Use of Speaker Location Features in Meeting Diarization

Scott Otterson

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Washington

2008

Program Authorized to Offer Degree:
Electrical Engineering
University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Scott Otterson

and have found that it is complete and satisfactory in all respects, and that any and all revisions required by the final examining committee have been made.

Chair of the Supervisory Committee:

________________________________________
Mari Ostendorf

Reading Committee:

________________________________________
Dan Ellis

________________________________________
Maya Gupta

________________________________________
Mari Ostendorf

________________________________________
John Sahr

Date: ________________________________
In presenting this dissertation in partial fulfillment of the requirements for the doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this dissertation is allowable only for scholarly purposes, consistent with “fair use” as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to Proquest Information and Learning, 300 North Zeeb Road, Ann Arbor, MI 48106-1346, 1-800-521-0600, to whom the author has granted “the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform.”

Signature_________________________

Date_____________________________
University of Washington

Abstract

Use of Speaker Location Features in Meeting Diarization

Scott Otterson

Chair of the Supervisory Committee:
Professor Mari Ostendorf
Electrical Engineering

This thesis proposes several improvements to the correlation-based location features recently used in meeting speaker diarization (answering the question, "Who spoke when?"). The problem of leveraging time delay information is examined for multi-microphone meeting environments, where microphones are placed at unknown, widely spaced, and ad-hoc locations. In addition, conversational speech is challenging because of the many short utterances and speaker overlaps. Finally, assuming no room constraints, the microphone configuration and acoustic environment changes from meeting to meeting. Together, these conditions make it impractical to apply standard localization and beamforming techniques. To address these challenges, we first consider what combination of channel pairs and signal processing to use for location information extraction. Initially, we consider all pairs, then de-emphasizing low quality pairs with feature vector dimension reduction. We also develop an approach for fusing speaker ID information as viewed by different physical processes. Two views are a new time delay estimate and multi-band energy ratios (cues to location) and a third is a vector of mel-warped cepstral coefficients (MFCC's), related to vocal tract characteristics. We find that both MFCC's and energy ratios can improve time delay information when jointly transformed using canonical correlation analysis (CCA). Oracle experiments show that the location feature dimension producing the best di-
arization error varies with meeting. Therefore, we evaluate automatic methods for determining feature reduction output dimension. In addition, we separately consider reducing the feature dimension by explicitly selecting subsets of channel pairs using estimated signal to noise ratio (SNR) and information-theoretic feature selection methods. Location features are also employed to detect speaker overlap, a significant cause of increased speaker diarization error. First, monaural overlap features are developed for a single channel beamformer output. These features are then compared to overlap detector features which make use of location information, but neither type provides good performance due to a high degree of variation across meetings. We also develop a simple, nearest-neighbor overlap processing scheme which, when given accurate overlap detection, improves diarization accuracy. Together, these results underscore the need for dynamic models to handle variable room and recording configurations.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Figures</td>
<td>iv</td>
</tr>
<tr>
<td>List of Tables</td>
<td>v</td>
</tr>
<tr>
<td>Chapter 1: Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Speaker Diarization Challenges in Meetings</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Contributions</td>
<td>4</td>
</tr>
<tr>
<td>1.3 Thesis Overview</td>
<td>6</td>
</tr>
<tr>
<td>Chapter 2: Background</td>
<td>7</td>
</tr>
<tr>
<td>2.1 Meeting Diarization</td>
<td>7</td>
</tr>
<tr>
<td>2.1.1 Speaker Boundary Detection</td>
<td>7</td>
</tr>
<tr>
<td>2.1.2 Clustering</td>
<td>10</td>
</tr>
<tr>
<td>2.2 Multi-Microphone Localization Features for Diarization</td>
<td>11</td>
</tr>
<tr>
<td>2.2.1 Localization By Time Delay Estimation</td>
<td>12</td>
</tr>
<tr>
<td>2.2.2 Localization by Energy Ratios</td>
<td>16</td>
</tr>
<tr>
<td>2.2.3 Localization by Speaker Separation</td>
<td>17</td>
</tr>
<tr>
<td>2.3 Overlap Detection</td>
<td>21</td>
</tr>
<tr>
<td>Chapter 3: Corpus</td>
<td>23</td>
</tr>
<tr>
<td>3.1 Dev and Test Meeting Partitioning</td>
<td>23</td>
</tr>
<tr>
<td>3.2 Reported Diarization Performance on the Source Corpora</td>
<td>25</td>
</tr>
<tr>
<td>3.3 Sources of Meeting variability</td>
<td>28</td>
</tr>
<tr>
<td>Chapter 4: New Location Features</td>
<td>32</td>
</tr>
<tr>
<td>4.1 Limits of Single-Reference Correlations</td>
<td>32</td>
</tr>
<tr>
<td>4.2 Limitations of GCC-PHAT</td>
<td>35</td>
</tr>
</tbody>
</table>
4.3 Energy Ratios ........................................... 37
4.4 Dimension Reduction .................................. 39
  4.4.1 Principal Components Analysis ....................... 40
  4.4.2 Canonical Correlation Analysis ....................... 43
4.5 Experiments .......................................... 45
  4.5.1 Experimental Paradigm .............................. 45
  4.5.2 Results ........................................... 48
  4.5.3 Summary .......................................... 52

Chapter 5: Automatic Dimension Choice .................... 54
  5.1 Automatic choice of PCA dimension .................... 57
  5.2 Feature Selection .................................... 62
  5.3 Experiments .......................................... 65
    5.3.1 PCA dimension choice ............................ 66
    5.3.2 Feature Selection ................................ 71
  5.4 Summary ............................................. 74

Chapter 6: Overlap Handling .................................. 78
  6.1 Use of Overlap Information ............................ 78
    6.1.1 Questions in Overlap Handling ..................... 79
    6.1.2 Experimental Paradigm ............................ 80
    6.1.3 Cluster quality and location feature robustness .... 80
    6.1.4 Nearest-2 cluster result post-processing .......... 81
    6.1.5 Summary ........................................ 84
  6.2 Monaural Overlap Detection ........................... 85
    6.2.1 Pitch Detection and Hough Transform ............... 86
    6.2.2 Voiced Speech Spectral Overlap Detection Features .. 91
    6.2.3 Classifier Design ................................ 95
    6.2.4 Monaural overlap detection on synthetic TIMIT data . 98
    6.2.5 Monaural overlap detection on meeting data ....... 102
  6.3 Overlap Detection Using Location Information .......... 104
    6.3.1 Speaker pair model fit ................................ 104
    6.3.2 Delay Triplet Sums ................................ 106
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure Number</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>2</td>
</tr>
<tr>
<td>2.1</td>
<td>8</td>
</tr>
<tr>
<td>2.2</td>
<td>9</td>
</tr>
<tr>
<td>4.1</td>
<td>34</td>
</tr>
<tr>
<td>4.2</td>
<td>35</td>
</tr>
<tr>
<td>4.3</td>
<td>39</td>
</tr>
<tr>
<td>4.4</td>
<td>42</td>
</tr>
<tr>
<td>4.5</td>
<td>47</td>
</tr>
<tr>
<td>5.1</td>
<td>58</td>
</tr>
<tr>
<td>6.1</td>
<td>81</td>
</tr>
<tr>
<td>6.2</td>
<td>82</td>
</tr>
<tr>
<td>6.3</td>
<td>83</td>
</tr>
<tr>
<td>6.4</td>
<td>87</td>
</tr>
<tr>
<td>6.5</td>
<td>96</td>
</tr>
<tr>
<td>6.6</td>
<td>100</td>
</tr>
<tr>
<td>6.7</td>
<td>111</td>
</tr>
<tr>
<td>6.8</td>
<td>112</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table Number</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 NIST Meeting Microphones</td>
<td>24</td>
</tr>
<tr>
<td>3.2 Dev Meetings</td>
<td>26</td>
</tr>
<tr>
<td>3.4 Test Meetings</td>
<td>27</td>
</tr>
<tr>
<td>3.6 Diarization Error Performance in Previous NIST competitions</td>
<td>28</td>
</tr>
<tr>
<td>3.7 Factors Conceivably Affecting Diarization Error (XC+ER)</td>
<td>30</td>
</tr>
<tr>
<td>3.8 Diarization Error / SNR Correlation Coefficient</td>
<td>30</td>
</tr>
<tr>
<td>4.1 Delay / Energy Ratio Complementarity</td>
<td>38</td>
</tr>
<tr>
<td>4.2 Error vs. Features and Dimension Reduction</td>
<td>50</td>
</tr>
<tr>
<td>5.1 Diarization error loss due to fixed PCA dimension (Dev Set)</td>
<td>55</td>
</tr>
<tr>
<td>5.2 Diarization error loss due to fixed PCA dimension (Test Set)</td>
<td>56</td>
</tr>
<tr>
<td>5.3 Diarization Error vs. PCA dimension choice: standard techniques and regression</td>
<td>68</td>
</tr>
<tr>
<td>5.4 Diarization error for minimum entropy PCA dimension choice</td>
<td>71</td>
</tr>
<tr>
<td>5.5 Max-Min SNR Feature Selection</td>
<td>72</td>
</tr>
<tr>
<td>5.6 Diarization error for mRMR feature reduction ((nChans - 1 \text{ XC pairs}))</td>
<td>74</td>
</tr>
<tr>
<td>5.7 Diarization error for mRMR feature reduction (fixed or oracle number of XC pairs, Dev)</td>
<td>75</td>
</tr>
<tr>
<td>5.8 Diarization error for mRMR feature reduction (fixed or oracle number of XC pairs, Test)</td>
<td>76</td>
</tr>
<tr>
<td>6.1 Effect of input overlap processing on single-talker diarization accuracy</td>
<td>81</td>
</tr>
<tr>
<td>6.2 Effect of input/output overlap processing on full diarization accuracy</td>
<td>85</td>
</tr>
<tr>
<td>6.3 Hough Transform Pitch Detection Accuracy vs. Input Preprocessing</td>
<td>91</td>
</tr>
<tr>
<td>6.4 Overlap Detection Performance Using All Features</td>
<td>99</td>
</tr>
<tr>
<td>6.5 Overlap Detection Performance vs. Left-Out Pre-processing or Feature</td>
<td>101</td>
</tr>
<tr>
<td>6.6 Monaural Overlap Detection Performance on Meeting Data (GMM classifiers)</td>
<td>103</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENTS

Thanks, Mari, for ensuring quality work, and thanks, Folks, for ensuring me.
Chapter 1

INTRODUCTION

In some multi-party meetings such as legislative proceedings, board meetings, and courtrooms, a rich transcription is produced – a written document describing what was said and who said it. Human-produced transcription is time consuming and expensive; many important meetings are not transcribed. Once automatic transcription becomes available, new applications are possible. Transcriptions can be automatically summarized or translated, made available to remote participants, and archived for later public or corporate search.

High quality acoustic information can be obtained when there are several microphones, arranged in an array designed for optimal beamforming. But in many practical situations, microphone placement will be haphazardly, being located for convenience instead of acoustic quality. We concentrate on this ad-hoc placement scenario.

Figure 1.1 outlines the steps that might be involved in an automatic transcription system. Speech from several talkers arrives mixed in the waveforms from multiple microphones. The speaker diarization function detects speech start and stop times and identifies the speaker (speaker tracking). It also combines the waveforms from multiple waveforms into one time series for each single talker segment. Since speakers overlap frequently, this task also involves some form of overlap handling. The result is fed to an automatic speech recognition (ASR) system which decodes waveforms into words. Finally, a meta-data extractor identifies structural information such as sentence and topic boundaries, speaker roles, etc. and converts the words and accompanying speaker labels into a meeting transcript. This thesis focuses on the first step
in this process, speaker diarization, which is a limiting factor for all subsequent steps.

This thesis will examine algorithms for identifying the speech of arbitrarily positioned talkers in spontaneous meeting conversation, given the signals from a set of arbitrarily positioned microphones. Speaker diarization is an important step for several reasons. First, the resulting signal matches single-talker ASR acoustic models. Second, the knowledge of speaker transitions can provide important cues to a multi-talker ASR language model \[63\] and to automatic classification of different types of meetings \[68\]. Finally, the talker sequence is itself useful when trying to understand the semantic meaning of the recognized words. For example, in a negotiation, it is important to know who said, “yes.”

### 1.1 Speaker Diarization Challenges in Meetings

Most state-of-the-art diarization work has been done for the NIST meetings project \[44\], where several labs \[108, 65, 89, 42, 24, 82\] competed to transcribe multi-microphone meetings under two main microphone conditions: using the signals of personal microphones – heads or lapel microphones attached to each speaker – or from several distant omni directional microphones in varying configurations. The results were scored in
terms of ASR word error rate (WER), and a “diarization” (speaker tracking) score that measured whether or not segments containing words were assigned to the correct talker.

For either type of scoring, speaker overlap is a serious problem. First, overlaps were extremely common: In meeting conversation, one study has found that 17% of words and 54% of sentence-like “spurts” are interrupted. When words were assigned to the smallest silence bounded region that contained them, 78% occurred in a region with speaker overlap \[104\]. Second, when overlaps occur, performance dropped significantly. On state-of-the-art transcription systems, WER increased from 45% in non-overlapped regions to 56% in overlapped regions; diarization errors increased from 22.8% to 39.5%.

Speaker diarization is particularly challenging for meeting recordings in which a variable (meeting-dependent) number of microphones are placed at unknown distances from the speakers in an unknown (ad hoc) configuration. In the NIST meeting recognition competition \[44\], this is known as the multiple-distant-microphone (MDM) test condition, and it is difficult for several reasons. First, conversational speech switches rapidly from speaker to speaker, limiting the time window available for speaker tracking and separation to as little as 0.5s. Most existing speaker tracking and separation algorithms have poor performance for utterances shorter than sentence-length. This is acceptable for some conversational speech applications, where a “yeah” is frequently ignorable back-channel information. But in a business meeting, the “yeah” may be significant.

The central assumption of this thesis is that each of these problems can be mitigated if more use is made of location. Specifically, the goals are:

- to develop new pattern recognition features which provide cues related to speaker location in order to enhance diarization;

- to assess the impact on diarization of those features in various conditions; and
• to also use the features to detect and process speaker overlaps.

For a given talker, the time delay between microphones must be estimated for two reasons. First, inter-microphone delays will be one component of the location features mentioned above, and second, beamforming requires these delays.

In the multiple distant microphone test condition, time delay estimation is especially difficult because microphone distances are unknown and are nearly always farther apart than half the wavelength of even the lowest speech frequencies. This causes spatial aliasing, violating assumptions that underlie many multi-microphone speech techniques. Or standard techniques may require side information not available for meeting data. For example, the spatial aliasing correction method found in [34] requires a known calibration signal. These meeting data idiosyncrasies, combined with the requirement of an extremely short analysis window, prevent us from using sophisticated time delay estimation techniques such as those based on eigen analysis [16], either because of unmet assumptions, or because of the numerical stability issues inherent in estimating a long correlation function with little data.

Nevertheless, crude delay estimates are useful. Empirically, they have been proven adequate for the simple delay-sum beamformer used in this work. Also, since diarization does not require exact speaker locations, they have been found to be useful location features.

1.2 Contributions

The main contributions of this work are new location features for use in diarization, new methods for dimensionality reduction for location features, and new algorithms for handling speaker overlaps. We show how a speech-specific time delay estimator can improve diarization over the most commonly used method, and how all-pairs correlations, when combined with feature reduction, can improve results over smoothed single reference correlations, as in other work [9]. In addition to new location fea-
tures based on time delay, we introduce a complimentary location-dependent energy ratio feature which in some cases, results in further improvement in diarization. We find that location feature dimension reduction improves noise robustness more effectively than dynamic programming approaches. In the process, we also show how canonical correlation analysis (CCA) dimension reduction makes it possible to enhance location-dependent delay information with traditional speech features related to speaker vocal tract characteristics. Using the same technique, delay information can also be improved with energy ratios.

Using oracle experiments, we show that the dimension reduction output dimension yielding the best diarization performance is meeting-dependent. Therefore, we evaluate several dimension choice methods, the best of which is a regression upon certain meeting acoustic properties.

Instead of using a transform to reduce the all-pairs feature dimension, it is also possible to simply select the best pairs. We show that an information-theoretic approach to this feature selection problem is superior to a method based on signal-to-noise ratio (SNR). We also show how to use this typically supervised technique in this unsupervised application. Feature reduction and feature transformation are similarly effective.

Location features are also employed to detect speaker overlap, a significant cause of increased speaker diarization error. First, monaural overlap features are developed for a single channel beamformer output. These features are then compared to overlap detector features which make use of location information, but neither type provides good performance due to a high degree of variation across meetings. We also develop a simple, nearest-neighbor overlap processing scheme which, when given accurate overlap detection, improves diarization accuracy.
1.3 Thesis Overview

In the Chapter 2, we review past work relevant to meeting speaker diarization, specifically, past diarization systems, indicators of speaker location, and methods for detecting and handling overlaps. In Chapter 3, we describe the data used for this thesis, as well as results reported on it by other researchers. We then develop several new location features derived from measures of time delay, position-dependent attenuation, and traditional speaker vocal tract characteristics (Chapter 4). In Chapter 5, we explore automatic methods for determining the best dimension, comparing transformation-based dimension reduction and feature selection. In Chapter 6, we propose new algorithms for overlap detection. First, we develop new monaural (single channel) features on synthetic data, where overlap timing and phonetic content is known, and then we test on overlaps found in real meeting data. We then develop a technique employing both monaural and location features. In Chapter 7, we conclude the thesis with a summary of results and a discussion of future work.
Chapter 2

BACKGROUND

Meeting diarization is a fairly new research problem, but much work has already been done. In this chapter, we first introduce the functions typically present in diarization systems built to date. Then, since location features are prominent in this thesis, we review the speaker localization literature where it is applicable to the meeting diarization task presented in the NIST meeting data. Finally, past work in overlap detection is discussed.

2.1 Meeting Diarization

Figure 2.1 outlines the functions present in a typical speaker diarization system. Waveforms are used to generate pattern recognition features for clustering. Most diarization systems then remove features from the data stream during times when speech is not present, according to a voice activity detector (VAD) algorithm. The next step in the process is to estimate the boundaries between speaker changes. An agglomerative clustering algorithm then uses those boundaries to iteratively build and refine single-speaker models, eventually converging on an estimate of the number of speakers and the time when each one of them spoke. Provisions for speaker overlap are absent from typical diarization systems.

The details of these functions are discussed in the following sections.

2.1.1 Speaker Boundary Detection

Speaker clustering systems often attempt to automatically detect breaks between regions of acoustic similarity. Subsequent speaker modeling steps may re-merge similar
segments but, since most systems do not generate new boundary candidates after the initial segmentation, speaker change detectors are designed to generate far more boundary insertion than deletion errors [76].

The longer the speech segments produced by a voice activity detector, the more likely it becomes that speakers have changed with almost no silence between them [44]. Therefore, it is often necessary to check for speaker change with a method in addition to speech/non-speech detection. The most common method is to create a window with a sliding partition dividing the segment into two sections – $X_a$ and $X_b$ in Figure 2.2. The boundary between $X_a$ and $X_b$ is advanced from left to right across the segment being examined, and for each time step of the advancement, speech feature statistics are collected on both sides. When a distance metric between the two distributions is large, a speaker change boundary candidate is declared.

As first proposed in [46], most systems with explicit speaker change detection use the generalized likelihood ratio (GLR) distance metric. In this approach, Gaussian speech models are trained individually on $X_a$ and $X_b$ and also on $X_{ab}$; the GLR is the ratio of the likelihood product, $L(X_a)L(X_b)$, as calculated by each segment’s own model, to the likelihood, $L(X_{ab})$, as calculated by the joint model trained on $X_{ab}$. Authors have added terms to the GLR metric which are intended to intended
to compensate for differences in the length of $X_a$ and $X_b$ [87], and have extended the speaker likelihood model from single Gaussians to GMM’s, adding a back-off algorithm [35], which replaced large distance or low probability values with a constant in cases where there were few training samples.

The GLR distance metric provides a score for the merit of modeling the data with one vs. two statistical models. Another standard model selection metric is the Bayesian Information Criterion (BIC), which trades off model complexity and model fit by subtracting a model complexity penalty from the log GLR. In [29, 36, 67], single Gaussian speaker models were built for evenly spaced segments; a boundary was declared when the BIC value reached a threshold tuned on separate training data [76]. The BIC provides very accurate boundaries for longer segments [36, 87] but it is vulnerable to short segment Gaussian covariance matrix ill-conditioning; as many as 78% of segments shorter than 1s have been missed [29]. One of the most systematic boundary detection studies, [67], compared boundary candidates derived from energy-based VAD, GMM models, KL-distance, BIC, and GLR. The study concluded that the GLR is the best of the tested alternatives. In a successful attempt at getting the benefits of both the GLR and BIC methods, the authors of [36] devised a hybrid GLR boundary generator with BIC refinement.

In early speaker diarization experiments, it was thought that explicit speaker change detection was necessary. However, the best performing recent systems do not
explicitly detect change boundaries, but instead determine them during the process of clustering, as discussed in Section 2.1.2.

2.1.2 Clustering

Clustering collects acoustically similar time segments, removing false initial boundary candidates, and grouping together distant same-talker segments. For ASR, the advantage of clustering is that it allows ASR speech models to adapt over longer speech segments, yielding 10-25% relative WER improvement over non-clustered adaptation [64]. For speaker diarization, the clustering is the mechanism used to obtain speaker ID's. Purely in terms of speaker change boundary identification accuracy, clustering has improved the speaker change boundary equal error rate (EER) from 26.5% to 18% [76]. Even when speakers are separated with state-of-the-art microphone array beamforming, speaker ID's have been crucial to improving WER [71].

Once speaker change boundary candidates are defined by VAD, speaker change detection, or by simple assignment of linearly spaced segments of fixed duration (the approach used in this thesis) most speaker clustering systems use iterative agglomerative clustering. In the first iteration, each segment defined by the initial segment boundary candidates is placed in its own cluster. On the first clustering iteration the two segments (clusters) most similar according to a distance metric, e.g. the GLR, are merged into one cluster. GLR robustness to short segments has been improved in [52] by segment-length sensitive methods.

At the next merging iteration, the inter-cluster distances are re-calculated and, again, the two most similar clusters are merged. Merging ceases when a stopping criteria is met, at which point there will ideally be one cluster for each speaker and each cluster will contain only data from that speaker. Some systems have stopped merging when the summed inter-cluster distance cannot be decreased by grouping clusters but most have stopped based on the change in BIC. BIC methods are undesirably sensitive to speaker cluster size and speaker order, so a promising new method based
on information change rate has been proposed [51], although the method has not been tested on a large meeting corpus with separate development and training sets.

Almost all clustering algorithms require tuning of the stopping criteria parameters. An exception is [2], where cluster quality was improved (in terms of accurate speaker identification) without this tuning. In that work, each initial segment is assigned a state in a hidden Markov model (HMM). The states corresponding to the two most similar segments according the GLR test are merged; the HMM is retrained using an expectation-maximization algorithm; and the speech is re-segmented using the Viterbi algorithm. The process repeats until merging yields no improvement in the GLR. One important difference between this algorithm and the others is that during merge distance testing, the merged distributions are estimated with the same total number of speech model mixtures as with the two unmerged distributions. This accomplishes a BIC-like model fit vs. complexity trade-off, and on Broadcast News data, it has been especially successful at correctly determining the number of speakers. Because correctly determining the number of speakers is crucial for meeting transcription, and because HMM clustering has already been used for the related problem of known microphone position speaker and location clustering [4], a variant of this algorithm is used in this thesis.

