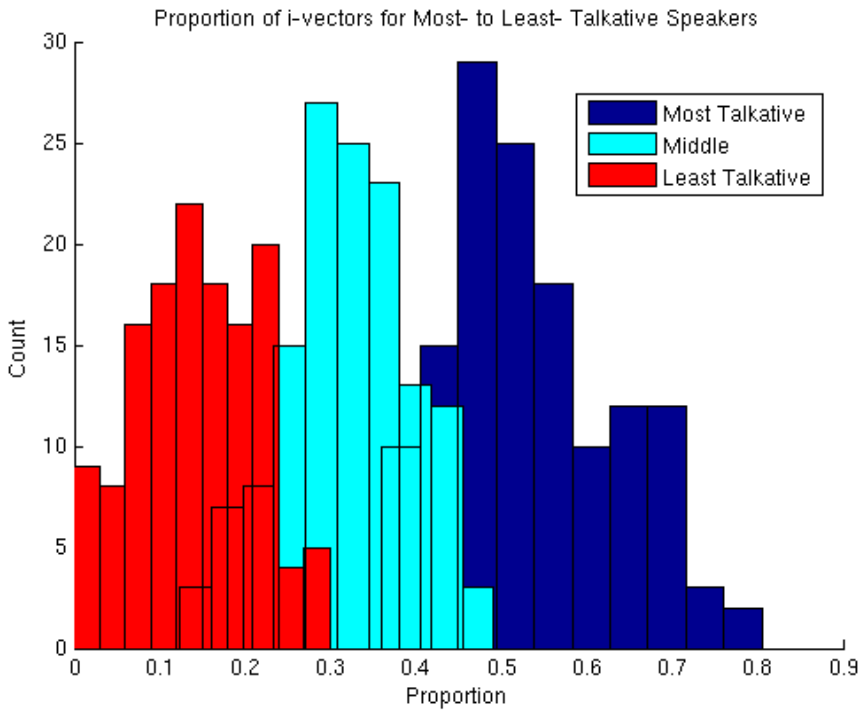

Figure 5-2: *Histograms showing the proportion of i-vectors associated with the most, second, and least talkative participants in a given three speaker conversation.*

length and, within each slice, utilizes streams of speaker factors [19] to do within-slice clustering. The system then takes advantage of speaker recognition techniques to compare and cluster speakers across slices. Because the least talkative speaker is probably involved for only one short interval of the conversation, this method of divide-and-conquer effectively gives that speaker a better local representation within his or her respective slice.

5.3.1 Duration-Proportional Sampling

The challenge of data imbalance is one that cannot be avoided. We can, however, try to balance out this effect in the best way possible. Our current system uses i-vectors extracted from each segment as a single data vector; it does not take into account the duration of the segment that was used. Recall the posterior covariance of an i-vector, Equation (3.26), and notice that it is inversely proportional to the zeroth order Baum-Welch statistics $N(u)$, the soft counts of the UBM components. Thus the longer the segment used, the higher the values of $N(u)$, the smaller the covariance (uncertainty) of the i-vector and the more robust of a speaker estimate we have.

To make use of durational and covariance information, we propose the following

sampling scheme. For a given i-vector w and its covariance $\text{cov}(w, w)$, we draw a number of samples n from this distribution proportional to the duration t of the segment used to estimate $\mathcal{N}(w, \text{cov}(w, w))$. This technique makes use of durational information in two ways: (a) a shorter segment results in relatively fewer i-vector samples, and (b) a shorter segment results in a covariance $\text{cov}(w, w)$ that is relatively large, thus its samples will be noisier. What this effectively does is increase the relative importance of longer, more reliable segments for the estimation of our respective speaker clusters.

5.3.2 Experiments

For our experiments, we draw samples at a rate of 10 per second; for example, if a segment is 1.2 seconds in length, we draw 12 samples from its corresponding i-vector distribution. Thus, we increase the amount of data with which we can use to cluster by a factor of 10.

Figure 5-3 compares the First Pass Clustering results obtained using K-means as usual with those obtained from applying K-means clustering to the duration-proportional sampling scheme described previously. To provide some relative referene, we have also included the benchmark results from [19]. We can see that there is really no significant difference between the results using duration-proportional K-means and those obtained using the original K-means clustering. This suggests the possibility that our current limitations may not only lie in the data, but also in our method of clustering.

5.4 Soft Clustering on the Unit Hypersphere

Until now, we have relied solely on a cosine distance-based K-means clustering algorithm to do our First Pass Clustering. And while it has been reasonably effective thus far, it might be useful to explore probabilistic alternatives. For data defined in Euclidean space, the probabilistic generalization of K-means is the Gaussian Mixture Model [32]. For the cosine distance metric, our i-vector data is defined on the surface of a unit hypersphere. Unfortunately, GMMs are inadequate for characterizing such data. Instead, there has been work exploring the use of von Mises-Fisher distributions, which arises naturally for data distributed on the unit hypersphere [34].

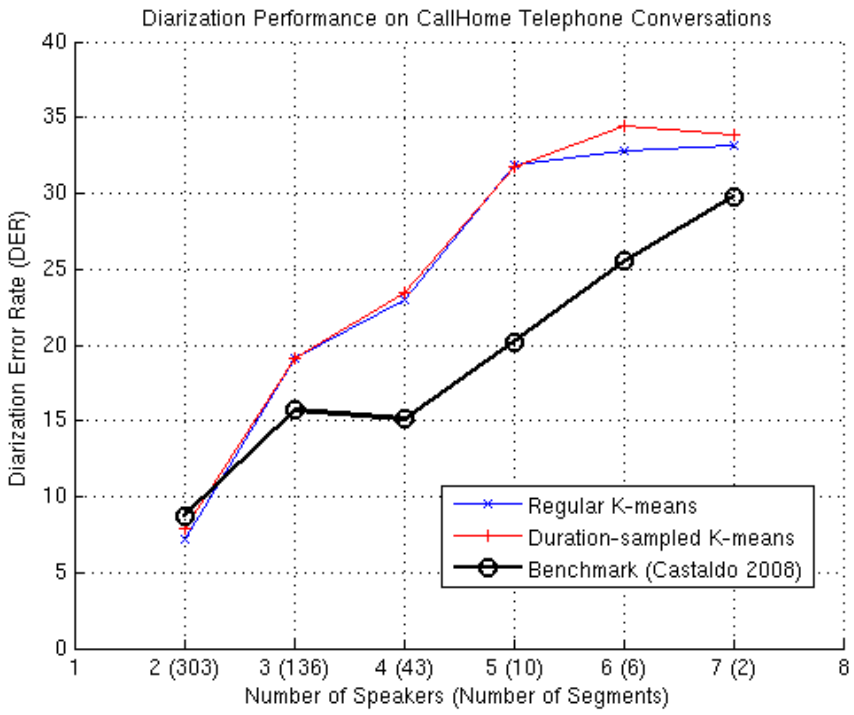

Figure 5-3: *A plot of First Pass Clustering results after incorporating a duration-proportional sampling scheme before K-means clustering. The baseline standard K-means results are shown, as well as the benchmark results for reference.*

5.4.1 The von Mises-Fisher Distribution

A d-dimensional unit random vector x (i.e. $x \in \mathbb{R}^d$ and $\|x\| = 1$) is said to have a d-variate von Mises-Fisher (vMF) distribution if its probability density function is given by

$$f(x|\mu, \kappa) = c_d(\kappa) e^{\kappa \mu^* x} \tag{5.1}$$

where $\|\mu\| = 1$, $\kappa \geq 0$, and $d \geq 2$. The normalizing constant $c_d(\kappa)$ is given by

$$c_d(\kappa) = \frac{\kappa^{\frac{d}{2}-1}}{(2\pi)^{\frac{d}{2}} I_{\frac{d}{2}-1}(\kappa)} \tag{5.2}$$

where $I_r(\cdot)$ represents the modified Bessel function of the first kind and order r. Thus, the parameters of our density function $f(x|\mu, \kappa)$ consist of a mean direction μ and a so-called *concentration* parameter κ, as it characterizes how strongly the unit vectors are concentrated about the direction μ. Larger values of κ imply a stronger concentration about the mean direction. Thus as $\kappa \to \infty$, $f(x|\mu, \kappa)$ tends to a point density; conversely, the case where $\kappa = 0$ reduces to the uniform density over the

entire hypersphere [34].

