Partially Supervised Speaker Clustering

Hao Tang, *Member, IEEE,* Stephen Chu, *Member, IEEE,* Mark Hasegawa-Johnson, *Senior Member, IEEE,* and Thomas Huang, *Life Fellow, IEEE*

Abstract—Content-based multimedia indexing, retrieval and processing as well as multimedia databases demand the structuring of the media content (image, audio, video, text, etc.), one significant goal being to associate the identity of the content to the individual segments of the signals. In this paper, we specifically address the problem of speaker clustering, the task of assigning every speech utterance in an audio stream to its speaker. We offer a complete treatment to the idea of partially supervised speaker clustering, which refers to the use of our prior knowledge of speakers in general to assist the unsupervised speaker clustering process. By means of an independent training data set, we encode the prior knowledge at the various stages of the speaker clustering pipeline via 1) learning a speaker-discriminative acoustic feature transformation, 2) learning a universal speaker prior model, and 3) learning a discriminative speaker subspace, or equivalently, a speaker-discriminative distance metric. We study the directional scattering property of the Gaussian mixture model (GMM) mean supervector representation of utterances in the high-dimensional space, and advocate exploiting this property by using the cosine distance metric instead of the Euclidean distance metric for speaker clustering in the GMM mean supervector space. We propose to perform discriminant analysis based on the cosine distance metric, which leads to a novel distance metric learning algorithm – linear spherical discriminant analysis (LSDA). We show that the proposed LSDA formulation can be systematically solved within the elegant graph embedding general dimensionality reduction framework.
Our speaker clustering experiments on the GALE database clearly indicate that 1) our speaker clustering methods based on the GMM mean supervector representation and vector-based distance metrics outperform traditional speaker clustering methods based on the "bag of acoustic features" representation and statistical model based distance metrics, 2) our advocated use of the cosine distance metric yields consistent increases in the speaker clustering performance as compared to the commonly used Euclidean distance metric, 3) our partially supervised speaker clustering concept and strategies significantly improve the speaker clustering performance over the baselines, and 4) our proposed LSDA algorithm further leads to the state-of-the-art speaker clustering performance.

Index Terms—Speaker clustering, partial supervision, distance metric learning.

1 Introduction

Content-based multimedia indexing, retrieval and processing as well as multimedia databases are active fields of research in the information era [1]. In many situations it is highly demanded that we structure the media content (image, audio, video, text, etc.) so that the identity of the content (face, voice, keywords, etc.) can be associated with the individual segments of the data. Often, clustering multimedia data is a first step to multimedia content analysis as well as multimedia database construction, mining, search, and visualization [2].

In this paper, the problem of speaker clustering [3], [4], [5], [6], [7], [8] is specifically addressed. Speaker clustering aims to assign every speech utterance in an audio stream to its respective speaker, and is an essential part of a task known as speaker diarization [9], [10], [11], [12], [13], [14]. Also referred to as speaker segmentation and clustering, or "who spoke when", speaker diarization is the process of partitioning an input audio stream into temporal regions of speech signal energy contributed from the same speakers. A typical speaker diarization system consists of three stages. The first is the speech detection stage, where we find the portions of speech in the audio stream. The second is the segmentation stage, where we find the locations in the audio stream likely to be change points between speakers. At this stage, we often over-segment the audio stream, resulting in only one single speaker in each segment. The last is the clustering stage, where we associate the segments from the same speakers together. Figure 1 illustrates the process of speaker diarization. In this paper, we mainly focus on the clustering stage, not only because the clustering stage is the most important part of speaker diarization, but also most techniques developed for the clustering stage can be readily applied to the segmentation stage (for example, with the help of a sliding window of fixed or variable length).

Unlike speaker recognition (i.e. identification and verification), where we have training data for the speakers and thus training can be done in a supervised fashion, speaker clustering is usually performed in a completely unsupervised manner. The output of

- *H. Tang, M. Hasegawa-Johnson, and T. Huang are with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, 61821 USA. E-mail: {haotang2,jhasegaw,t-huang1}@uiuc.edu.*
- *S. Chu is with the Human Language Technologies Group at the IBM T. J. Watson Research Center, Yorktown Heights, NY, 10598 USA. E-mail: schu@us.ibm.com.*

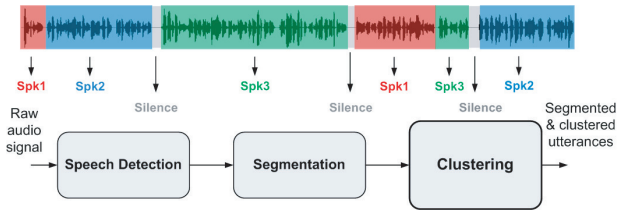

Fig. 1. The process of speaker diarization. A typical speaker diarization system consists of a speech detection stage, a segmentation stage, and a clustering stage.

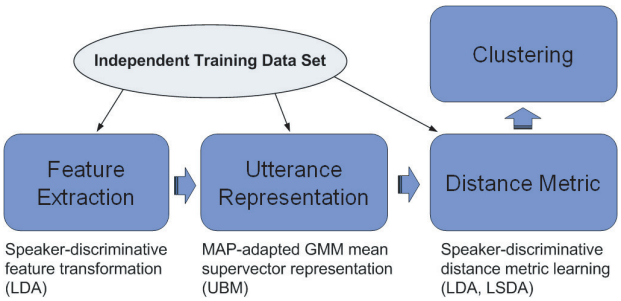

Fig. 2. The general speaker clustering pipeline. There are four essential elements in any speaker clustering algorithm. This paper transfers knowledge from an independent training set in order to improve every stage in the speaker clustering pipeline.

speaker clustering is a unique arbitrary code for each speaker (e.g., spk1, spk2, etc.) rather than his or her real identity (e.g., Tom, Mary, etc.). An interesting question is: Can we do speaker clustering in a somehow supervised manner? That is, can we make use of all available prior information that may be helpful for speaker clustering?

Our answer to this question is positive. It is worth noting that a few researchers in the field of speaker diarization have already tried to incorporate some prior knowledge into their methods and indeed gained noticeable improvements in the performance. For example, the use of a universal background model (UBM) for adapted Gaussian mixture model (GMM) based clustering was attempted in [9] and [10], and the GMM mean supervector as the utterance representation was recently adopted in [13] and [14]. However, none of the previous work addresses every facet of the problem. In this paper, we offer a complete treatment to the conceptually new idea of partially supervised speaker clustering, which refers to the use of our prior knowledge of speakers in general to assist the unsupervised speaker clustering process. By means of an independent training data set, we acquire prior knowledge about speakers in general by 1) learning a speaker-discriminative acoustic feature transformation, 2) learning a universal speaker prior model (i.e. a UBM) which is then adapted to the individual utterances to form the GMM mean supervector representation, whose directional scattering properties we study and exploit, and 3) learning a discriminative speaker subspace, or equivalently, a speaker-discriminative distance metric.

Figure 2 is a general speaker clustering pipeline. Basically, there are four critical elements in any speaker clustering algorithm and it is these elements that make a difference. We incorporate our prior knowledge of speakers into the various stages of this pipeline through an independent training data set. First, at the feature extraction stage, we learn a speaker-discriminative acoustic feature transformation based on linear discriminant analysis (LDA) [15]. Second, at the utterance representation stage, we adopt the *maximum a posteriori* (MAP) adapted GMM mean supervector representation [16] based on a UBM [17], which can be considered as a universal speaker prior model. Third, at the distance metric stage, we learn a speaker-discriminative distance metric through a novel algorithm – linear spherical discriminant analysis (LSDA). Note that at the clustering stage, conventional clustering techniques such as k-means [18] and hierarchical clustering [19] can be naturally employed.

The contribution of our paper is at least three-fold. First, we propose methods that allow the transfer of learning from an independent training set to the unsupervised clustering problem, and show that these strategies significantly improve the speaker clustering performance over the baselines. Our methods outperform traditional speaker clustering methods based on the "bag of acoustic features" representation and statistical model based distance metrics. Second and in particular, we study the directional scattering property of the GMM mean supervector representation of utterances in the high-dimensional space, and advocate exploiting this property by using the cosine distance metric instead of the Euclidean distance metric for GMM mean supervector speaker clustering. Last but not least, we propose to perform discriminant analysis based on the cosine distance metric, which leads to a novel distance metric learning algorithm – linear spherical discriminant analysis (LSDA). We show that the proposed LSDA formulation can be systematically solved within the elegant graph embedding [20] general dimensionality reduction framework. We demonstrate that the LSDA algorithm leads to the state-of-the-art speaker clustering performance.

This paper is organized as follows. In Sections 2, 3, 4, and 5, the four stages of the speaker clustering pipeline, namely feature extraction, utterance representation, distance metric, and clustering, are described. In each section, we first review the current state-of-the-art approaches, and then present the strategies that incorporate our partially supervised speaker clustering concept into the corresponding stage of the speaker clustering pipeline. In Section

6, we describe our experiment setup and protocol, introduce the performance evaluation metrics, and present the experiment results as well as provide a discussion of the results. Finally, we conclude the paper in Section 7.

2 Feature Extraction

2.1 Acoustic Features

The first stage of the speaker clustering pipeline is feature extraction. Feature extraction is the process of identifying the most important cues from the measured data while removing unimportant ones for a specific task or purpose based on domain knowledge. For speaker clustering, the most widely used features are the short-time spectrum envelope based acoustic features such as the mel-frequency cepstral coefficients (MFCC) and perceptual linear predictive (PLP) coefficients [21]. Although MFCC and PLP were not originally designed for representing the information relevant to distinguishing among different speakers, and in fact, their primary use is in speech recognition, they work reasonably well for speaker clustering in practice. The higher-order MFCCs (e.g., 13-19) are known to correspond to the source characteristics in the source-filter model of speech production [22], and thus convey speaker information. In order to account for the temporal dynamics of the spectrum, the basic MFCC or PLP features are usually augmented by their first-order derivatives (i.e. the delta coefficients) and second-order derivatives (i.e. the acceleration coefficients). Higher-order derivatives may be used too, although that is rarely seen. These derivatives incorporate the time-evolving properties of the speech signal and are expected to increase the robustness of the acoustic features.