On a small data set, diarization error comparable to the baseline system used here was obtained in [115]. In this approach speakers were clustered by maximum mutual information using the agglomerative information bottleneck method [105] with a minimum description length [105] stopping scheme. The main advantage of this methods is that is uses one third of the computational resources of our baseline system.

2.2 Multi-Microphone Localization Features for Diarization

In traditional speaker tracking, cepstral-based speech models are used to cluster speakers by how they spoke; in multi-microphone meetings, speakers can also be identified by where they spoke. Since the talker’s location varies more slowly than
speech characteristics, it is possible to build statistical location models with less data than that required for cepstral speech models, and thus improve diarization results on shorter segments.

Some diarization systems \cite{11} have already used speaker azimuths estimated by beamforming signals from microphone arrays in known locations. The most successful MDM system in the NIST 2006 competition \cite{11} used “location features,” the set of correlation lags computed between microphones placed at unknown location. The use of location features reduced diarization error by 15.1\% \cite{93}.

A problem inherent with multiple distant microphone channels is how to convert several channels of incoming waveforms to the single channel required by an ASR system. Initially, meeting transcription systems simply chose a single channel. Some chose the most centrally located microphone throughout the entire meeting, but the single system with the best distant microphone performance \cite{65} selected a different channel for each segment based on an energy and signal-to-noise quality factor. However, better results were obtained in \cite{108}, which combined all channels with a delay and sum beamformer, with the delays being calculated by microphone pair cross-correlation on long segments up to several seconds in duration. On that system, delay and sum beamforming improved WER by 6.6\% over selecting a fixed channel. In this system, time delay estimates were used for improving traditional acoustic speaker ID features, as well as being a feature themselves.

In the following section, we review time-based localization alternatives in order to motivate directions that we explore in feature extraction.

2.2.1 Localization By Time Delay Estimation

In reverberant environments such as a meeting room, the generalized cross-correlation phase transform algorithm (GCC-PHAT) is frequently proposed \cite{69, 48} for time difference of arrival (TDOA) estimation. In GCC-PHAT, it is assumed that coherence breaks down for microphones far apart from each other and therefore, time delays
are computed for correlation peaks between only adjacent microphone pairs. Since destructive interference may occur at only certain frequencies, the correlations are computed in individual frequency bands, yielding a series of phase delay estimates. The phases are then combined to form a single TDOA estimate. The TDOA estimates for each pair may then be triangulated across microphone pairs to obtain a speaker location.

The GCC-PHAT time delay estimator is popular for use in acoustic localization in reverberant environments. Although it is general purpose and not tailored to speech, it is employed by many researchers concentrating on localizing speech, including [11]. In GCC-PHAT, the cross-correlation is calculated in the frequency domain, after the cross-spectra whitening. However, other researchers [100] have found that, for speech, GCC-PHAT is outperformed by speech-specific algorithms which correlate the Hilbert envelopes of inverse linear-prediction-filtered waveforms. Inverse filtering performs roughly the same function as GCC-PHAT whitening; frequencies are still effectively being divided by their magnitudes, but with inverse filtering, the denominator is smoothed by the spectral envelope of speech, thus avoiding noise pumping in spectral zeros. Viewed in the time domain, the Hilbert envelope calculation, which smooths out easily aliased periodic information in the glottal transient oscillations, is similarly speech-specific. Finally, the delay features are the lags at the peak of standard correlations between the Hilbert envelopes of the inverse-filtered channel waveforms.

Several researchers [49, 10] have found that diarization performance is improved if waveforms are noise reduced with a Wiener filter based algorithm before the correlation calculation. In the Hilbert transform / inverse filtering results to follow, we did not find this to be the case. We speculate that the speech-specific correlation feature processing is itself a type of noise reduction, and therefore does not benefit from this step.

Other, older TDOA estimates include an adaptive filter approach, which predicts
the signal from one microphone with the signal from the other microphone in the pair [26]. Techniques such as the above have been criticized in [22, 47] in the case of speech in reverberant rooms. They maintain that the non-stationarity and wide bandwidth of speech, when combined with room reverberation causes algorithms like these to break down. They propose instead, extracting and tracking pitch pulses with wavelet and/or linear predictive coding (LPC).

Several newer approaches to TDOA have demonstrated superior results to the GCC-PHAT algorithm. In [16], reverberation is explicitly modeled as a finite impulse response (FIR) filter, which is solved for via an eigen computation over the microphone pair cross-correlation matrix. Unfortunately, this method is not ideal for the up to 3m microphone spacing present in the meeting data used for this thesis, as it requires estimating an extremely long filter. A frequency domain approach similar to GCC-PHAT but which uses all microphone pairs was studied in [27], but it is not usable for this data since microphone locations must be known. Another frequency domain technique which uses only a single microphone pair, and which avoids a long time domain eigen computation, was demonstrated in [27]. This approach also does an eigen approximation, but utilizes a non-stationarity property to distinguish speech from stationary noise. This technique does not require known microphone locations, but would require a very long discrete Fourier transform (DFT) window, and was thus not considered in this study.

Other techniques using more than one microphone pair include [14], in which a threshold scheme determines location based on a histogram of time delay estimates, and [109], in which a neural net is trained on correlation peak histograms for each position. The work most closely related to the problem posed for this thesis is [4], where location features were derived from microphones in a known ring configuration, combined with other speaker ID features, and then clustered in order to segment speakers. All of these techniques could benefit from the speech-specific knowledge incorporated into the algorithms of [22, 47], where delays are calculated from the
linear prediction (LP) residual.

The algorithms above break down when there is more than one speaker. For a standard microphone array – microphones closely spaced and in known positions – it is possible to use the ROOT-MUSIC algorithm, where the number and location of sources are estimated jointly. Usually applied to radar or sonar signals, ROOT-MUSIC has also been used in speaker localization [31]. Another reverberance and overlap resistant approach is found in [80], in which a two-microphone phase delay is estimated in each time/frequency bin of microphone pair cross-coherence spectra; a time-frequency mask is then estimated with an expectation maximization algorithm (EM).

There are one or two problems with the above algorithms. First, most require known microphone positions, something not available in this work. Second, they are all vulnerable to “spatial aliasing,” where the microphone cross-correlation waveforms will have maxima corresponding to more than one source location. Spatial aliasing occurs when sensors are spaced at distances less than half the wavelength of the source signal. In the ICSI meeting data [80], a subset of the data to be used for this thesis, inter-microphone spacing is approximately 1m. This limits the maximum alias-free speech frequency to approximately 175Hz. Hence, spatial aliasing is inevitable.

An example of a beamforming-type localization algorithm which can handle spatial aliasing to some degree is the algebraic maximum beamformer output method in [28]. Unfortunately, this method uses an eigen decomposition to derive time domain filters of length proportional to the maximum inter-microphone delay. For widely spaced microphones present in the NIST meeting data, this approach is computationally infeasible. Other techniques which localize by finding maximum beamformer output power eschew the eigen calculation. Instead, they employ a brute force search – in [130], a hierarchical, multi-resolution search in Cartesian space, and in [97], a search in spatial frequency. However, they again require known microphone locations.
A technique which may be robust to multiple speakers as well as widely spaced microphones in unknown locations is found in [80], where the phase delay estimation is done in time/frequency cells of a microphone pair cross-spectrum. The “microphone pair” in this system was a simulated pair of human ears. Given the close sensor spacing, spatial aliasing would be less of a problem than with meeting data, however, the technique models delay noise in such a way that spatial aliasing becomes part of the speaker location characteristic. An EM algorithm is developed for training Gaussian mixture model (GMM) location models, an approach which would fit in well with the clustering system used for this thesis. Unfortunately, there was not enough time to experiment with this approach.

Given the limitations of many of the above approaches, and given the empirical success of the time delay estimation algorithms described in [11], we take the same approach for our diarization baseline.

2.2.2 Localization by Energy Ratios

Given the problem of widely spaced microphones in unknown positions – unanticipatable reverberation and spatial aliasing – some degradation of correlation features is unavoidable. Sound intensity decreases with distance, and in addition, room reverberation imposes location and frequency-dependent zeros over the sound spectrum. For cross-correlations, these are both problems because they make channel waveforms dissimilar, lowering correlation peaks. For energy ratios, however, this dissimilarity is a speaker location feature. Humans are known to use inter-aural intensity difference – a kind of energy ratio – for localization [55].

A class of algorithms use a human-inspired energy difference between two microphones to localize sound, e.g. [88]. Others researchers have used energy ratios for diarization [62, 50]. However, these were with personal microphones, known to be close to the mouths of the speakers – an easier case – and were not multi-band, meaning that the energy ratios did not characterize detailed frequency and position-
dependent room reverberation.

2.2.3 Localization by Speaker Separation

Another view of localization is as a speaker separation problem. In the case of single talker speech, the problem is to estimate microphone delays and weights such that speech is separated from noise; for multi-talker speech, these are adjusted to separate speakers. In either case, the delays and weights could also be considered speaker localization features, which could be used in diarization.

There are at least two main approaches to separating speech using the signals from multiple microphones: computational auditory scene analysis (CASA), which models human hearing; and independent component analysis (ICA), which adaptively filters speech to maximize the statistical independence of the separated outputs.

CASA techniques employ models of human hearing, especially some aspects of the cochlea. The waveform from each microphone is usually fed into an array of bandpass filters of non-linearly increasing bandwidth and center frequency. At each filter output, a half-wave rectification is performed, which has the effect of extracting pitch pulses in each band. Auto-correlations in each band are used to estimate pitch, and per-band microphone pair cross-correlations are used to estimate time delay, a form of location. The pitch and location estimates are then used to assign time-frequency regions to one of the talkers to be separated; these are used to build sinusoid-plus-noise speech models from which separated speech is synthesized [78 88 66]. The strength of CASA is its robustness to reverberation, but it is usually limited to two closely spaced microphones. In addition, it cannot separate speech when there is a time-frequency overlap [113].

Independent component analysis (ICA) techniques make no assumption about microphone spacing and often make no assumption about the sound sources other than that they are statistically independent. Early ICA techniques required that there were the same number of sensors as sources, and that there was no time delay when
the sources arrived mixed at the sensor outputs. In so-called square, instantaneously mixed ICA \cite{32}, there are $R$ mutually independent and individually i.i.d. sources, $x_i[n]$, or, $\mathbf{x}[n] = (x_1[n], x_2[n], \ldots x_R[n])$, where $n$ is the time index. The sources jointly produce $R$ sensor signals, $y[n] = (y_1[n], y_2[n], \ldots y_R[n])$, which are modeled as having been generated by the mixing matrix, $\mathbf{V}$:

$$\mathbf{y}[n] = \mathbf{Vx}[n]$$

If the assumptions and models are correct, then the sources can be recovered by multiplying the sensor signals by a reconstruction matrix, $\mathbf{W} \approx \mathbf{V}^{-1}$

$$\hat{\mathbf{x}}[n] = \mathbf{W}y[n]$$  \hspace{1cm} (2.1)

It is tempting to employ a second-order least-squared fit to solve for $\mathbf{W}$, that is, solve for $\mathbf{W}$ such that $(\hat{\mathbf{x}}[n] - \mathbf{W}y[n])^2$ is minimized. For stationary sources, this has the effect of rotating the estimated source signals so that they are uncorrelated but, because decorrelation of stationary signals does not guarantee statistical independence, separation is not necessarily achieved. However, for non-stationary sources evaluated over a long time window, decorrelation does yield statistical independence; several ICA variants employ decorrelation \cite{123, 75, 85}. When ICA algorithms start with decorrelation, they are either solving the problem for a non-stationary signal, or are at least providing a partially separated starting point for another stage of processing that explicitly maximizes statistical independence.

Many techniques for estimating $\mathbf{W}$ employ information-theoretic principles. In one of the earliest, the infomax approach \cite{15}, the goal was to maximize the mutual information between the input vector, $\mathbf{x}$, and the corresponding output of a neural network. For nonlinear output units, it was shown that maximizing mutual information was equivalent to simply maximizing the entropy of the neural network output. Algorithms for several nonlinear outputs were derived, one example of which was the sigmoidal output unit, where the $i$th neural network output was $g(y_i) = (1 + e^{-y_i})^{-1}$. 
For this choice, a learning rule was developed to adjust $V = W^{-1}$, yielding a $V$ matrix element update of

$$\Delta v_{ij} \propto \frac{\text{cof} v_{in}}{\det V} + x_i(1 - 2y_i)$$

where $\text{cof} v_{in}$ was the cofactor of $v_{ij}$, i.e. $(-1)^{i+j}$ times the determinant of the matrix obtained by removing the $i$th row and $j$th column of $V$. During learning, when the weight vectors became too similar (equivalently, when $y_i$ became too similar) $\det V$ became small, driving apart the separated outputs, $y_i$. Although the learning rule was not explicitly designed to separate outputs, it has been empirically shown to have that effect. A more direct information-theoretic approach is found in [6], where the mutual information between the separated outputs is minimized. The infomax and minimum mutual information approaches were related in [124], where it was shown that infomax learning would stop adjustment of $V$ at the stable points found by a mutual information minimizing gradient search. Infomax was also related to source model likelihood maximization in [23], where the infomax gradient search was found to be equivalent to searching for $V$ such that there was a maximal match between the distributions of separated outputs and those of probabilistic source models, which may be estimated during the iterative search.

It is surprising that many ICA methods do not require an $a$ priori model of source statistics. One exception is found in [33], in which a large improvement in ASR WER was obtained when an instantaneous mixing matrix was adjusted with cost functions derived from separated signal filterbank energy, entropy and the entropy of HMM phone model posterior probabilities.

Instantaneous ICA algorithms fail in the presence of convolutive mixing – an example of which is room reverberation – so a more complicated approach would be necessary for meetings. One way to preserve some of the simplicity of the instantaneous mixing model is to transform the problem to the frequency domain [122, 7], where within each frequency bin of a DFT, the convolutive mixing due to reverberation can be modeled with an instantaneous mixing matrix. Individual reconstruction matrices
are estimated for each bin, and they are used to estimate the separated spectra. After appropriate frequency domain scaling and permutation, time-domain-separated signals are synthesized. A problem with many frequency domain ICA methods is that they are equivalent to adaptive beamforming algorithms which separate sources with only simple time-delay differences; they are often inefficient at unmixing reverberation with realistically long room impulse responses [13].

Although the computational load is increased significantly, reverberation can be unmixed directly in the time domain. If $g_{im}$ is the impulse response between the $ith$ source and the $mth$ sensor, and if the sensor input is $y_m = \sum_{i=1}^{R} g_{im} \ast x_i$, then the sources can be recovered with individually estimated reconstruction filters, $h_{im}$:

$$\hat{x}_i = \sum_{m=1}^{M} h_{im} \ast y_m.$$ 

Several authors have taken this approach. For example, in [37], the reconstruction filters were combined into a single block-structured convolution matrix, which was adjusted by signal decorrelation at the same time as GMM source models were estimated. Others have chosen to use explicit speech models. In [101], simulated convolutionally mixed speech was enhanced with factorial HMM speech models, although the separation was not truly blind because speech transcripts were utilized. In [1], speech mixed in a naturally reverberant environment was modeled with quantized linear prediction coefficients (LPC), and an EM algorithm adjusted the reconstruction filters to optimize the fit of the separated output to pre-trained speech models.

A method which makes very high level use of speech side information is found in [56], where localization and separation are combined by forming a joint state variable specifying the combination of an ASR speech model and one of the large number of discrete locations at which a beamformer is focused. The correct focal point is then selected by an ASR system, which (for each time instant) searches for the optimal HMM state encoding both speech and location.

A common theme for all of these speaker separation techniques is that separation is performed over sentence-length utterances, much too long for the short overlaps
in meeting speech (we have calculated that the median overlap length, averaged over 54 meetings in [86], is 250ms). Also common is the assumption that the number of talkers is known in advance, and that they are all speaking during the ICA analysis window. This situation rarely occurs in meetings, so it is likely that regions of overlap must be detected before any separation algorithm is applied. Beamforming and source separation methods found in the literature have either not been tested under realistic acoustic environments with overlapping speech, or they require known microphone positions [71].

2.3 Overlap Detection

As discussed above, neither the cepstral nor currently realizable location features are robust to overlaps; as will be shown, diarization can be improved if overlaps are detected and then subjected to special processing. In this section, we discuss methods of overlap detection.

In the monaural channel vocoder work of [70], spikes in speech amplitude kurtosis were found to bracket 83-92% of single talker speech, where single talker speech was defined as speech with a greater than 10dB talker to interferer ratio (TIR). Closely related overlap detection algorithms in [77 25 125] were applied to speaker identification. On synthetically overlapped TIMIT data, up to 75% of “usable” speech was detected, where “usable” segments were defined as those in which the level of one talker was 20 dB or more greater than that of the other. The authors of [125] used the spectral autocorrelation peak valley ratio (SAPVR) as a proxy for TIR. After removing speaker overlap with a SAPVR threshold, they found that a speaker identification system got results equivalent to removing speech with a TIR less than 20dB. However, speaker identifications (ID) were assigned based on a known TIR, a quantity unavailable in practice. SAPVR features yielded poor overlap detection performance for meeting speech personal microphone voice activity detection, but good performance was obtained using kurtosis and cross-correlations [121].
In [31], clustered “eigen locations” derived from ROOT-MUSIC beamforming outputs of a linear microphone array were able to find location peaks in projected space due to simultaneous talkers. Unfortunately, ROOT-MUSIC is inappropriate for meetings, as it requires known microphone positions, and is highly sensitive to microphone position errors. Also, the test data did not feature quickly changing speakers with short utterances, as is common in meetings.

Overlap detectors that were tested on meeting data, and which improved diarization or ASR results, include [65, 73, 74, 72], where the main features were functions of the ratio of microphone cross-correlation peaks. However, the data came from headset microphones, a much easier case than the distant microphones we focus on here.

Two published overlap detector algorithms were tested on distant microphone meeting speech. In [114], speaker segments obtained from a single-speaker diarization system were used to train hidden Markov model (HMM) speaker states corresponding to every possible overlap pair between detected speakers. Meeting data was then re-segmented using the combined single-speaker and overlap HMM. While the authors do not cite accuracy numbers, they state that, while the system was capable of detecting overlap, it correctly identified the overlapped speakers only a third of the time, and that the approach did not reduce overall diarization error. Meeting diarization error was successfully reduced (by about 1.4%) in [20], where an HMM overlap detector was trained on MFCC, energy, speaker model posterior entropy, and linear prediction error features. The strongest features were MFCC’s, with the next best being entropies. A small gain was obtained with the addition energy while the addition of linear prediction error caused a performance drop. The authors consider this exploratory work, as results were reported only over detector training data.
Chapter 3

CORPUS

Experiments were conducted on data from the NIST Rich Meeting Transcription Project. In these meetings, held at five locations in conventional conference rooms, 3 to 16 participants were recorded with 1 to 16 distant, omni-directional microphones. Sound was acquired at 16bits and 16KHz on multi-track digital recording systems. No information about microphone or speaker location is available.

The data used included the NIST 2004 development (rt04s) test and evaluation (rt04se) sets, and the eval sets for 2005 and 2006 (rt05se, rt06se). The final data set contained 31 meetings. As shown in Table 3.1, the number and type of microphones varied between years, site and meeting.

NIST provided hand-generated references identifying the correct speaker for 10-12 minute segments near the center of each of the meetings. However, for this work, performance was measured against more accurate forced alignments provided by ICSI.

3.1 Dev and Test Meeting Partitioning

Since there was only one meeting for some sites, and few meetings with a large number of microphones or circular arrays, we decided to not use the NIST corpora as boundaries for dividing the data into development (dev) and test (test) partitions (Tables 3.2 and 3.4). The partitioning rationale was as follows:

- Meetings with fewer than two microphones were omitted, since location features cannot be computed in the single-microphone case, and their use was of interest in this study.
Table 3.1: NIST Meeting Microphones

<table>
<thead>
<tr>
<th>Site</th>
<th>Corpus</th>
<th>Distant Microphones</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMU</td>
<td>rt04*</td>
<td>no farfield mics (not used)</td>
</tr>
<tr>
<td>CMU</td>
<td>rt05se</td>
<td>3 omni</td>
</tr>
<tr>
<td>CMU</td>
<td>rt06se</td>
<td>3 omni</td>
</tr>
<tr>
<td>LDC</td>
<td>rt04*</td>
<td>1 omni, one condenser</td>
</tr>
<tr>
<td>ICSI</td>
<td>rt04*</td>
<td>6 PZM omni, or 4 PZM omni and 2 PDA electrets</td>
</tr>
<tr>
<td>ICSI</td>
<td>rt05se</td>
<td>4 PZM omni, 2 PDA electrets</td>
</tr>
<tr>
<td>NIST</td>
<td>rt04*</td>
<td>3 omni 1 (missing in one meeting), 1 4-elem cardioid boundary</td>
</tr>
<tr>
<td>NIST</td>
<td>rt05se</td>
<td>3 omni, 1 4-elem directional cardioid</td>
</tr>
<tr>
<td>AMI</td>
<td>rt05se</td>
<td>either 2 8-mic circular arrays (omni elements),</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or 1 array plus 4 omni desktops</td>
</tr>
<tr>
<td>VT</td>
<td>rt05se</td>
<td>2 omni mics</td>
</tr>
<tr>
<td>VT</td>
<td>rt06se</td>
<td>4 omni mics</td>
</tr>
<tr>
<td>TNO</td>
<td>rt06se</td>
<td>8-mic circular array (omni elements), 2 omni</td>
</tr>
<tr>
<td>EDI</td>
<td>rt06se</td>
<td>2 8-mic circular arrays (omni elements)</td>
</tr>
</tbody>
</table>
If a site had only one meeting, its meeting was assigned to the dev corpora.

The remainder of the meetings were assigned randomly so that there were 19 dev meetings and 12 test meetings.

Although the number of speakers in a meeting was not considered, this partitioning process distributed the number of speakers fairly evenly across the dev and test corpora ("Num Speakers" column in Tables 3.2 and 3.4).

A disadvantage of this approach is that systems in this thesis are trained on all NIST corpora, making it unfair to compare against published test results on, say, the NIST rt06 corpora, which would have been trained only on earlier data. On the other hand, the full training data used by other researchers was not available, making that comparison suspect in any case. Therefore, the approach taken is to use the best performing diarization system in the NIST 2006 competition [11], and then to create a new baseline system by tuning its parameters on the partitions shown in Tables 3.2 and 3.4.