We briefly summarize the maximum likelihood estimates for the parameters of a single vMF distribution. A full derivation can be found in [34]. Let $\mathcal{X} = \{x_1, ..., x_n\}$ be a set of independent, identically distributed samples drawn from $f(x|\mu, \kappa)$. Given $\mathcal{X}$, we want to find maximum likelihood estimates for the parameters μ and κ. Writing the log-likelihood of $\mathcal{X}$ to be

$$\log P(\mathcal{X}|\mu, \kappa) = n \log c_d(\kappa) + \kappa \mu^* r \tag{5.3}$$

where $r = \sum_i x_i$, we obtain maximum likelihood estimates of our parameters by maximizing (5.3) subject to the constraints $\|\mu\| = 1$ and $\kappa \geq 0$. With some algebraic manipulation [34], we find our parameter estimates (μ, κ) to be as follows:

$$\hat{\mu} = \frac{r}{\|r\|} = \frac{\sum_i x_i}{\|\sum_i x_i\|} \tag{5.4}$$

$$\frac{I_{\frac{d}{2}}(\hat{\kappa})}{I_{\frac{d}{2}-1}(\hat{\kappa})} = \frac{\|r\|}{n} = \bar{r} \tag{5.5}$$

Because solving for $\hat{\kappa}$ involves an implicit equation (5.5) that is a ratio of modified Bessel functions, it is not possible to obtain an analytic solution. Thus, we resort to the following approximation obtained via numerical methods [34]:

$$\hat{\kappa} = \frac{\bar{r}d - \bar{r}^3}{1 - \bar{r}^2} \tag{5.6}$$

5.4.2 An EM Algorithm on a Mixture of vMFs

We can extend from a single vMF distribution into a generative model consisting of a mixture of K von Mises-Fisher distributions with a probability density function given by

$$g(x|\Theta) = \sum_{h=1}^{K} \alpha_h f_h(x|\mu_h, \kappa_h) \tag{5.7}$$

where $\Theta = \{\alpha_1, ..., \alpha_K, \mu_1, ..., \mu_K, \kappa_1, ..., \kappa_K\}$, $\alpha_h \geq 0$ and $\sum_h \alpha_h = 1$.

Using $h = 1, ..., K$ to index the mixture components, an EM algorithm can be

derived for this distribution to obtain the following update equations:

$$\alpha_h = \frac{1}{n}\sum_{i=1}^{n} p(h|x_i, \Theta) \tag{5.8}$$

$$r_h = \sum_{i=1}^{n} x_i p(h|x_i, \Theta) \tag{5.9}$$

$$\hat{\mu}_h = \frac{r_h}{\|r_h\|} \tag{5.10}$$

$$\frac{I_{\frac{d}{2}}(\hat{\kappa}_h)}{I_{\frac{d}{2}-1}(\hat{\kappa}_h)} = \frac{\|r_h\|}{\sum_{i=1}^{n} p(h|x_i, \Theta)} = \bar{r}_h \tag{5.11}$$

$$\implies \kappa_h \approx \frac{\bar{r}_h d - \bar{r}_h^3}{1 - \bar{r}_h^2} \tag{5.12}$$

where the posterior distribution of the hidden variables α_h is given by

$$p(h|x_i, \Theta) = \frac{\alpha_h f_h(x_i|\Theta)}{\sum_{l=1}^{K} \alpha_l f_l(x_i|\Theta)} \tag{5.13}$$

Note also that we once again resort to numerical approximation in our updates for κ_h. Further details of this derivation are beyond the scope of this thesis, but can be found in [34].

5.4.3 Experiments

Figure 5-4 displays the results obtained using mixtures of von Mises-Fisher distributions (Mix-vMF) as a method for First Pass Clustering. Once again, the i-vectors were not subject to any PCA-based dimensionality reduction. The EM algorithm was initialized using K-means, which was initialized by hierarchical clustering (as in Section 2.2); incorporating the soft-clustering of Mix-vMF so discussed is just an additional processing step. We have also included results from applying both Mix-vMF and the duration-proportional sampling methods discussed in Section 5.3.1. Interestingly enough, it seems as though using both techniques simultaneously somewhat averages out the differences in results obtained between the original K-means and standard Mix-vMF applied to the non-sampled data.

Only looking at First Pass results can be misleading, however; there is a lot still to happen in the subsequent steps of our diarization system. Figure 5-5 shows the final results obtained using standard Mix-vMF as well as duration-proportional sampling with Mix-vMF. With the exception of the six-speaker case, we can see that both methods involving Mix-vMF perform very competitively with the standard

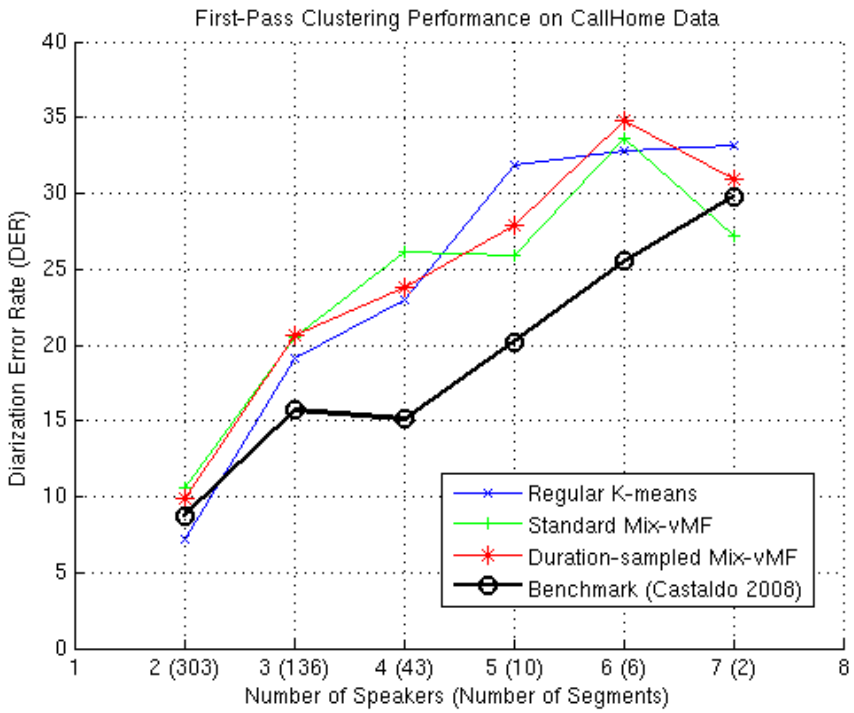

Figure 5-4: *A plot of First Pass Clustering results after incorporating an EM algorithm for Mixtures of von Mises-Fisher distributions. Results are reported with (red) and without (green) the use of the duration-proportional sampling scheme discussed in Section 5.3.1. Shown also are First Pass results given by the standard K-means approach, as well as the final results obtained by our benchmark system for reference.*

K-means clustering method. In the five-speaker case, both methods involving Mix-vMF perform quite a bit better than the K-means initialization, with the duration-proportional Mix-vMF doing nearly as well as our benchmark. For the remaining scenarios the duration-proportional scheme seems to have a slight edge in performance over its less complex Mix-vMF counterpart, but not in a way that is too significant.

5.5 Discussion

In this chapter, we have extended our initial work on two-speaker telephone diarization into the K-speaker setting. We explored techniques to combat issues related to data sparsity and the underrepresentation of less talkative speakers, and we have also provided a probabilistic generalization to the spherical K-means that was used in our original work. As it stands, however, it seems as though doing less is more. On average, the newly proposed methods and enhancements fail to bring significant performance improvements; further investigation on this front is necessary in order

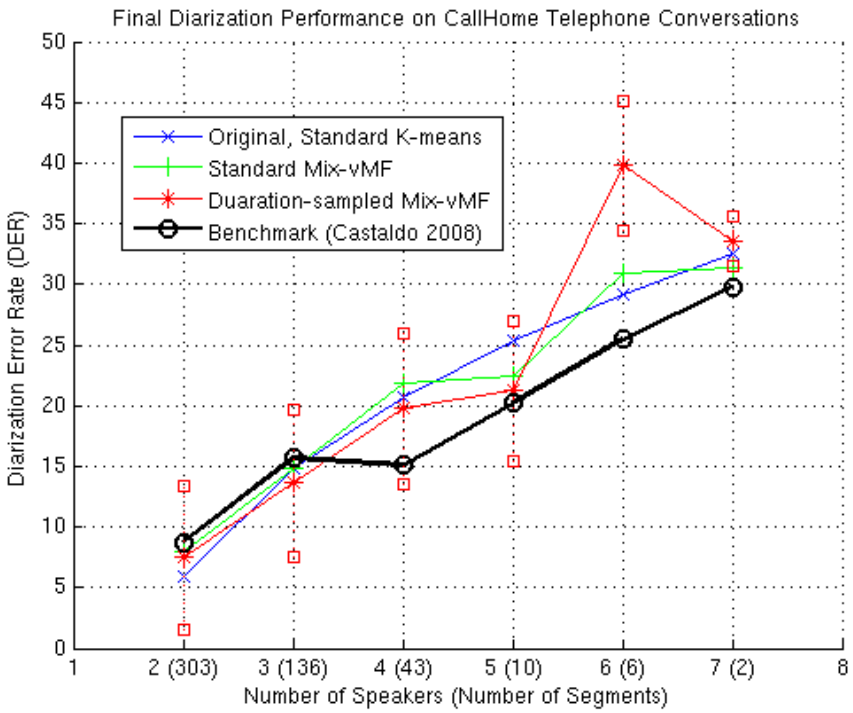

Figure 5-5: *A plot of Final Diarization results after incorporating an EM algorithm for Mixtures of von Mises-Fisher distributions. Results are reported with (red) and without (green) the use of the duration-proportional sampling scheme discussed in Section 5.3.1. Shown also are the error intervals ($\pm\frac{1}{2}$ a standard deviation) of the duration-proportional Mix-vMF, as well as the Final Second Pass results given by the standard K-means approach and the results obtained by our benchmark system for comparison.*

to better understand these sorts of behaviors.