2.2 Speaker-discriminative Acoustic Feature Transformation

The use of first and second order derivatives of the basic acoustic features introduces the characterization of the temporal dynamics of the spectrum. However, such characterization is completely unsupervised, and thus lacks the potential to discriminate between speakers. Using an independent training data set, we can simultaneously characterize the temporal dynamics of the spectrum and maximize the discriminative power of the augmented acoustic features based on a discriminative learning framework. Specifically, we first compute 13 PLP features for every speech frame with cepstral mean subtraction (CMS) and cepstral variance normalization (CVN) to compensate for the inter-section and inter-channel variability [23]. Then, instead of augmenting the basic PLP features by their first and second order derivatives, we augment them by the basic PLP features of the neighboring frames spanning a window centered on the current frame. More precisely, the PLP features of the current frame, those of the K_L (e.g., 4) frames to the left and those of the K_R (e.g., 4) frames to the right are concatenated to form a high-dimensional feature vector, referred to as the context-expanded feature vector. In the context-expanded feature vector space, we learn a speaker-discriminative acoustic feature transformation by LDA based on the known speaker labels of the independent training data set. The context-expanded feature vectors can then be projected onto a low-dimensional (e.g., 40) speaker-discriminative feature subspace, which is expected to provide optimal speaker separability. In this way we transfer knowledge about the speakers in one corpus to improve clustering of the speakers in a different corpus.

In the experiment section, we specifically compare the proposed LDA transformed acoustic features with the acoustic features traditionally augmented with the first and second order derivatives and show that the LDA transformed acoustic features outperform the traditional acoustic features on speaker clustering under the same clustering conditions. This validates that the proposed speaker-discriminative acoustic feature transformation strategy can provide a better frontend to speaker clustering as compared to traditional ones.

3 Utterance Representation

3.1 "Bag of Acoustic Features" Representation

The second stage of the speaker clustering pipeline is utterance representation. Utterance representation, as its name suggests, is the task of compactly encoding the acoustic features of an utterance. In the literature on speaker clustering, the mainstream utterance representation is the so-called "bag of acoustic features" representation where the acoustic feature vectors are described by a time-independent statistical model such as a Gaussian or GMM. The rationale behind this representation is that in speaker clustering the linguistic content of the speech signal is considered to be irrelevant and normally disregarded. Thus, temporal independence between inter-frame acoustic features is assumed.

Most often, due to its unimodal nature a single Gaussian is far from being sufficient to model the probability distribution of the acoustic features of an utterance, and a GMM is preferred. The theoretical property that a GMM can approximate any continuous probability density function (PDF) arbitrarily closely given a sufficient number of Gaussian components makes the GMM a popular choice for parametric PDF estimators.

The acoustic features of an utterance are modeled by an m-component GMM, defined as a weighted sum of m component Gaussian densities

$$p(\mathbf{x}|\lambda) = \sum_{i=1}^{m} w_i N(\mathbf{x}|\mu_i, \Sigma_i) \quad (1)$$

where $\mathbf{x}$ is a d-dimensional random vector, w_i is the i^{th} mixture weight, and $N(\mathbf{x}|\mu_i, \Sigma_i)$ is a multivariate Gaussian PDF

$$N(\mathbf{x}|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mu_i)^T \Sigma_i^{-1}(\mathbf{x}-\mu_i)} \quad (2)$$

with mean vector μ_i and covariance matrix Σ_i. w_i can be interpreted as the *a priori* probability that an observation of $\mathbf{x}$ comes from the source governed by the i^{th} Gaussian distribution. Thus it satisfies the properties $0 \leq w_i \leq 1$ and $\sum_{i=1}^{m} w_i = 1$. A GMM is completely specified by its parameters $\lambda = \{w_i, \mu_i, \Sigma_i\}_{i=1}^{m}$ and the estimation of the PDF reduces to finding the proper values of λ.

A central problem of the GMM is how to estimate the model parameters λ. This problem can be practically solved by maximum likelihood estimation (MLE) techniques such as the expectation-maximization (EM) algorithm [24]. However, it is widely known that MLE easily over-fits with insufficient training data. The number of free parameters of a GMM, p, depends on the feature dimension d and the number of Gaussian components m. More precisely, $p = md^2/2 + 3md/2 + m - 1$, which grows linearly in m but quadratically in d. In order to alleviate this "curse of dimensionality" [25], diagonal covariance matrices are often used in the Gaussian components. In this case, $p = 2md + m - 1$, which grows linearly in both m and d.

3.2 GMM Mean Supervector Representation

A relatively new utterance representation that has emerged in the speaker recognition area is the GMM mean supervector representation, which is obtained by concatenating the mean vectors of the Gaussian components of a GMM trained on the acoustic features of a particular utterance [26].

3.2.1 UBM and MAP Adaptation

When an utterance is short, the number of acoustic feature vectors available for training a GMM is small. To avoid over-fitting, we can first train a single GMM on an independent training data set, leading to a well-trained GMM known as the UBM in the speaker recognition literature [27]. Since the amount of data used to train the UBM is normally large, and the data is fairly evenly distributed across many speakers, the UBM is believed to provide a good representation of speakers in general. Therefore, it can be considered as a universal speaker prior model in which we encode the common characteristics of different speakers. Given a specific utterance, we can then derive a target GMM by adapting the UBM to the acoustic features of the utterance. This is done by MAP adaptation [28].

MAP adaptation starts with a prior model (i.e. the UBM), and iteratively performs EM estimation. In the E step, the posterior probability of a training vector falling into every Gaussian component is computed

$$p(i|\mathbf{x}_t) = \frac{w_{0i} N(\mathbf{x}_t|\mu_{0i}, \Sigma_{0i})}{\sum_{j=1}^{m} w_{0j} N(\mathbf{x}_t|\mu_{0j}, \Sigma_{0j})}, \quad i = 1, 2, \cdots, m \quad (3)$$

Note that Equation 3 is the probability that we re-assign the training vector $\mathbf{x}_t$ to the i^{th} Gaussian component of the UBM $\lambda_0 = \{w_{0i}, \mu_{0i}, \Sigma_{0i}\}_{i=1}^{m}$. Based on these posterior probabilities, we compute the sufficient statistics of the training data

$$n_i = \sum_{t=1}^{T} p(i|\mathbf{x}_t), \quad E_i = \frac{1}{n_i} \sum_{t=1}^{T} p(i|\mathbf{x}_t)\mathbf{x}_t \quad (4)$$

$$E_i^2 = \frac{1}{n_i} \sum_{t=1}^{T} p(i|\mathbf{x}_t)\mathbf{x}_t^2 \quad (5)$$

In the M step, the sufficient statistics of the training data are combined with the prior model sufficient statistics by interpolation. The new model parameters are obtained as follows

$$w_i' = [\alpha_i n_i/T + (1 - \alpha_i) w_i]\delta \quad (6)$$

$$\mu_i' = \beta_i E_i + (1 - \beta_i)\mu_i \quad (7)$$

$$\sigma_i'^2 = \gamma_i E_i^2 + (1 - \gamma_i)(\sigma_i^2 + \mu_i^2) - \mu_i'^2 \quad (8)$$

where δ is a scaling factor computed over all new mixture weights to ensure that they sum to unity. The interpolation coefficients in Equations 6-8 are data dependent and automatically determined for every Gaussian component using the empirical formula $\nu_i = n_i/(n_i + r^\nu)$ where $\nu \in \{\alpha, \beta, \gamma\}$ and r^ν is a fixed relevance factor for ν. This empirical formula offers a smart mechanism to control the balance between the new and old sufficient statistics. Figure 3 demonstrates the basic idea of MAP adaptation for a GMM.

3.2.2 GMM Mean Supervectors

The GMM mean supervector representation of an utterance is obtained by first MAP adapting the UBM to the acoustic features of the utterance and then concatenating the component mean vectors of the target GMM to form a long column vector. Figure 4

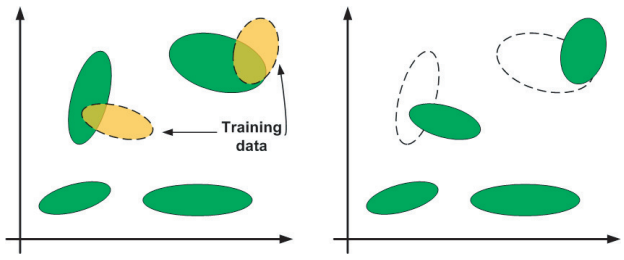

Fig. 3. The basic idea of MAP adaptation. MAP adaptation starts with a prior model and iteratively performs regularized EM estimation.

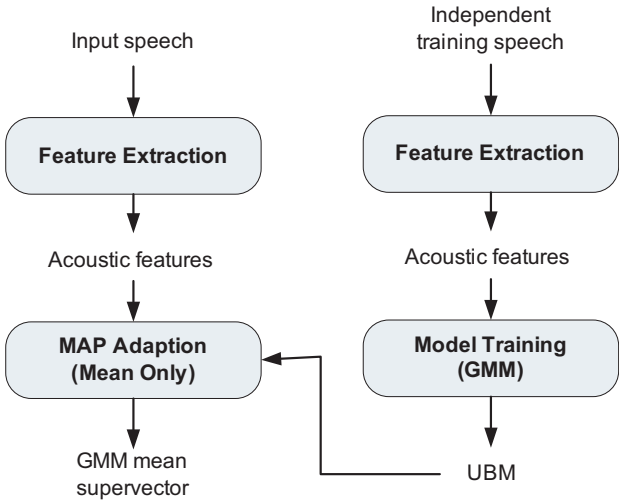

Fig. 4. The generation of a GMM mean supervector. A GMM mean supervector is obtained by MAP adapting only the component means of a UBM.

gives a block diagram that shows how a GMM mean supervector is generated.