### 3.2 Reported Diarization Performance on the Source Corpora

For the purposes of orientation, Table 3.6 shows the best results reported for the NIST competitions by other researchers on the original corpora used to build the dev and test corpora used here. In 2004, results were reported only for non-overlapping speech. The top row of the table shows the effect on the score of using more accurate forced alignment references for the rt06se corpus. The results in Table 3.6 came from mostly for diarization systems built during the year of the corresponding corpus, meaning the the results for earlier corpora were obtained with less advanced

---

1It should be noted that the baseline system was improved by ICSI and was, again, the best system in the NIST 2007 competition. The main changes were a new configuration of parameters and a better VAD. Some of the settings were used in this work, but the VAD used here is the original 2006 version.
Table 3.2: Dev Meetings

<table>
<thead>
<tr>
<th>Meeting</th>
<th>Thesis Corpus</th>
<th>NIST Corpus</th>
<th>Num Mics</th>
<th>Num Speakers</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICSI20010208-1430</td>
<td>dev</td>
<td>rt04s</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>ICSI20010322-1450</td>
<td>dev</td>
<td>rt04s</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>NIST20020214-1148</td>
<td>dev</td>
<td>rt04s</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>LDC20011116-1500</td>
<td>dev</td>
<td>rt04s</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>ICSI20011030-1030</td>
<td>dev</td>
<td>rt04se</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>NIST20030925-1517</td>
<td>dev</td>
<td>rt04se</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>LDC20011121-1700</td>
<td>dev</td>
<td>rt04se</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>VT20050318-1430</td>
<td>dev</td>
<td>rt05se</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>CMU20050301-1415</td>
<td>dev</td>
<td>rt05se</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>NIST20050427-0939</td>
<td>dev</td>
<td>rt05se</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>ICSI20010531-1030</td>
<td>dev</td>
<td>rt05se</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>AMI20041210-1052</td>
<td>dev</td>
<td>rt05se</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>NIST20050412-1303</td>
<td>dev</td>
<td>rt05se</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>CMU20050912-0900</td>
<td>dev</td>
<td>rt06se</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>VT20051027-1400</td>
<td>dev</td>
<td>rt06se</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>NIST20051102-1323</td>
<td>dev</td>
<td>rt06se</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>NIST20051024-0930</td>
<td>dev</td>
<td>rt06se</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>TNO20041103-1130</td>
<td>dev</td>
<td>rt06se</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>EDI20050216-1051</td>
<td>dev</td>
<td>rt06se</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>Meeting</td>
<td>Thesis Corpus</td>
<td>NIST Corpus</td>
<td>Num Mics</td>
<td>Num Speakers</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------</td>
<td>-------------</td>
<td>----------</td>
<td>--------------</td>
</tr>
<tr>
<td>NIST20020305-1007</td>
<td>test</td>
<td>rt04s</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>LDC20011116-1400</td>
<td>test</td>
<td>rt04s</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>LDC20011207-1800</td>
<td>test</td>
<td>rt04se</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>ICSI20000807-1000</td>
<td>test</td>
<td>rt04se</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>NIST20030623-1409</td>
<td>test</td>
<td>rt04se</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>VT20050304-1300</td>
<td>test</td>
<td>rt05se</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>CMU20050228-1615</td>
<td>test</td>
<td>rt05se</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>ICSI20011113-1100</td>
<td>test</td>
<td>rt05se</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>AMI20050204-1206</td>
<td>test</td>
<td>rt05se</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>CMU20050914-0900</td>
<td>test</td>
<td>rt06se</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>VT20050623-1400</td>
<td>test</td>
<td>rt06se</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>EDI20050218-0900</td>
<td>test</td>
<td>rt06se</td>
<td>16</td>
<td>4</td>
</tr>
</tbody>
</table>
Table 3.6: Diarization Error Performance in Previous NIST competitions

<table>
<thead>
<tr>
<th>Corpus/year</th>
<th>Best reported DER</th>
<th>Overlaps scored</th>
<th>Refs</th>
</tr>
</thead>
<tbody>
<tr>
<td>rt06se</td>
<td>20.3 [94]</td>
<td>yes</td>
<td>forced</td>
</tr>
<tr>
<td>rt06se</td>
<td>35.8 [94]</td>
<td>yes</td>
<td>NIST</td>
</tr>
<tr>
<td>rt05se</td>
<td>18.6 [41]</td>
<td>yes</td>
<td>NIST</td>
</tr>
<tr>
<td>rt04se</td>
<td>16.22 [9]</td>
<td>no</td>
<td>NIST</td>
</tr>
<tr>
<td>rt04s</td>
<td>16.92 [9]</td>
<td>no</td>
<td>NIST</td>
</tr>
</tbody>
</table>

algorithms. Although the state of the art improved, the data also became harder to accurately diarize in later years.

Two results are shown for the rt06se data, in which the same diarization was scored against the official NIST references and against the forced alignments provided by the ICSI laboratory. This NIST references were quite inaccurate, so the ICSI forced alignments are used for this thesis.

3.3 Sources of Meeting variability

A main problem encountered in this work is acoustic variability across meetings. As will be shown in subsequent chapters, performance for a given diarization system is highly meeting specific. In an early pilot study, we examined factors which might have conceivably contributed to diarization error. The data were 75 meetings in the NIST rt04 training corpus, a very early meeting data set, which is now seldom used. The features were concatenated energy ratio (ER) and correlation features (XC) (see Chapter 4). Here, they were concatenated in a single feature vector, an approach which performed well on that data set and with the 2005 version of the speaker clustering software used for this work. The factors studied are in the first column of Table 3.7: the number of distant microphones, the number of speakers in the meeting, the percentage of time during which the most frequent speaker was talking,
the speaker identification error for the lowest scoring talker, and the percentage of speech that was overlapped in the whole meeting. The correlation coefficient of these factors with respect to diarization error is shown in the second column; the third and forth columns are the factor values for the meetings with the lowest and highest diarization errors; and the last three columns are the minimum, maximum and mean of the factors.

Surprisingly, the number of channels in a meeting had almost no effect on diarization error, and the correlation with the number of speakers was also very weak. Errors were generally lowest when the meeting was dominated by a single speaker, perhaps not surprising since there was a lot of training data for that speaker, and errors on other speakers accounted for less of the total meeting time. We had thought that some meetings had high errors because of a difficult speaker (for example one that moved) but this did not appear to be the case. In fact, diarization errors were lowest when most of the error came from a single speaker; in bad meetings, errors were distributed more equally.

One expected result is that a high amount of overlapped speech generally implies a higher diarization error, indicating that a successful method of detecting and handling overlaps would pay off.

In a different pilot study, we examined the effect of signal to noise ratio (SNR) on diarization. These experiments were done with XC+ER features, as well as with XC and ER features individually. We used the 2005 version of the diarization system used as a baseline for this thesis, and data from the rt05se corpus. For each meeting in rt05se, we used a standard NIST SNR measurement tool [107] to calculate the min, median and max channel SNR across that meeting’s channels. Then we calculated the correlation coefficient of the min, max and median SNR against meetings diarization error. The results are shown in Table 3.8.

The results generally support the hypothesis that a few low SNR channels hurt diarization performance on some meetings. The maximum SNR is weakly and in-
Table 3.7: Factors Conceivably Affecting Diarization Error (XC+ER)

<table>
<thead>
<tr>
<th>Factor</th>
<th>ρ</th>
<th>Value for Best Meeting</th>
<th>Value for Worst Meeting</th>
<th>Min</th>
<th>Max</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td># Distant Mics</td>
<td>0.01</td>
<td>6</td>
<td>6</td>
<td>2</td>
<td>7</td>
<td>5.6</td>
</tr>
<tr>
<td># Speakers</td>
<td>0.09</td>
<td>3</td>
<td>7</td>
<td>2</td>
<td>8</td>
<td>5.2</td>
</tr>
<tr>
<td>% Speech, max</td>
<td>-0.43</td>
<td>73</td>
<td>20</td>
<td>18</td>
<td>93</td>
<td>45</td>
</tr>
<tr>
<td>% Errors, worst</td>
<td>-0.41</td>
<td>65</td>
<td>20</td>
<td>17</td>
<td>88</td>
<td>45</td>
</tr>
<tr>
<td>% Overlapped</td>
<td>0.23</td>
<td>2.6</td>
<td>13</td>
<td>2.6</td>
<td>20</td>
<td>9.8</td>
</tr>
</tbody>
</table>

Table 3.8: Diarization Error / SNR Correlation Coefficient

<table>
<thead>
<tr>
<th>Feature</th>
<th>SNR corr. coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min</td>
</tr>
<tr>
<td>ER</td>
<td>-0.33</td>
</tr>
<tr>
<td>XC</td>
<td>-0.54</td>
</tr>
<tr>
<td>XC+ER</td>
<td>-0.66</td>
</tr>
</tbody>
</table>
consistently correlated with diarization error across the different features, and the minimum SNR is highly correlated. This is easiest to see in the cross-correlation features, where the median SNR is almost uncorrelated while the minimum SNR correlation coefficient is -0.54.
Chapter 4

NEW LOCATION FEATURES

As discussed in Chapter 2, the most successful MDM system in the NIST 2006 competition [11] used “location features,” the set of correlation lags computed between microphones placed at unknown locations. In this thesis, we will use the clustering algorithm employed by this system, and will use its correlation features as a baseline location feature. The following sections discuss the limitations of this feature, and propose several alternatives.

4.1 Limits of Single-Reference Correlations

In meetings recorded with ad hoc microphone placement, the microphones are generally widely spaced, meaning that the maximum time-of-flight difference between them can be several times longer than the fundamental period of speech, $T_0$. When coherence is calculated over a time extent long enough to capture the time-of-flight difference, the analysis window will contain several pitch periods. This does not usually cause a problem for sentence-length (multi-second) correlation analysis windows, in which $T_0$ and its harmonics can vary widely enough to produce a single strong coherence peak. However, the goal here is to segment utterances as brief as 0.5s (“yeah”). When coherences are computed over this short analysis window, speech can appear nearly stationary and periodic. For a single location, there will be several coherence peaks of nearly equal magnitude, causing the location feature to be ambiguous. The peak ambiguity due to widely spaced microphones is similar to the peak ambiguity problem often encountered in correlation-based pitch detectors. In addition, widely spaced microphones are more likely to be subject to different room
reverberation effects, making the pair waveforms quite dissimilar, resulting in a weak true-delay correlation peak. If a speaker is too far from one or both microphones in a pair, differences in reverberation or poor SNR for that speaker can make the waveforms so dissimilar that a correlation peak is invisible.

It should be noted that the idiosyncrasies of the meeting data, namely widely spaced microphones at unknown positions, combined with the requirement for an extremely short analysis window, prevent us from using sophisticated time delay estimation techniques, such as those based on eigen analysis [16], either because of unmet assumptions, or numerical stability issues inherent in estimating a long correlation function with little data. Hence, we rely on simple correlation-based algorithms.

To assess the potential impact of low quality correlations, a series of speaker ID cheating experiments were performed in which known speaker segmentation information was used to extract training data for each speaker. For each correlation pair, a Gaussian mixture model (GMM) is trained for each speaker – the 1-dimensional feature being the lag at maximum correlation. For each pair, a maximum likelihood speaker identification is made and speaker diarization error is calculated. The result provides an indication of the suitability of a given microphone pair for use in an unsupervised speaker clustering system.

Figure 4.1 is a composite view of the per-pair cheating experiment errors across all the meetings in the RT05SE corpus [41]. The horizontal axis is the meeting number and the vertical axis is the per-pair diarization error. Three lines are drawn: one for the pair with the lowest diarization error; one for the pair with the median diarization error; and one for the pair with the highest diarization error. Comparing the dashed maximum and minimum lines, it is clear that there are good and bad pairs, although it is impossible to know from this data if they are good or bad all of the time. In addition, there is wide variability in the suitability of various microphone pairs in most meetings.

If a speaker is too far from one or both microphones in a pair, differences in
Figure 4.1: Minimum, median and maximum diarization error using single cross-correlation pair but known (oracle) speaker segmentations in training. Diarization error is shown as a function of the different meetings in the RT05S corpus.

reverberation or poor SNR for that speaker can make the waveforms so dissimilar that a correlation peak is invisible.

That certain pairs are good for certain speakers is illustrated for a sample meeting in Figure 4.2 In the figure, the vertical coordinate of each pixel is the true speaker ID; the horizontal coordinate is the channel pair; and the intensity is the speaker ID F-number (geometric mean of precision and recall) for speaker identification based on the correlation for that channel pair. A light region indicates that a certain channel pair is especially good at distinguishing a particular speaker from all of the others in the meeting. Clearly, the best microphone pair depends upon the speaker (there are no solid white vertical lines), and there are no microphone pairs which are bad for every speaker (there are no solid black vertical lines).

Ideally, the clustering algorithm would be fed only correlation features from closely spaced microphones, but with ad hoc microphone placement the spacing is unknown. If all channels are correlated only against a single reference channel, it is likely that the set of pairs does not include the best pair for every speaker.

For these reasons, we explore the use of all $N(N - 1)/2$ channel pairs, rather
than the $N - 1$ channel pairs matched to a single reference for $N$ microphones.

### 4.2 Limitations of GCC-PHAT

The GCC-PHAT time delay estimator [48] is popular for use in acoustic localization in reverberant environments. Although it is general purpose and not tailored to speech, it is employed by many researchers concentrating on localizing speech, including [11], which in addition preprocesses the signal with a Wiener filter-based noise reduction algorithm. In GCC-PHAT, the cross-correlation is calculated in the frequency domain after the cross-spectra whitening: given two microphone channel spectral components, $Z_i(\omega)$ and $Z_j(\omega)$, a frequency domain phase transform, $\Psi_{i,j}^{PHAT}(\omega)$, is calculated as:

$$
\Psi_{i,j}^{PHAT}(\omega) = \frac{1}{|Z_i(\omega)Z_j(\omega)^*|}
$$

The phase transform weights the cross-spectrum before and inverse Fourier transformation is used to calculate the time domain cross-correlation waveform. The GCC-PHAT delay estimate, in samples, is then:

$$
\hat{\tau}_{i,j} = \arg\max_\tau \int_\omega Z_i(\omega)Z_j(\omega)^*\Psi_{i,j}^{PHAT}(\omega)e^{i\omega\tau}d\omega.
$$
In addition, in [11], errors in the delay estimates are filtered using a dynamic programming smoothing algorithm.

Other researchers [100] have found that, for speech, GCC-PHAT is outperformed by speech-specific algorithms which correlate the Hilbert envelopes of inverse linear-prediction-filtered waveforms. For each microphone waveform, an inverse-filtered Hilbert-envelope is calculated as:

$$z_k^{IH} = |H\{LP^{-1}(x_k(t))\}|$$

where $LP^{-1}(\cdot)$ is the inverse linear predictive filter, and $H(\cdot)$ is the Hilbert transform. After this pre-processing the delay estimate is calculated with the standard cross-correlation peak picking, as with GCC-PHAT.

Inverse filtering performs roughly the same function as GCC-PHAT whitening; frequencies are still effectively being divided by their magnitudes, but with inverse filtering, the denominator is smoothed by the spectral envelope of speech, thus avoiding noise pumping in spectral zeros. Viewed in the time domain, the envelope calculation, which smooths out easily aliased periodic information in the glottal transient oscillations, is similarly speech-specific. Finally, the delay features are the lags at the peak of standard correlations between the Hilbert envelopes of the inverse filtered channel waveforms.

In this thesis, no noise reduction is applied before the correlation calculation, as it was found that per-channel noise reduction reduced the pair similarity needed for a good correlation peak. Several researchers, for example, [49, 10], have found that Wiener filter noise reduction before correlation improves diarization performance. We speculate that our speech-specific correlation feature processing is itself a type of noise reduction, and therefore does not benefit from this step.
4.3 Energy Ratios

Given the problems of meeting data microphone placement – unknown locations and spatial aliasing – some correlation feature inaccuracy is unavoidable. Two other factors that make distant microphone pair correlation features undesirable are the decrease in sound intensity with distance, and the location and frequency-dependent zeros that room reverberation imposes over the sound spectrum. The power received by microphone $\text{mic}$ when speaker $\text{spkr}$ is active can be approximated as:

$$|X_{\text{mic}}(\omega)|_{\text{spkr}} = |S_{\text{spkr}}(\omega)| \cdot \alpha(|\vec{r}_{\text{spkr}} - \vec{r}_{\text{mic}}|, \omega) \cdot g(\vec{r}_{\text{spkr}}, \vec{r}_{\text{mic}}, \omega)$$

where $|S_{\text{spkr}}(\omega)|$ is the sound pressure magnitude at the speaker’s mouth at frequency $\omega$, $\alpha(|\vec{r}_{\text{spkr}} - \vec{r}_{\text{mic}}|, \omega)$ is the main path distance attenuation between the speaker position $\vec{r}_{\text{spkr}}$ and the microphone position $\vec{r}_{\text{mic}}$, and $g(\vec{r}_{\text{spkr}}, \vec{r}_{\text{mic}}, \omega)$ is the attenuation due to interference with non-main-path sound propagation, i.e. the position- and frequency-dependent reverberation.

For cross-correlations, main-path attenuation and reverberation are both problems because they make waveforms from the microphone pairs dissimilar, lowering correlation peaks. However, the dissimilarity can be turned into a pattern recognition feature giving another clue to speaker location. For this thesis, energy ratios were calculated over a sliding time windows spanning $t + \Delta t$ for each microphone, $l$, within $m$ frequency bands, $BW_m$:

$$E_{lm}(t) = \int_{t' \in (t,t+\Delta t)} \int_{\omega \in BW_m} |X_l(\omega)|^2 d\omega dt'$$

Then, the log energy ratio was calculated for each microphone pair, $(i, j)$, and each frequency band, $m$:

$$\xi_{ijm}(t) = \log E_{im}(t) - \log E_{jm}(t)$$

The energy measurements capture the frequency dependent power coming from a location, while the ratios remove the dependence upon the speaker voice loudness.
Table 4.1: Delay / Energy Ratio Complementarity

<table>
<thead>
<tr>
<th>Hypothesized Diarization Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delay Est.</td>
</tr>
<tr>
<td>-------------</td>
</tr>
<tr>
<td>Near Mics</td>
</tr>
<tr>
<td>Far Mics</td>
</tr>
<tr>
<td>Reverberation</td>
</tr>
</tbody>
</table>

For this thesis, each channel was fed into a filter bank with linearly spaced band-pass filters of width 1250Hz centered about 875, 2125 and 3375Hz, yielding a \(3 \cdot N_{pairs}\) feature vector, where \(N_{pairs}\) is the number of channel pairs in the meeting. The filter bank parameters were chosen by a grid search optimizing for cheating experiment diarization accuracy (as in Section 4.1). The overall bandwidth covers regions of the speech spectrum where there is significant path-length attenuation for distances typical of meeting room environments. The number of bands chosen by the search is a trade-off between a desirably small feature vector dimension and the goal of capturing frequency-dependent room reverberation differences. Although noise reduction did not help with the proposed cross-correlation features, it was found to improve results as a preprocessing step to the energy ratio calculation.

The hypothesized complimentary relationship of delay estimates and energy ratios is summarized in Table 4.1.

Figure 4.3 illustrates a meeting where energy ratios are expected to provide improved results for speakers who are difficult to identify with cross-correlation features. The figure shows confusion images, where pixel intensity is proportional to the fraction of cases where an active speaker (vertical axis) is labeled as one of the possible set of speakers in the meeting (horizontal axis). In the cheating experiment, speakers are identified with GMM’s trained in a supervised manner on the two different types of location features. As can be seen in the figure, a correlation feature classifier has
Figure 4.3: Cheating experiment speaker ID confusion image for cross-correlation (xcpair) and energy ratio (eratio) features. A correlation feature classifier make errors not made by a classifier trained on energy ratios.

a higher degree of confusability among candidate speakers.

Others researchers have used energy ratios for diarization [62, 50]. However, these were with personal, close-talking microphones – an easier case – and were not multi-band, meaning that the energy ratios did not characterize detailed frequency and position-dependent room reverberation.

4.4 Dimension Reduction

High dimensionality is a major disadvantage when location features are computed for all microphone pairs – in some meetings in the NIST corpora, the combined energy and cross-correlation location feature vector dimension is 480. The high dimension has high computational cost in diarization and can dominate the score in clustering when compared to MFCC’s which, in this work, have a dimension of 19. Therefore, both accuracy and efficiency benefit from dimensionality reduction. Ideally, dimensionality reduction also allows us to use a single lag per channel pair, rather than using several hypothesized correlation lags from each channel pair associated with a single reference microphone and then selecting the best via a continuity-enforcing
dynamic-programming algorithm, as in [11].

Since the number and position of microphones can vary across meetings, it is impossible to train a supervised discriminant transform on a dev set, and then use it on meetings in a test set. Therefore, this thesis considers two unsupervised dimensionality reduction techniques.

The first is principal components analysis (PCA) [39], in which feature vectors are projected upon lower dimensional orthogonal subspaces such that each transformed dimension has maximum variance. The lower dimension approximation thus gives the minimum mean squared error reconstruction for a given dimension. PCA is commonly employed in a variety of speech processing tasks.

The second dimensionality reduction method is canonical correlation analysis (CCA) [53], in which two “views” of a phenomena are transformed to orthogonal subspaces such that the correlation between the transformed views is maximized. CCA has been applied in meteorology, statistical signal processing, economics, and other fields in which it is desirable to identify relations between two multi-dimensional variables. In some applications, the goal has been to discover the inter-view correlations themselves, but in this work, the desired result is the data after the correlation maximizing transform. As with PCA, the transformed space is often of lower dimension. In related work, [103], CCA was used to fuse audio and video features for speaker identification.

4.4.1 Principal Components Analysis

In PCA [39], feature vectors are projected upon lower dimensional orthogonal subspaces such that each transformed dimension has maximum variance.

Figure 4.4 illustrates PCA noise removal. In the figure, delay estimates that have been collected over a NIST meeting are arranged in columns, one column for each speaker. Each row represents the delay estimate histograms for a given microphone pair; the thick lines show the count of raw delays, extracted by selecting
cross-correlation peaks. Speaker 1’s delays were fairly consistent over the meeting, although there are some peaks at erroneous delays. The delay histograms for Speaker 2 are much more broadly spread. This could have been due to spatial aliasing, a change in the acoustic environment, or speaker movement.

Next, PCA was performed on all of the delay estimates collected over the entire meeting. At each sample time, \( k \), a feature vector, \( x_k = [x_0, x_1 \ldots x_D] \), was composed, where \( x_l \) is the delay estimate for the \( l \)th of \( D \) channel pairs. Typically – and for the clustering experiments done in this thesis – the meeting’s \( N \) feature vectors are collected in the matrix \( X = [x_0 x_1 \ldots x_N]^T \) and PCA is used to compute a linear transform:

\[
Y = (X - \mu)Q_{1:d}
\]

where \( \mu \) is the feature vector mean, and \( Q_{1:d} \) is the \( D \times d \) PCA weighting matrix corresponding to the largest \( d \) eigenvalues (the minimum mean squared error representation for dimension \( d \)). In addition to mean normalization, many researchers divide each column of \( X \) by its variance, but we found that variance normalization did not improve diarization performance. Limiting the transform to \( d \) eigenvectors throws away information in the original data, but the hope is that this information is not useful for speaker classification. To see what has been eliminated, we can reconstruct an approximation of the original data:

\[
\hat{X} = YQ_{1:d}^{-1} + \mu
\]  

(4.1)

In Figure 4.4, where \( \hat{X} \) histograms are displayed in thin lines, we see that PCA smooths the delay estimate roughly in proportion to the variance of a given speaker’s delay distribution, and that it can eliminate delay peaks which are not supported by co-occurring peaks in other channel pairs (bottom plot for Speaker 2).
Figure 4.4: PCA noise removal on lag estimates
4.4.2 Canonical Correlation Analysis

The second dimensionality reduction method explored in this thesis is canonical correlation analysis (CCA) [53], in which two “views” of a phenomena are linearly transformed to subspaces such that the correlation between the transformed views is maximized. The rationale is that if correlations and energy ratios are both related to location, then they should be highly correlated; non-correlated components can be expected to contain noise. Additionally, if a pattern of location features is associated with a speaker’s voice—a fundamental assumption of this work—then a reduction technique maximizing the correlation with MFCC’s should also remove noise.