Looking ahead, we consider the diarization problem in general where the number of speakers K is unknown *a priori*. While attempts have been made using Bayesian Information Criterion-based methods [9], Variational Bayes [10], and Hierarchical Dirichlet Processes [15, 16], this remains an open problem in speaker diarization. Thusfar, the methods mentioned have all relied solely on low-level acoustic features; we will propose an attempt using intermediate-level i-vectors as features for clustering and identifying the number of speakers in a conversation.

Chapter 6

An Unknown Number of Speakers

We have been able to achieve competitive results with the configurations described in the previous chapter; however, the task was limited to the problem in which we knew exactly how many speakers are present in the given telephone conversation. To solve the speaker diarization problem in general, we need to be able to handle the case in which we do not know how many speakers are present.

This problem becomes one of Bayesian model selection. One advantage of the Bayesian framework is that it benefits from the interesting property of Occam's razor, which states that "Entities should not be multiplied unnecessarily." In the probabilistic modeling domain, this can be interpreted as the preference for the simplest model amongst all different possible models that can explain a given phenomenon. To this end, Bayesian model learning naturally embeds a form of penalty on increased model complexity; as such, Bayesian methods prefer simpler models for explaining data, making this framework appealing for many model selection problems [10].

Unfortunately, the richness of Bayesian theory often renders exact probabilistic inference computationally intractable. In such cases, approximation methods must be considered in order to produce a solution as close as possible to the real one. An example of such is the Bayesian Information Criterion (BIC) based on the Laplace approximation [35]. Because of its relative simplicity, BIC-based methods have been widely used for the problem of speaker diarization. As discussed in Chapter 2, however, this method often requires a bit of parameter tuning and engineering for optimal performance [9].

There are a number of ways to work around the intractability of Bayesian computations. One methodology that has energized the field for more than a decade is that of Markov Chain Monte Carlo (MCMC), which provides a systematic sampling approach to the computation of likelihoods and posterior distributions [17]. Despite its success, however, MCMC methods can be slow to converge and their convergence

can be difficult to diagnose. Further research on statistical sampling is needed, and it is also important to explore alternatives.

The rest of this chapter examines one class of such alternatives, known as *variational inference* [18]. We follow the footsteps of research done previously in [12] and [10] in an attempt to arrive at a Total Variability-based method of determining the number of speakers present in a given audio stream. Ultimately, the ideas discussed here are mostly exploratory and very much still a work in progress. Our goal is to present the intuition behind our proposed methods and some preliminary results.

6.1 The Variational Approximation

The basic idea of variational inference is to formulate the computation of a marginal or conditional probability distribution in terms of an optimization problem [17]. This (generally still intractable) problem is then "relaxed," yielding a simplified optimization problem that can be solved in iterative (EM-like) fashion.

In this section, we follow the explanations of [10] and provide a mathematical overview of variational methods for Bayesian learning. Consider some data $Y = \{y_1, ..., y_L\}$, a hidden variable set $X = \{x_1, ..., x_M\}$ and a parameter set $\theta = \{\theta_1, ..., \theta_N\}$ for a given model m. The quantity of interest is the marginal log-likelihood:

$$\log P(Y|m) = \log \int P(Y, X, \theta|m) dX d\theta \tag{6.1}$$

In a fully Bayesian framework, we consider both hidden variables and model parameters to be random variables with respective distributions. Without some simplifying assumptions, however, it can be seen that the exact computation of (6.1) is not tractable in general. For example, suppose there were inter-dependencies within the set of hidden variables X as well as the model parameters θ. Then integrating over every possible configuration of those variables and parameters would potentially scale exponentially with each inter-dependency.

Let us introduce a distribution $q(X, \theta)$ to approximate the true (but unknown) posterior distribution $P(X, \theta|Y, m)$ of hidden variables and model parameters given the observed data Y. By applying Jensen's inequality[1], it is possible to define an

[1]Jensen's inequality in general: If Z is a random variable and ϕ is a convex function, then $\phi(\mathbb{E}[Z]) \leq \mathbb{E}[\phi(X)]$. In our case, since $\log(\cdot)$ is a concave function, the inequality is reversed.

upper bound on the marginal log-likelihood as

$$\log P(Y|m) = \log \int P(Y, X, \theta|m) dX d\theta = \log \int q(X, \theta) \frac{P(Y, X, \theta|m)}{q(X, \theta)} dX d\theta \quad (6.2)$$

$$\geq \int q(X, \theta) \log \frac{P(Y, X, \theta|m)}{q(X, \theta)} dX d\theta \quad (6.3)$$

The bound represented in (6.3) is still an intractable integral. For the *variational approximation*, we assume that our distribution $q(X, \theta)$ can be factorized independently over model parameters θ and hidden variables X; that is,

$$q(X, \theta) = q(X)q(\theta) = \prod_{j=1}^{M} q(x_j) \cdot \prod_{k=1}^{N} q(\theta_k) \quad (6.4)$$

where we have explicitly written that each of the individual hidden variables x_j and model parameters θ_k can factorize independently. This is not a requirement, however; the factorizations can be done as desired, and it should actually be noted that model parameters have natural dependencies (e.g. mean and covariance of a Gaussian distribution) that require joint estimation but will not render our computation intractable. As we will see later, these dependencies can be expressed through the use of graphical models [18, 24].

To keep notation simple, we will continue under the assumption that our distribution $q(\cdot)$ factorizes over the parameters θ and hidden variables X, i.e. $q(X, \theta) = q(X)q(\theta)$. Then (6.3) can be broken down and simplified as follows:

$$\log P(Y|m) \geq \int q(X)q(\theta) \log \frac{P(Y, X, \theta|m)}{q(X)q(\theta)} dX d\theta \quad (6.5)$$

$$= \int q(\theta) \left[\int q(X) \log \frac{P(Y, X|\theta, m)}{q(X)} dX + \log \frac{P(\theta|m)}{q(\theta)} \right] d\theta \quad (6.6)$$

$$= \int q(\theta) \left[\int q(X) \left(\log P(Y, X|\theta, m) - \log q(X) \right) dX - \log \frac{q(\theta)}{P(\theta|m)} \right] d\theta \quad (6.7)$$

$$= \int q(\theta)q(X) \log P(Y, X|\theta, m) d\theta dX - \int q(X) \log q(X) dX - \int q(\theta) \log \frac{q(\theta)}{P(\theta|m)} d\theta \quad (6.8)$$

$$= \int q(X)q(\theta) \log P(Y, X|\theta, m) dX d\theta + \mathbf{H}(X) - \mathbf{KL}\left(q(\theta) \| P(\theta|m)\right) \quad (6.9)$$

$$= F_m(q(X), q(\theta)) \quad (6.10)$$

where $\mathbf{H}(X)$ denotes the entropy[2] of the set of hidden variables X, and the expression $\mathbf{KL}\left(q(\theta)\|P(\theta|m)\right)$ denotes the Kullback-Leibler divergence[3] between the distributions $q(\theta)$ and $P(\theta|m)$.

The expression denoted by (6.10) is also known as the variational energy, or *free energy*. The goal of variational learning is to find optimal distributions $q(X)$ and $q(\theta)$ that maximize this free energy. Furthermore, we can see the sort of "Bayesian mentality" that is incorporated in this expression. To maximize (6.10), we strive to maximize the first term, which is the expectation under $q(X,\theta)$) of the complete data log-likelihood (including both hidden X and observed variables Y). And in order to prevent overfitting, our second term tries to maximize the entropy, or randomness associated with our hidden variable set X. Thirdly, we want to minimize the KL-divergence between the variational parameters $q(\theta)$ and the exact model priors $P(\theta|m)$. Finally, we can also view the value of this KL-divergence as a penalty term that becomes larger with the number of parameters that need to be estimated [10]. Thus, this formulation penalizes models that are overly complex and intuitively follows the essence of Occam's razor, as desired.