Once a target GMM is obtained for an utterance, its component means are stacked to form a GMM mean supervector

$$\mathbf{s} = [\mu_1{}^T \quad \mu_2{}^T \quad ... \quad \mu_m{}^T]^T \tag{9}$$

It is numerically beneficial to subtract the mean supervector of the UBM from a GMM mean supervector, namely,

$$\mathbf{s}' = \mathbf{s} - \mathbf{s}_0 \tag{10}$$

where $\mathbf{s}_0$ is the mean supervector of the UBM. Without causing any ambiguity, we call $\mathbf{s}'$ (instead of $\mathbf{s}$) a GMM mean supervector. A complete set of GMM mean supervectors forms a high-dimensional space called the GMM mean supervector space.

A supervector created by stacking component means, as shown in Equation 9, represents local first-order differences between the UBM and adapted GMM: shifts in local modes and regional centers of mass. Concatenating variances to the supervector would allow us to also represent local second-order adaptation, e.g., changes in the local compactness of the GMM. The scattering patterns of second-order information in supervector space are very different from those of first-order information, however, so we choose to include only first-order information.

3.2.3 Property of GMM Mean Supervectors

The GMM mean supervector is an effective utterance representation that has been applied to speaker recognition. However, it has come to our attention that the use of the GMM mean supervector representation for speaker clustering is still rare. The GMM mean supervector representation allows us to represent an utterance as a single data point in a high-dimensional space, where conventional clustering techniques such as k-means and the hierarchical clustering can be naturally applied.

Figure 5 visualizes the GMM mean supervectors of many utterances from five different speakers using 2D scatter plots of their two principal components. In each plot, the different speakers are shown in different colors. For each speaker, there are about 150 utterances, denoted by small dots. As one can see, the data points belonging to the same speaker tend to cluster together. Thus, the Euclidean distance metric is a reasonable choice for speaker clustering in the GMM mean supervector space. However, one can also observe that the data points show very strong directional scattering patterns. The directions of the data points seem to be more informative and indicative than their magnitudes. This observation motivated us to favor the cosine distance metric over the Euclidean distance metric for speaker clustering in the GMM mean supervector space.

A reasonable explanation as to why the GMM mean supervectors show strong directional scattering patterns is that when we perform mean-only MAP adaptation, only a subset of the UBM component means is adjusted, and the particular subset that is adjusted seems to be rather speaker-dependent. Hence, the speaker-specific information is encoded in those component means which are adapted. Therefore, the utterances from the same speaker tend to yield a cluster of GMM mean supervectors that scatter in a particular direction in the GMM mean supervector space.

As presented later in the experiment section, our experiment results on all speaker clustering tasks clearly demonstrate that the cosine distance metric consistently outperforms the Euclidean distance metric when using the GMM mean supervector as the utterance representation. This strongly supports our discovery of the directional scattering property of the GMM mean supervectors and forms the foundation of our original motivation to perform discriminant analysis in the cosine distance metric space.

4 Distance Metric

The third stage of the speaker clustering pipeline is distance metric. The distance metrics that can be used for speaker clustering are closely related to the particular choice of utterance representations. Two popular categories of distance metrics, namely likelihood-based distance metrics and vector-based distance metrics, are widely used for the two corresponding utterance representations, respectively.

4.1 Likelihood-based Distance Metrics

For the "bag of acoustic features" utterance representation, the distance metric should represent some

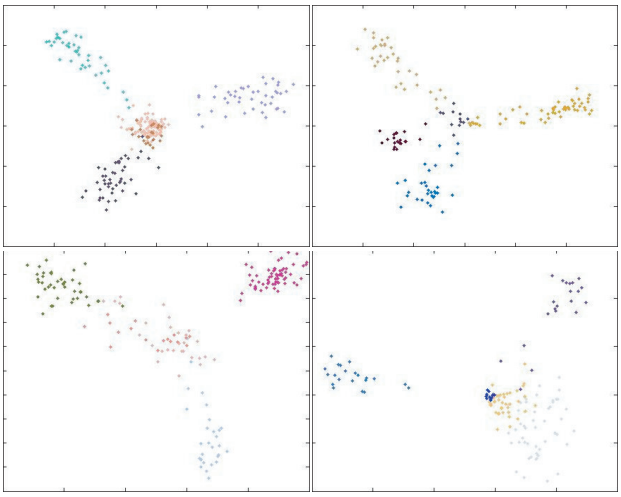

Fig. 5. The property of GMM mean supervectors. The data points show strong directional scattering patterns.

measure of the distance between two statistical models. A famous likelihood-based distance metric extensively used for speaker clustering is the Bayesian information criterion (BIC) [29]. For a given utterance, the BIC value indicates how well a model fits the utterance and is given by

$$BIC(M_i) = \log L(X_i|M_i) - \frac{\lambda}{2} k_i \log(n_i) \qquad (11)$$

where $L(X_i|M_i)$ is the likelihood of the acoustic features X_i given the model M_i, λ is a design parameter, k_i is the number of free parameters in M_i, and n_i is the number of feature vectors in X_i. The distance between two utterances X_i and X_j is given by the ΔBIC value. If we assume M_i and M_j are both Gaussian, then ΔBIC is given by

$$\Delta BIC(X_i, X_j) = n \log \Sigma - n_i \log \Sigma_i - n_j \log \Sigma_j - \lambda P \qquad (12)$$

where $n = n_i + n_j$, Σ_i and Σ_j are the covariance matrices of X_i and X_j, respectively, Σ is the covariance matrix of the aggregate of X_i and X_j, and P is a penalty term given by

$$P = \frac{1}{2}(d + \frac{1}{2}d(d+1)) \log n \qquad (13)$$

with d being the dimension of the acoustic feature vectors.

Other likelihood-based distance metrics include the generalized likelihood ratio (GLR), Gish distance, Kullback-Leibler divergence (KLD), divergence shape distance (DSD), Gaussian divergence (GD), cross BIC (XBIC), cross log likelihood ratio (XLLR), and so forth [30]. All these metrics have been proposed for the "bag of acoustic features" utterance representation.

4.2 Vector-based Distance Metrics

For the GMM mean supervector utterance representation, since an utterance can be represented as a single data point in a high-dimensional vector space, the most often used distance metric is the Euclidean distance metric

$$d(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T (\mathbf{x} - \mathbf{y}) \qquad (14)$$

As we discussed earlier, in the GMM mean supervector space, the data points belonging to the same speaker tend to cluster together. Thus, the Euclidean distance metric is a reasonable choice for speaker clustering in the GMM mean supervector space. However, we observe that the data points show very strong directional scattering patterns. The directions of the data points seem to be more informative and indicative than their magnitudes. This observation motivated us to advocate the use of the cosine distance metric instead of the Euclidean distance metric for speaker clustering in the GMM mean supervector space. The cosine distance metric is a measure of the angle between two vectors in the space and is irrelevant to the norms of the vectors. It is defined as

$$d(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathbf{x}^T \mathbf{y}}{\sqrt{\mathbf{x}^t \mathbf{x}} \sqrt{\mathbf{y}^T \mathbf{y}}} \qquad (15)$$

Our experiments show that the cosine distance metric consistently outperforms the Euclidean distance metric for speaker clustering in the GMM mean supervector space.

4.3 Distance Metric Learning versus Linear Subspace Learning

Although the Euclidean and cosine distance metrics can be directly used, they are optimal only if the data points are uniformly distributed in the entire space. In a high-dimensional space, most often the data points lie in or near a low-dimensional manifold, or preferably a linear subspace, of the original space. In this case, it is extremely advantageous if we can learn an optimal distance metric for the data.

We define a generalized Euclidean distance metric between two data points $\mathbf{x}$ and $\mathbf{y}$ as

$$d(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T A (\mathbf{x} - \mathbf{y}) \qquad (16)$$

where the positive definite matrix A is aimed to compensate for the non-uniform data distribution. If A coincides with the covariance matrix of the data, this generalized Euclidean distance metric reduces to the Mahalanobis distance [31]. Equation 16 can be rewritten as

$$\begin{aligned} d(\mathbf{x}, \mathbf{y}) &= (A^{\frac{1}{2}}\mathbf{x} - A^{\frac{1}{2}}\mathbf{y})^T (A^{\frac{1}{2}}\mathbf{x} - A^{\frac{1}{2}}\mathbf{y}) \\ &= (W\mathbf{x} - W\mathbf{y})^T (W\mathbf{x} - W\mathbf{y}) \end{aligned} \qquad (17)$$

That is, the generalized Euclidean distance metric between $\mathbf{x}$ and $\mathbf{y}$ can be re-organized as the Euclidean distance metric between two linearly transformed data points $W\mathbf{x}$ and $W\mathbf{y}$ where $W = A^{\frac{1}{2}}$.

Similarly, we define a generalized cosine distance metric between $\mathbf{x}$ and $\mathbf{y}$ as

$$\begin{aligned} d(\mathbf{x},\mathbf{y}) &= 1-\frac{\mathbf{x}^T A\mathbf{y}}{\sqrt{\mathbf{x}^T A\mathbf{x}}\sqrt{\mathbf{y}^T A\mathbf{y}}} \\ &= 1-\frac{(A^{1/2}\mathbf{x})^T(A^{1/2}\mathbf{y})}{\sqrt{(A^{1/2}\mathbf{x})^T(A^{1/2}\mathbf{x})}\sqrt{(A^{1/2}\mathbf{y})^T(A^{1/2}\mathbf{y})}} \\ &= 1-\frac{(W^T\mathbf{x})^T(W^T\mathbf{y})}{\sqrt{(W^T\mathbf{x})^T(W^T\mathbf{x})}\sqrt{(W^T\mathbf{y})^T(W^T\mathbf{y})}} \end{aligned} \quad (18)$$

Likewise, the generalized cosine distance metric between $\mathbf{x}$ and $\mathbf{y}$ is the cosine distance metric between two linearly transformed data points $W\mathbf{x}$ and $W\mathbf{y}$ where $W = A^{\frac{1}{2}}$. In this sense, it is clear that learning an optimal distance metric is equivalent to learning an optimal linear transformation of the original high-dimensional space. There exist various linear subspace learning methods that can fit into this context.