Given two feature vectors “viewing” a phenomenon, \( u \) and \( v \), CCA seeks a pair of linear transforms
\[
\begin{align*}
    u & \rightarrow w_u \cdot u \\
    v & \rightarrow w_v \cdot v
\end{align*}
\]

such that
\[
    r = \arg\max_{w_u, w_v} \frac{E \{(w_u \cdot u) \cdot (w_v \cdot v)\}}{\sqrt{E\{w_u \cdot u\}^2 E\{w_v \cdot v\}^2}}
\]
is maximized. Defining the joint covariance matrix as:
\[
    C = \begin{bmatrix}
        C_{uu} & C_{uv} \\
        C_{vu} & C_{vv}
    \end{bmatrix} = E \left\{ \begin{bmatrix} u \\ v \end{bmatrix} \begin{bmatrix} u^T \\ v^T \end{bmatrix} \right\}
\]
where \( C_{uu} \) and \( C_{vv} \) are the covariance matrices for \( u \) and \( v \), and where \( C_{vu} \) and \( C_{uv} \) are the cross-covariance matrices, the maximization is accomplished by solving the following generalized eigenvalue equations:
\[
\begin{align*}
    C_{uu}^{-1}C_{uv}^{-1}C_{vu}w_u &= \rho^2 w_u \\
    C_{vv}^{-1}C_{vu}^{-1}C_{uv}w_v &= \rho^2 w_v
\end{align*}
\]
where \( \rho^2 \) are the eigenvalues of \( C \) (and also the squared correlations between transformed dimensions). The maximum dimension of the CCA-transformed outputs is
\[
d_{max} = \min(\text{rank}(u), \text{rank}(v)),
\]
but further dimension reduction can often be obtained by removing transformed dimensions associated with smaller eigenvalues, as in PCA.

In this thesis, there are three sets of speaker diarization features: cross-correlation delays (XC), energy ratios (ER), and traditional speech feature MFCC's. All of them are candidates for CCA views. Here, several view pairs are examined.

1. XC-MFCC: We assume that a set of delays (a location) should be associated with a speaker's voice; one CCA view is the delay vector, and the other is the vector of speaker ID MFCC's. CCA can produce two transformed views (equation 4.2) but since the intent is to reduce location feature noise, only the XC view is transformed for use as a diarization feature.

2. XC-XC: In “split CCA,” the elements of the delay vector are partitioned into two disjoint sets constituting the two CCA views. The rationale is that transformed delay subspaces should be correlated with each other. That is, as speakers change, the transformed delay pattern should also change consistently. The elements of delay are partitioned into two groups using a spectral clustering algorithm [57], with an affinity matrix composed of the correlation coefficients between each pair of delay vector elements. In this case, the smaller dimension transformed view was chosen as the diarization feature.

3. XC-ER: Since energy ratios and delays are both views of a speaker location, these features should also be correlated. Pilot studies showed that the transformed ER view of this combination was a poor diarization feature, so only the XC transformed view was studied in depth.

4. ER-MFCC: By the same reasoning as with delays, a good dimension reducing transform should result in MFCC's being maximally correlated with energy ratios. For completeness, we use the transformed ER view.
4.5 Experiments

4.5.1 Experimental Paradigm

The baseline for comparison in evaluating our features is the system described in [11]. The authors made the feature extraction and clustering software available for this work. The diarization algorithm uses an agglomerative clustering scheme with a Bayesian information criterion (BIC) segment merging criteria and a hidden Markov model (HMM) to enforce minimum length constraints. In a single stream HMM, the probability, $b_j(o_t)$, of state $j$ generating an output vector $o_t$ at time $t$ is modeled with a mixture of Gaussians. In an $N_s$ stream multi-stream HMM, the probability is modeled as the product of $N_s$ independent Gaussian mixtures:

$$b_j(o_t) = \prod_{s=1}^{N_s} \left[ \sum_{m=1}^{M_{js}} c_{jsm} \mathcal{N}(o_{st}; \mu_{jsm}, \Sigma_{jsm}) \right]^{\gamma_s}$$  \hspace{1cm} (4.3)

where $M_{js}$ is the number of mixtures for the $j$th state in stream $s$, and where $c_{jsm}$, $\mu$, and $\Sigma_{jsm}$ are the Gaussian mixture weights, means and variances. $o_{st}$ is a sub-vector of $o_t$, and $\gamma_s$ is the weight for stream $s$.

As illustrated in Figure 4.5, the baseline system was modified so that the proposed location features replaced the original delay features, although the original delay estimates were still used to perform delay-sum beamforming. Modifications were also made to extract baseline reference channel information, and to allow the baseline correlation features to be used directly instead of being processed by the dynamic programming smoother. Depending upon the test being performed, different location features were fed into one or two of the HMM streams.

Speaker diarization results were measured against references with word times determined by forced alignments, using the NIST diarization scoring software.\footnote{www.nist.gov/speech/tests/rt/rt2006/spring/code/md-eval-v21.pl} For this tool, the diarization error is the sum of time over all reference speakers for which
speech is either missed or falsely detected – including during overlaps and silences – divided by the total speech time of the scored region. During overlapped periods, the total speech time is incremented by the duration of the overlap multiplied the number of speakers in the overlap.

There are many tunable parameters in this system, but an attempt was made to keep as many of them as possible fixed to those used by ICSI for the 2007 NIST meeting evaluation [120]. Grid searches were employed to adjust the following parameters for optimal performance on the dev set:

**HMM stream weights** For this thesis, up to three HMM streams are active during clustering. Since we introduce features of varying reliability, it is important to adjust their stream weights appropriately. GMM posterior probabilities in the stream HMM's are weighted in two places. In the first, fixed weights are applied to likelihoods during iterative re-segmentation ($\gamma_s$ in equation [4.3]). In the second, the weights are applied during merging decisions. When automatic weight calculation is turned on, merge decisions are made using a normalized BIC metric, with weights calculated over each segment to be considered for a merge. As discussed in [12], this is equivalent to a variance normalization, assuming that each stream is modeled by a zero mean Gaussian. This assumption is an especially poor match to the energy ratio distribution, and degraded diarization performance. Therefore, automatic merge weight calculations were turned off and fixed weights were used for some location feature combinations containing energy ratios. During optimization on the dev set, a grid search is performed over the fixed stream weights, with the same weight being applied to the BIC metric weight during the initial round of agglomerative clustering. Weights were varied over the range (0.05, 0.1, 0.2..., 0.9, 0.95) and were restricted so that the sum of weights was always 1.
Number of initial stream mixtures  This parameter controls the initial number of mixtures in each stream’s GMM. After each stage of merging, uninformative mixtures may be removed, as described in [10]. For correlation delay features, a single mixture was found to be optimal, and was therefore not adjusted. Similarly, 5 mixtures were optimal for MFCC HMM stream, and the number was therefore fixed. Streams containing energy ratios were scanned with mixtures ranging from 3 to 25.

Dimension reduction output dimension  For location features, dimension reduction was usually applied. For each stream, the output dimension was chosen to minimize the total average diarization error over the dev set. The dimensions were held constant across all meetings, except for those with too few microphones, in which case the dimension was set to the maximum attainable. PCA dimensions were varied between 2 and 11 and CCA dimensions were varied between 2 and 9.
4.5.2 Results

A series of diarization experiments were run to assess the usefulness of the different transformations. In all experiments, MFCC’s were fed into one HMM stream of the clustering HMM, and location features were fed into one or two additional streams. The following conditions were tested:

0: X0: baseline system; the system used in [11] with the configuration used in the NIST 2007 evaluation, except for having the above parameters optimized on the dev set. This system used GCC-PHAT correlations computed against a single reference channel.

1: PCA(XC): all-pairs, inverse filtered, Hilbert/Inverse correlations, with PCA dimension reduction.

2: PCA(XC,ER): Hilbert/Inverse correlations as in 1, in one HMM stream, PCA dimension reduced energy ratios in a second HMM stream.

3: CCA(XC,XC): correlation features partitioned into two CCA views; the first transformed view fed into one HMM stream (the other view is ignored).

4: CCA(ER,XC): energy ratios CCA-transformed with correlation features as in (1); the transformed energy ratios fed into one HMM stream.

5: CCA(ER, MFCC): energy ratios CCA-transformed with MFCC’s; the transformed energy ratios fed into one HMM stream.

6: CCA(ER, XC): energy ratios CCA-transformed with correlations as in (1); the transformed correlations fed into one HMM stream.

7: CCA(XC,MFCC): correlations as in (1) CCA-transformed with MFCC’s, with the transformed energy ratios fed into one HMM stream.
CCA(ER, XC), CCA(XC, MFCC): The features described in (6) and (7), each fed into a separate HMM stream.

XCo: Baseline, GCC-PHAT correlations (single reference) with dynamic programming smoothing removed.

PCA(XCo): XCo features but with dynamic programming removed and then PCA transformed.

PCA(XC-1): Inverse-filtered, Hilbert envelope correlations calculated against the single reference determined by the baseline system. The result is PCA’ed and fed into a single HMM stream.

PCA(ER): PCA’ed energy ratio features.

Diarization error for the different combinations of features is shown in Table 4.2, where “tot” indicates results for all meetings combined; “worst” and “best” are the scores for the meeting with the highest and lowest error.

Comparing expts. 0 and 10 of Table 4.2, we see that the baseline system’s dynamic programming smoother actually hurts performance, although it was used in the diarization systems that won the NIST 2006 and 2007 competition [11, 120]. A close reading of [8] explains some of the mystery: detailed tests found that the improvement due to the smoother was meeting dependent, for example, it yielded 0.7% worse results on the rt06 eval corpus than doing nothing. The loss in performance found here is much larger (3%) but we have used a different combination of meetings. Meeting variability is extremely high in this data, which causes dev and test set mismatch for many meeting permutations.

Comparing expts. 0, 1 and 3, we see that the combination of all-pairs correlations and dimension reduction is a more effective type of location feature smoothing that the baseline dynamic programming algorithm, and that PCA is perhaps superior to
<table>
<thead>
<tr>
<th>Expt.</th>
<th>Dev</th>
<th>Test</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>worst</td>
<td>best</td>
<td>tot</td>
</tr>
<tr>
<td>0</td>
<td>44.4</td>
<td>3.3</td>
<td>16.0</td>
</tr>
<tr>
<td>1</td>
<td>36.4</td>
<td>2.6</td>
<td>15.2</td>
</tr>
<tr>
<td>2</td>
<td>36.3</td>
<td>3.1</td>
<td>15.8</td>
</tr>
<tr>
<td>3</td>
<td>36.4</td>
<td>2.4</td>
<td>15.2</td>
</tr>
<tr>
<td>4</td>
<td>46.5</td>
<td>2.5</td>
<td>19.4</td>
</tr>
<tr>
<td>5</td>
<td>56.3</td>
<td>6.3</td>
<td>20.4</td>
</tr>
<tr>
<td>6</td>
<td>36.2</td>
<td>2.8</td>
<td>14.9</td>
</tr>
<tr>
<td>7</td>
<td>35.0</td>
<td>2.8</td>
<td>15.0</td>
</tr>
<tr>
<td>8</td>
<td>36.0</td>
<td>2.9</td>
<td>14.5</td>
</tr>
<tr>
<td>9</td>
<td>36.8</td>
<td>2.9</td>
<td>14.5</td>
</tr>
<tr>
<td>10</td>
<td>37.1</td>
<td>3.0</td>
<td>15.6</td>
</tr>
<tr>
<td>11</td>
<td>36.4</td>
<td>2.3</td>
<td>14.6</td>
</tr>
<tr>
<td>12</td>
<td>57.0</td>
<td>2.9</td>
<td>20.4</td>
</tr>
</tbody>
</table>
“split CCA”. However, PCA is not helpful for low dimensional data vectors (expts. 9 and 10). On the dev data, PCA and CCA applied to XC features produce similar results on the dev set but CCA works better on the test set (expts. 1, 3, 6, 7), which could indicate more robustness to meeting variation.

Results using all-pairs XC's vs. single reference are mixed although expts. 1 and 11 suggest that all-pairs are useful. The performance of Hilbert envelope vs. GCC-PHAT features is also inconclusive. This varies from results in [91], where positive gains were found, but those experiments were done on smaller data sets and with limited dev set tuning; it is possible that, here, we have overtrained on the dev set.

Energy ratios are not a reliable diarization feature, as is shown in expts. 2, 4 and 5. However they are somewhat effective when used with CCA to dimension reduce the all-pairs correlations, as shown in expts. 6 and 8. This confirms that fusing location cues arising from different physical properties (time delay and attenuation) can result in a more robust location feature.

The other type of fusion – location and the sound of a voice, as manifested in MFCC’s – is also successful for both types of location information (for XC, compare expts. 1 and 7; for ER, compare expts. 5 and 12).

For XC features, MFCC’s are a much more effective CCA view for XC dimension reduction, as is shown in expt. 7. However, the resulting CCA transformed features may not capture all location information available. This is apparent when comparing expt. 7 with expt. 8, where both energy and MFCC fused delays are fed into separate HMM streams, yielding the overall best diarization result. This result is slightly better than that obtained with single reference GCC-PHAT features when smoothing is removed and no dimension reduction is applied (expt. 9).

Given the small number of meetings available, these conclusions are tentative. The statistical significance of the differences in diarization error between experiments was assessed using the Wilcoxon signed-rank test [59, 45], as implemented in the Matlab statistics toolbox. First, the performance gain obtained when the dynamic
programming smoother is significant, either when it is simply removed (XCOu features) or replaced by CCA’ed all-pairs correlations (CCA(ER,XC),CCA(XC,MFCC) features); p=0.1 when expt. 0 is compared with expt. 8, and p=0.03 for expt. 0 vs. 9. Comparing expts. 8 and 9, we see that the small improvement due to CCA processing is significant at the 80% level (p=0.2). For many other results, we can say that we need more meetings to make statistically significant comparisons.

4.5.3 Summary

It is unclear if the inverse filtered correlation algorithm provides a performance gain over the GCC-PHAT delay estimator. But when the inverse filtered features are used, it is more effective to calculate them over all pairs and then apply PCA than it is to apply PCA to pairs calculated with only a single reference.

The PCA approach to diarization feature reduction [91] is an effective method of removing information not helpful to diarization from microphone pairs which happen to be of low quality, although CCA, through the use of side information available in MFCC’s is superior for both ER and XC location features.

Energy ratios are not useful by themselves as a diarization feature, although they can be used to improve correlations via CCA. Correlations CCA’ed with energy ratios appear to contain useful information not present in the MFCC/XC CCA output, as performance is best with two correlation HMM streams, one with correlations CCA’ed with correlations, and one with energy ratios.

It is worth noting that the range between the best and worst single-meeting diarization scores using the same features is as large as 50% and is never lower than 20.7%. Given the wide meeting variability and the fact that there are only 31 meetings in the dev and test corpus, it is clear that large changes in test set results can be caused by swapping only one or two meetings between the dev and test sets. In sparse data situations like this one, leave-one-out evaluation is often employed, where a single point in the data is declared to be the “test” and the rest are used for tuning.
a classifier. Then a new test point is selected and the tuning procedure repeats until all points have been tested. The merit of a classifier is then judged on the average of the single point test results. Unfortunately, a leave-one-out scheme was not practical for meeting data, where a diarization run on a single meeting can take up to an hour of CPU time on the computers available for this work.
Chapter 5

AUTOMATIC DIMENSION CHOICE

In Chapter 4, we calculated correlations and energy ratios across all microphone pairs (and three frequency bands in the case of energy ratios). The result was projected into a lower dimensional space, which was fixed to the same size for all meetings. Because of speaker and microphone variation, a fixed choice is not optimal for each meeting, as is shown in Tables 5.1 and 5.2. In the first column of the table are the PCA dimensions that yielded the best diarization error for each meeting (MFCC and PCA’ed all-pairs correlation lag features); the next column is the difference between the optimal dimensions and the fixed dimension which yielded the best total dev corpus diarization error. The “Diarization Error” columns of Tables 5.1 and 5.2 show the diarization error degradation due to the fixed dimension choice, which was optimal in only one dev and one test corpus meeting. Correctly selecting the PCA dimension for each meeting would result in a 1.6% and 2.0% (absolute) diarization error improvement on the dev and test corpora, respectively.

Besides linear projection to a lower dimensional space, the feature dimension can be reduced by feature selection – removing inputs which are uninformative. Although we have argued that all-pairs location features provide benefits, it is possible that a subset of channel pairs may be especially noisy and thus eliminating them altogether might be preferred over giving them a small transform weight. Therefore, we will also explore feature selection as an alternative to dimension reduction transforms. In the following sections, we begin with automatic methods for choosing the PCA dimension.

\footnote{The entries are filled in with a '*' when the meeting had fewer microphone pairs than the fixed choice, in this case, 7. For example, meeting CMU20050912-0900 had only one microphone pair, so the entry in the best PCA dim column is '1*'.}
Table 5.1: Diarization error loss due to fixed PCA dimension (Dev Set). The '*' indicates that the meeting had fewer than 7 microphone pairs.

<table>
<thead>
<tr>
<th>Meeting</th>
<th>PCA dim best</th>
<th>ΔDim</th>
<th>Diarization Error best dim=7 loss, dim=7</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMI20041210-1052</td>
<td>12</td>
<td>5</td>
<td>2.2 2.6 0.4</td>
</tr>
<tr>
<td>CMU20050301-1415</td>
<td>2</td>
<td>-5</td>
<td>6.0 7.5 *</td>
</tr>
<tr>
<td>CMU20050912-0900</td>
<td>1*</td>
<td>*</td>
<td>26.7 26.7 *</td>
</tr>
<tr>
<td>EDI20050216-1051</td>
<td>3</td>
<td>-4</td>
<td>19.7 24.2 4.5</td>
</tr>
<tr>
<td>ICSI20010208-1430</td>
<td>2</td>
<td>-5</td>
<td>7.4 7.8 0.3</td>
</tr>
<tr>
<td>ICSI20010322-1450</td>
<td>12</td>
<td>5</td>
<td>7.0 11.3 4.3</td>
</tr>
<tr>
<td>ICSI20010531-1030</td>
<td>2</td>
<td>-5</td>
<td>9.9 11.3 1.4</td>
</tr>
<tr>
<td>ICSI20011030-1030</td>
<td>3</td>
<td>-4</td>
<td>9.8 10.0 0.2</td>
</tr>
<tr>
<td>LDC20011116-1500</td>
<td>5</td>
<td>-2</td>
<td>12.2 13.2 1.0</td>
</tr>
<tr>
<td>LDC20011121-1700</td>
<td>10</td>
<td>3</td>
<td>4.7 7.6 2.9</td>
</tr>
<tr>
<td>NIST20020214-1148</td>
<td>12</td>
<td>5</td>
<td>16.7 19.9 3.2</td>
</tr>
<tr>
<td>NIST20030925-1517</td>
<td>11</td>
<td>4</td>
<td>22.3 25.9 3.5</td>
</tr>
<tr>
<td>NIST20050412-1303</td>
<td>11</td>
<td>4</td>
<td>35.6 36.4 0.7</td>
</tr>
<tr>
<td>NIST20050427-0939</td>
<td>7</td>
<td>0</td>
<td>6.9 6.9 0.0</td>
</tr>
<tr>
<td>NIST20051024-0930</td>
<td>5</td>
<td>-2</td>
<td>5.9 9.7 3.8</td>
</tr>
<tr>
<td>NIST20051102-1323</td>
<td>11</td>
<td>4</td>
<td>9.1 9.3 0.2</td>
</tr>
<tr>
<td>TNO20041103-1130</td>
<td>9</td>
<td>2</td>
<td>23.6 26.1 2.5</td>
</tr>
<tr>
<td>VT20050318-1430</td>
<td>1*</td>
<td>*</td>
<td>20.1 20.1 *</td>
</tr>
<tr>
<td>VT20051027-1400</td>
<td>6*</td>
<td>*</td>
<td>14.41 14.4 *</td>
</tr>
<tr>
<td>dev corpus total</td>
<td></td>
<td></td>
<td>13.6 15.2 1.6</td>
</tr>
</tbody>
</table>
Table 5.2: Diarization error loss due to fixed PCA dimension (Test Set). The '*' indicates that the meeting had fewer than 7 microphone pairs.

<table>
<thead>
<tr>
<th>Meeting</th>
<th>PCA dim</th>
<th>Diarization Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>best</td>
<td>ΔDim</td>
</tr>
<tr>
<td>AMI20050204-1206</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>CMU20050228-1615</td>
<td>3</td>
<td>-4</td>
</tr>
<tr>
<td>CMU20050914-0900</td>
<td>1*</td>
<td>*</td>
</tr>
<tr>
<td>EDI20050218-0900</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>ICSI20000807-1000</td>
<td>4</td>
<td>-3</td>
</tr>
<tr>
<td>ICSI20011113-1100</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>LDC20011116-1400</td>
<td>2</td>
<td>-5</td>
</tr>
<tr>
<td>LDC20011207-1800</td>
<td>6</td>
<td>-1</td>
</tr>
<tr>
<td>NIST20020305-1007</td>
<td>5</td>
<td>-2</td>
</tr>
<tr>
<td>NIST20030623-1409</td>
<td>5</td>
<td>-2</td>
</tr>
<tr>
<td>VT20050304-1300</td>
<td>1*</td>
<td>*</td>
</tr>
<tr>
<td>VT20050623-1400</td>
<td>6*</td>
<td>*</td>
</tr>
<tr>
<td><strong>test corpus total</strong></td>
<td><strong>11.0</strong></td>
<td><strong>13.0</strong></td>
</tr>
</tbody>
</table>
and then proceed to feature selection.

5.1 Automatic choice of PCA dimension

Many PCA dimension choice techniques operate under the assumption that the PCA input is a linear, instantaneous mixture of independent signals and noise. That is, they assume that the PCA input is:

\[ x = At + n \]  

(5.1)

where \( x \) is a \( D \times 1 \) vector, \( A \) is a \( D \times d \) matrix, \( t \) is a \( d \times 1 \) random vector composed of \( d \) independent signals, and \( n \) is a \( D \times 1 \) stationary and ergodic Gaussian noise vector with zero mean and variance \( \sigma^2 I \). For this model, the correlation matrix of \( x \) can be shown to be [116]:

\[ R = ASA^H + \sigma^2 I \]

where \( S \) is the covariance matrix of \( t \), and \( H \) is the Hermitian transpose. Assuming that the columns of \( A \) are linearly independent - a property which is satisfied by PCA - then largest \( p \) eigenvalues of \( R \) are \( \lambda_0 \geq \lambda_1 \ldots \lambda_{d-1} \). Assuming \( \sigma^2 \) is small compared to the signal covariance, the remaining eigen vectors are then small and constant: \( \lambda_{d:D-1} = \sigma^2 \leq \lambda_{d-1} \).

This provides some theoretical justification for the widely used “scree” heuristic, in which PCA eigenvalues, when plotted in order of decreasing of magnitude, are inspected for a steep drop-off which would indicate a transition between \( \lambda_{d-1} \) and \( \lambda_d \). If the assumptions above are met, the drop-off resembles a cliff, with the noise eigenvalues looking like rubble, or scree, at the bottom. Although inspection is usually done visually, the procedure has been automated, for example, in [129], where eigenvalues above and below the cliff are modeled as two Gaussians with different means, and where the drop-off is located by maximum likelihood estimation.

Unfortunately, the assumptions in (5.1) do not match the distribution of correlation lags found in the NIST meetings (Figure 5.1). As is typically the case, almost
all of the variance ends up in the first few principal components, revealing that the signal/noise model is a poor fit. In addition, when searching for the drop-off point, the scree heuristic is attempting to the minimize reconstruction error of a “true” signal by separating it from noise. However, for this work, the goal is not to minimize reconstruction error, but classification error. In any case, although there is often a second drop-off in the scree plots, visual inspection shows that it is almost never at the dimension yielding the lowest diarization error, which is at $d = 12$ for the meeting in Figure 5.1.