Through (6.10), we have defined a lower bound to the marginal log-likelihood and intend to maximize this value. However, we have not yet seen exactly how maximizing the expression for free energy helps us model the data. To do so, we consider the following algebraic construction:

$$\log P(Y|m) = \int q(X,\theta)\log P(Y|m)dXd\theta \tag{6.11}$$

$$= \int q(X,\theta)\log\frac{P(Y,X,\theta|m)}{P(X,\theta|Y,m)}dXd\theta \tag{6.12}$$

$$= \int q(X,\theta)\log\left(\frac{P(Y,X,\theta|m)}{P(X,\theta|Y,m)}\cdot\frac{q(X,\theta)}{q(X,\theta)}\right)dXd\theta \tag{6.13}$$

$$= \int q(X,\theta)\log\frac{P(Y,X,\theta|m)}{q(X,\theta)}dXd\theta + \int q(X,\theta)\log\frac{q(X,\theta)}{P(X,\theta|Y,m)}dXd\theta \tag{6.14}$$

$$= F_m(q(X,\theta)) + \mathbf{KL}\left(q(X,\theta)\|P(X,\theta|Y,m)\right) \tag{6.15}$$

We can see that the marginal log-likelihood differs from our free energy $F_m(q(X,\theta))$ lower bound by the value of the KL-divergence between the variational posterior distribution $q(X,\theta)$ and the true posterior distribution $P(X,\theta|Y,m)$. Thus, maximizing

[2]In general, entropy is a measure of the uncertainty associated with a random variable X and is defined as $\mathbf{H}(X) = -\int dXq(X)\log q(X)$.

[3]The KL-divergence is a non-symmetric measure of the difference between two probability distributions Q and P and is defined as $\mathbf{KL}(Q\|P) = \int d\theta Q(\theta)\log\frac{Q(\theta)}{P(\theta|m)}$.

the lower bound F_m by optimization with respect to the distribution $q(X, \theta)$ is equivalent to minimizing the KL-divergence between the variational and true posteriors, which is our ultimate goal [32].

Using (6.10), it is possible to derive an EM algorithm to iteratively optimize the free energy. Known as *Variational Bayesian EM* (VBEM), the process involves taking functional derivatives with respect to $q(\theta)$ and $q(X)$ and incorporating Lagrange multipliers to enforce probability constraints, then equating the resulting expression to zero, and solving. Now, because we factorize $q(X, \theta)$ according to (6.4), the required derivatives are easy to take. As a result, each hidden variable x_j and parameter θ_k can be optimized independently while holding all other values constant, ultimately giving us the iterative nature of the VB-EM algorithm. To be clear, the order by which the variables and parameters are updated still depend on the structure of the corresponding graphical model, but so as to not belabor this thesis with the exact algorithmic details, we refer the interested reader to [10, 32, 36] for a more complete treatment of this topic.

6.2 Variational Bayes, Factor Analysis, and Telephone Diarization

We briefly summarize the key ideas behind the use of Variational Bayes in the development of Kenny's factor analysis-based diarization system [12]. In doing so, we motivate the incorporation of Variational Bayesian methods into our approach of using i-vectors for diarization.

The work in [12] begins by assuming there are just two speakers in the audio file and partitions the file into short segments, each of which is assumed to contain speech of just one of the speakers. This partitioning does not need to be very accurate; a uniform partition into one second intervals can be used to start.

We define two types of posterior distributions, known as *speaker posteriors* and *segment posteriors*. For each of the two speakers, the speaker posterior is simply a Gaussian distribution, or an i-vector of sorts[4] with some associated covariance. The mean of this distribution can be thought of as a point estimate of the speaker factors and the covariance matrix as a measure of the uncertainty in the point estimate. And for each segment, there are two segment posteriors q_1 and q_2 (where $q_1 + q_2 = 1$), where q_1 is the posterior probability of the event that the speaker in the segment is speaker 1 [20].

[4]This work used eigenvoices instead of i-vectors, whose differences are discussed in Section 3.3.

As discussed in the previous section, the Variational Bayes algorithm consists in estimating our two types of posterior distribution alternately. We summarize the algorithm as follows [20]:

- Initialize by partitioning the file into 1-second segments. Extract Baum-Welch statistics from each segment. Randomly initialize the segment posteriors; there is no need to initialize the speaker posteriors.
- On each iteration of Variational Bayes, synthesize Baum-Welch statistics for each speaker i by weighting the Baum-Welch statistics of each segment by the corresponding segment posterior q_i. Using these synthetic Baum-Welch statistics, update the speaker posterior via (3.25) and (3.26) as previously discussed. Then for each segment, update the segment posteriors for each speaker.
- Once convergence is reached, standard post-processing algorithms such as the Viterbi re-segmentation from the baseline system (Section 2.2) can be applied. Additionally, it might help to do yet another pass of Variational Bayes after the re-segmentation.

We can see that using the segment posteriors to weight the segment Baum-Welch statistics offers a sort of soft-speaker clustering, which is very much unlike the hard decision-making of agglomerative clustering in the baseline system. At convergence, q_1 and q_2 will usually take values of 0 or 1 for each segment [12].

In this approach, each segment is represented by a set of Baum-Welch statistics, which is then weighted according to some segment posterior distribution and pooled together with the Baum-Welch statistics of all other segments in order to update the speaker posterior. This gives us an elegant and fully probabilistic approach to two-speaker diarization that can be easily extended to the K-speaker problem; however, it is unclear how we might generalize this method even further to the problem of estimating an unknown number of speakers. As discussed in Chapter 4, our system represents each segment with a corresponding i-vector. Though the difference between our respective representations is minimal - the Baum-Welch statistics are summary statistics that contain all the information necessary to extract an i-vector - working with the i-vector representation allows us to use them as feature vectors to perform the necessary clustering.

6.3 Variational Bayesian GMM

Our approach seeks to incorporate Variational Bayes as a method for model selection. Because we work at the i-vector level, we can treat each i-vector as an independent

and identically distributed observation and attempt to identify the number of clusters (i.e. speakers) in addition to associating each i-vector (i.e. segment) with a cluster. Our previous approach used spherical K-means to do this clustering, which was then generalized probabilistically in Section 5.4 to an EM algorithm on mixtures of von Mises-Fisher distributions (Mix-vMF). Ideally, we would want to apply Variational Bayes to Mix-vMF, but because the concentration parameters κ_h of the distribution cannot be calculated analytically, a straightforward derivation of VBEM for Mix-vMF is more difficult to obtain [37]. Nevertheless, the next section will address some preliminary work on this front.

To begin, we resort to the use of Variational Bayesian Gaussian Mixture Models (VB-GMM). Although mixtures of Gaussians are not exactly appropriate for modeling data on a unit hypersphere, the derivation of VBEM for GMMs is well-defined and straightforward, making this a good starting point for our experiments. The full details of the VBEM derivation can be found in Chapter 4 of [10] as well as in Section 10.2 of [32]. Without dwelling too much on the exact mathematics, we outline the essential ideas to provide the necessary intuition.

6.3.1 VBEM Learning

Let us consider a Gaussian Mixture Model with K components and parameter set $\theta = \{\theta_1, ..., \theta_K\}$, where each $\theta_k = \{\pi_k, \mu_k, \Sigma_k\}$, giving us

$$p(y_t|\theta) = \sum_{k=1}^{K} \pi_i \mathcal{N}(y_t|\mu_k, \Sigma_k) = \sum_{k=1}^{K} p(x_t = k|\theta) p(y_t|x_t = k, \theta) \tag{6.16}$$

where $Y = \{y_1, ..., y_L\}$ is the observed set of independent, identically distributed data samples and $X = \{x_1, ..., x_L\}$ are the hidden variables. For a given time t, x_t indicates which mixture $k \in \{1, ..., K\}$ generated data point y_t. We assign priors to the parameters as follows:

$$p(\theta) = p(\{\pi_1, \mu_1, \Sigma_1\}, ..., \{\pi_K, \mu_K, \Sigma_K\}) = p(\{\pi_k\}) \prod_{k=1}^{K} p(\mu_k|\Sigma_k) p(\Sigma_k) \tag{6.17}$$

$$p(\{\pi_k\}) = \text{Dir}(\lambda_0) \tag{6.18}$$

$$p(\mu_k|\Sigma_k) = \mathcal{N}(\rho_0, \xi_0 \Sigma_k) \tag{6.19}$$

$$p(\Sigma_k) = \mathcal{W}(a_0, B_0) \tag{6.20}$$

where $\text{Dir}(\cdot)$, $\mathcal{N}(\cdot)$, and $\mathcal{W}(\cdot)$ respectively denote the Dirichlet, Normal, and Wishart distributions with associated hyperparameter set $\{\lambda_0, \rho_0, \xi_0, a_0, B_0\}$. Figure 6-1 po-

vides a graphical model representation of the Bayesian GMM showing the dependencies between our parameters and hidden variables.