4.4 Distance Metric Learning in Euclidean Space

Linear subspace learning can be classified into two distinct categories: unsupervised learning and supervised learning. For unsupervised linear subspace learning, principal component analysis (PCA) [15] may be the most early developed and prevailing technique, and when applied to speech or speaker recognition, is known as the eigenvoice approach [32]. Other more recent unsupervised learning techniques include the locality preserving projection (LPP) [33], neighborhood preserving embedding (NPE) [34], etc. All these techniques may be applied to speaker clustering. However, we are most interested in supervised learning since the goal of speaker clustering is related to classification. It is natural that we prefer a learning technique that is discriminative rather than generative. The most famous technique for supervised linear subspace learning is Fisher's LDA. LDA has been applied to speaker clustering, and the resulting technique is termed the fishervoice approach [35]. The term "fishervoice" is analogous to "fisherface" in the face recognition literature, where the fisherface approach refers to the face recognition method based on LDA while the eigenface approach refers to the face recognition method based on PCA.

4.5 Distance Metric Learning in Cosine Space

Most existing linear subspace learning techniques (e.g., PCA and LDA) are implicitly based on the Euclidean distance metric. As we mentioned earlier, due to the directional scattering property of the GMM mean supervectors, we favor the cosine distance metric over the Euclidean distance metric for speaker clustering in the GMM mean supervector space. Therefore, we propose to perform discriminant analysis in the cosine distance metric space.

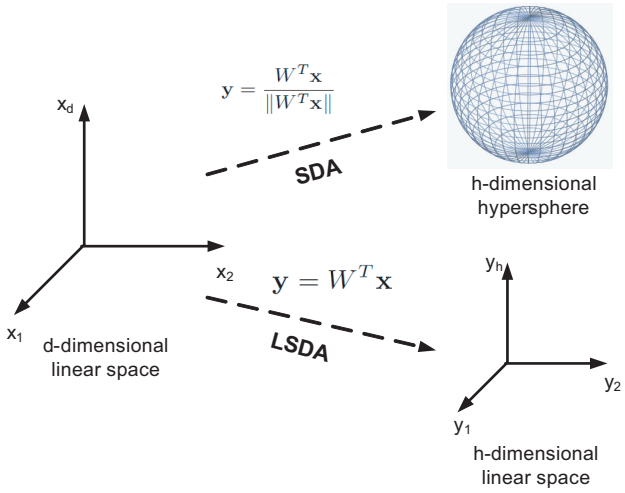

Fig. 6. The schematic illustration of SDA and LSDA. Under two mild conditions, the nonlinear projection can be linearized and thus SDA reduces to LSDA.

4.5.1 Linear Spherical Discriminant Analysis

We coined the phrase "spherical discriminant analysis" (SDA) to denote discriminant analysis in the cosine distance metric space. We define a projection from a d-dimensional linear space to an h-dimensional hypersphere where $h < d$

$$\mathbf{y} = \frac{W^T\mathbf{x}}{\|W^T\mathbf{x}\|} \quad (19)$$

We note that such a projection is nonlinear. However, under two mild conditions, this projection can be linearized. One condition is that the objective function for learning the projection only involves the cosine distance metric. The other condition is that only the cosine distance metric is used in the projected space. In this case, the norm of the projected vector $\mathbf{y}$ has impact on neither the objective function nor distance computation in the projected space. Thus, the denominator term of Equation 19 can be safely dropped, leading to a linear projection $\mathbf{y} = W^T\mathbf{x}$, which is called "linear spherical discriminant analysis" (LSDA). Figure 6 illustrates the basic ideas of SDA and LSDA.

Formally speaking, the goal of LSDA is to seek a linear projection W such that the average within-class cosine similarity of the projected data is maximized while the average between-class cosine similarity of the projected data is minimized. Assuming that there are c classes, the average within-class cosine similarity is defined to be the average of the class-dependent average cosine similarities between the projected data vectors. It can be written in terms of the unknown projection matrix W and original data points $\mathbf{x}$

$$S_W = \frac{1}{c}\sum_{i=1}^{c} S_i \quad (20)$$

$$S_i = \frac{1}{|D_i||D_i|}\sum_{\mathbf{y}_j,\mathbf{y}_k\in D_i}\frac{\mathbf{y}_j^T\mathbf{y}_k}{\sqrt{\mathbf{y}_j^T\mathbf{y}_j}\sqrt{\mathbf{y}_k^T\mathbf{y}_k}} = \frac{1}{|D_i||D_i|}\sum_{\mathbf{x}_j,\mathbf{x}_k\in D_i}\frac{\mathbf{x}_j^T WW^T\mathbf{x}_k}{\sqrt{\mathbf{x}_j^T WW^T\mathbf{x}_j}\sqrt{\mathbf{x}_k^T WW^T\mathbf{x}_k}} \quad (21)$$

where $|D_i|$ denotes the number of data points in the i^{th} class. Similarly, the average between-class cosine similarity is defined to be the average of the average cosine similarities between any two pairs of classes. It can be likewise written in terms of W and $\mathbf{x}$

$$S_B = \frac{1}{c(c-1)}\sum_{m=1}^{c}\sum_{n=1}^{c}S_{mn} \quad (m\neq n) \quad (22)$$

$$S_{mn} = \frac{1}{|D_m||D_n|}\sum_{\substack{\mathbf{y}_j\in D_m\\ \mathbf{y}_k\in D_n}}\frac{\mathbf{y}_j^T\mathbf{y}_k}{\sqrt{\mathbf{y}_j^T\mathbf{y}_j}\sqrt{\mathbf{y}_k^T\mathbf{y}_k}} = \frac{1}{|D_m||D_n|}\sum_{\substack{\mathbf{x}_j\in D_m\\ \mathbf{x}_k\in D_n}}\frac{\mathbf{x}_j^T WW^T\mathbf{x}_k}{\sqrt{\mathbf{x}_j^T WW^T\mathbf{x}_j}\sqrt{\mathbf{x}_k^T WW^T\mathbf{x}_k}} \quad (23)$$

where $|D_m|$ and $|D_n|$ denote the number of data points in the m^{th} and n^{th} classes, respectively.

The LSDA criterion is to maximize S_W while minimizing S_B, which can be written in the trace difference form

$$W = arg\max_W(S_W - S_B) \quad (24)$$

Note that there are various forms of the criterion that may be adopted. We choose the trace difference form, which is similar to the work of Ma et al. [36]. However, we systematically solve our LSDA formulation in an elegant general dimensionality reduction framework known as graph embedding [20], [37].

4.5.2 Graph Embedding Solution to LSDA

Graph embedding is a general framework for dimensionality reduction, where, an undirected weighted graph, $G = \{X, S\}$, with vertex set X and similarity matrix S, is used to characterize certain statistical or geometrical properties of a data set. A vertex $\mathbf{x}_i$ in X represents a data point in the high-dimensional space. An entry s_{ij} in S, denoted as the weight of the edge connecting $\mathbf{x}_i$ and $\mathbf{x}_j$, represents the similarity between the two corresponding data points. The purpose of graph embedding is to represent each vertex of the graph as a low dimensional vector that preserves the similarities in S.

Graph embedding unifies most dimensionality reduction algorithms into a general framework. For a specific dimensionality reduction algorithm, we often use two graphs: the intrinsic graph $\{X, S^{(i)}\}$, which characterizes the data properties that the algorithm aims to preserve, and the penalty graph $\{X, S^{(p)}\}$, which characterizes the data properties that the algorithm aims to avoid. These two graphs share the same vertex set but have different similarity matrices. The graph similarity preserving criterion is given by

$$W = arg\min_W\sum_{i\neq j}\|f(\mathbf{x}_i, W) - f(\mathbf{x}_j, W)\|^2(s_{ij}^{(i)} - s_{ij}^{(p)}) \quad (25)$$

where $f(\mathbf{x}, W)$ is a general projection with parameters W. Note that the above objective function integrates the two aforementioned graphs through the subtraction of the similarities in the penalty graph from the similarities in the intrinsic graph. One can easily see that minimizing this objective function ensures that if the data points $\mathbf{x}_i$ and $\mathbf{x}_j$ are close in the sense of the similarities in $S^{(i)}$ and $S^{(p)}$ then their projections in the low-dimensional space are close, too.