Several other methods attempt to detect a transition between signal and noise
spaces using only the eigenvalues of the correlation matrix, \( \mathbf{R} \), and eigenvalues alone. In the imbedded (sic) error function (IEF) method \[79\], the transition is detected at the minimum of:

\[
IEF(l) = \left( \frac{l \sum_{j=l+1}^{D-1} \lambda_j}{N D (D - l)} \right)^{1/2}
\]

where \( N \) is the number of samples used to calculate the correlation matrix. The idea is that, from \( l = 0 \) to \( l = d - 1 \), the projection is capturing a decreasing amount of signal power, causing \( IEF(l) \) to drop. For \( l \geq d \) the eigenvalues will be constant, causing \( IEF(l) \) to begin increasing.

Abandoning the assumption of constant low eigenvalues, some authors have tried thresholding the sum of eigenvalues (at 90-95% of the total) and the eigenvalues themselves \[112\].

More theoretically justified techniques \[116\] based on the signal/noise model in (5.1) employ minimum description length (MDL) and the Akaike information criteria (AIC) \[102, 5\] which choose between models by trading off model fit and complexity, as with the BIC metric discussed in Section \[2.1.1\]. The AIC and MDL criteria to be minimized are:

\[
AIC(l) = \gamma + 2M
\]

\[
MDL(l) = \gamma + 2M \log N
\]

\[
\gamma = -2 \log \left( \prod_{j=l}^{D-1} \lambda_j^{1/(d-l)} \frac{1}{1/(D-l) \sum_{j=l}^{D-1} \lambda_j} \right)^{(D-l)N}
\]

where \( M \) is the number of free parameters in the PCA signal model: \( M = l(2D - l) \) for complex eigenvectors and \( M = \frac{1}{2} (D - l) \) for real eigenvectors. MDL was found to be consistent for large \( N \), while AIC tended to overestimate the PCA dimension.

With the added assumption that both the noise and the low dimensional sources, \( t \), have spherical Gaussian distributions, the PCA input observation probability is
also a Gaussian:

\[ p(x | A, m, \sigma) \sim \mathcal{N}(m, AA^T + \sigma I) \]

where \( m \) is the \( D \times 1 \) mean of \( x \), \( n \sim \mathcal{N}(0, \sigma I_D) \) and \( t \sim \mathcal{N}(0, I_d) \). In probabilistic PCA (PPCA), \( A \) and \( m \) can be calculated using an expectation maximization algorithm \[\text{[11]}\]. Although the parameters estimated by PPCA converge to those of standard PCA, the probabilistic framing enables new methods for estimating the proper PCA dimension. In \[\text{[19]}\], the PPCA parameters were calculated with Bayesian model selection, which estimates parameters by maximizing data likelihood, integrated over the universe of model choices, each of which is weighted by a prior probability. The advantage of Bayesian techniques is that they choose simpler models when sparse data provides less support for complex models, thus avoiding over-generalization. In the original Bayesian PCA (BPCA) algorithm, the parameter prior is:

\[ p(A | \alpha) = \prod_{i=1}^{D-1} \left( \frac{\alpha_i}{2\pi} \right)^{D/2} \exp \left\{ -\frac{1}{2} \alpha_i \|a_i\|^2 \right\} \]

where \( \{a_i\} \) are the columns of \( A \) and where \( \alpha \) is a \( D - 1 \) dimensional hyper-parameter which is estimated iteratively to maximize the data likelihood. When \( \alpha_i \) becomes large, \( a_i \) becomes improbable, driving that column of \( A \) towards zero, in effect, reducing the dimensionality of the low dimensional source vector, \( t \). The effective dimensionality is then the number of non-zero columns of \( A \).

The computational properties of Bayesian PCA were improved in \[\text{[83]}\], where a new prior was designed to prefer an \( A \) with few effective dimensions. The Laplace method was used to approximate the attendant Bayesian integral, an exact solution of which was analytically intractable. The end result is a non-iterative and computationally inexpensive dimension estimating algorithm, which performed better than \( \text{MDL}^2 \) cross-validation techniques, and the original BPCA algorithm in cases where the data closely or fairly closely matched the model assumptions. However,

\(^2\text{Specifically, the authors tested a BIC approach, which can be considered an approximation of MDL for parametric models.}\)
the author of [83] notes that, because the probabilistic model is essentially chosen to minimize reconstruction error, the method may not be suitable for detecting salient features. He also found that it was prone to highly overestimating the dimension on certain real world data.

The PCA dimension is also frequently chosen by minimizing reconstruction error in cross-validation tests. The idea common to these techniques is that the ability to reconstruct noisy inputs using (4.1) is optimal at the correct PCA dimension. Two example methods are the predicted sum of residual squares (PRESS) [118, 112] and the variance of reconstruction error (VRE) [99].

In PRESS, PCA input dimensions are randomly declared “missing” and then a PCA transform is calculated using a PCA technique capable of estimating $A$ in the presence of missing data e.g. PPCA or non-linear iterative partial least squares (NIPALS) [119]. In other words, several randomly selected input frames, $n$, are chosen, and within each frame, one or more inputs, $x_l[n] \in \{x_0[n], x_1[n], \ldots, x_d[n]\}$ are selected at random and are declared missing. The PCA transform is computed and then, for varying PCA output dimensions, the “missing” inputs are reconstructed using (2.1). The correct PCA dimension is estimated to be the point at which the average reconstruction error is minimized.

In VRE method, $A$ is estimated only once, using all inputs. Next, assuming a candidate output dimension, $p$, each input is reconstructed using only data from other inputs. VRE then selects the PCA dimension which minimizes the variance of reconstruction error across the inputs. In [112] VRE was tested against several real data sets, and was found to be more reliable than AIC, MDL, IEF, and a scree test based on summed eigenvalues. Performance was comparable to PRESS, but VRE required much less computation. However, since VRE and PRESS use reconstruction-based objective functions, they may not be applicable to diarization, where the objective is to minimize classification error.

For this thesis, we compared two representative techniques; the automated scree
method and Laplacian-approximated BPCA. We chose the former because it is widely used and is the simplest of the methods above. The later was selected because the literature suggests that it is the most accurate technique of those discussed in this section.

5.2 Feature Selection

As discussed in Chapter 4, all pairs in a meeting contain location information which provides a clue to speaker identity, but location features derived from certain channel pairs are more reliable than others. It is possible that the classification error induced by unreliable pairs outweighs the benefit of including them in the location feature vector. In this section, we discuss methods for selecting only pairs that are likely to improve diarization error.

Measurements of microphone signal quality – signal power, signal-to-noise ratio, correlation peak value – are clearly related to the value of a microphone pair for classification. However, pilot experiments showed that it was difficult to use these measures to identify the utility of pairs for classification. The problem was that, in the NIST data, speaker and microphone properties are vastly different from one meeting to the next; thresholds on these measures tuned on the development set failed on the test set, as did “bad pair” classifiers trained on the development set. It was easy to measure signal power, etc. but it was difficult to determine the impact of such a measure on diarization accuracy for a particular meeting.

Therefore, we decided to study class-based feature selection methods, which directly measure classification impact with the use of class labels. These labels are not actually available in diarization – speaker ID’s are unknown – so we will test the method using estimated labels.

A property shared by all supervised feature selection methods is that they seek to determine if there is a relationship between a feature and the class label. In gene selection work, the F-test is commonly employed [40]. The F-test assumes that the
data is composed of multiple, normally distributed populations with different means but the same variance. The null hypothesis is that the means are the same for each class. Features are selected when the F-test rejects the null hypothesis, indicating that the feature has a different distribution mean for different classes. Unfortunately, the F-test is not robust to non-normally distributed data, a condition present in diarization location features (see Figure 4.4).

More general relationships between class labels and features can be detected with mutual information:

\[
I(x; y) = \sum_{i,j} p(x^i, y^j) \log \frac{p(x^i, y^j)}{p(x^i)p(y^j)}
\]  

(5.2)

Here, \(p(z^i)\) is the prior probability of an input or class label having a discrete value \(z^i\), and where \(p(x^i, y^j)\) is the joint probability of \((x^i, y^j)\). For this work, the individual elements of the feature vector, \(x_k\), are continuous. However, as will be shown, good results can be obtained by discretizing them and estimating the probability distribution from histograms. An example of the use of mutual information can found in [30], where Bayesian network conditional independence was assessed using thresholded mutual information between features. The determination of the threshold was improved in [128], where the distribution of mutual information estimated in (5.2) was itself estimated, leading to a threshold determination method which selects only features highly certain to be relevant to the class label.

The problem with selecting all of the features relevant to the class label is that they may be redundant. Many researchers have observed that the best features for classification are both relevant and non-redundant. One method for ensuring maximum relevance and minimum redundancy (mRMR) is found in [38], where features are accumulated in a greedy fashion. The first feature selected, \(f_0 \in \{x_0, x_1, \ldots, x_{D-1}\}\), is the one with the highest class label mutual information, and subsequent features...
are selected based upon following criterion:

\[ f_k = \arg\max_{x_k \notin f_{0...k-1}} I(x_k; y) - \frac{1}{k-1} \sum_{k'=0}^{k-1} I(x_k; f_{k'}) ; k = 1 \ldots D - 1 \]

Features that are highly relevant to the class label yet not redundant with those already selected are preferred. The two variable mutual information calculations express neither the true discriminatibility nor the redundancy of the features chosen at step \( k \), but limiting the distribution estimation problem to two dimensions makes the problem computationally tractable, and reduces distribution estimation problems inherent in a mutual information calculation with more variables. The authors of [95] have proven that, even though mRMR is a first order greedy search, it is equivalent to finding the set of features which \textit{jointly} have the largest dependency upon the class label. Empirically, mRMR features have yielded lower classification error on real data than features derived with the F-test [38] and with maximum mutual information.

Two measures similar to mRMR are the explaining away residual (EAR) and Redundancy Compensated KL distance (RCKL). In EAR [18], nodes in a Bayesian network are connected if

\[ EAR_{xy} = \sum_{c \in C} I(x; y|c) - I(x; y) \]

exceeds a threshold, where \( c \) is a class label in the set of all class labels, \( C \). The idea is that connections are made when there is high class-dependent mutual information between variables \( x \) and \( y \), and when there is also more class-conditional dependence than marginal independence. EAR could conceivably be used to identify good features.

In [21], a document topic classifier feature vector is composed utilizing the following measure of class-label relevance:

\[ KL_w = D(p(c|w)||p(c)) = \sum_{c \in C} p(c|w) \log \frac{p(c|w)}{p(c)} \]

where \( D(p(c|w)||p(c)) \) is the Kullback–Leibler distance between the conditional and marginal distributions of the class label. When selecting features, the algorithm first
adds individual words, \( w_k \), and then decides whether or not to add pairs of consecutive words, \( w_k, w_{k+1} \) (bigrams). The bigram decision is based on the RCKL measure:

\[
RCKL_{w_k w_{k+1}} = KL_{w_k w_{k+1}} - KL_{w_k} - KL_{w_{k+1}}
\]

which gives high scores to bigrams only if they provide more information than the individual word counts, \( \{w_k, w_{k+1}\} \) alone. In [21], RCKL is considered to be superior to EAR because the KL distance is more robust to estimation problems due to infrequently occurring words. In both EAR and RCKL, redundancy is estimated only on pairs of variables, rather than the full set of selected variables (as in mRMR). But perhaps these methods could be extended by calculating averages in a manner similar to the way that mean mutual informations are calculated in mRMR.

The techniques mentioned so far are all “filter techniques,” which select features assuming no knowledge of the algorithm that eventually does the actual classification. Two advantages of such approaches are computational simplicity, and that the features are believed to cover a more representative space [38]. Two disadvantages are that it is not clear how to stop the iterative processes through the use of a threshold tuned on a development set (a big disadvantage for widely variable NIST meetings) and that features are still not being selected purely on classifier discriminatibility.

“Wrapper” techniques do consider discriminatibility. In this class of algorithms, features are iteratively selected, used to train a classifier, and then the classification error is used to assist in selecting the next set of features. A good example using mRMR is found in [96]. Since wrapper techniques require class labels, they are not directly applicable to the unsupervised diarization problem, and therefore, we tested only the mRMR filter technique.

5.3 Experiments

In this section, we test new algorithms, inspired by the previous work outlined above, but tailored to the particular unsupervised classification problem that presents itself
in diarization. First, we test two approaches to PCA dimension determination, and then we test a new, unsupervised feature selection algorithm.

5.3.1 PCA dimension choice

PCA dimension choice by reconstruction error: Scree and Laplace BPCA

As representatives of the error of reconstruction approaches (Section 5.1) we selected the PCA dimension based on the scree and Laplacian BPCA methods. The results are compared with the fixed and oracle dimension choice in the first four rows of Table 5.3. The scree method performs considerably worse than a fixed dimension choice tuned on the dev set. Laplacian PCA matches the test set performance of a fixed dimension choice, but it is worse on the dev set since the fixed dimension is explicitly optimized on the dev set.

PCA dimension choice by linear regression

Here, we attempt to use the microphone and speaker properties present in a meeting to predict the choice that will produce the lowest diarization error. Several regression variables found to be correlated with the lowest error PCA dimension were chosen as independent regression variables:

- **number of channel pairs**: Number of channel pairs divided by the number of channels in a meeting

- **number of estimated speakers**: Number of speakers estimated by the clusterer.

- **best scree PCA dimension**: The best PCA dimension estimated by the scree method of [129], divided by the number of channels
• **number of closely spaced microphones**: Number of microphones that were probably within 10cm of another microphone, based on the information in Table 3.1 (divided by the number of channels).

• **number of scattered microphones**: Number of microphones that were probably further than 10cm from the closest microphone, based on the information in Table 3.1 (divided by the number of channels).

• **sum of coherence distance**: The sum of the estimated inter-microphone pair distances, as estimated by their coherence using the algorithm in [81], divided by the number of channels.

Linear regression was then used to predict the PCA dimension with the lowest diarization error. Two types of regression were tried. The first was standard linear regression, where the regression cost was the squared difference between the regression output and the dimension with the lowest diarization error for a given meeting. For certain meetings, a large error in predicting the best dimension did not cause a large change in diarization error. Therefore, a second regression technique was developed, where regression coefficients were tuned to explicitly minimize diarization error. The Nelder-Mead simplex (direct search) method was employed (via that Matlab’s fminsearch() function). Within this iterative algorithm the normalized PCA dimension was predicted as \( \hat{d} = a \cdot z \), where \( a \) is a vector of regression coefficients, and \( z \) is the vector of features listed above. The algorithm adjusted \( a \) to minimize the following cost: \( C(\hat{d}) = derr(\hat{d}) - derr(d^*) \), where \( d^*\) is the dimension choice which yielded the lowest diarization error.

The results for the two regression methods are shown in Table 5.3. Linear regression results in a 0.1% decrease in test corpus error over a fixed dimension chosen by tuning on the dev corpus, but the error on the dev corpus is 0.7% higher. As expected, optimizing the regression with a dev set diarization error reduces the dev
Table 5.3: Diarization Error vs. PCA dimension choice: standard techniques and regression

<table>
<thead>
<tr>
<th>Method</th>
<th>error (dev)</th>
<th>error (test)</th>
</tr>
</thead>
<tbody>
<tr>
<td>oracle (Tables 5.1 and 5.2)</td>
<td>13.6</td>
<td>11.0</td>
</tr>
<tr>
<td>fixed dimension (Tables 5.1 and 5.2)</td>
<td>15.2</td>
<td>13.0</td>
</tr>
<tr>
<td>scree</td>
<td>16.9</td>
<td>17.0</td>
</tr>
<tr>
<td>Laplacian BPCA</td>
<td>15.9</td>
<td>13.0</td>
</tr>
<tr>
<td>regression (mean squared dimension error cost)</td>
<td>15.9</td>
<td><strong>12.9</strong></td>
</tr>
<tr>
<td>regression (diarization error cost)</td>
<td>15.5</td>
<td>13.4</td>
</tr>
</tbody>
</table>

set error relative to mean squared error regression, but not below that obtained by simply picking a fixed dimension, which yielded 0.4% worse results on the test corpus.

Possible reasons for the regression failure are numerous. First, a linear regression model may be too simple to express the complex relationship between meeting parameters and PCA dimension. Second, the training data (19 meetings in the dev corpus) is extremely limited, even when training such an uncomplicated model. Indeed, we tried training a support-vector based ridge regression model – arguably only slightly more complicated – and got considerably worse results, probably due to data sparsity. Finally, it is possible that, even given enough training data, the regression variables are not sufficiently informative proxies for diarization error.

**PCA dimension choice by cluster model entropy**

A more direct proxy for diarization error is the set of speaker model posteriors produced at the end of a diarization run. If a run has been successful, one would expect that, for each speech frame, the GMM speaker models should be relatively certain about which speaker is active, that is, the posterior probability for one of the M detected speakers should be high, and the rest, low. A convenient measure of certainty
is entropy:

\[
H(x) = -\frac{1}{M} \sum_{m=0}^{M-1} p(x = x_m) \log p(x = x_m)
\]

where \( p(x = x_m) \) is the probability of a discrete random variable, \( x \), having value \( x_m \). \( H(x) \) is at a maximum when all values \( \{x_m\} \) are equally probable, and is zero if \( x \) has only one value. In speech research \[117, 3, 20, 17, 33\], the term “entropy” is often used to refer to an analogous measure based on model posterior probabilities:

\[
\bar{H}(x|s) = \frac{1}{N} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} p(x[n]|s_m) \log p(x[n]|s_m)
\]

where \( x[n] \) is a feature vector for frame \( n \)th of \( N \) frames (for this work, frames are MFCC’s or one of the location features), and where \( p(x[n]|s_m) \) is the posterior probability of the \( m \)th speech model, \( s_m \) (usually a phone model, but for this work, it will be the diarization system GMM for speaker \( m \), normalized so that the sum of probabilities for a given frame sum to one). \( \bar{H} \), which is low when one model has a much higher posterior probability than the others, and is high when all models have roughly the same posteriors, is a related to conditional entropy, \( H(x|s) \):

\[
H(x|s) = -\sum_{x,s} p(x,s) \log(p(x|s)) = -\sum_{s} p(s) \sum_{x} p(x|y) \log(p(x|s))
\]

For \( \bar{H}(x|s) \), frame averaging is used to approximate the empirical distribution of \( s \), \( p(s) \).

In the dimension choice method tested here, the diarization system is run \( K \) times, with dimension choices of \( d_k \in \{2, 3\ldots 11\} \). On the \( k \)th run, a diarization system given feature vectors reduced to dimension \( d_k \) detects some number of speakers, \( M_k \), and, for each frame, \( n \), it produces an \( M_k \) dimensional vector of model posteriors, \( p^k[n] = [p^k(x[n]|S_0), p^k(x[n]|S_1), \ldots p^k(x[n]|M_k)] \). The run estimated to have had the lowest diarization error, \( k^* \), is the one that produces the most certain ensemble of model posteriors during speech, as measured by some entropy-based function, \( \mathcal{H}(\cdot) \),
of the posteriors over all speech:

\[ k^* = \text{argmin}_k \mathcal{H}(p^k[n_s, n_s \in \text{speech}]) \]

where the \( \mathcal{H} \) notation is used generically here. Finally, the dimension choice is \( d^* = d_{k^*} \). Several considerations figure into the choice of \( \mathcal{H}(\cdot) \):

- **What posterior?** In [5.3], the per-frame entropy is calculated and then averaged over the frames. Would it be better to average the model posteriors first, and then compute the entropy on the mean posteriors?

- **Which feature?** The GMM’s are built for MFCC’s and one or more location features. Which feature(s) should be used to calculate the entropy?

Several versions of \( \mathcal{H}(\cdot) \) were tested, and results are reported here for the three most successful variants. The first was the averaged frame entropy calculated over the frame posteriors:

\[
\bar{H}^p(x) = -\frac{1}{N \log M_k} \sum_{n=0}^{N-1} \sum_{m'=0}^{M-1} p(x[n]|s_m) \log p(x[n]|s_m)
\]  

(5.4)

where \( \log M_k \) normalizes by the maximum entropy possible, given the number of speaker detected on the \( k \)th run. The second variant was the averaged frame entropy, weighted by the number of frames the clusterer has assigned to a speaker:

\[
\bar{H}^p_s(x) = -\frac{1}{M} \sum_{m=0}^{M-1} \frac{1}{|\text{frames}(m)|} \sum_{n \in \text{frames}(m)} \frac{1}{\log M} \sum_{m'=0}^{M-1} p_{m'}(x[n]) \log p_{m'}(x[n])
\]  

(5.5)

The third variant was the average entropy, with probabilities averaged across each speaker’s frames:

\[
H^p_s(x) = -\frac{1}{M} \sum_{m=0}^{M-1} \frac{1}{\log M} \sum_{m'=0}^{M-1} \bar{p}_{m'}^m \log \bar{p}_{m'}^m
\]  

(5.6)
Table 5.4: Diarization error for minimum entropy PCA dimension choice

<table>
<thead>
<tr>
<th>$\mathcal{H}()$ feature</th>
<th>Diarization Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tilde{H}_p(x)$ MFCC</td>
<td>15.0 13.5</td>
</tr>
<tr>
<td>$\tilde{H}_r(x)$ XCC</td>
<td>15.4 13.5</td>
</tr>
<tr>
<td>$\tilde{H}_s^p(x)$ MFCC</td>
<td>18.6 14.5</td>
</tr>
<tr>
<td>$\tilde{H}_s^r(x)$ XCC</td>
<td>16.8 14.1</td>
</tr>
<tr>
<td>$H_s^p(x)$ MFCC</td>
<td>14.7 13.6</td>
</tr>
<tr>
<td>$H_s^r(x)$ XCC</td>
<td>15.3 13.7</td>
</tr>
</tbody>
</table>

where $\bar{p}_m$ is the average posterior probability for a speaker $m$:

$$\bar{p}_m^k = \frac{1}{|frames(k)|} \sum_{n \in frames(k)} p_{m'}(x[n])$$

The results are shown in Table 5.4. For both MFCC and XCC features, the best approach is the simple averaged frame entropy, and PCA dimension works about as well for either feature. Unfortunately, none of the algorithms beat the fixed dimension choice or regression (Tables 5.2 and 5.3).

5.3.2 Feature Selection

An alternative to PCA or CCA transformations for dimension reduction is to simply select the best channel pairs for clustering. The baseline system does this by selecting the most correlated reference channel and then correlating it against the rest, the assumption being that these will be the best channel pairs. Here, we explore other pair selection methods.
Table 5.5: Max-Min SNR Feature Selection

<table>
<thead>
<tr>
<th>Location Feature</th>
<th>Diarization Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dev</td>
</tr>
<tr>
<td>ER</td>
<td>23.7</td>
</tr>
<tr>
<td>XC</td>
<td>19.3</td>
</tr>
</tbody>
</table>

Selection based on max-min SNR

For pair correlation features, it is reasonable to expect that correlations where one or more channel is noisy will be unreliable. So, this most basic level of feature selection is tested here. First, the signal-to-noise ratio for each channel is estimated using the standard NIST SNR measuring tool also used in Section 3.3. Then, channel pairs are sorted by the minimum SNR channel in the pair. The SNR estimating tool occasionally breaks down for noisy channels, yielding an erroneous zero SNR estimate; in those cases, pairs were instead sorted by the maximum SNR estimate. The first \( nChans - 1 \) pairs are then selected as diarization inputs. Considering only channels with two good estimates, we find the minimum SNR for each pair and then select the pairs with the maximum minimum SNR. If there are fewer than \( nChans - 1 \) pairs with two good SNR estimates, then selection continues on the remaining list of pairs, selecting the pairs with the highest maximum SNR first. \( nChans - 1 \) pairs are chosen to match the baseline system.