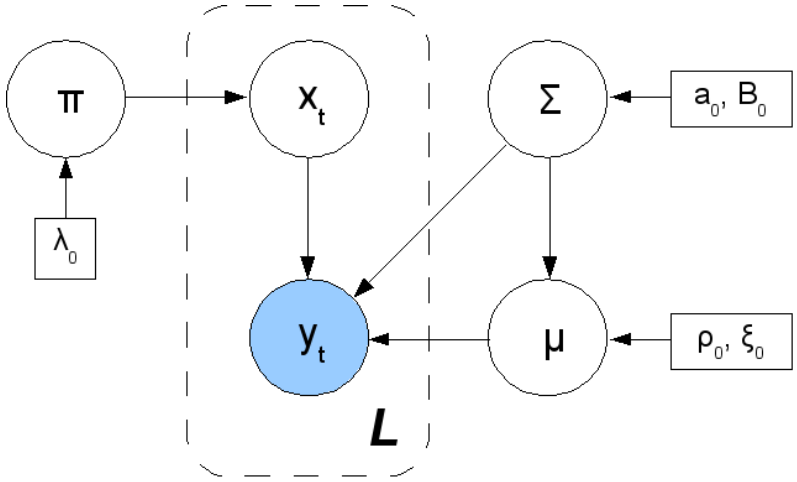

Figure 6-1: *A directed acyclic graphical model representing a Bayesian GMM. The dotted plate representation denotes a set of L repeated occurrences, while the shaded node y_t denotes an observation. For the parameters, $\boldsymbol{\Sigma}$ represents $\{\Sigma_1, ..., \Sigma_K\}$ and μ represents $\{\mu_1, ..., \mu_K\}$, while the hyperparameters are shown in boxes.*

We pick our variational posterior distributions to have the same form as our prior distributions with new hyperparameters, which we later update and optimize:

$$q(\theta) = q(\{\pi_1, \mu_1, \Sigma_1\}, ..., \{\pi_K, \mu_K, \Sigma_K\}) = q(\{\pi_k\}) \prod_{k=1}^{K} q(\mu_k|\Sigma_k) q(\Sigma_k) \tag{6.21}$$

$$q(\{\pi_k\}) = \text{Dir}(\lambda_k) \tag{6.22}$$

$$q(\mu_k|\Sigma_k) = \mathcal{N}(\rho_k, \xi_k \Sigma_k) \tag{6.23}$$

$$q(\Sigma_k) = \mathcal{W}(a_k, B_k) \tag{6.24}$$

where $\{\lambda_k, \rho_k, \xi_k, a_k, B_k\}$ are our updated hyperparameters. Notice that the factorization over the parameters in (6.21) is taken directly from the factorization of the prior distribution in (6.17). As such, we have not yet made any sort of approximation in our variational distribution [10].

For notational convenience, let $\{\Sigma_k\} = \{\Sigma_1, ..., \Sigma_K\}$ and $\{\mu_k\} = \{\mu_1, ..., \mu_K\}$. According to the graphical model of Figure 6-1, the true joint distribution over all

the random variables is given by

$$p(Y, X, \{\pi_k\}, \{\mu_k\}, \{\Sigma_k\}) = p(Y|X, \{\mu_k\}, \{\Sigma_k\})p(X|\{\pi_k\}) \cdot p(\{\pi_k\}) \prod_{k=1}^{K} p(\mu_k|\Sigma_k)p(\Sigma_k) \tag{6.25}$$

$$= \prod_{t=1}^{L} p(y_t|x_t, \{\mu_k\}, \{\Sigma_k\})p(x_t|\{\pi_k\}) \cdot p(\{\pi_k\}) \prod_{k=1}^{K} p(\mu_k|\Sigma_k)p(\Sigma_k) \tag{6.26}$$

where (6.26) results from the i.i.d assumption of the observed data Y. The subsequent variational distribution that factorizes over all the latent variables X and parameters θ is simply

$$q(X, \theta) = q(X) \cdot q(\theta) = \prod_{t=1}^{L} q(x_t) \cdot q(\{\pi_k\}) \prod_{k=1}^{K} q(\mu_k|\Sigma_k)q(\Sigma_k) \tag{6.27}$$

It is remarkable that this is the *only* assumption that we need to make in order to obtain a tractable practical solution to our Bayesian mixture model [32].

From here, a derivation of the VBEM algorithm is straightforward. We first assume that some initialization of the hyperparameters $\{\lambda_k, \rho_k, \xi_k, a_k, B_k\}$ has been provided. Then to obtain the VB E-step, consider the correspondence between each $q(x_t)$ in (6.27) and the expression $p(y_t|x_t, \{\mu_k\}, \{\Sigma_k\})p(x_t|\{\pi_k\})$ from (6.26). The latter is simply the joint distribution of two hidden and observed variables $p(x_t, y_t|\theta)$; moreover, we know from Bayes' rule that the posterior distribution is proportional to the joint distribution (i.e. $p(x_t|y_t, \theta) \propto p(x_t, y_t|\theta)$). Thus, for our variational posterior $q(x_t)$ to best approximate the true posterior as desired, we would also want

$$q(x_t) \propto p(y_t|x_t, \{\mu_k\}, \{\Sigma_k\}) \cdot p(x_t|\{\pi_k\}) \tag{6.28}$$

This ultimately leads us to the following expression for the VB E-step, which can be seen as equating $\log q(x_t)$ with the expected log likelihood of the observed data:

$$\log q(x_t) = \mathbb{E}_{\{\mu_k\},\{\Sigma_k\}} \left[\log p(y_t|x_t, \{\mu_k\}, \{\Sigma_k\})\right] + \mathbb{E}_{\{\pi_k\}} \left[\log p(x_t|\{\pi_k\})\right] + \text{constant} \tag{6.29}$$

where the expectations are to be taken over the current values for the parameters. Substituting for the two conditional distributions and absorbing any terms that are

independent of x_t into the additive constant gives us

$$\log q(x_t) = \sum_{k=1}^{K} x_{tk} \log z_{tk} + \text{constant} \tag{6.30}$$

where $x_{tk} \in \{0, 1\}$ are binary-valued components of the indicator vector x_t such that $\sum_{k=1}^{K} x_{tk} = 1$. In particular, x_{tk} indicates whether or not mixture component k was used to generate y_t. We define

$$\log z_{tk} = \mathbb{E}\left[\log \pi_k\right] - \frac{1}{2}\mathbb{E}\left[\log |\Sigma_k|\right] - \frac{D}{2}\log(2\pi) - \frac{1}{2}\mathbb{E}_{\mu_k,\Sigma_k}\left[(y_t - \mu_k)^* \Sigma_k^{-1} (y_t - \mu_k)\right] \tag{6.31}$$

where D is the dimensionality of the observed variable y_t. Taking the exponential of both sides of (6.30) and performing the proper normalization, we obtain

$$q(x_t) = \prod_{k=1}^{K} \gamma_{tk}^{x_{tk}} \tag{6.32}$$

where

$$\gamma_{tk} = \frac{z_{tk}}{\sum_{i=1}^{K} z_{ti}}. \tag{6.33}$$

The choice of the letter $\gamma_{(\cdot)}$ to denote these probabilities is due to their analogous relationship to the Baum-Welch sufficient statistics discussed previously in Sections 3.1.4 and 3.3.1. Once again, we can see that these are essentially the posterior probabilities of a given observation being generated by some specified mixture component. This can also be interpreted as the amount of responsibility given to each mixture component for a particular observation [32].

The VB M-step utilizes these values of $\gamma_{(\cdot)}$ to update the model hyperparameters in a way that is similar to the MAP adaptation discussed in Section 3.1.4. We begin by computing the following quantities for each component $k = 1, ..., K$:

$$N_k = \sum_{t=1}^{L} \gamma_{tk} \tag{6.34}$$

$$\bar{F}_k = \frac{1}{N_k} \sum_{t} \gamma_{tk} \cdot y_t \tag{6.35}$$

$$\bar{S}_k = \frac{1}{N_k} \sum_{t} \gamma_{tk} \cdot (y_t - \bar{F}_k)(y_t - \bar{F}_k)^* \tag{6.36}$$

Then the variational posterior distribution hyperparameters are updated for each

mixture component k as follows [10]:

$$\lambda_k = N_k + \lambda_0 \tag{6.37}$$

$$a_k = N_k + a_0 \tag{6.38}$$

$$\xi_k = N_k + \xi_0 \tag{6.39}$$

$$\rho_k = \frac{N_k \bar{F}_k + \xi_0 \rho_0}{N_k + \xi_0} \tag{6.40}$$

$$B_k = B_0 + \bar{S}_k + \frac{\xi_0 N_k}{\xi_0 + N_k} (\bar{F}_k - \rho_0)(\bar{F}_k - \rho_0)^* \tag{6.41}$$

We iterate these VBEM updates until the free energy (6.10) converges. Section 4.1.2 of [10] and Section 10.2.2 of [32] provide the exact details on how to calculate the free energy for the VB-GMM; however, its mathematical intricacies are beyond the scope of this thesis. Having now seen how to derive Variational Bayesian updates for Gaussian Mixture Models, we evaluate the performance of this method for model selection in speaker diarization.