In Equation 25, if we use a spherical projection of the form of Equation 19, we obtain the following criterion

$$W = arg\min_W\sum_{i\neq j}\left\|\frac{W^T\mathbf{x}_i}{\|W^T\mathbf{x}_i\|} - \frac{W^T\mathbf{x}_j}{\|W^T\mathbf{x}_j\|}\right\|^2(s_{ij}^{(i)} - s_{ij}^{(p)}) \quad (26)$$

Although there is no closed-form solution to the optimization problem of Equation 26, as shown in [37], this problem can be solved using a steepest descent algorithm, with the gradient derived as

$$G = 2\sum_{i\neq j}\left\{\frac{f_{ij}W^T\mathbf{x}_i\mathbf{x}_i^T}{f_i^3 f_j} + \frac{f_{ij}W^T\mathbf{x}_j\mathbf{x}_j^T}{f_j^3 f_i} - \frac{W^T(\mathbf{x}_i\mathbf{x}_j^T + \mathbf{x}_j\mathbf{x}_i^T)}{f_i f_j}\right\}(s_{ij}^{(i)} - s_{ij}^{(p)}) \quad (27)$$

where $f_i = \sqrt{\mathbf{x}_i^T WW^T\mathbf{x}_i}$, $f_j = \sqrt{\mathbf{x}_j^T WW^T\mathbf{x}_j}$, and $f_{ij} = \mathbf{x}_i^T WW^T\mathbf{x}_j$. If we expand the L_2 norm term of Equation 26, by some simple manipulations we obtain

$$\left\|\frac{W^T\mathbf{x}_i}{\|W^T\mathbf{x}_i\|} - \frac{W^T\mathbf{x}_j}{\|W^T\mathbf{x}_j\|}\right\|^2 = 2\left(1 - \frac{\mathbf{x_i}^T WW^T\mathbf{x_j}}{\|W^T\mathbf{x_i}\|\,\|W^T\mathbf{x_j}\|}\right) \quad (28)$$

Thus, the criterion in Equation 26 is equivalent to the following criterion

$$W = arg\max_W\sum_{i\neq j}\frac{\mathbf{x_i}^T WW^T\mathbf{x_j}}{\sqrt{\mathbf{x}_i^T WW^T\mathbf{x}_i}\sqrt{\mathbf{x}_j^T WW^T\mathbf{x}_j}}(s_{ij}^{(i)} - s_{ij}^{(p)}) \quad (29)$$

By comparing Equation 29 to Equations 20–24, we conclude that the graph embedding criterion of Equation 26 is equivalent to the LSDA criterion of Equation 24 if the entries of the similarity matrices $S^{(i)}$ and $S^{(p)}$ are set to proper values, as follows

$$\begin{aligned} s_{jk}^{(i)} &\leftarrow \frac{1}{c|D_i||D_i|} \quad \text{if } \mathbf{x}_j, \mathbf{x}_k \in D_i, \quad i = 1, ..., c \\ s_{jk}^{(p)} &\leftarrow \frac{1}{c(c-1)|D_m||D_n|} \quad \text{if } \mathbf{x}_j \in D_m, \mathbf{x}_k \in D_n \\ & \quad m, n = 1, ..., c, m \neq n \end{aligned} \quad (30)$$

That is, by assigning appropriate values to the weights of the intrinsic and penalty graphs, our LSDA formulation can be systematically solved within the elegant graph embedding general dimensionality reduction framework.

5 CLUSTERING

The last stage of the speaker clustering pipeline is clustering. We are interested in conventional clustering techniques which can be applied to the GMM mean supervector space. We mainly focus on two traditional classes of algorithms. One is "flat" clustering – clustering by partitioning the data space. The other is hierarchical clustering, where we try not to construct a partition but a nested hierarchy of partitions. In most of real-world applications, one of these two classes of algorithms is employed.

The representative flat clustering algorithm is k-means whose objective is to partition the data space in such a way that the total intra-cluster variance is minimized. It iterates between a cluster assigning step and a mean updating step until convergence. Spherical k-means [38] is an extension of k-means that is based on the cosine distance metric.

The representative hierarchical clustering algorithm is agglomerative clustering [15] whose objective is to obtain a complete hierarchy of clusters in the form of a dendrogram. The algorithm adopts a bottom-up strategy. First, it starts with each data point being a cluster. Then, it checks which clusters are the closest and merges them into a new cluster. As the algorithm proceeds, it always merges the two closest clusters until there is only one single cluster left.

A remarkable question related to agglomerative clustering is how to determine which clusters are the closest. There exist several methods that measure the distance between two clusters, for instance, the single linkage, complete linkage, average linkage, "ward" linkage, and so on [19], [39]. We empirically discover that the "ward" linkage yields the best performance for speaker clustering. The "ward" linkage is a function that specifies the distance between two clusters X and Y by the increase in the error sum of squares (ESS) after the merging of X and Y ($Z = X \cup Y$):

$$\begin{aligned} \bar{\mathbf{x}} &= \frac{1}{|X|}\sum_{\mathbf{x}\in X}\mathbf{x}, \quad ESS(X) = \sum_{\mathbf{x}\in X}|\mathbf{x}-\bar{\mathbf{x}}|^2 \\ d(X,Y) &= ESS(Z) - [ESS(X) + ESS(Y)] \end{aligned} \quad (31)$$

6 EXPERIMENTS

6.1 Experiment Setup and Protocol

We conduct extensive speaker clustering experiments on the GALE Mandarin database [40]. The GALE database contains about 1900 hours of broadcast news speech data collected from various Mandarin TV programs at various times. The waveforms were sampled at 16 KHz and quantized at 16 bits per sample, and were automatically over-segmented into short utterances using the BIC criterion, with each utterance being as pure as possible, namely, each utterance being from a single speaker. A random sample of the results were further verified by human listeners.

TABLE 1
Experiment settings.

	Test set	Indep. training set
#speaker	630	498
#utterance	19024	18327
#utt/spk (ave.)	$30 \sim 40$	$30 \sim 40$
utt duration (ave.)	$3 \sim 4$s	$3 \sim 4$s

Our experiments are based on a test set of 630 speakers and 19024 utterances extracted from the GALE database. In order to implement our partially supervised speaker clustering strategies at the various stages of the speaker clustering pipeline, we employ an independent training set, which was also extracted from the GALE database. Note that the test set and the independent training set were chosen in such a way that speakers in the independent training set do not exist in the test set. Table 1 lists the detailed experiment settings.

To guarantee a statistically significant performance comparison, we carry out our experiments as follows.

1. A case is an experiment associated with a specific number of test speakers, namely 2, 5, 10, 20, 50, and 100, respectively;
2. For each case, this number of speakers are drawn randomly from the test set, and all the utterances from the selected speakers are used in the experiment;
3. For each case, 10 independent trials are run, each of which involves a random draw of the test speakers;
4. For each case, the mean and standard error of the clustering results over the 10 independent trials are reported.

6.2 Performance Evaluation Metrics

We report our experiment results based on two performance evaluation metrics, namely the clustering accuracy and the normalized mutual information (NMI) [41]. These two metrics are standard for evaluating (general) data clustering results [42]. The clustering accuracy is given by

$$r = \frac{1}{N}\sum_{i=1}^{N}[c_i = l_i] \quad (32)$$

where N denotes the number of test utterances, c_i is the cluster label of the i^{th} utterance returned by the algorithm, l_i is the true cluster label, and $[v]$ is an indication function which returns 1 if v is true and 0 otherwise.

The NMI is another popular, information-theoretically interpreted metric given by

$$r = \frac{I(C,L)}{[H(C)+H(L)]/2} \quad (33)$$

where $I(C, L)$ is the mutual information

$$I(C, L) = \sum_i \sum_j |c_i \bigcap l_j| \log \frac{|c_i \bigcap l_j|}{|c_i||l_j|} \quad (34)$$

and $H(C)$ and $H(L)$ are the entropy

$$H(C) = -\sum_i \frac{|c_i|}{N} \log \frac{|c_i|}{N}, \quad H(L) = -\sum_i \frac{|l_i|}{N} \log \frac{|l_i|}{N} \quad (35)$$

In the above formulas, $|c_i|$, $|l_j|$ and $|c_i \bigcap l_j|$ are the number of utterances from speaker c_i, l_j, and $c_i \bigcap l_j$, respectively.

The above two metrics are used for utterance-based evaluations. We extend them to frame-based evaluations by simply replacing the number of utterances in the above formulas with the corresponding number of frames. This allows us to investigate how the duration of an utterance affects the clustering performance.

6.3 Experiment Results and Discussions

Our main experiment results are presented in Tables 2–5. Specifically, we conduct speaker clustering 1) in the GMM mean supervector space with the Euclidean and cosine distance metrics; 2) in the PCA subspace with the Euclidean distance metric(i.e. the eigenvoice approach); 3) in the LPP subspace with the Euclidean distance metric; 4) in the NPE subspace with the Euclidean distance metric; 5) in the LDA subspace with the Euclidean distance metric (i.e. the fishervoice approach); 6) in the kernel-LDA nonlinear manifold with the Euclidean distance metric; 7) in the LSDA subspace with the cosine distance metric. In each experiment, we utilize both k-means (or spherical k-means) and agglomerative clustering. In order to compare our methods to the traditional "bag of acoustic features" methods, we employ the "Gaussian+BIC" method as the baseline. The experiment results are presented in four forms – utterance-based clustering accuracies (Table 2), utterance-based NMIs (Table 3), frame-based clustering accuracies (Table 4), and frame-based NMIs (Table 5). For each case, we present the mean of the results over 10 trials as well as the standard error of the mean, $se = s/\sqrt{n}$, in parentheses, where s is the standard deviation of the results and n the number of trials (10). In all tables, Orig stands for the original GMM mean supervector space, k for k-means, and h for hierarchical clustering.

Additionally, we compare the proposed LDA transformed acoustic features with the acoustic features traditionally augmented with the first and second order derivatives. Specifically, 13 basic PLP features augmented by their first and second order derivatives form a 39-dimensional traditional acoustic feature vector. Table 6 gives a comparison of the results (clustering accuracies) of both kinds of acoustic features on speaker clustering under the same clustering conditions. In this table, "traditional" stands for traditional acoustic features, and "LDA" for the proposed LDA transformed acoustic features.

Finally, to further demonstrate the statistical significance of our performance improvements, we perform a paired t-test (at the default 5% significance level) between our proposed method and the "Gaussian+BIC" method for every case. The p-values of all tests turn out to be so close to zero that Matlab rounds most of them to zero. This clearly indicates that the results of our method are significantly different from the results of the "Gaussian+BIC" method.