The resulting diarization errors are shown in Table 5.5. Max-min SNR pair selection yielded a very poor energy ratio (ER) feature result. For correlation lag features (XC), performance is considerably worse than for the best PCA dimension reduction choice (see Tables 5.1 and 5.2).
**mRMR Feature Selection**

Here, we calculate all pairs and then select the best based on the mRMR method discussed in Section 5.2. One difficult aspect of the iterative mRMR method is knowing when to stop the iteration. For the first series of experiments, we simply chose the same number of pairs as was chosen by the baseline system: $nChans - 1$. The other difficult aspect is that the technique requires class labels, which are not available. So, here, we use the estimated class labels from the diarization run with the lowest error in Table 4.2. Then we run mRMR and make another diarization run with these features. No dimension reduction is done after feature selection. The final difficulty is in estimating the probability distributions required for the entropy calculations. For this work, the continuous variables were quantized, and then the distributions were estimated from histograms with $3\sqrt{N}$ X $3\sqrt{N}$ bins between the minimum and maximum value of the variables. This follows the practice of [84], where mutual information bias due to the quantization is corrected based on maximum likelihood Gaussian assumptions, or a with a factor designed to minimize the maximum mean square error (MSE) of the mutual information estimate. In addition, when there is a good chance of having empty histogram bins due to a low number of samples, it is common to smooth the histogram counts using the Laplacian prior:

$$\hat{p}_L = \frac{c + 1}{N + M}$$

where $c$ is the original bin count.

The results for XC features are shown in Table 5.6. The best performance, which comes with no smoothing and the maximum MSE bias correction, is slightly better than that obtained by selecting the PCA dimension with regression or entropy (Tables 5.3 and 5.4). It is 0.3% better than the result obtained with a fixed PCA dimension trained on the dev set but is 2.7% greater than the oracle result (Tables 5.2 and 5.2).

Clearly, the selection of $nChans - 1$ features is arbitrary. To see what is lost
Table 5.6: Diarization error for mRMR feature reduction ($nChans - 1$ XC pairs)

<table>
<thead>
<tr>
<th>histogram</th>
<th>bias</th>
<th>dev</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td>smoothing</td>
<td>compensation</td>
<td>derr</td>
<td>derr</td>
</tr>
<tr>
<td>none</td>
<td>unbiased</td>
<td>20.9</td>
<td>15.9</td>
</tr>
<tr>
<td>none</td>
<td>MSE</td>
<td>15.3</td>
<td>12.7</td>
</tr>
<tr>
<td>Laplacian</td>
<td>unbiased</td>
<td>15.8</td>
<td>13.2</td>
</tr>
<tr>
<td>Laplacian</td>
<td>MSE</td>
<td>15.8</td>
<td>13.2</td>
</tr>
</tbody>
</table>

by that choice, an oracle experiment was performed where diarization was run over varying numbers of the best mRMR-ranked features. The results are shown in Tables 5.7 and 5.8. If, instead of choosing $nChans - 1$ features, the oracle number of features was chosen for each meeting, the diarization error would improve by 2.1% on the dev set and 2.3% on the test set. Looking down columns 2 to 4, the improvement does not seem to be the result of $nChans - 1$ being either systematically too large or too small. This makes sense because the best number of pairs should depend upon the microphone configuration and the number and location of speakers, rather than the number of microphones.

Comparing the feature selection oracle results in Tables 5.7 and 5.2 to the oracle PCA dimension choices in Tables 5.7 and 5.2, we also see that a larger gain is possible with an oracle number of mRMR features than with an oracle PCA dimension choice.

5.4 Summary

In this chapter, it was shown that, when XC features are dimension reduced with PCA, automatic PCA dimension selection can yield slightly better results than those obtained with a fixed dimension choice. The best dimension selection method was also the simplest: linear regression on meeting acoustic parameters, although the Laplacian BPCA algorithm was nearly as good, matching the performance obtained
Table 5.7: Diarization error for mRMR feature reduction (fixed or oracle number of XC pairs, Dev)

<table>
<thead>
<tr>
<th>Meeting</th>
<th>nChans-1</th>
<th>numFeats</th>
<th>oracle best nChans-1</th>
<th>Diariz. Error (Dev)</th>
<th>Total</th>
<th>Oracle Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMI20041210-1052</td>
<td>11</td>
<td>4</td>
<td>7</td>
<td>3.1</td>
<td>2.5</td>
<td></td>
</tr>
<tr>
<td>CMU20050301-1415</td>
<td>2</td>
<td>3</td>
<td>-1</td>
<td>11.5</td>
<td>7.2</td>
<td></td>
</tr>
<tr>
<td>CMU20050912-0900</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>27.1</td>
<td>27.1</td>
<td></td>
</tr>
<tr>
<td>EDI20050216-1051</td>
<td>15</td>
<td>4</td>
<td>11</td>
<td>24.3</td>
<td>17.1</td>
<td></td>
</tr>
<tr>
<td>ICSI20010208-1430</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>7.3</td>
<td>7.3</td>
<td></td>
</tr>
<tr>
<td>ICSI20010322-1450</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>11.0</td>
<td>5.9</td>
<td></td>
</tr>
<tr>
<td>ICSI20010531-1030</td>
<td>5</td>
<td>12</td>
<td>-7</td>
<td>10.3</td>
<td>10.1</td>
<td></td>
</tr>
<tr>
<td>ICSI20011030-1030</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>12.9</td>
<td>9.3</td>
<td></td>
</tr>
<tr>
<td>LDC20011116-1500</td>
<td>7</td>
<td>7</td>
<td>0</td>
<td>12.6</td>
<td>12.6</td>
<td></td>
</tr>
<tr>
<td>LDC20011121-1700</td>
<td>9</td>
<td>11</td>
<td>-2</td>
<td>4.6</td>
<td>4.5</td>
<td></td>
</tr>
<tr>
<td>NIST20020214-1148</td>
<td>6</td>
<td>12</td>
<td>-6</td>
<td>18.4</td>
<td>12.8</td>
<td></td>
</tr>
<tr>
<td>NIST20030925-1517</td>
<td>6</td>
<td>5</td>
<td>1</td>
<td>26.7</td>
<td>19.6</td>
<td></td>
</tr>
<tr>
<td>NIST20050412-1303</td>
<td>6</td>
<td>8</td>
<td>-2</td>
<td>44.2</td>
<td>35.8</td>
<td></td>
</tr>
<tr>
<td>NIST20050427-0939</td>
<td>6</td>
<td>13</td>
<td>-7</td>
<td>8.6</td>
<td>7.2</td>
<td></td>
</tr>
<tr>
<td>NIST20051024-0930</td>
<td>6</td>
<td>5</td>
<td>1</td>
<td>5.6</td>
<td>5.5</td>
<td></td>
</tr>
<tr>
<td>NIST20051102-1323</td>
<td>6</td>
<td>15</td>
<td>-9</td>
<td>9.0</td>
<td>8.7</td>
<td></td>
</tr>
<tr>
<td>TNO20041103-1130</td>
<td>9</td>
<td>14</td>
<td>-5</td>
<td>22.5</td>
<td>22.1</td>
<td></td>
</tr>
<tr>
<td>VT20050318-1430</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>25.0</td>
<td>25.0</td>
<td></td>
</tr>
<tr>
<td>VT20051027-1400</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>14.0</td>
<td>14.0</td>
<td></td>
</tr>
<tr>
<td><strong>ALL</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>15.3</strong></td>
<td></td>
<td><strong>13.2</strong></td>
</tr>
</tbody>
</table>
Table 5.8: Diarization error for mRMR feature reduction (fixed or oracle number of XC pairs, Test)

<table>
<thead>
<tr>
<th>Meeting</th>
<th>nChans-1</th>
<th>numFeats</th>
<th>oracle best</th>
<th>nChans-1 - oracle</th>
<th>Diariz. Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMI20050204-1206</td>
<td>15</td>
<td>17</td>
<td>-2</td>
<td>6.2</td>
<td>6.0</td>
</tr>
<tr>
<td>CMU20050228-1615</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>12.7</td>
<td>12.0</td>
</tr>
<tr>
<td>CMU20050914-0900</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>17.9</td>
<td>17.9</td>
</tr>
<tr>
<td>EDI20050218-0900</td>
<td>15</td>
<td>4</td>
<td>11</td>
<td>21.4</td>
<td>8.8</td>
</tr>
<tr>
<td>ICSI20000807-1000</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>7.5</td>
<td>7.3</td>
</tr>
<tr>
<td>ICSI20011113-1100</td>
<td>5</td>
<td>13</td>
<td>-8</td>
<td>22.5</td>
<td>17.4</td>
</tr>
<tr>
<td>LDC20011116-1400</td>
<td>7</td>
<td>2</td>
<td>5</td>
<td>5.4</td>
<td>5.3</td>
</tr>
<tr>
<td>LDC20011217-1800</td>
<td>3</td>
<td>5</td>
<td>-2</td>
<td>19.6</td>
<td>14.7</td>
</tr>
<tr>
<td>NIST20020305-1007</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>8.8</td>
<td>8.2</td>
</tr>
<tr>
<td>NIST20030623-1409</td>
<td>6</td>
<td>5</td>
<td>1</td>
<td>2.6</td>
<td>1.7</td>
</tr>
<tr>
<td>VT20050304-1300</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>4.9</td>
<td>4.9</td>
</tr>
<tr>
<td>VT20050623-1400</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>18.9</td>
<td>18.9</td>
</tr>
<tr>
<td><strong>ALL</strong></td>
<td></td>
<td></td>
<td></td>
<td>12.7</td>
<td>10.4</td>
</tr>
</tbody>
</table>
with a fixed dimension. Unfortunately, none of the model entropy approaches were better than a fixed dimension choice. It is possible that the relative merits of these algorithms would change if applied to CCA dimension choice, a study left for future work.

Better results were obtained by using the mRMR feature selection technique to automatically choose a subset of microphone XC pairs as diarization features, and then doing no PCA dimension reduction. For these tests, the number of pairs was constrained to \(\text{numChans} - 1\) but oracle results show that this was suboptimal. Left for future work is the development of algorithms to automatically select the number of pairs, possibly similar to those one of those used to select the PCA dimension. It is possible that additional gains are available when features are first selected with an algorithm like mRMR and then dimension reduced with PCA or CCA. We also leave this study to future work.

The large variations in diarization performance across meetings, along with various oracle results and analyses obtained in this thesis, indicate that meeting-level adaptive algorithms are needed to adjust to the different acoustic conditions found in meeting recordings. Examples include the number of microphones, average microphone signal to noise ratio, the unknown microphone spacing, the unknown number of speakers, etc. The improved performance found in this thesis with meeting-dependent feature selection is a first step in this direction, and we have described some extensions. Meeting-dependent adaptation could be developed for other aspects of the diarization system. For example, it may be possible to detect meetings where location features tend to lock on fan noise, or where energy ratios are useful speaker ID features. In practice, there may be several recordings for a given site, providing an opportunity to adapt over a data set longer than one meeting.
Chapter 6

OVERLAP HANDLING

As was mentioned in Chapter 2, the handling of the short overlaps that occur naturally in conversational speech is an open research problem, and one, if addressed, would benefit both speaker diarization and ASR. One way of dealing with overlaps would be to separate speakers with a beamforming algorithm designed to always search for and null interference, or with a source separation that that constantly searches for a way to make its outputs statistically independent. However, these methods have difficulties with one or more of the conditions present in realistic, multi-microphone meeting speech, or they are not designed for arbitrarily spaced microphones [71].

Another approach to overlaps is to detect simply detect that an overlap has occurred, and then to process the region separately. Assuming that accurate overlap detection is available, Section 6.1 explores the best way to use that information for diarization and shows that a simple nearest neighbor scheme can approach the diarization gains available from “perfect” overlap assignment. In Section 6.2, a monaural overlap detector is proposed and tested, while in Section 6.3, a detector is proposed which makes use of the location information available in multi-microphone meetings. Finally, the work is summarized in Section 6.4.

6.1 Use of Overlap Information

Speaker overlap in meetings is thought to be a significant contributor to error in speaker diarization, but it is not clear if overlaps are problematic for speaker clustering and/or if errors could be addressed by assigning multiple labels in overlap regions.
In this section, we look at these issues experimentally, assuming perfect detection of overlaps, to assess the relative importance of these problems and the potential impact of overlap detection. In addition, we distinguish between potential causes of errors at different stages of the diarization process and look at the role of location features such as microphone pair correlation lags.

6.1.1 Questions in Overlap Handling

As discussed in Section 2.3, analyses have shown that overlaps contribute to diarization error but the reasons for the degradation are not clear. Most current systems (including our baseline) assign only one speaker label to any region of speech. Is this alone what leads to errors, or is the overlapped speech corrupting the single-speaker models learned in the process of diarization? It may be that very high accuracy overlap detection is needed to successfully leverage overlap information, or it may be that there is little gain from such detection.

In order to better understand the potential impact of overlap detection, we factor out the effect of overlap detection errors by using an “oracle” overlap detector and evaluating alternatives for using this information in the diarization process.

In a typical diarization system, a stream of speech is initially broken into short time segments, either at detected speaker change points or at uniform intervals. Then these segments are grouped and assigned to a speaker by agglomerative clustering. Overlaps cause errors in at least two ways. First, clustering assigns segments to only one speaker during an overlap; other speakers during the overlap will be missed. Second, clusterer speaker models can be corrupted when overlapped speech is included in their training data.

In diarization experiments, we ask the following questions:

1. Would diarization be improved if overlaps were detected and removed before speaker clustering?
2. After the initial single-speaker diarization has been completed, can the assignment of two speaker labels, based on the temporal adjacency of other speakers to an overlap, lead to significant improvements in diarization scores?

To answer these questions, we use an oracle overlap detector, derived from forced alignments.

6.1.2 Experimental Paradigm

Experiments in this section were performed on NIST dev and test sets defined in Section 3.1 (all 31 meetings, since the algorithms evaluated here have no tunable parameters). Tunable parameters in the clustering system (number of mixtures, HMM states, etc.) were fixed to those found to be optimal by ICSI during the NIST 2007 evaluation. For simplicity, we have limited the location feature choice to the all-pairs correlation lag features, transformed to a fixed lower dimension using PCA.

6.1.3 Cluster quality and location feature robustness

Figure 6.1 illustrates the test used to est the effect of overlap removal cluster model quality and Table 6.1 shows the effect of overlaps on single talker speech accuracy, that is, the subset of speech where there are no overlaps. For this series of experiments, references derived from forced alignments were used to exclude overlaps from the final diarization outputs, so that the diarization error was calculated only over single talker speech. We then compared the cases where the input to the diarization clustering includes vs. excludes the overlap regions. To explore the effect of overlaps on different features, clustering experiments were conducted using: i) MFCC’s alone, and ii) MFCC’s in one observation stream of the HMM and cross-correlation features (XC) in a second stream, as in [91].

In the first two rows of Table 6.1 we see that an ideal overlap detector, which excludes overlaps from MFCC-only clustering input data, would improve single-speaker-
Table 6.1: Effect of input overlap processing on single-talker diarization accuracy

<table>
<thead>
<tr>
<th>Features</th>
<th>Overlaps in Input?</th>
<th>Diarization Error (%)</th>
<th>Improvement (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC</td>
<td>Y</td>
<td>18.5</td>
<td>*</td>
</tr>
<tr>
<td>MFCC</td>
<td>N</td>
<td>17.9</td>
<td>0.6</td>
</tr>
<tr>
<td>MFCC+XC</td>
<td>Y</td>
<td>11.9</td>
<td>*</td>
</tr>
<tr>
<td>MFCC+XC</td>
<td>N</td>
<td>11.6</td>
<td>0.3</td>
</tr>
</tbody>
</table>

only diarization performance by 0.6% absolute (3.2% relative). Adding XC features reduces the error by about 35% relative in both cases, shrinking the gap in performance.

6.1.4 Nearest-2 cluster result post-processing

Figure 6.2 illustrates the overlap post-processing tests, the results of which are shown in Table 6.2. Here, we study the effect of different overlap output processing approaches on the full diarization score, i.e. including overlap regions in the diarization score. As in the previous experiments, we use oracle overlap regions, i.e., assuming perfect detection, in this case to obtain different input and output processing conditions.
The first row of Table 6.2 gives the diarization baseline performance with MFCC features, with no special overlap handling.

The next two rows of Table 6.2 show the effect of replacing the speaker label associated with an overlap with the labels associated with the two speakers detected closest to the overlap (referred to here as “nearest-2”). Two examples of nearest-2 processing are shown in Figure 6.3. In the top example, two speakers have been detected by the clusterer (black lines, with high indicating that the clusterer has detected the speaker). An overlap detector (blue line) has detected one overlap. The nearest-2 algorithm assigns two speakers to the overlapped region, the first of which is the speaker found by the clusterer during the overlap time (speaker 1). The algorithm then searches left and right of the overlap for the nearest detected speaker different than the one already assigned to the overlap (speaker 2). During the overlap time, the nearest-2 algorithm generates a new speaker segment for this speaker of the width of the overlap (shown in red). In the bottom example of Figure 6.3, three speakers have been detected, and nearest-2 has inserted an extra overlap segment for speaker 3, the unique speaker nearest to the overlap. As shown in Table 6.2, nearest-2 yields a 2% absolute improvement (10% relative) if overlaps are not excluded at the clusterer input. If overlaps are excluded at the clustering input, a 2.5% absolute improvement results.

In the fourth row of Table 6.2, we show the result of “perfect” overlap post pro-
Figure 6.3: Nearest-2 overlap assignment
cessing, where oracle overlap regions are assigned to the true speaker labels. Here, “perfect” means “the best you can do.” If the clusterer had underestimated the number of speakers so that some overlap segments contained speakers which were not detected anywhere in the meeting, then no segment for that speaker was inserted into the overlap region. This causes a diarization error, but it matches the condition of the nearest-2 strategy, which can only select from speakers detected by the clusterer. For MFCC features, this "perfect" overlap scheme yields a 1.1% absolute improvement over the nearest-2 approach; the nearest-2 approach obtained 70% of the improvement ideally possible with input and post-processing overlap detection.

The remaining rows of Table 6.2 show that, with MFCC+XC features, the nearest-2 approach comes much closer to the ideal. Comparing the baseline performance in the fifth row with the next two, we see that the simple nearest-2 post processing step improves performance by 2.2% absolute (15% relative), regardless of whether or not overlaps were excluded at the clusterer input. From the last two rows, we see that nearest-2 approach obtains 75% of the ideal improvement.

6.1.5 Summary

In this study, we have demonstrated that a simple nearest-2 post-processing step will yield most of diarization performance improvement possible from overlap detection, given the speaker clusters detected by the diarization system under study. We show that, for MFCC+XC features, removing overlap segments from the input of the diarization clusterer yields no further improvement when cross-correlation features are included in clustering. The results suggest that one way in which cross-correlation features help diarization is to improve the overlap-robustness of single-speaker model building.
Table 6.2: Effect of input/output overlap processing on full diarization accuracy

<table>
<thead>
<tr>
<th>Features</th>
<th>Overlaps in Input?</th>
<th>Output Post-process</th>
<th>Diarization Error (%)</th>
<th>Improvement (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC</td>
<td>Y</td>
<td>none</td>
<td>21.6</td>
<td>*</td>
</tr>
<tr>
<td>MFCC</td>
<td>Y</td>
<td>nearest-2</td>
<td>19.6</td>
<td>2.0</td>
</tr>
<tr>
<td>MFCC</td>
<td>N</td>
<td>nearest-2</td>
<td>19.1</td>
<td>2.5</td>
</tr>
<tr>
<td>MFCC</td>
<td>N</td>
<td>perfect</td>
<td>18.0</td>
<td>3.6</td>
</tr>
<tr>
<td>MFCC+XC</td>
<td>Y</td>
<td>none</td>
<td>15.1</td>
<td>*</td>
</tr>
<tr>
<td>MFCC+XC</td>
<td>Y</td>
<td>nearest-2</td>
<td>12.9</td>
<td>2.2</td>
</tr>
<tr>
<td>MFCC+XC</td>
<td>N</td>
<td>nearest-2</td>
<td>12.9</td>
<td>2.2</td>
</tr>
<tr>
<td>MFCC+XC</td>
<td>N</td>
<td>perfect</td>
<td>12.2</td>
<td>2.9</td>
</tr>
<tr>
<td>MFCC+XC</td>
<td>Y</td>
<td>perfect</td>
<td>12.5</td>
<td>2.6</td>
</tr>
</tbody>
</table>

6.2 Monaural Overlap Detection

Here, we look at spectral features that are useful for monaural overlap detection. Monaural data can be either from a single microphone or from the beamformer output of a multi-microphone system. Since it is single-channel, it contains information only about how a speaker sounds; location information is lost. We will examine overlap detection with location information in Section 6.3.

First we test various features on synthesized overlaps derived from the well-known, phonetically-transcribed TIMIT corpus. It was hoped that designing classifiers in this environment would allow us to isolate properties that might give a clue to overlaps without the confounding and variable acoustic environments present in the NIST data. Synthesized overlaps also allowed us to know exactly when the overlaps occurred, which is only roughly known in the NIST data, where the best available timing information comes from imprecise forced alignments – for short overlaps (which is most
of them), forced alignment timing errors can be comparable to total overlap width. After proposing several new pitch based features, we evaluate them on TIMIT, and then move on to real meeting data.

6.2.1 Pitch Detection and Hough Transform

Cepstral pitch detectors find the fundamental period of speech ("pitch," or \( T_0 \)) by searching for a peak in a single cepstrum. In the three cepstral images of [6,4], the vertical direction is period and the horizontal direction is time. The top two images are single talker speech, while the bottom image is the cepstrum that results when the waveforms of the two speakers are added.

A typical cepstral pitch tracker, which searches vertically at each time instant, for the highest cepstral value, would frequently make erroneous \( T_0 \) estimates when noise or natural speech variability cause this peak to occur at a location other than the true pitch period. However, since true pitch is generally continuous, the correct peak periods over a short time window of cepstral vectors lie roughly along a straight line; if detected pitch is forced to fit a straight line, \( T_0 \) detection is improved. Hough transform \( T_0 \) continuity enforcement has substantially increased pitch detection accuracy in several types of noise [61].

The Hough transform [60] is most commonly used to search an image for straight lines of arbitrary slope and intercept. Along a large number of lines, \( l \), each with slope, \( m_l \), and intercept, \( b_l \), the average pixel brightness is calculated and the result is stored in a “Hough image.” In the Hough image, the \( x \) and \( y \) coordinates corresponding to line \( l \) are the indices of the quantized \( m_l \) and \( b_l \), and the brightness at the point \((x_l,y_l)\) is the average input image brightness along that line; the most pronounced line in the input image is the one with the brightest pixel in the Hough image. In Hough transform pitch detection, the image is a time window of cepstral vectors; the brightest spot in the Hough image generally corresponds to the line of the true pitch track.
Figure 6.4: Overlapped Speech Cepstrum
The starting plan for this study was to adapt a Hough transform pitch detector to the problem of overlap detection, the thought being that, if a single bright spot in the Hough image indicated one pitch, then two bright spots would indicate two pitches, i.e. a speaker overlap. Unfortunately, it was difficult to distinguish between reduced peakiness due to harmonics of a single pitch and reduced peakiness due to an actual second pitch.

Therefore, we explore how to improve the usefulness of Hough image entropy with filtering designed to sharpen the cepstra. Since other overlap detection features will require knowledge of pitch, the filters are also intended to improve Hough pitch estimation.

**Hough Transform Input Processing: Smoothing and Spectral Fit**

Frequently, the input to the Hough transform is preprocessed to enhance line detectability, although in [61], pitch detection was improved even without pre-processing. The standard Hough preprocessing steps – edge detection and morphological operations – are inappropriate for pitch detection because the cepstral image was only nine pixels wide, too narrow for morphological operations; because cepstral noise is more severe than in typical Hough input images, making high pass filtering undesirable; and finally, because the cepstral pixel amplitude is strongly related to true pitch, while the traditional Hough transform input preprocessing algorithms deemphasize pixel amplitude in favor of pixel amplitude gradient.