6.3.2 Preliminary Results

In practice, we apply the implementation of VBEM for GMMs provided by [38] to our system. Given a set of i-vectors, we initialize a GMM using far too many components (i.e. $K = 15$, though our CallHome test set contains conversations with no more than 7 speakers). Then we run the VBEM updates, which automatically prunes out the unnecessary components.

In Figure 6-2, we first compare the ideal and actual results as obtained by VB-GMM on a three-speaker conversation. All plots were generated in three dimensions, and the top two plots display the ideal output from VB-GMM, while a visualization of the actual outputs are directly below. The plots on the left were generated using length-normalized data to emulate the cosine distance, while the plots on the right are generated by running VB-GMM in Euclidean space.

Though it does not affect our decision to emulate the use of cosine similarity for our clustering methods, it is readily apparent how unsuited GMMs are for modeling spherical data. We also see in the actual results for both Euclidean and spherical spaces how there are still extra Gaussians left over after VBEM converges. Some of these leftover components are degenerate and insignificant, suggesting the potential need for some sort of a heuristic to prune out the components whose responsibility (i.e. proportion of data points it accounts for) is below some certain threshold.

We do, however, want to keep in mind instances in which a conversation par-

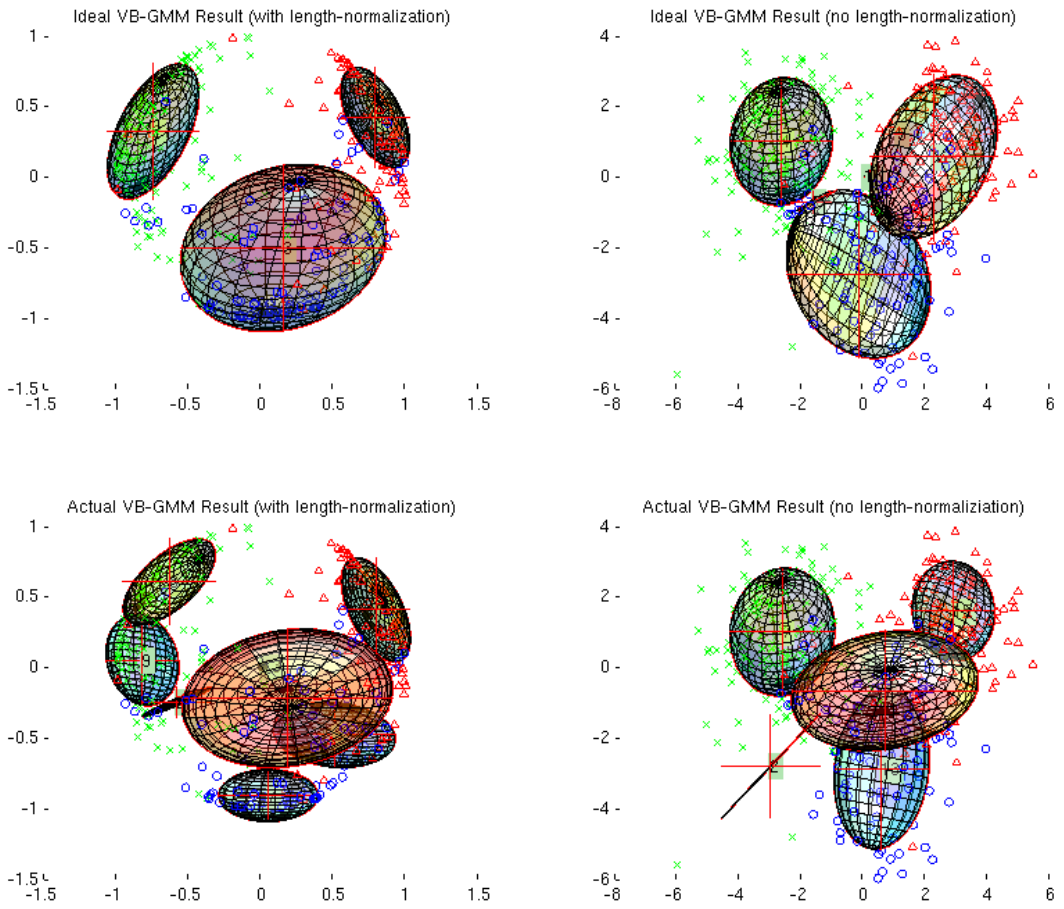

Figure 6-2: *Top: Ideal clustering results obtained by VB-GMM applied to length-normalized and non-length-normalized i-vectors. Bottom: Actual clustering results obtained by VBGMM.*

ticipant seldom speaks. Thus, we incorporate the duration-proportional sampling method discussed in Section 5.3.1 and run VBEM on length-normalized i-vectors to obtain the model selection results shown in Figure 6-3. For a given Actual Number of Speakers (x-axis), the colormap histogram shows the proportion of those conversations containing x speakers that resulted in the corresponding Number of Speakers Found. For example, perfect model selection would result in an image that is blue everywhere except for dark red blocks along the diagonal running from the bottom-left to the top-right.

Though still far from perfect, our initial results seem reasonable. We are able to correctly detect the three-speaker scenario nearly 60% of the time, and even our errors

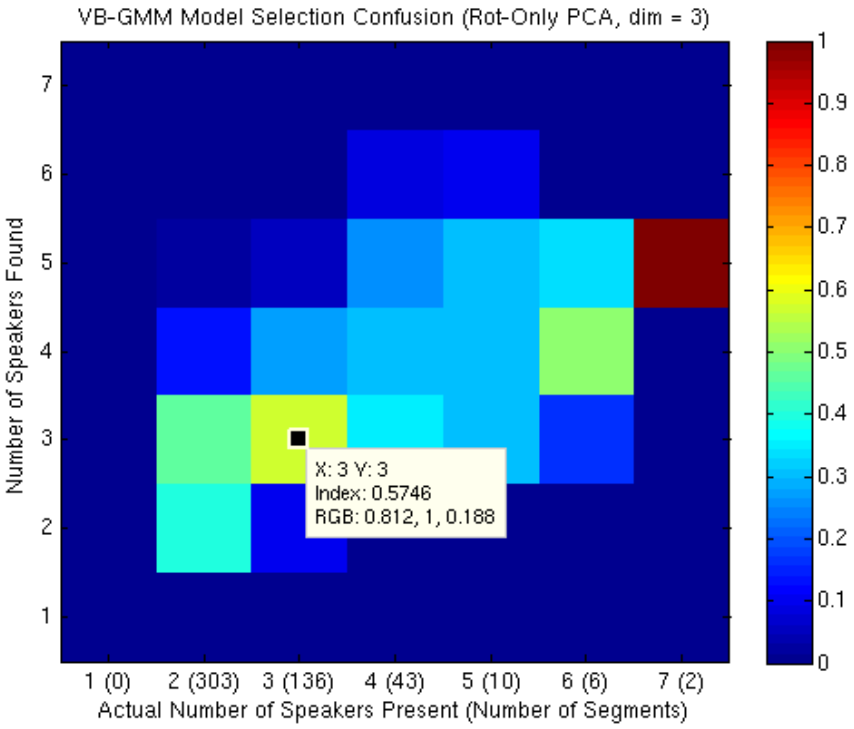

Figure 6-3: *Model selection results obtained by VB-GMM applied to length-normalized i-vectors. For a given Actual Number of Speakers (x-axis), the colormap shows the proportion of those conversations that resulted in the corresponding Number of Speakers Found (y-axis).*

are often off by no more than 1 speaker. Unfortunately, these results are not robust; they were obtained only after processing each utterance using rotation-only PCA and keeping only the first three principal components of the i-vectors. Possibly attributed to the curse of dimensionality, retaining more dimensions caused this method to "find" many more clusters (speakers) than was appropriate. Such a phenomenon could also be explained by the previously mentioned fact that Gaussians do not properly model data on a hypersphere.

6.4 VB Mixtures of the von Mises-Fisher Distribution

The more natural approach to modeling data on a unit hypersphere was introduced in Section 5.4, and it would be ideal to bring Variational Bayesian methods to bear on mixtures of von Mises-Fisher distributions. Unfortunately, because we are forced to use numerical approximation to estimate the concentration parameters $\{\kappa_1, ..., \kappa_K\}$

for each mixture component, no conjugate prior exists for this parameter and thus the derivation of VBEM is no longer straightforward.

Nevertheless, an attempt was made in [37] to apply Variational Bayes to Mix-vMF. In this derivation, the concentration parameters $\{\kappa_1, ..., \kappa_K\}$ were not given priors but were directly re-estimated in the VB M-step using the sufficient statistics obtained from the VB E-step. This maximum likelihood-esque modification made the resulting algorithm unstable at times; not having a prior distribution for each κ_h allowed the concentration parameters to vary freely from one iteration to the next. In an effort to mitigate this effect, our implementation enforced a non-memoryless prior of sorts that essentially prevented each κ_h from varying too much between subsequent iterations of VBEM.