Next, we perform a step-by-step verification and discussion of the improvements that our partially supervised speaker clustering strategies have made to the experiment results, as follows:

In Tables 2–5, our experiment results show that our speaker clustering methods based on the GMM mean supervector representation and vector-based distance metrics significantly outperform traditional speaker clustering methods based on the "bag of acoustic features" representation and statistical model based distance metric such as the BIC (Rows: Orig vs. Baseline). It is worth mentioning that in the "Gaussian+BIC" method, if we utilize agglomerative clustering, the computational load can become prohibitive as the number of speakers gets larger. This is because at each iteration, along with a new cluster being formed by merging the two closest ones, a new statistical model representing the new cluster has to be retrained, and the distance between the new model and any other model updated. On the contrary, agglomerative clustering can be done very efficiently in the GMM mean supervector space by using the "ward" linkage. In addition, in the GMM mean supervector space, although speaker clustering based on the Euclidean distance metric achieves reasonably good results, the cosine distance metric consistently outperforms the Euclidean distance metric (Rows: COS/Orig vs. EU/Orig), thanks to the directional scattering property of the GMM mean supervectors, which is discussed in Section 3.2.3.

In Table 6, our experiment results show that the LDA transformed acoustic features consistently outperform the traditional acoustic features by 1%-3%. The proposed speaker-discriminative acoustic feature transformation implements one of our partially supervised speaker clustering strategies, which can provide a better frontend to speaker clustering as compared to traditional ones.

Due to the difficulty of handling high-dimensional data, and in order to alleviate the "curse of dimensionality", linear subspace learning methods are used to derive various subspaces in which the final speaker clustering is performed. From the experiment results in Tables 2–5, we see that the unsupervised methods (i.e. PCA/eigenface approach, LPP, NPE) more or less improve the performance, but significant improvements of the performance are achieved by

TABLE 2
Performance comparison of speaker clustering based on utterance-based clustering accuracies.

×100%		2 spk ∼ 60 utt		5 spk ∼ 150 utt		10 spk ∼ 300 utt		20 spk ∼ 600 utt		50 spk ∼ 1500 utt		100 spk ∼ 3000 utt	
		k	*h*	*k*	*h*	*k*	*h*	*k*	*h*	*k*	*h*	*k*	*h*
EU	Orig	94.7 (0.28)	96.0 (0.25)	81.6 (0.84)	85.0 (0.81)	77.3 (1.01)	82.6 (0.98)	70.5 (1.07)	78.1 (1.02)	58.4 (1.31)	69.4 (1.24)	47.2 (1.47)	57.7 (1.45)
	PCA	96.6 (0.26)	96.2 (0.26)	84.8 (0.77)	85.5 (0.74)	81.3 (0.82)	82.9 (0.84)	78.5 (1.02)	79.3 (1.00)	69.7 (1.26)	69.9 (1.28)	59.4 (1.43)	58.5 (1.45)
	LPP	98.3 (0.19)	98.1 (0.19)	92.5 (0.39)	92.8 (0.38)	87.9 (0.69)	88.6 (0.70)	85.6 (0.75)	84.7 (0.76)	77.4 (1.09)	77.0 (1.08)	70.1 (1.27)	69.8 (1.28)
	NPE	97.4 (0.20)	97.0 (0.18)	84.3 (0.78)	85.0 (0.74)	83.2 (0.81)	83.9 (0.82)	78.7 (1.01)	79.3 (1.00)	70.4 (1.25)	69.8 (1.23)	58.1 (1.39)	58.3 (1.45)
	LDA	98.3 (0.20)	98.4 (0.18)	94.1 (0.32)	94.0 (0.32)	89.9 (0.49)	90.8 (0.50)	87.2 (0.83)	86.6 (0.84)	79.5 (1.01)	79.6 (0.97)	73.1 (1.19)	72.3 (1.19)
	KLDA	98.2 (0.18)	98.1 (0.19)	93.0 (0.38)	92.8 (0.37)	87.3 (0.58)	87.5 (0.57)	84.6 (0.79)	83.9 (0.82)	75.2 (1.05)	75.3 (1.04)	70.4 (1.26)	70.1 (1.24)
COS	Orig	99.0 (0.13)	99.1 (0.13)	88.3 (0.51)	90.7 (0.51)	84.1 (0.73)	86.5 (0.72)	80.6 (0.99)	82.2 (0.98)	74.7 (1.07)	77.7 (1.09)	66.4 (1.30)	69.3 (1.26)
	LSDA	**99.2** (0.12)	**99.1** (0.12)	**97.8** (0.19)	**98.0** (0.20)	**95.0** (0.31)	**94.7** (0.32)	**90.3** (0.51)	**90.0** (0.50)	**84.3** (0.77)	**85.9** (0.77)	**77.9** (1.01)	**79.4** (1.00)
Baseline (Gaussian+BIC)		82.5 (0.84)	83.8 (0.85)	71.6 (1.19)	72.0 (1.21)	58.3 (1.44)	60.5 (1.42)	53.1 (1.59)	52.7 (1.47)	43.2 (1.72)	44.1 (1.69)	35.0 (1.87)	37.4 (1.91)

TABLE 3
Performance comparison of speaker clustering based on utterance-based NMIs.

×100%		2 spk ∼ 60 utt		5 spk ∼ 150 utt		10 spk ∼ 300 utt		20 spk ∼ 600 utt		50 spk ∼ 1500 utt		100 spk ∼ 3000 utt	
		k	*h*	*k*	*h*	*k*	*h*	*k*	*h*	*k*	*h*	*k*	*h*
EU	Orig	91.9 (0.41)	93.7 (0.37)	79.0 (0.96)	82.7 (0.97)	74.4 (1.11)	80.3 (1.05)	67.7 (1.16)	75.3 (1.09)	55.9 (1.27)	66.4 (1.46)	44.3 (1.45)	55.3 (1.36)
	PCA	93.9 (0.37)	93.5 (0.37)	82.7 (0.79)	83.0 (0.79)	78.9 (0.99)	79.9 (0.96)	76.2 (1.08)	76.7 (1.05)	66.9 (1.31)	67.6 (1.32)	56.9 (1.45)	56.3 (1.47)
	LPP	96.1 (0.25)	95.4 (0.25)	89.9 (0.61)	89.9 (0.61)	85.1 (0.83)	86.2 (0.84)	83.1 (0.77)	82.0 (0.78)	74.8 (1.09)	74.0 (1.09)	67.2 (1.32)	67.3 (1.27)
	NPE	95.3 (0.26)	94.3 (0.25)	81.6 (0.80)	82.3 (0.79)	81.1 (0.90)	81.3 (0.88)	76.0 (1.06)	77.1 (1.03)	67.8 (1.24)	67.0 (1.35)	55.7 (1.43)	55.9 (1.46)
	LDA	96.2 (0.25)	95.8 (0.25)	92.0 (0.41)	91.5 (0.41)	87.5 (0.64)	88.2 (0.63)	85.0 (0.83)	84.0 (0.83)	77.4 (1.04)	77.6 (1.09)	71.1 (1.27)	70.2 (1.28)
	KLDA	96.0 (0.25)	95.6 (0.26)	91.8 (0.40)	91.7 (0.43)	86.0 (0.63)	86.2 (0.63)	83.3 (0.90)	82.9 (0.88)	74.1 (1.11)	74.0 (1.09)	66.9 (1.30)	67.2 (1.33)
COS	Orig	96.6 (0.25)	96.8 (0.25)	86.0 (0.60)	88.0 (0.57)	81.4 (0.78)	83.6 (0.79)	78.2 (1.02)	79.2 (1.01)	72.1 (1.12)	75.2 (1.11)	63.8 (1.32)	67.1 (1.27)
	LSDA	**97.2** (0.16)	**97.0** (0.16)	**95.3** (0.22)	**95.3** (0.23)	**92.2** (0.38)	**92.6** (0.38)	**87.3** (0.60)	**87.9** (0.61)	**82.3** (0.85)	**83.2** (0.80)	**75.3** (1.07)	**77.4** (1.08)
Baseline (Gaussian+BIC)		79.9 (0.91)	81.5 (0.90)	69.3 (1.26)	69.5 (1.28)	56.2 (1.46)	58.4 (1.45)	50.9 (1.62)	50.4 (1.58)	41.0 (1.77)	41.5 (1.81)	32.2 (1.93)	35.2 (1.93)

supervised methods (i.e. LDA/fishervoice approach, LSDA). Notably, our proposed LSDA algorithm leads to the state-of-the-art speaker clustering performance. This clearly indicate that LSDA benefits significantly from the directional scattering property of the data in the GMM mean supervector space.

Owing to the non-linearity of the data, a kernel discriminant analysis method may be preferable over LDA. In Tables 2–5, we present the results obtained via kernel LDA [43] where the affinities used in LSDA are used as the Gram matrix in kernel LDA. Our experiment results show that kernel LDA is comparable to LDA when applied to smaller numbers of test speakers (2,5,10). When the number of test speakers gets larger, kernel LDA performs slightly worse than LDA (20,50,100). The reason behind this observation may be explained as follows: kernel LDA is based on an implicit nonlinear mapping from the original data space to a much higher dimensional feature space where linear projections are found. As the number of clusters increases, linear separation in the higher dimensional feature space increasingly comes at the cost of very small margins. In order to avoid small margins, the kernel LDA algorithm may choose suboptimal clustering strategies.

Also noted is that the frame-based performance is better than the utterance-based performance. The rationale behind this is that longer utterances tend to be more often correctly classified than shorter utterances, which is reasonable because longer utterances can provide more speaker-discriminative information than shorter ones.

In every experiment, we employ two clustering algorithms, namely k-means and agglomerative clus-

TABLE 4
Performance comparison of speaker clustering based on frame-based clustering accuracies.