Therefore, two application-specific cepstral filtering steps were applied. The first dealt with the horizontal time smearing present in the cepstral images due to a long FFT analysis window. These were erroneously detected as pitch lines by the Hough transform. To reduce this problem, the cepstral image was smoothed by convolution with a disc-shaped, 2-D linear averaging filter. Cepstral images have much higher dynamic range than typical images so the averaging did not significantly reduce resolution near a cepstral peak. At the same time, when two cepstral line artifacts were
near each other, the gap between was effectively filled. A disc radius of 4 pixels was found to be optimal on cepstral images derived from 1024 point FFT’s, calculated with a window skip of 32 waveform samples.

The second cepstral filter was intended to minimize false pitch harmonics in single-talker speech. The idea was that the idealized spectrum corresponding to an erroneous pitch estimate should be a poor match to the measured spectrum. If cepstral coefficients are multiplied with some measure of the match (high for a good match, low for a bad match), then the cepstral image should be more suitable for Hough single talker pitch detection. The problem with such a pre-processing approach is that can potentially eliminate the pitch of a second talker, reducing the value of the Hough image entropy. The benefit is better pitch detection which can be used for independent spectral mismatch features covered in the next section.

The procedure for calculating the idealized spectrum is as follows. First, measured spectra are calculated with a Hanning windowed, short term Fourier transform of a monaural waveform. Let \( \{ X_1, X_2 \ldots X_{N_F} \} \) denote the short term Fourier transform magnitudes at the \( N_F \) positive frequency points between 250 and 3750 Hz. Subsequent processing operates on single spectra so, for notational simplicity, the time index for the current spectral magnitude, \( X \), is omitted.

Next, a single-talker spectrum is synthesized from the Hough transform pitch estimate, \( F_0 = 1/T_0 \), and a spectral envelope, \( E \), estimated by LPC analysis. The vector of idealized pitch excitation harmonics, \( P_{F_0} \), are generated:

\[
P_{F_0} = \text{comb}(f, F_0) \ast W(f)
\]

where \( \ast \) is discrete convolution over frequency, \( \text{comb}(f, F_0) = \sum_k \delta(f - kF_0) \), and where \( W(f) \) is the Hanning window frequency response. The \( F_0 \) estimate is quantized: \( F_0 = \frac{1}{nT_s} \), where \( n \) is an integer and where \( T_s \) is the waveform sample period. By definition, the \( F_0 \) estimate is incorrect for unvoiced frames, but no attempt is made to remove the estimates based on some voicing indicator.
Finally, the voiced spectrum corresponding to $F_0$ and $E$ is synthesized:

$$S_{F_0} = GP_{F_0} \cdot E \quad (6.2)$$

where $G$ is set to yield the least mean squared difference between the $S_{F_0}$ and the measured spectrum, $X$.

Given $S_{F_0}$ and the measured spectrum, $X$, the normalized mean squared error is calculated:

$$nse_{F_0} = \frac{\sum_{i=1}^{N_F} (X_i - S_{F_0}^i)^2}{\left(\sum_{i=1}^{N_F} X_i^2 \sum_{j=1}^{N_F} (S_{F_0}^j)^2\right)^{1/2}} \quad (6.3)$$

where the summations occur over $N_F$ discrete frequencies, and where $S_{F_0}^i$ is the synthesized spectral magnitude at the $i$th discrete frequency.

Since $nse_{F_0}$ is small for correct pitch estimates and large for erroneous pitch estimates, a search for its minima over a set of $K$ possible pitches, $F_0^k$, yields another type of $F_0$ estimate which is somewhat independent of a cepstral peak pitch estimate. Instead of using it to make a separate pitch estimate, $nse_{F_0^k}$ was used to enhance the Hough transform: before the Hough transform, the cepstral coefficient corresponding to $F_0^k$ was multiplied by $1/nse_{F_0^k}$ for $k = 1 \ldots N_F$. This had the effect of attenuating harmonic peaks and sharpening the Hough transform. The final pitch estimate used in subsequent processing was then determined at the Hough transform output, as in [61].

As shown in Table 6.3, both preprocessing steps improved pitch detection accuracy. A zero in one of the first two columns indicates that the pre-processing step was not present. A pitch error was declared whenever the pitch estimate deviated from the hand labeled truth by more than 5%. The test set, which was not corrupted by noise, consisted of the 50 female and 50 male Japanese digit utterances used in [61] to test pitch detection accuracy. The sample rate was 20KHz.
Table 6.3: Hough Transform Pitch Detection Accuracy vs. Input Preprocessing

<table>
<thead>
<tr>
<th>Spectral Fit</th>
<th>Cepstral Image Disc Filter</th>
<th>% Pitch Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>11.0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>10.0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>9.9</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>9.3</td>
</tr>
</tbody>
</table>

6.2.2 Voiced Speech Spectral Overlap Detection Features

In this section, we discuss overlap detection features designed to detect the presence to two pitches (an overlap).

Hough Image Entropy

It was conjectured that single talker Hough images might be more “peaky” than overlapped speech Hough images. To that end, a sort of Hough image entropy was calculated:

$$H(I) = -\sum_{i=1}^{N_p} p(I_i) \log p(I_i)$$  \hspace{1cm} (6.4)

where $N_p$ is the number of pixels in the Hough image, $I$, and $p(I_i) = \frac{I_i}{\sum_{j=1}^{N_p} I_j}$ is a normalized pixel magnitude instead of the usual probability. The entropy is large for a uniform image and close to zero for a very peaky image so it was hoped that the entropy for two-pitched speech would be lower than that of single-pitched speech.

Synthesized Spectral Match

If a spectrum synthesized for spectral envelope and a pitch estimate matches that actually measured, it is likely that the speech came from a single talker. If the match
is poor, the measured spectrum may have come from either unvoiced or overlapped speech.

Given \( S_{F_0} \) and \( X \) (see previous section) we calculated several metrics indicating the quality of their match. The first was the correlation coefficient, \( \rho_{XS_{F_0}} \), between the synthesized and measured spectrum. In general, the correlation coefficient is defined as:

\[
\rho_{XY} = \frac{\sigma_{XY}}{(\sigma_{XX}\sigma_{YY})^{1/2}} \tag{6.5}
\]

where:

\[
\sigma_{XY} = \frac{1}{M-1} \sum_{m=1}^{N_F} (X_m - \mu_X) (Y_m - \mu_Y),
\]

\( \mu \) is the sample mean, and \( X \) and \( Y \) are \( N_F \) dimensional vectors.

**Harmonic Screen**

Another spectral match feature attempted to measure the amount of spectral power within the harmonic screen corresponding to \( F_0 \):

\[
SP_{F_0} = \frac{\sum_{i=1}^{N_F} X_i X_i}{\sum_{i=1}^{N_F} X_i} \sum_{j=1}^{N_F} P_{F_0}^j \tag{6.6}
\]

where \( P_{F_0}^i \) and \( X_i \) are the \( i \)th frequency bins of the discrete Fourier transforms of the idealized and measured spectra. This feature was again vulnerable to pitch doubling and halving, and for similar reasons as for \( \rho_{XS_{F_0}} \). However, since the errors were not exactly the same, the measure was retained as an input to the overlap classifier.

**Normalized Squared Synthesis Error**

The final spectral match feature was:

\[
nse_{F_0} = \min_k nse_{F_0}^k
\]
Based on inspection of one male and one female utterance, this quantity was most sensitive to pitch doubling than $\rho_{XS_{F0}}$ or $SP_{F0}$, that is, it produced a high error for erroneous pitch estimates during single talker speech. While this property is an advantage for single talker pitch estimation, it may make $NSE_{F0}$ less useful for detecting overlap, since $NSE_{F0}$ for single talker pitch estimation errors can be as high as $NSE_{F0}$ during overlapped speech. However, $NSE_{F0}$ was included as a classifier feature because its values were not perfectly correlated with $\rho_{XS_{F0}}$ or $SP_{F0}$, and therefore may have provided an overlap detector classifier with unique overlap information.

**Spectral Flatness Measure**

Since all of the spectral match features assume a voiced spectrum, they are inappropriate for unvoiced speech. For this reason, an overlap detector classifier given these features classifier must be given some indication of voicing. A commonly used indicator of voicing is the spectral flatness measure (SFM), the ratio between the geometric and arithmetic means of the speech power spectrum [126]:

$$SF M = \frac{(\prod_{i=1}^{N_F} X_i)^{1/N_F}}{1/N_F \sum_{i=1}^{N_F} X_i}$$  \hfill (6.7)

In other work [54, 127, 125], SFM was thresholded to generate a binary voicing decision but here, it was fed into the classifier as a continuously valued input.

**Peak-to-Valley Ratio Spectral Flatness Measure**

For low $F0$, the harmonic peaks were crowded closely together, smearing out the valleys and making the $SF M$ numerator high relative to the denominator. Therefore, a second measure of spectral flatness was designed to accentuate the valleys and peaks.

First, peaks and valleys were located: The spectrum was flattened by dividing $X$ by the LPC envelope estimate. Then the general trend was extracted by filtering the spectrum with a median filter of width equal to the period of the $F0$ estimate.
Contiguous regions above and below the median line were considered “peaks” and “valleys,” respectively. The spectral flatness measure, $SFM_{pvr}$, was then the power-weighted peak-power-to-valley-power ratio:

$$SFM_{pvr} = \frac{1}{N_{pks}} \sum_{p=1}^{N_{pks}} \frac{\sum_{k \in \text{peaks}(p)} X_k^2}{\sum_{k \in \text{valleys}(p)} X_k^2} V_p$$  \hspace{1cm} (6.8)

where $N_{pks}$ peak-valley pairs were found and where the points in the $p$th pair are denoted by $\text{peaks}(p)$ and $\text{valleys}(p)$. $V_p$ is the normalized power weighting for the $p$th peak/valley pair:

$$V_p = \frac{\sum_{k \in \text{peaks}(p) \cup \text{valleys}(p)} X_k^2}{\sum_{i=1}^{N_F} X_i^2}$$  \hspace{1cm} (6.9)

$SFM_{pvr}$ was developed on the 20KHz sampled, hand labeled pitch data used in to evaluate $F0$ extraction in [61]. On this data, it was a substantially better indicator of voicing than $SFM$. However, on the 16KHz sampled TIMIT data, it was not clearly better.\footnote{Since hand labeled pitch was not available for TIMIT, the Xwaves \texttt{get\_f0} program \cite{110} voicing output (with default parameters) was used as a reference.} Nevertheless, the areas where it performed poorly were sometimes different than those for $SFM$. Therefore, it was retained as a continuously valued classifier input.

\textbf{Envelope Correlation}

When speech is voiced, the pitch pulses affect all frequencies, and because of vocal tract filtering and non-stationarity, speech from only one talker may dominate the spectrum at certain frequencies while at the same time, speech from a second talker may dominate at others. The Hilbert envelope correlation feature attempts to detect these cases. To accentuate the pitch pulses, the time domain waveform is inverse-filtered to generate the LPC residual. The residual is then filtered by a bank of $M = 20$ FIR bandpass filters, each with 100 taps. The bandpass filter center frequencies are
linearly spaced, ranging from DC to Nyquist, and they have bandwidths such that the filters’ pass-bands overlap by 50%. Each of the bandpass filter’s outputs are then transformed \[106\] to obtain phase, $\phi_m$, and envelope, $A_m$, components:

$$Z_m = A_m \cos \phi_m$$  \hspace{1cm} (6.10)

where $Z_m$ is a vector of outputs from the $m$th filterbank, spanning a sliding analysis window of length 1024 points. For a single pitch, the envelope coefficients are highly correlated across filter banks; for two pitches they are generally less correlated. An example is shown Figure 6.5., where the horizontal axis is time, and the waveforms are the Hilbert envelopes of the filterbank outputs. In this case, the pitch pulses of one speaker dominate the lower frequencies, while another speaker dominates frequencies near the middle.

To summarize pitch dominance, the average envelope correlation coefficient is calculated:

$$CC_{env}(Z) = \frac{2}{M^2 - M} \sum_{m=1}^{M-1} \sum_{n=m+1}^{M} \rho_{Z_mZ_n}$$  \hspace{1cm} (6.11)

This number tends to high for voiced, single talker speech, and low for unvoiced speech, or during overlaps when two talkers dominate different parts of the spectrum.

### 6.2.3 Classifier Design

Three GMM’s – one each for single talker speech, overlapped speech, and non-speech – were trained on the training set. Covariances were diagonal, and 30 mixtures was chosen, as more mixtures did not improve performance. As a baseline, overlap decisions were then made individually for each frame:

$$\hat{\theta}_t = \arg\max_i \log p(O_t | \theta_i)$$  \hspace{1cm} (6.12)
Figure 6.5: Spectral envelopes during overlap
where \( O_t \) is the feature vector at frame \( t \) and where \( \log p(O_t|\theta_i) \) is the log likelihood of \( O_t \) according to the \( i \)th GMM with parameters \( \hat{\theta}_i \in (\theta_0, \theta_1, \theta_2) \) for non-speech, single-talker speech, and overlapped speech, respectively. The rule does not make use of prior probabilities because the proportion of speech types in a conversation is generally unknown.

The MFCC features were generated at the typical sample rate of 100Hz. The Hough features were calculated at 625 Hz (the same as in [61]) because tests performed during feature design showed that Hough transform pitch tracking broke down for longer time windows. To match the two rates, the Hough features were decimated by averaging. It is possible that a better scheme would have been to generate MFCC’s at 625Hz but this option was not explored.

In an attempt to exploit temporal dependence, the log likelihoods calculated by the GMM’s built above were combined with the GLM softmax function,

\[
r(Y_t)_i = \frac{e^{Y_t w_i + b_i}}{\sum_{k=1}^{3} e^{Y_t w_k + b_k}}
\]

where \( r(Y_t)_i \) is the GLM output for the \( i \)th speech type, and where the row vector \( Y_t \) is the \( 1 \times 2(L + 1) \) concatenation of GMM log likelihood ratio pairs within a neighborhood of \( \pm L \) frames of the current frame, \( t \):

\[
Y_t = [\gamma_{t-L}, \gamma_{t-L+1}, \ldots \gamma_{t+L}]
\]

where \( \gamma_{t'} \) is the \( 1 \times 2 \) row vector,

\[
\gamma_{t'} = -\log \left[ \frac{p(O_{t'}|\theta_2)}{p(O_{t'}|\theta_1)}, \frac{p(O_{t'}|\theta_0)}{p(O_{t'}|\theta_1)} \right]
\]

and where the column vector \( w_i \) and scalar \( b_i \) for the \( i \)th speech type are obtained using iterative re-weighted least squares training [58]. The speech type decision was then \( \hat{\theta}_t = \arg \max_i r(Y_t)_i \). The neighborhood size was chosen as \( L = 7 \), corresponding to 150ms, a value found to be optimal for meeting speech detection in another study using the GLM structure [90]. The 150ms neighborhood is significantly shorter than a typical 250ms overlap.
6.2.4 Monaural overlap detection on synthetic TIMIT data

Overlapped speech was synthesized from the 16KHz TIMIT \cite{43} speech corpus. Selected utterance pairs were time aligned such that 50% of the first utterance was overlapped with the second and then the waveforms were added. Utterance selection was random so that sexes, regions and dialects were roughly equally represented. For testing and training truths, the entire region where the utterances overlapped in time was defined as overlapped. This introduced a small amount of noise in the inference labels because, when a talker paused briefly during this time, there was truly only one speaker. Most TIMIT utterances have very little pausing so the noise was considered tolerable. There were 400 overlap pairs synthesized from the TIMIT test set and 200 from the train set.

Classifiers were first trained and tested on the pitch-dependent features of Section 6.2.1., collectively called the “Hough features.” Since these features require at least one voiced speaker for accurate overlap detection, MFCC’s, which operate only on the spectral envelope, were also included. MFCC’s were either appended to the Hough feature vector or tested separately, making it apparent which features contribute most to correct overlap detection.

Overlap detection results are reported in terms of frames missed and detected, and in terms of the composite $F$:

$$F = \frac{2PR}{P + R} = \left(\frac{1}{2} \left( \frac{1}{P} + \frac{1}{R} \right) \right)^{-1} \quad (6.16)$$

where $P$ is the precision and $R$ is the recall.

Combined Feature Performance

A set of GMM’s were trained on MFCC’s alone ($\text{GMM-MFCC}$); $O_t$ in \eqref{6.14} was the 22 MFCC coefficients for each frame. Another set of GMM’s ($\text{GMM-Hough}$) were trained on the Hough features; $O_t$ was the 7-D vector obtained by concatenating
Finally, a third set of GMM’s were trained on the combined MFCC and Hough features (MFCC-both), where the MFCC and Hough features were concatenated into a single vector. As can be seen in the table, the MFCC and the Hough features are about equally successful at detecting overlaps but are more effective when they are combined.

<table>
<thead>
<tr>
<th>Model</th>
<th>% Missed</th>
<th>% False</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMM-MFCC</td>
<td>34.7</td>
<td>51.4</td>
<td>0.56</td>
</tr>
<tr>
<td>GMM-Hough</td>
<td>34.2</td>
<td>53.2</td>
<td>0.55</td>
</tr>
<tr>
<td>GMM-both</td>
<td>27.1</td>
<td>48.5</td>
<td>0.60</td>
</tr>
<tr>
<td>window-GLM-MFCC</td>
<td>38.7</td>
<td>20.2</td>
<td>0.69</td>
</tr>
<tr>
<td>window-GLM-Hough</td>
<td>29.1</td>
<td>46.7</td>
<td>0.61</td>
</tr>
<tr>
<td>window-GLM-both</td>
<td>26.6</td>
<td>13.6</td>
<td>0.79</td>
</tr>
</tbody>
</table>

GLM’s were trained for each feature combination and the results are shown in Table 6.4. The MFCC features by themselves (window-GLM-MFCC) benefit much more from the GLM decision than do the Hough features by themselves (window-GLM-Hough). When the features are combined, the GLM structure yields a substantial improvement (window-GLM-both), bringing the F measure from 0.60 (GMM-both) to F=0.79.

Figure 6.6 shows how combining the MFCC and pitch-based features improves performance of the window-GLM classifier. The nonSpeech bar shows that the classifier has no problem detecting non-speech (during non-speech, the classifiers almost never falsely detect any kind of speech).

The bars below nonSpeech show the percentage error for five speech conditions. During non-overlapped speech, a single talker may be voiced or unvoiced: the Voice1 bar indicates that all features are good at detecting single speaker cases when the speech is voiced (non-speech over overlapped speech was rarely detected). The fea-
The remaining bars show detector accuracy during overlap. Most errors are due to overlapped frames being falsely classified as single talker speech. But it is here where combining pitch and MFCC features results in a better overall classifier. As expected, when one or both of the overlapping talkers are unvoiced (\textit{oneVoiceOv} and \textit{unVoiceOv}, respectively), the MFCC features are superior to the pitch features. When both speakers are voiced, however, the pitch based features are better than the MFCC features. In all cases, the combination of pitch and MFCC features still results in a classifier that is as good or better than either feature set alone.
Hough Feature Selection

The Hough features were all intended to help in detecting voiced overlaps so it was possible that they were redundant. If so, performance may in fact improve by eliminating unreliable or redundant features to reduce the problem of overfitting.

In order to identify the most useful feature combination, a simple leave-one-out study was performed. Eleven versions of the \texttt{window-GLM-both} classifier were trained and tested on feature vectors with Hough feature or Hough transform input pre-processing steps removed. The results are shown in Table 6.5 where the first row shows the original, complete Hough feature \texttt{window-GLM-both} performance, for reference. Rows 2 and 3 show the result of removing one of the image processing steps; rows 4-10 show the result when one feature at a time is removed; and rows 11-12 show the performance obtained when combinations thereof are removed.

Table 6.5: Overlap Detection Performance vs. Left-Out Pre-processing or Feature

<table>
<thead>
<tr>
<th>Feature and/or Pre-Processing Removed</th>
<th>% Missed</th>
<th>% False</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>None Removed (complete \texttt{window-GLM-both})</td>
<td>26.6</td>
<td>13.6</td>
<td>0.79</td>
</tr>
<tr>
<td>cepstral image disk filter</td>
<td>28.3</td>
<td>14.3</td>
<td>0.78</td>
</tr>
<tr>
<td>$1/nse_{F_0}$ pre-multiply</td>
<td>26.6</td>
<td>13.4</td>
<td>0.80</td>
</tr>
<tr>
<td>$H$</td>
<td>29.4</td>
<td>14.8</td>
<td>0.77</td>
</tr>
<tr>
<td>$\rho x s_{F_0}$</td>
<td>25.3</td>
<td>12.3</td>
<td>0.81</td>
</tr>
<tr>
<td>$SP_{F_0}$</td>
<td>26.8</td>
<td>13.5</td>
<td>0.79</td>
</tr>
<tr>
<td>$nse_{F_0}$</td>
<td>27.3</td>
<td>13.2</td>
<td>0.79</td>
</tr>
<tr>
<td>$SFM$</td>
<td>27.6</td>
<td>15.3</td>
<td>0.78</td>
</tr>
<tr>
<td>$SFM_{pvr}$</td>
<td>28.6</td>
<td>28.6</td>
<td>0.77</td>
</tr>
<tr>
<td>$CC_{env}$</td>
<td>29.7</td>
<td>14.4</td>
<td>0.77</td>
</tr>
<tr>
<td>$\rho x s_{F_0}$, $1/nse_{F_0}$ pre-multiply, $SP_{F_0}$, $nse_{F_0}$</td>
<td>26.8</td>
<td>14.0</td>
<td>0.79</td>
</tr>
</tbody>
</table>
Two of the largest performance drops occur when the \( H \) and \( CCenv \) features are removed, indicating their usefulness. The voicing features \( SFMpv\) and \( SFM \) also seem important, with \( SFMpv\) being the most useful.

Performance improves when \( \rho_{xS_{F_0}} \) or the \( 1/nse_{F_0} \) pre-multiply are removed. That \( \rho_{xS_{F_0}} \) removal helps is perhaps not surprising, as in separate tests, it frequently indicated a close spectral match when pitch estimates were erroneously doubled. However, the slight performance increase due to \( 1/nse_{F_0} \) pre-multiply removal is surprising, given its effectiveness at improving pitch detection accuracy. When we eliminate all features whose individual removal either improves or does not degrade performance, we see that performance is the same as with the complete window–GLM–both features (last line of Table 6.5). Clearly, some features are either redundant or noisy.

Conclusions from TIMIT synthetic overlap data

Several new pitch-based overlap detection features were proposed, which performed roughly as well as MFCC features in detecting the synthesized overlaps used to study this monaural classifier design. As would be expected, the pitch features were better able to detect overlaps when one or both speakers were voiced, while MFCC features were better for unvoiced speech. When both features were used, overall performance was better than with either feature set by itself. Regardless of the feature type, the window–GLM classifier structure significantly improved results over the GMM structure. The end result was a fairly successful overlap detector capable of acting on the very short overlaps typical of conversational speech. On a subset of the TIMIT corpus, it detected 74.7% of overlapped frames with a false detect rate of 12.3% (F=0.81).

6.2.5 Monaural overlap detection on meeting data

In this section, we apply the monaural overlap features in Sections 6.2.1–6.2.4 to real meeting data (given the results in Table 6.4, the \( \rho_{xS_{F_0}} \) feature was removed). Features were calculated from the waveform at the output of the baseline system’s
delay-sum beamformer and classification was done with 30 mixture GMM’s, as in equation (6.12).