Figure 6-4 shows the results obtained using our implementation of VB Mix-vMF. The instability of this method still manifests in the cases where 8+ speakers are found in two- or three-speaker conversations, but at least those occurrences are relatively infrequent. These results are dependent on the use of data-centered PCA dimensionality reduction on the i-vectors. Both the use of higher dimensions and the act of not centering the i-vectors at the origin would often cause the data to be modeled by a single component (i.e. only one speaker found). Nevertheless, even though the results are not yet as clean as in the VB-GMM case, this method shows promise. Some additional study is needed, but with further theoretical developments and a few more algorithmic refinements, there is potential for VB Mix-vMF to improve our model selection performance.

6.5 Discussion

In this chapter, we introduced the Variational Bayesian framework for model selection and applied it to the problem of using i-vectors to estimate the number of speakers in a conversation. We derived the updates for VB-GMM and found it to work surprisingly well despite the inherent inability of Gaussians to model spherical data. We also experimented with a modified version of VB Mix-vMF and saw promising preliminary results on that front.

Some recurring issues were present in our exploration of these methods. The results for VB-GMM were obtained using rotation-only PCA, which does not center the i-vectors at the origin. On the other hand, VB Mix-vMF failed consistently if the i-vectors were not centered. This notion of data pre-processing is a topic of wide discussion in Bayesian learning, as many methods require the data to be in some sense "standardized" beforehand. Yet in our case, the use of the cosine similarity metric

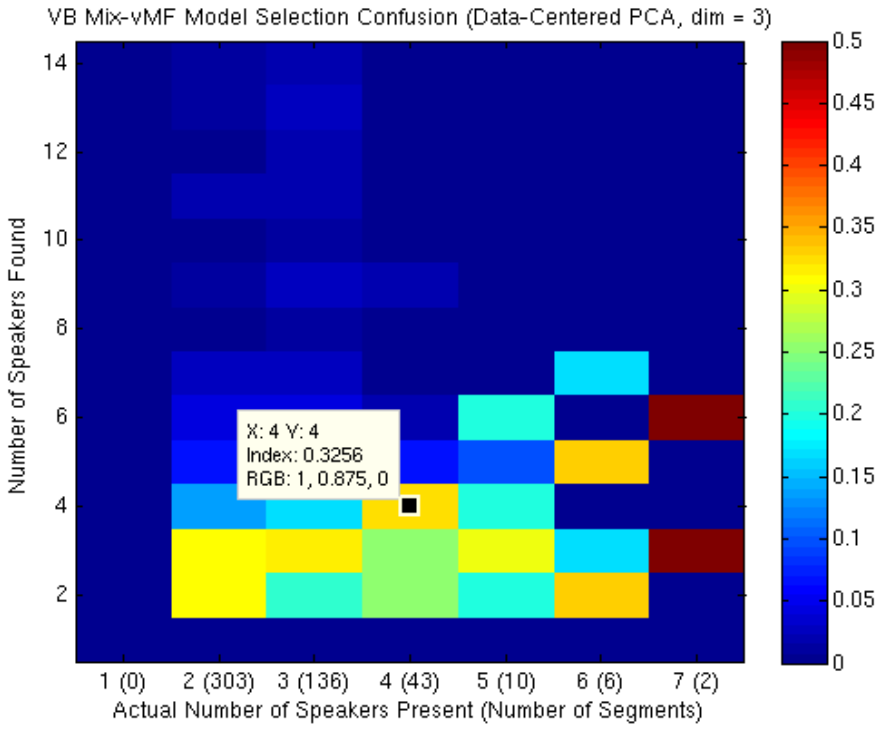

Figure 6-4: *Model selection results obtained by VB Mix-vMF applied to length-normalized i-vectors. For a given Actual Number of Speakers (x-axis), the colormap shows the proportion of those conversations that resulted in the corresponding Number of Speakers Found (y-axis).*

is not invariant to all types of pre-processing (e.g. translational bias); thus a better understanding of the inherent structure of our data is required for further progress.

The other issue is that of dimensionality. We were unable to obtain decent results on data-dimensions greater than three. This potentially boils down to the curse of dimensionality, as our initial i-vectors live in a high dimensional space (e.g. 100) and we have very few data points with which to perform our model selection and clustering. One avenue we could explore is that of low-dimensional embeddings (e.g. Johnson-Lindenstrauss Lemma [39]), which map our high dimesional dataset into a much lower dimensional space while preserving Euclidean or cosine distances.

Lastly, there exists a laundry list of Variational Bayesian methods, many of which are potentially effective in the diarization setting. For example, Variational Bayesian mixtures of Probabilistic PCA or mixtures of Factor Analyzers could model each speaker in his or her own subspace of the Total Variability space [36]. Also, we have yet to address the temporal aspects of a conversation. Previous approaches using Hierarchical Dirichlet Processes (HDP) applied to Hidden Markov Models (HMM)

have attained good results on the diarization task where the number of participating speakers is unknown [15]; it might also be worth extending the use of i-vectors to build upon previous work on VB-HMMs as an approach to this problem [10].

Chapter 7

Conclusion

In this thesis, we have taken an incremental approach to tackling the main challenges of speaker diarization. We utilized the effectiveness of factor analysis for extracting low-dimensional speaker-specific features to build a diarization system that achieves state-of-the-art results on two-speaker telephone conversations. That approach was extended to handle conversations containing K speakers, where the number of speakers K is given. Upon evaluation, that system was shown to be competitive with the current state-of-the-art benchmark. And finally, promising initial results were obtained in our approach to estimating the number of speakers present in a given conversation.

7.1 Summary of Methods and Contributions

Throughout our discussion, a wide variety of methods were introduced and developed. From the ground up, we explained the mechanics behind Total Variability and the i-vector representation starting from the GMM-UBM-MAP approach to speaker recognition, which then motivated the use of factor analysis on speaker supervectors. The general diarization algorithm was initialized using a spherical K-means algorithm that was later generalized probabilistically to involve mixtures of von Mises-Fisher distributions on a unit hypersphere. Subsequent steps in our system included a Viterbi-based Re-segmentation algorithm, and a Second Pass Refinement stage that was essentially a modified version of K-means using i-vectors and the cosine similarity metric.

In our work on two-speaker diarization, we explored the use of Principal Components Analysis to find the directions of maximum variability between the i-vectors of our given utterance, and in doing so, we analyzed the detrimental effect of introducing a translational bias on our angular metric. For K-speaker diarization, we introduced

a sampling scheme that increases the importance of longer, more reliable segments for the estimation of our speaker clusters.

Exploring ways to determine the number of speakers in a conversation required the introduction of an entire family of methods for Bayesian model selection. After providing the necessary background for understanding Variational Bayesian learning, we applied this framework to both Gaussian Mixture Models and mixtures of von Mises-Fisher distributions to obtain promising initial results.

The presentation of this thesis strives to tie together the various contributions of our work in a logical and cohesive manner. Via a thorough use of visualizations and simplified derivations, we seek to provide an intuitive explanation for our chosen methods and results. Ultimately, despite the relative complexity of some of our experiments, we found the techniques that worked best to be those that - to paraphrase Occam's razor - "were not unnecessarily complex." Indeed, most of our best results were obtained using the methods that were less intricate and involved, such as K-means without PCA-based dimensionality reduction and VB-GMMs instead of VB mixtures of vMF distributions. On one hand, this suggests fruitful areas for continued investigation; on the other, it shows that models need not be overly complex to be effective.

7.2 Directions for Future Research

In some ways due to the open and yet unsolved nature of speaker diarization, the work presented in this thesis raises more questions than it answers. Let us review some of the extensions mentioned previously and develop additional ideas for further work.

7.2.1 Overlapped Speech

Though there has been work on the separation of simultaneous speech, relatively little work has been done on the actual detection of overlapped speech segments [33]. It would be interesting to see if i-vectors can be used as features for this task; one could imagine building a simple two-class classifier via a Support Vector Machine, where one class consists of i-vectors extracted from clean, one-person speech, and another class is composed of i-vectors of overlapped segments. That said, the extraction of i-vectors assumes some form of preliminary segmentation; thus, it is not immediately clear how to handle cases in which a segment contains one second of clean speech and half a second of overlapped speech. To that end, one could imagine developing some

sort of an iterative procedure similar to that of the Viterbi Re-segmentation step that makes classifications at the acoustic feature level.