×100%		2 spk ∼ 60 utt		5 spk ∼ 150 utt		10 spk ∼ 300 utt		20 spk ∼ 600 utt		50 spk ∼ 1500 utt		100 spk ∼ 3000 utt	
		k	h	k	h	k	h	k	h	k	h	k	h
EU	Orig	95.9 (0.24)	97.2 (0.22)	83.4 (0.77)	87.7 (0.75)	80.6 (0.91)	85.8 (0.78)	74.4 (1.01)	82.5 (0.99)	63.3 (1.19)	74.7 (1.17)	53.5 (1.38)	64.4 (1.31)
	PCA	97.4 (0.22)	97.7 (0.23)	86.2 (0.70)	86.6 (0.72)	84.4 (0.73)	85.7 (0.74)	81.9 (0.97)	83.1 (0.98)	74.5 (1.19)	74.8 (1.20)	66.1 (1.38)	67.2 (1.30)
	LPP	99.1 (0.13)	99.0 (0.13)	94.7 (0.31)	94.8 (0.31)	90.5 (0.56)	90.6 (0.56)	88.7 (0.60)	88.9 (0.58)	82.6 (0.97)	82.3 (0.89)	76.7 (1.19)	76.9 (1.20)
	NPE	98.8 (0.16)	98.5 (0.19)	87.1 (0.70)	87.7 (0.68)	86.1 (0.74)	86.7 (0.77)	82.5 (0.93)	83.6 (0.96)	74.9 (1.20)	74.7 (1.20)	64.5 (1.36)	64.7 (1.32)
	LDA	99.2 (0.13)	99.4 (0.12)	96.2 (0.26)	96.0 (0.26)	92.5 (0.38)	92.6 (0.40)	91.0 (0.63)	90.8 (0.64)	84.4 (0.89)	84.3 (0.87)	80.2 (1.06)	81.7 (1.13)
	KLDA	98.6 (0.19)	98.4 (0.19)	94.8 (0.35)	94.6 (0.34)	90.1 (0.50)	90.4 (0.54)	88.0 (0.64)	87.6 (0.63)	80.3 (1.04)	80.1 (1.02)	76.7 (1.26)	77.4 (1.20)
COS	Orig	99.4 (0.13)	99.5 (0.13)	89.9 (0.45)	92.5 (0.42)	87.3 (0.73)	88.1 (0.70)	85.1 (0.89)	86.5 (0.86)	79.3 (1.08)	81.2 (0.99)	72.3 (1.21)	74.9 (1.20)
	LSDA	**99.7** (0.10)	**99.5** (0.09)	**98.9** (0.17)	**99.2** (0.15)	**97.1** (0.22)	**96.8** (0.25)	**92.5** (0.44)	**92.1** (0.44)	**87.0** (0.68)	**88.2** (0.71)	**82.6** (0.93)	**83.2** (0.96)
Baseline (Gaussian+BIC)		84.7 (0.77)	85.1 (0.74)	72.9 (1.17)	73.4 (1.13)	61.6 (1.43)	63.2 (1.38)	57.7 (1.46)	56.9 (1.40)	48.6 (1.63)	49.4 (1.57)	42.7 (1.83)	44.1 (1.92)

TABLE 5
Performance comparison of speaker clustering based on frame-based NMIs.

×100%		2 spk ∼ 60 utt		5 spk ∼ 150 utt		10 spk ∼ 300 utt		20 spk ∼ 600 utt		50 spk ∼ 1500 utt		100 spk ∼ 3000 utt	
		k	h	k	h	k	h	k	h	k	h	k	h
EU	Orig	93.4 (0.36)	95.0 (0.31)	81.2 (0.95)	84.9 (0.90)	78.4 (1.00)	80.1 (0.93)	71.5 (1.08)	80.2 (1.01)	60.7 (1.27)	72.4 (1.38)	51.0 (1.39)	61.6 (1.36)
	PCA	95.0 (0.31)	95.6 (0.30)	83.5 (0.77)	84.5 (0.76)	81.9 (0.87)	82.7 (0.88)	78.9 (0.98)	80.1 (1.02)	72.2 (1.24)	72.1 (1.35)	63.4 (1.45)	65.1 (1.40)
	LPP	96.8 (0.24)	96.8 (0.26)	92.4 (0.47)	92.3 (0.52)	87.6 (0.74)	88.2 (0.67)	86.2 (0.78)	86.9 (0.79)	80.1 (1.04)	80.1 (0.98)	74.0 (1.16)	74.6 (1.22)
	NPE	96.2 (0.25)	95.7 (0.29)	84.7 (0.78)	85.0 (0.76)	83.5 (0.84)	84.6 (0.83)	80.2 (0.91)	80.8 (1.01)	72.0 (1.26)	72.5 (1.23)	61.8 (1.42)	62.0 (1.45)
	LDA	97.0 (0.22)	96.5 (0.23)	93.6 (0.37)	93.2 (0.37)	89.5 (0.59)	90.4 (0.57)	88.7 (0.72)	88.0 (0.76)	81.6 (1.04)	81.7 (0.99)	77.2 (1.13)	79.2 (1.16)
	KLDA	96.6 (0.25)	96.1 (0.25)	93.0 (0.38)	92.8 (0.38)	87.1 (0.58)	88.0 (0.63)	85.2 (0.86)	84.9 (0.87)	77.1 (1.06)	77.1 (1.09)	74.3 (1.29)	75.4 (1.23)
COS	Orig	97.1 (0.22)	97.0 (0.21)	87.8 (0.61)	89.9 (0.58)	84.7 (0.68)	85.8 (0.68)	82.2 (0.93)	84.4 (0.95)	76.8 (1.06)	78.7 (1.06)	69.5 (1.30)	72.6 (1.25)
	LSDA	**97.4** (0.16)	**97.3** (0.15)	**96.1** (0.22)	**96.5** (0.22)	**94.9** (0.36)	**94.5** (0.37)	**89.9** (0.59)	**89.4** (0.62)	**84.5** (0.83)	**86.2** (0.82)	**79.8** (1.07)	**81.0** (1.10)
Baseline (Gaussian+BIC)		82.5 (0.85)	82.8 (0.82)	70.2 (1.23)	71.3 (1.23)	59.4 (1.39)	60.7 (1.46)	55.4 (1.52)	54.1 (1.54)	46.3 (1.71)	47.0 (1.64)	40.2 (1.78)	41.7 (1.85)

tering. Although there are pros and cons in each algorithm, we observe that in the same subspace, the speaker clustering performance of the two algorithms is comparable. However, k-means is sensitive to initialization, which means the results across multiple runs may not be identical. Thus, we need to restart the k-means algorithm many times (e.g., 50) with a different initialization at each time, and record the best result. Therefore, k-means is normally much slower than agglomerative clustering. On the other hand, agglomerative clustering with the "ward" linkage method runs very fast. For the case of 100 speakers (about 3000 utterances), it takes our Matlab program less than one minute to complete the job on a Linux machine with a mainstream configuration.

An issue that is not addressed in this paper is the determination of the number of speakers. Automatically finding the number of clusters in a dataset in a completely unsupervised manner is still an open research problem. Many speaker diarization systems deal with this problem through hierarchical clustering using a BIC-based stopping criterion [11]. A similar method could have been used to determine the number of speakers automatically in our paper. However, the exact number of speakers is not accurately computed by this simple method. In general, clustering results may vary dramatically for different numbers of speakers determined. In order to eliminate the influence of the number of speakers and single out the extent to which the proposed partially supervised strategies may improve the speaker clustering performance, we assume that the number of test speakers is known a priori, and defer the investigation of this issue to our future work.

TABLE 6
Performance comparison of the proposed LDA transformed acoustic features with the traditional acoustic features based on clustering accuracies.

×100%		2 spk ∼ 60 utt		5 spk ∼ 150 utt		10 spk ∼ 300 utt		20 spk ∼ 600 utt		50 spk ∼ 1500 utt		100 spk ∼ 3000 utt	
		k	h	k	h	k	h	k	h	k	h	k	h
EU	Traditional	93.5 (0.28)	94.7 (0.28)	80.1 (0.85)	83.7 (0.85)	76.2 (1.07)	81.3 (1.03)	69.0 (1.10)	76.6 (1.04)	57.5 (1.29)	68.4 (1.27)	46.0 (1.55)	56.6 (1.47)
	LDA	**94.7** (0.29)	**96.0** (0.25)	**81.6** (0.82)	**85.0** (0.79)	**77.3** (1.03)	**82.6** (0.99)	**70.5** (1.05)	**78.1** (1.04)	**58.4** (1.31)	**69.4** (1.27)	**47.2** (1.51)	**57.7** (1.41)
COS	Traditional	97.4 (0.16)	97.7 (0.16)	87.3 (0.56)	89.4 (0.58)	82.9 (0.77)	85.4 (0.80)	79.6 (1.02)	81.2 (1.02)	73.5 (1.10)	76.4 (1.10)	64.9 (1.34)	67.8 (1.30)
	LDA	**99.0** (0.13)	**99.1** (0.12)	**88.3** (0.49)	**90.7** (0.51)	**84.1** (0.71)	**86.5** (0.72)	**80.6** (1.00)	**82.2** (1.00)	**74.7** (1.06)	**77.7** (1.09)	**66.4** (1.25)	**69.3** (1.28)

7 Conclusions

In this paper, we propose the conceptually new idea of partially supervised speaker clustering and offer a complete treatment of the speaker clustering problem. By means of an independent training data set, our strategies are to encode the prior knowledge of speakers in general at the various stages of the speaker clustering pipeline via 1) learning a speaker-discriminative acoustic feature transformation, 2) learning a universal speaker prior model, and 3) learning a discriminative speaker subspace, or equivalently, a speaker-discriminative distance metric. We discover the directional scattering property of the GMM mean supervector representation of utterances and advocate the use of the cosine distance metric instead of the Euclidean distance metric. We propose to perform discriminant analysis based on the cosine distance metric, leading to a novel distance metric learning algorithm – LSDA. We show that the proposed LSDA formulation can be systematically solved within the elegant graph embedding framework. Our speaker clustering experiments indicate that 1) our speaker clustering methods based on the GMM mean supervector representation and vector-based distance metrics outperform traditional methods based on the "bag of acoustic features" representation and statistical model based distance metrics, 2) our advocated use of the cosine distance metric yields consistent increases in the speaker clustering performance as compared to the commonly used Euclidean distance metric, thanks to the directional scattering property of the GMM mean supervectors discovered, 3) our partially supervised speaker clustering concept and strategies significantly improve the speaker clustering performance over the baselines, and 4) our proposed LSDA algorithm further leads to the state-of-the-art speaker clustering performance.