As shown in Table 6.6, monaural features perform much worse on real overlap data than they do on TIMIT. The first two rows show results for GMM’s trained on MFCC’s and Hough features alone, the last shows results when the two feature vectors are concatenated. Numerous differences between the two data sets could explain the degradation:

- meeting data is much noisier than TIMIT, and is subject to reverberation;
- the Wiener filter noise-reduction step, which is designed to enhance single talker speech, may have obliterated overlap cues;
- the overlapped speakers will not have a consistent power ratio, like they do on TIMIT; and
- the spontaneous overlaps in meeting data may have different properties than synthesized TIMIT overlaps.

Table 6.6: Monaural Overlap Detection Performance on Meeting Data (GMM classifiers)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Dev precision</th>
<th>recall</th>
<th>F</th>
<th>Test precision</th>
<th>recall</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC</td>
<td>0.07</td>
<td>0.53</td>
<td>0.12</td>
<td>0.01</td>
<td>0.54</td>
<td>0.03</td>
</tr>
<tr>
<td>Hough</td>
<td>0.04</td>
<td>0.29</td>
<td>0.06</td>
<td>0.01</td>
<td>0.33</td>
<td>0.02</td>
</tr>
<tr>
<td>MFCC+Hough</td>
<td>0.06</td>
<td>0.53</td>
<td>0.11</td>
<td>0.01</td>
<td>0.51</td>
<td>0.02</td>
</tr>
</tbody>
</table>

These results are also much worse than those reported on meeting data in [20]. Besides the fact that we are using a different classifier and features, the discrepancy
can possibly be explained by the fact that we report results on a large test set with tuning done on a separate training set. Other differences were that we used meeting data from several corpora, rather than just the AMI meetings used in [20], and that we derived monaural features from a noise-reduced delay-sum beamformer output, rather than from a single channel. Finally, the results in Table 6.6 are for single-frame GMM classifiers, where other results are for classifiers based on HMM’s, which impose temporal smoothing on the detection output.

In any case, on our data, Hough features are not useful for overlap detection. Therefore, we moved on without trying to use them in experiments with location features.

6.3 Overlap Detection Using Location Information

As has been shown thus far, location features can significantly improve speaker diarization in non-overlapped regions. Just as location features turned out to be important indicators for speaker ID, we believe that certain location features can be a clue to the presence of overlap. In this section, we propose several location based overlap detectors, and test them on multi-microphone meeting data.

6.3.1 Speaker pair model fit

If two speakers are talking, then it is reasonable to expect that two speaker models will fit the data better than one. As the speakers go from voiced to fricative speech, or pause between syllables, the relative amplitude between them changes, and during certain time intervals, one speaker or the other will dominate. During the time when one speaker dominates, there will be a better match with the associated speaker model built during the diarization process (assuming accurate diarization). The approach tested here attempts to detect speaker overlap by examining pairs of model posteriors. Since speaker location is more stationary than speaker sound, we might expect location feature models to provide better detection of brief overlaps than detection
based only on MFCC speaker models.

For this work, we assume that an overlap consists of two speakers. One of them is the speaker identified by the clustering system, with model parameters \( \theta_C \); the other is the nearest-2 speaker, as defined in Section 6.1.4., with parameters \( \theta_N \). Assuming these two speakers, a joint speaker model, \( \theta_{C\cup N} = \{ \mu_{C\cup N}, \Sigma_{C\cup N}, c_{C\cup N} \} \), is built by merging the set of mixtures of the original model GMM’s:

\[
\begin{align*}
  \mu_{C\cup N} &= \{ \mu_C^0, \mu_C^1, \ldots, \mu_{M_C-1}^C, \mu_N^0, \mu_N^1, \ldots, \mu_{M_N-1}^N \} = \{ \mu_0^{C\cup N}, \mu_1^{C\cup N}, \ldots, \mu_{L_{C\cup N}}^{C\cup N} \} \\
  \Sigma_{C\cup N} &= \{ \Sigma_C^0, \Sigma_C^1, \ldots, \Sigma_{M_C-1}^C, \Sigma_N^0, \Sigma_N^1, \ldots, \Sigma_{M_N-1}^N \} = \{ \Sigma_0^{C\cup N}, \Sigma_1^{C\cup N}, \ldots, \Sigma_{L_{C\cup N}}^{C\cup N} \} \\
  c_{C\cup N} &= \{ \beta_c \{ c_0^C, c_1^C, \ldots, c_{M_C-1}^C \}, \beta_N \{ c_0^N, c_1^N, \ldots, c_{M_N-1}^N \} \} = \{ c_0^{C\cup N}, c_1^{C\cup N}, \ldots, c_{L_{C\cup N}}^{C\cup N} \}
\end{align*}
\]

where \( \mu_k^j \), \( \Sigma_k^j \), and \( c_k^j \), are the \( M_j \) mixture components for speaker model \( j \), and where:

\[
\beta_j = \frac{M_j}{M_c + M_N}
\]

is a multiplicative factor so that \( \sum c_{m}^{C\cup N} = 1 \). For the \( n \)th feature frame, \( o_n \), data likelihoods given parameters \( \theta_j \) are computed over a neighborhood of frames, \( w_n \):

\[
\bar{L}(o_n|\theta_j) = \log \left[ \prod_{n' \in w_n} \sum_{m=1}^{L_j} c_{m}^j N(o_{n'}; \mu_m^j, \Sigma_m^j) \right]^{1/|w_n|} \tag{6.17}
\]

where \( |w_n| \) is the number of frames in the window, and where \( L_j \) is the number of mixtures in either the single or joint model, as specified by \( j \). The window, \( w_n \), is designed to smooth the likelihoods while excluding non-speech frames (which can severely corrupt speech or location models). First, frames are selected within a 0.31s region centered around frame \( n \). Then, frames are removed which are detected as non-speech by the diarization system voice activity detector. Speech segments generated by this continuity-enforcing algorithm often contain a significant amount of non-speech. Therefore, an additional GMM-based voice activity detector was designed to eliminate single non-speech frames from \( w_n \).

Two overlap detection features are generated with different normalizations of the merged model likelihood. The first is the log likelihood ratio of the merged and
detected speaker models:

\[ R_{CN}(o_n) = \overline{L}(o_n | \theta_{C\cup N}) - \overline{L}(o_n | \theta_C) \]

It is sometimes the case that the merged model likelihood calculation is dominated by a single mixture component, making \( \overline{L}(o_n | \theta_{C\cup N}) \approx \overline{L}(o_n | \theta_C) \) even when the window contains two speakers. Therefore, a second feature is calculated:

\[ R_{CM}(o_n) = \overline{L}(o_n | \theta_{C\cup N}) - \max(\overline{L}(o_n | \theta_C), \overline{L}(o_n | \theta_N)) \]

Note that \( \overline{L}(o_n | \theta_C) \) is not guaranteed to be greater than \( \overline{L}(o_n | \theta_N) \) even though it is based on the clusterer’s maximum likelihood speaker model. This is because the clusterer made a total likelihood decision over a window estimated to be a single speaker segment, rather than \( w_n \). In addition, the clusterer’s window is more likely to contain non-speech, as it was not filtered by the frame-wise VAD applied to \( w_n \).

Speech sound and speaker location was fused by concatenating model fit parameters derived from MFCC’s and XC features:

\[ R_F = [R_{CN}(MFCC), R_{CM}(MFCC), R_{CN}(XC), R_{CM}(XC), R_{CN}(MFCC) + R_{CN}(XC)] \]

The XC features and speaker models came from Exp. 8 of Table 4.2. CCA(ER, XC) XC features were chosen because, in pilot studies, they produced better overlap detection results than CCA(XC, MFCC) features. This makes sense because the CCA(XC, MFCC) features would have been transformed closer to the MFCC space, possibly eliminating unique overlap information not present in MFCC’s.

### 6.3.2 Delay Triplet Sums

Assume that a sound plane wave hits three microphones, \( i \), \( j \), and \( m \). If there is only one plane wave (one speaker), then the time of arrival differences between the microphones must sum to zero:

\[ \tau_{i,j} + \tau_{j,m} + \tau_{m,i} = 0 \]
where $\tau_{i,j}$ is the delay between microphones $i$ and $j$, etc. In [98], this property was used as a time delay estimation confidence factor; here, we will use it as an overlap detector feature. Since, in some NIST meetings, there are 560 microphone triplets, it is desirable to first select a subset of them. The approach used here was to select individual microphone pairs by information-theoretic feature selection until enough have been selected to build the required number of triplets.

In Section 5.2, mRMR using diarization system class labels was found to be a useful feature selection method. However, because high quality delay estimates also have high mutual information, the redundancy eliminating properties of mRMR were undesirable. Therefore, microphone pairs were sorted purely by mutual information with the clusterer class labels. Pair selection started with those with the highest mutual information, working down the list of pairs until there were enough to compose three delay triplets or until no more triplets were available for a meeting. The final feature was:

$$\Delta_r = \frac{1}{|\text{triplets}|} \sum_{i,j,m \in \text{triplets}} |\tau_{i,j} + \tau_{j,m} + \tau_{m,i}|$$

where $|\text{triplets}|$ is the number of triplets selected. We averaged over triplets so that we could use information from more than one triplet yet could train a classifier on a feature with a fixed dimension across meetings with different numbers of microphones.

6.3.3 Results

The results over meetings with three or more channels are shown in Table 6.7, where feature vector concatenation is noted with a ’+’ in the feature column. Overlap detection was done by training 40 mixture, diagonal covariance matrix GMM’s on various features and then doing a maximum likelihood classification. Inputs were normalized by subtracting the mean value during speech (for each meeting, according to the diarization voice activity detector) and dividing by the speech variance. The delay triplet calculation requires at least three microphones; two dev and two test set
meetings were removed from the corpora because they had only two channels.

The first thing to note is that overlap detection with monaural MFCC features works much better on this set of meetings than with the dev and test sets shown in Table 6.6. The discrepancy could be due to the fact that it is possible to do better beamforming on meetings with more than 2 channels, thus yielding MFCC’s that are better for overlap detection. In support of this explanation are the baseline system diarization results with only MFCC features: the average diarization error on dev and test set meetings with more than 2 channels is 20.17, while the average for the four eliminated 2-channel meetings is 28.22. It is also likely that better location feature models are possible with more than 2 channels.

In any case, on the meetings with more than 2 channels, $R_F$ features are comparable to MFCC’s for overlap detection. When the two are concatenated, performance is better than with either alone. $\Delta_r$ features by themselves are not useful for overlap detection, and when combined with $R_F$ and $\Delta_r$ features, they provide only an insignificant improvement on the test set. The full $R_F+\Delta_r+\text{MFCC}$ feature vector yields a 16% relative F score improvement over the MFCC baseline.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Train</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>precision</td>
<td>recall</td>
<td>F</td>
<td>precision</td>
<td>recall</td>
<td>F</td>
<td></td>
</tr>
<tr>
<td>MFCC</td>
<td>0.08</td>
<td>0.56</td>
<td>0.15</td>
<td>0.22</td>
<td>0.55</td>
<td>0.32</td>
<td></td>
</tr>
<tr>
<td>$R_F$</td>
<td>0.07</td>
<td>0.52</td>
<td>0.12</td>
<td>0.24</td>
<td>0.49</td>
<td>0.32</td>
<td></td>
</tr>
<tr>
<td>$R_F+\text{MFCC}$</td>
<td>0.09</td>
<td>0.57</td>
<td>0.15</td>
<td>0.27</td>
<td>0.55</td>
<td>0.36</td>
<td></td>
</tr>
<tr>
<td>$\Delta_r$</td>
<td>0.04</td>
<td>0.06</td>
<td>0.05</td>
<td>0.13</td>
<td>0.06</td>
<td>0.08</td>
<td></td>
</tr>
<tr>
<td>$\Delta_r+\text{MFCC}$</td>
<td>0.07</td>
<td>0.57</td>
<td>0.13</td>
<td>0.20</td>
<td>0.51</td>
<td>0.29</td>
<td></td>
</tr>
<tr>
<td>$R_F+\Delta_r+\text{MFCC}$</td>
<td>0.09</td>
<td>0.56</td>
<td>0.15</td>
<td>0.28</td>
<td>0.56</td>
<td>0.37</td>
<td></td>
</tr>
</tbody>
</table>
In Table 6.8., results are separately shown for overlap detectors trained on the $R_F$ features that are derived purely from location information and MFCC’s:

$$R_F^L = [R_{CN}(XC), R_{CM}(XC)]$$

and

$$R_F^M = [R_{CN}(MFCC), R_{CM}(MFCC)]$$

respectively. In the top five rows, overlap decisions are made with GMM’s. The first two rows show that both model fit feature types contain some overlap information and that those based on MFCC’s are superior. In fact, comparing with the first row of Table 6.7., we see that MFCC model fit features based on MFCC’s are slightly better overlap detector features than MFCC’s themselves. As shown in the third row, there is a slight performance gain when the two MFCC feature types are combined, indicating that they contain some complimentary information. Rows four and five of the table suggest that, at least for GMM classifiers, location information does not improve overlap detection.

The main problem with this detector is the high false detect of overlap which leads to low precision – so high that tuning a threshold to eliminate it also eliminates almost all correct overlap detections. From anecdotal inspection, some false detects are actually laughter or reference alignment errors, but there are so many that this cannot account for the problem.

Looking at features for two individual meetings, the features seem to be providing the right overlap cues. A segment of an example meeting is shown in Figure 6.7., the top lines of which show the location-based features associated with delays. The bottom lines are indicators of false overlap detection, true overlap, and true single talker speech, respectively. Though the features peak during some of the overlap regions, there are many falsely detected overlaps because of other features which appear to be highly variable. Unfortunately, the feature/overlap relationship deteriorates in other meetings, as shown in Figure 6.8. Anecdotally, some of the problem seems to be due
Table 6.8: MFCC’s vs. Location Features For Overlap Detection (GMM and MLP classifiers)

<table>
<thead>
<tr>
<th>feature</th>
<th>class</th>
<th>precision</th>
<th>recall</th>
<th>F</th>
<th>precision</th>
<th>recall</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R^L_F$</td>
<td>GMM</td>
<td>0.08</td>
<td>0.62</td>
<td>0.15</td>
<td>0.19</td>
<td>0.56</td>
<td>0.29</td>
</tr>
<tr>
<td>$R^M_F$</td>
<td>GMM</td>
<td>0.06</td>
<td>0.56</td>
<td>0.11</td>
<td>0.23</td>
<td>0.66</td>
<td>0.35</td>
</tr>
<tr>
<td>$R^M_F$ + MFCC</td>
<td>GMM</td>
<td>0.09</td>
<td>0.61</td>
<td>0.16</td>
<td>0.26</td>
<td>0.58</td>
<td>0.36</td>
</tr>
<tr>
<td>$R^L_F + \Delta_r$ + MFCC</td>
<td>GMM</td>
<td>0.10</td>
<td>0.66</td>
<td>0.17</td>
<td>0.23</td>
<td>0.55</td>
<td>0.32</td>
</tr>
<tr>
<td>$R^M_F + \Delta_r$ + MFCC</td>
<td>GMM</td>
<td>0.09</td>
<td>0.60</td>
<td>0.16</td>
<td>0.24</td>
<td>0.58</td>
<td>0.34</td>
</tr>
<tr>
<td>$R^M_F$ + MFCC</td>
<td>MLP</td>
<td>0.39</td>
<td>0.13</td>
<td>0.20</td>
<td>0.70</td>
<td>0.20</td>
<td>0.31</td>
</tr>
<tr>
<td>$R^M_F + \Delta_r$ + MFCC</td>
<td>MLP</td>
<td>0.46</td>
<td>0.18</td>
<td>0.26</td>
<td>0.67</td>
<td>0.19</td>
<td>0.30</td>
</tr>
</tbody>
</table>

To the failure of the mean-variance feature normalization to maintain a consistent relationship with overlaps across different meetings, causing an inconsistent training set. But in some meetings, the features are apparently meaningless.

With the overlap features discussed in this chapter, good detector performance would probably require both a better normalization method, and some form of automatic feature selection, possibly leveraging methods in Chapter 5, with compensation for missing features.

The last two rows of Table 6.8 show the results when overlap decisions are made with a multi-layer perceptron (MLP). The MLP, which had one hidden layer of 90 nodes, was fed a 0.31s sliding window of GMM posteriors. The main effect of MLP smoothing is to increase precision at the expense of recall – a desirable trade-off, since the action taken for a false overlap detect (adding a new speaker) increases diarization error, while the action taken for a missed detection (do nothing) does not change diarization error. The last two rows also show that location information in the form of triplet sums substantially improves performance on the training set but
Figure 6.7: Location-based overlap detection features that peak during overlap for one meeting. The traces from bottom to top are true single speaker speech periods, true overlapped speech periods, false overlap detect, $R_{CN}(MFCC) + R_{CN}(XC)$, $R_{CN}(XC)$, and $R_{CM}(XC)$. 
Figure 6.8: Location-based overlap detection features that do not have meaningful peaks (same features as in Figure 6.7 but a different meeting). The sharp notches in the top three features are an artifact due to zeroing the time regions considered non-speech by the clusterer VAD; they do not affect overlap detector performance.
slightly degrades test set performance.

Because overlap detection remains an open research problem, it is difficult to compare these results with others in the literature. The most relevant published work is [20], where results are reported for a distant microphone on meetings in the AMI corpus, about 2/3 of which is not very spontaneous speech. Oracle voice activity detection was used, avoiding the noisy training data due to speech detection errors (that is dealt with in this thesis). On their training set, the authors of [20] report an F-score of 0.37, a precision of 0.66 and a recall of 0.26. They do not report a result on separate test data.

6.4 Summary

We find that detecting overlaps can indeed improve performance substantially, and that location features can help close the performance gap between diarization on overlapped and single talker speech. With a perfect overlap detector, special processing before diarization can help, but most of the potential gain is available after the clustering stage – a simple strategy of assigning speaker labels in overlap regions according to the labels of the neighboring segments could potentially result in a 15% relative diarization error improvement [92].

The overlap detection work done for this thesis is preliminary. A Hough transform and spectral fit detector algorithm developed on TIMIT data performed well on synthesized overlaps but unfortunately, this was was not helpful in developing features for use in real meeting environments. Features developed on real meeting data performed better, but were still not usable for diarization improvement; work remains on normalization techniques, and upon dynamic feature selection.

Another area for improvement is the weakening of a very strong assumption made by the speaker model fit features: that speakers are separable-enough in time that

\[\text{http://corpus.amiproject.org/documentations/overview/}\]
broadband features are capable of modeling the mixture. A weaker and more accurate assumption is that overlapping speakers are separable in time-frequency, that is, they dominate in separate time/frequency cells: modeling of speaker location in separate time-frequency cells, as in [80], may yield better results.

Finally, the delay triplet sum pair selection algorithm could likely be improved by selecting pairs that have high mutual information with clusterer output labels but also with high mutual information with other selected pairs, and with large, consistent time delay differences.

After additional overlap detector accuracy improvements, the last step would be to use the output to reduce diarization error.
Chapter 7

CONCLUSIONS

A main goal of this thesis was to illustrate methods for improving the use of location information in meeting speaker diarization, both as features used in clustering and for speaker overlap handling. So far, we have shown new ways they can be used to improve diarization, and it appears that they may also be useful for overlap detection.

7.1 Contributions

We have shown that PCA dimension reduction can eliminate uninformative noise admitted into a vector of time delay lags when correlations are calculated across all pairs. This method works slightly better than the single reference channel and dynamic programming smoothing employed in the baseline system, although we have also found that the dynamic programming algorithm can actually produce worse results than no smoothing at all. This was at first surprising, given the very competitive results produced by that system, but as we looked into the impact of meeting acoustic variability on diarization error, it became clear that the discrepancy is at least partly due to differences in test and training data.

We have also developed a CCA-based approach for fusing speaker ID information as viewed by different physical processes. We found that dimension reduction of time delays is enhanced when CCA is used with traditional MFCC speaker ID's. We also found that, although energy ratios are an unreliable diarization feature by themselves, they can be used as another way to improve time delay information via CCA. In fact, our best results were obtained when all three views were employed in diarization.

A major challenge in this research was the high variability between meetings.
In oracle experiments we found that using a fixed dimension dimension reduction for all meetings was suboptimal, and that results could be improved slightly with an automatic method employing linear regression on meeting acoustic properties. Another approach we took with location features was to develop automatic methods to explicitly select high quality channel pairs. For this task, channel SNR turned out to be an inadequate criteria so we tried maximum relevance, minimum redundancy information theoretic feature selection of channel pairs. This approach also turned out to yield better performance than the baseline system dynamic programming smoother. Improvements in this thesis have largely come via dynamic approaches that adapt to meeting variability. In addition, meeting variability – especially on the small set of meetings used here – makes performance assessment difficult, although we did expose some statistically significant results. These are the first example known to this author of significance testing on meeting diarization outputs.

Location features are also employed to detect and process speaker overlap, a significant cause of increased speaker diarization error. First, monaural overlap features are developed for a monaural delay-sum beamformer output based on synthesized overlap, but these features do not prove to be effective in actual meeting data. Location features are also explored for overlap detection and, while there is a performance improvement, the resulting detector is not accurate enough to be used for meeting diarization improvement. Finally, we have developed a simple, nearest-neighbor overlap processing scheme which, when given accurate overlap detection, can be used to improve diarization accuracy.

7.2 Future work

Experiments performed for this thesis suggest many areas for future work. While we had small successes with automatic PCA dimension determination, we did not explore this approach for CCA. Automatically selecting the CCA output dimension is a clear next step. Both PCA and CCA are linear techniques, which have the advantage of
simplicity. However, there is no reason to believe that the correlation or energy ratio manifold traced by meeting speaker locations actually are linear, so non-linear PCA or CCA are potential next steps.

Given the promising cheating experiments discussed in Section 4.3, it is surprising that energy ratios were not good diarization features by themselves. A possible reason for the discrepancy is that modeling location information found in energy ratios requires more complex models than are possible in the early stages of diarization clustering. Indeed, the cheating experiments used 60 mixture GMM’s, while for actual diarization experiments, the best number of initial mixtures was usually about 5. It may be possible to use the speaker segmentation at the end of a diarization run to build higher quality ER speaker models; these models could be used for resegmentation or for another round of diarization.

There is much to do for channel pair feature selection. Although we showed how to use mRMR to select a fixed number of features, we did not have the time to develop an automatic way of determining the best number of mRMR-selected features. An obvious approach would be to apply a “wrapper” technique, where iteration feature selection stops when a simple classifier finds no improvement. In this case, we would have to use estimated, rather than true class labels, so anticipated performance gains are speculative. Model entropy may be another way to determine the mRMR stopping point. We also speculate that, even after feature selection has removed low quality pairs, diarization error may be further improved if the selected features were dimension reduced with PCA, CCA or some other technique.

Further analysis is needed to understand why the existing overlap detector features are so variable. It may be that the solution to the problem lies in dynamic, permitting dimension reduction or feature selection, similar to that done for diarization. Although our overlap detector may not yet be accurate enough to be used to reduce diarization error, we believe that significant gains may still result if the one or two speaker model approach was extended to model speaker location not just in separate
frames, but in separate time/frequency cells.
BIBLIOGRAPHY


[114] David A. van Leeuwen and Marijn Huijbregts. The AMI speaker diarization system for NIST RT06s meeting data. In *NIST RT06 workshop*, 2006.


VITA

Scott Otterson was born in Janesville, Wisconsin and grew up in Minnesota. In 1984, he received a Bachelor’s degree in Electrical Engineering from the University of Minnesota, Minneapolis. He eventually returned to Wisconsin, where he received a Masters Degree in Electrical Engineering (Marquette University, Milwaukee, 1993). In 2008, received a PhD in Electrical Engineering from the University of Washington, Seattle.