7.2.2 Slice-based Diarization

Another potential thread for further progress is to perform, in the temporal sense, local modeling of the audio stream. The system developed in this thesis performs the initial clustering on a "bag of i-vectors." That is, once the i-vectors are extracted from the given segmentation, the temporal structure of the conversation is essentially ignored for the time being. Though we proposed a duration-proportional sampling mechanism to alleviate this effect, there is much more that can be done. One possibility would be to adopt a divide-and-conquer sort of approach that the stream-based factor analysis system did in [19] and perform an initial diarization on a short "slice" of the conversation (say 30-60 seconds). The result of slice diarization would be i-vectors for each speaker in the slice that have been extracted using more frames and hence represent the speaker much better. Pooling these "speaker slice i-vectors" together across the entire conversation and then incorporating the duration-proportional sampling once again, where the number of samples drawn is proportional to the number of frames used to estimate each speaker slice i-vector, could potentially provide a much more robust representation of each speaker. In a sense, since each speaker slice i-vector would have been estimated using more frames than our average segment ($\sim$1 second), the resulting i-vector covariance would have a smaller range and our samples would, hopefully, be more concentrated about the true speaker models. Unfortunately, this procedure is potentially unstable in the event that our within-slice diarization is done poorly; each speaker slice i-vector could then contain speech from more than one speaker, causing us to "lose" a speaker (or more) in the process.

7.2.3 From Temporal Modeling to Variational Bayes

Other ideas related to temporal modeling draw from recent work on the Hierarchical Dirichlet Process-based (HDP) Hidden Markov Model (HDP-HMM) [15] and Hidden Semi-Markov Model (HDP-HSMM) [16], which achieved good results despite using only acoustic features. If those approaches were applied to the robust speaker representations of the i-vector framework, perhaps even better results can be obtained. To extend upon our Variational Bayesian approaches, we mentioned previously the potential use for building VB-HMMs on each conversation, as well as the potential use for VB mixtures of PPCA/FA. In this setting, we model each speaker in his or

her own subspace of the original i-vector space. As in the case of VB-GMMs, the use of VB-mixPPCA/FAs may not be appropriate for modeling data constrained to the unit hypersphere, but this would be an area to explore.

7.3 Beyond Speaker Diarization

A diarization system is usually used as a pre-processing step to extract relevant meta-data, such as speaker turns and sentence boundaries, to help provide a richer transcription of the audio. Its goal is to serve as a utility to help with tasks including speaker adaptation for automatic speech recognition, speaker detection from multi-speaker conversations, and audio indexing. In this sense, a better measure of diarization performance might not be how low the DER is, but rather how much it improves our performance on these sorts of tasks. Doing this sort of evaluation would provide insight into the usefulness of our system for a wide array of applications.

This thesis focused on developing methods for speaker diarization. Yet in the field of audio classification, there are still many avenues to explore. The ideas we have developed here need not be constrained to the setting of recorded conversations; we can broaden our scope. There is a growing number of audio databases that contain all kinds of useful and interesting information. On a database of music for example, applying similar methods to those we discussed can potentially help determine the number of artists present and maybe even associate similar artists and musical styles. Or imagine the notion of an audio diary that automatically records a person's acoustic experience throughout the day. Diarization on that front might involve classifying different audio environments (e.g. street, restaurant, concert) and other spontaneous events (e.g. door slams, ringtones) in addition to the usual inter-personal interactions. Development of these resources is not far off, but progress these areas begins with effective speaker diarization. This is where it starts; let's see where it takes us.

Bibliography

[1] D. Reynolds and P. Torres-Carrasquillo, "The MIT Lincoln Laboratory RT-04F diarization systems: Applications to broadcast audio and telephone conversations," in *NIST Rich Transcription Workshop*, 2004.

[2] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, 2000.

[3] N. Dehak, "Discriminative and generative approaches for long- and short-term speaker characteristics modeling: Application to speaker verification," Ph.D. dissertation, Ecole de Technologie Superieure de Montreal, June 2009.

[4] L. Burget, N. Brummer, D. Reynolds, P. Kenny, J. Pelecanos, R. Vogt, F. Castaldo, N. Dehak, R. Dehak, O. Glembek, Z. Karam, J. N. Jr., E. Na, C. Costin, V. Hubeika, S. Kajarekar, N. Scheffer, and J. Cernocky, "Robust speaker recognition over varying channels," Johns Hopkins University, Center for Language and Speech Processing, Summer Workshop, Tech. Rep., 2008.

[5] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, July 2010.

[6] N. Dehak, Z. Karam, D. Reynolds, R. Dehak, W. Campbell, and J. Glass, "A channel-blind system for speaker verification," in *Proceedings of ICASSP (to appear)*, 2011.

[7] Z. Karam and W. Campbell, "Graph-embedding for speaker recognition," in *Proceedings of Interspeech*, 2010.

[8] J. Glass, T. Hazen, S. Cyphers, I. Malioutov, D. Huynh, and R. Barzilay, "Recent progress in the MIT spoken lecture processing project," in *Proceedings of Interspeech*, August 2007.

[9] S. Tranter and D. Reynolds, "An overview of automatic speaker diarisation systems," *IEEE Transactions on Audio, Speech, and Language Processing*, September 2006.

[10] F. Valente, "Variational bayesian methods for audio indexing," Ph.D. dissertation, Universite De Nice-Sophia Antipolis - UFR Sciences, September 2005.

[11] S. Chen and P. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the bayesian information criterion," in *DARPA Broadcast News Workshop*, 1998.

[12] P. Kenny, D. Reynolds, and F. Castaldo, "Diarization of telephone conversations using factor analysis," *IEEE Journal of Selected Topics in Signal Processing*, December 2010.

[13] S. Meignier, D. Moraru, C. Fredouille, J.-F. Bonastre, and L. Besacier, "Step-by-step and integrated approaches in broadcast news speaker diarization," *Computer Speech and Language*, 2006.

[14] M. Beal, Z. Ghahramani, and C. Rasmussen, "The infinite hidden markov model," in *Proceedings of Neural Information Processing Systems*, 2003.

[15] E. Fox, E. Sudderth, M. Jordan, and A. Willsky, "An HDP-HMM for systems with state persistence," in *Proceedings of the International Conference on Machine Learning*, 2008.

[16] M. Johnson and A. Willsky, "The hierarchical dirichlet process hidden semi-markov model," in *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2010.

[17] D. Blei and M. Jordan, "Variational inference for dirichlet process mixtures," *Bayesian Analysis*, 2006.

[18] M. Wainwright and M. Jordan, "Graphical models, exponential families, and variational inference," *Foundations and Trends in Machine Learning*, 2008.

[19] F. Castaldo, D. Colibro, E. Dalmasso, P. Laface, and C. Vair, "Stream-based speaker segmentation using speaker factors and eigenvoices," in *Proceedings of ICASSP*, 2008.

[20] D. Reynolds, P. Kenny, and F. Castaldo, "A study of new approaches to speaker diarization," in *Proceedings of Interspeech*, 2009.

[21] A. Martin and M. Przybocki, "Speaker recognition in a multi-speaker environment," in *Proceedings of Eurospeech*, 2001.

[22] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, 1977.

[23] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Speaker and session variability in GMM-based speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, 2007.

[24] M. Jordan, "An introduction to probabilistic graphical models," 2003, Chapter 14 - Factor Analysis.

[25] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of inter-speaker variability in speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, July 2008.

[26] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE Transactions on Speech and Audio Processing*, May 2005.

[27] S. Shum, N. Dehak, R. Dehak, and J. Glass, "Unsupervised speaker adaptation based on the cosine similarity for text-independent speaker verification," in *Proceedings of IEEE Odyssey*, 2010.

[28] A. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," in *Proceedings of ICSLP*, 2006.

[29] O. Glembek, L. Burget, N. Dehak, N. Brummer, and P. Kenny, "Comparison of scoring methods used in speaker recognition with joint factor analysis," in *Proceedings of ICASSP*, 2009.

[30] E. Chuangsuwanich, S. Cyphers, J. Glass, and S. Teller, "Spoken command of large mobile robots in outdoor environments," in *Proceedings of the IEEE Spoken Language Technologies Workshop*, 2010.

[31] NIST, "Diarization error rate (DER) scoring code," 2006, www.nist.gov/speech/tests/rt/2006-spring/code/md-eval-v21.pl.

[32] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[33] K. Boakye, O. Vinyals, and G. Friedland, "Two's a crowd: Improving speaker diarization by automatically identifying and excluding overlapped speech," in *Proceedings of Interspeech*, 2008.

[34] A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra, "Clustering on the unit hypersphere using von mises-fisher distributions," *Journal of Machine Learning Research*, 2005.

[35] G. Schwarz, "Estimation of the dimension of a model," *Annals of Statistics*, 1978.

[36] M. Beal, "Variational algorithms for approximate bayesian inference," Ph.D. dissertation, University College London, May 2003.

[37] A. Tanabe, K. Fukumizu, S. Oba, and S. Ishii, "Variational bayes inference of mixtured von mises-fisher distribution," in *Proceedings of Workshop on Information-Based Induction Sciences*, 2004, in Japanese.

[38] E. Khan and J. Bronson, "Variational bayesian EM for gaussian mixture models," 2008, http://www.cs.ubc.ca/ murphyk/Software/VBEMGMM/index.html.

[39] W. Johnson and J. Lindenstrauss, "Extensions of lipschitz mappings into a hilbert space," *Contemporary Mathematics*, 1984.