Acknowledgments

This work was supported in part by the DARPA contract HR0011-06-2-0001 and in part by the NSF Grant 08-03219. The authors would like to thank Dr. Lidia Mangu and Dr. Michael Picheny at the IBM T. J. Watson Research Center for their continuous encouragement and support.

References

[1] M. Lew, N. Sebe, C. Djeraba, R. Jain. Content-based multimedia information retrieval: State of the art and challenges. ACM Trans. on Multimedia Computing, Communications, and Applications, 2(1):1-19, 2006.
[2] Y. Gong, W. Xu. Machine Learning for Multimedia Content Analysis. Springer, 2007.
[3] H. Jin, F. Kubala, and R. Schwartz, "Automatic speaker clustering," Proc. DARPA Speech Recognition Workshop'97.
[4] S. Chen, and P. Gopalakrishnan, "Clustering via the Bayesian information criterion with applications in speech recognition," Proc. ICASSP'98.
[5] D. Reynolds, E. Singer, B. Carson, G. O'Leary, J. McLaughlin, and M. Zissman, "Blind clustering of speech utterances based on speaker and language characteristics," Proc. ICSLP'98.
[6] A. Solomonoff, A. Mielke, M. Schmidt, H. Gish. Clustering speakers by their voices. Proc. ICASSP, 2:757-760, 1998.
[7] R. Faltlhauser, and G. Ruske, "Robust speaker clustering in eigenspace," Proc. ASRU'01.
[8] W. Tsai, S. Cheng, and H. Wang, "Speaker clustering of speech utterances using a voice characteristic reference space," Proc. ICSLP'04.
[9] M. Ben, M. Betser, F. Bimbot, and G. Gravier. Speaker diarization using bottom-up clustering based on a parameter-derived distance between adapted GMMs. Proc. ICSLP, 2004.
[10] Barras, C., Zhu, X., Meignier, S. and Gauvain, J.L., "Multistage speaker diarization of broadcast news," IEEE Trans. ASLP. vol. 14 no. 5, pp. 1505-1512, Sept. 2006.
[11] S. Tranter, D. Reynolds. An Overview of Automatic Speaker Diarization Systems. IEEE T-ASLP, 14(5):1557-565, 2006.
[12] C. Wooters and M. Huijbregts, "The ICSI RT07s Speaker Diarization System," Lecture Notes in Computer Science, 2007.
[13] P. Kenny. Bayesian analysis of speaker diarization with eigenvoice priors. CRIM, 2008.
[14] D. Reynolds, P. Kenny, F. Castaldo. A study of new approaches to speaker diarization. Interspeech, 2009.
[15] R. Duda, P. Hart, D. Stork. Pattern Classification (2nd ed.). Wiley Interscience, 2000.
[16] W. Campbell,D. Sturim, D. Reynolds. Support vector machines using GMM supervectors for speaker verification. Signal Processing Letters 13(5):308-311, 2006.
[17] D. Reynolds, T. Quatieri, R. Dunn. Speaker Verification using Adapted Gaussian Mixture Models. Digital Signal Processing, 10:19-41, 2000.
[18] J. MacQueen. Some Methods for classification and Analysis of Multivariate Observations. Proc. 5th Berkeley Symposium on Math. Stat. and Prob. 1:281-297, 1967.
[19] A. Jain, R. Dubes. Algorithms for clustering data. Prentice-Hall, Inc., 1988.
[20] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, S. Lin. Graph Embedding and Extensions: A General Framework for Dimensionality Reduction. IEEE T-PAMI 29(1):40-51, 2007.

[21] H. Hermansky. Perceptual linear predictive (PLP) analysis of speech. J. Acoust. Soc. Amer., 87:1738-1752, 1990.
[22] G. Fant. Acoustic Theory of Speech Production. Mouton De Gruyter, 1970.
[23] P. Jain, H. Hermansky. Improved mean and variance normalization for robust speech recognition. Proc. ICASSP, 2001.
[24] A. Dempster, N. Laird, D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Stat. Soc. B, 39(1):1-38, 1977.
[25] C. Bishop. Pattern Recognition and Machine Learning. Springer, 2006.
[26] P. Kenny, G. Boulianne, and P. Dumouchel. Eigenvoice modeling with sparse training data. IEEE Trans. Speech and Audio Processing, 13(3):345–354, 2005.
[27] Douglas A. Reynolds and Richard C. Rose. Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models. IEEE Trans. Speech and Audio Processing, 3(1):72–83, 1995.
[28] J-L. Gauvain, C.-H. Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. IEEE Trans. Speech and Audio Processing, 2:291–298, 2004.
[29] G. Schwarz. Estimating the dimension of a model. Annals of Statistics 6(2):461-464, 1978.
[30] X. Miro. Robust Speaker Diarization for Meetings. PhD thesis, Universitat Politecnica de Catalunya, 2006.
[31] P. Mahalanobis. On the generalised distance in statistics. National Institute of Sciences of India 2(1):49-55, 1936.
[32] R. Kuhn, P. Nguyen, J. Junqua, L. Goldwasser, N. Niedzielski, S. Fincke, K. Field, M. Contolini. Eigenvoices for speaker adaptation. Proc. ICSLP, 1771-1774, 1998.
[33] X. He, P. Niyogi. Locality Preserving Projections. NIPS 2003.
[34] X. He, D. Cai, S. Yan, H. Zhang. Neighborhood Preserving Embedding. Proc. ICCV, 2005.
[35] S. Chu, H. Tang, T. Huang. Fishervoice and Semi-supervised Speaker Clustering. Proc. ICASSP, 2009.
[36] Y. Ma, S. Lao, E. Takikawa, M. Kawade. Discriminant Analysis in Correlation Similarity Measure Space. Proc. ICML, 227:577-84, 2007.
[37] Y. Fu, S. Yan, T. Huang. Correlation Metric for Generalized Feature Extraction. IEEE T-PAMI 30(12):2229-235, 2008.
[38] I. Dhillon, D. Modha. Concept decompositions for large sparse text data using clustering. Machine Learning, 42(1):143-75, 2001.
[39] A. Jain. Data Clustering: 50 Years Beyond K-Means. Technical Report TR-CSE-09-11. Under review by the Pattern Recognition Letters.
[40] S. Chu, H. Kuo, L. Mangu, Y. Liu, Y. Qin, Q. Shi. Recent advances in the IBM GALE mandarin transcription system. Proc. ICASSP, 2008.
[41] P. Viola, W. Wells III. Alignment by maximization of mutual information. Int. J. Comput. Vision, 24(2):137-154, 1997.
[42] J. Moore, E.-H. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, B. Mobasher. Web Page Categorization and Feature Selection Using Association Rule and Principal Component Clustering. Workshop on Information Technologies and Systems, 1997.
[43] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, K. Mullers. Fisher discriminant analysis with kernels. IEEE Neural Networks for Signal Processing Workshop, pp. 41-48, 1999.

Hao Tang received the Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign in 2010, the M.S. degree in electrical engineering from Rutgers University in 2005, the M.E. and B.E. degrees, both in electrical engineering, from the University of Science and Technology of China, in 2003 and 1998, respectively. Since 2010, he has been with HP Labs. His broad research interests include statistical pattern recognition, machine learning, computer vision, speech and multimedia signal processing.

Stephen Mingyu Chu was born in Beijing, China, in 1970. He studied Physics at Peking University before coming to the United States, where he continued his education and received the M.S. degree and the Ph.D. degree in Electrical Engineering from the University of Illinois at Urbana-Champaign, in 1999 and 2003, respectively. Since May, 2003, he has been with the speech group at the IBM T. J. Watson Research Center. His research interests include speech recognition, multimodal signal processing, graphical models, and machine learning.

Mark Hasegawa-Johnson (M'97-SM'04) received the S.B., S.M., and Ph.D. degrees in electrical engineering and computer science from MIT in 1989, 1989, and 1996 respectively. From 1986-1990 he was a speech coding engineer, first at Motorola Labs, Schaumburg, IL, then at Fujitsu Laboratories Limited, Kawasaki, Japan. From 1996-9 he was a post-doctoral fellow at the Speech Processing and Auditory Physiology Lab, UCLA, funded by the NIH, and by the 19th annual F.V. Hunt Post-Doctoral Fellowship of the Acoustical Society of America. Since 1999, Prof. Hasegawa-Johnson has been on the faculty of the University of Illinois. He is currently an Associate Professor in the Department of Electrical and Computer Engineering, with Affiliate appointments in Speech and Hearing Sciences, Computer Science, and Linguistics, and with a research appointment at the Beckman Institute for Advanced Science and Technology.

Thomas Huang (S'61-M'63-SM'76-F'79-LF'01) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, China, and the M.S. and Sc.D. degrees in electrical engineering from the Massachusetts Institute of Technology (MIT), Cambridge. He was at the faculty of the Department of Electrical Engineering at MIT from 1963 to 1973, and at the faculty of the School of Electrical Engineering and Director of its Laboratory for Information and Signal Processing at Purdue University from 1973 to 1980. In 1980, he joined the University of Illinois at Urbana-Champaign, where he is now William L. Everitt Distinguished Professor of Electrical and Computer Engineering, Research Professor at the Coordinated Science Laboratory, Head of the Image Formation and Processing Group at the Beckman Institute for Advanced Science and Technology, and Co-chair of the Institute's major research theme: human-computer intelligent interaction. Dr. Huang's professional interests lie in the broad area of information technology, especially the transmission and processing of multidimensional signals. He is a Member of the National Academy of Engineering, a Foreign Member of the Chinese Academies of Engineering and Science, and a Fellow of the International Association of Pattern Recognition, IEEE, and the Optical Society of America.