A Study of the Cosine Distance-Based Mean Shift for Telephone Speech Diarization

Article *in* IEEE/ACM Transactions on Audio, Speech, and Language Processing · January 2014

DOI: 10.1109/TASLP.2013.2285474

4 authors, including:

Mohammed Senoussaoui
Institut National de la Recherche Scientifique
21 PUBLICATIONS **509** CITATIONS

SEE PROFILE

Patrick Kenny
Centre de recherche informatique de Montréal
143 PUBLICATIONS **9,252** CITATIONS

SEE PROFILE

Pierre Dumouchel
École de Technologie Supérieure
104 PUBLICATIONS **6,588** CITATIONS

SEE PROFILE

A Study of the Cosine Distance-Based Mean Shift for Telephone Speech Diarization

Mohammed Senoussaoui, Patrick Kenny, Themos Stafylakis and Pierre Dumouchel

Abstract—**Speaker clustering is a crucial step for speaker diarization. The short duration of speech segments in telephone speech dialogue and the absence of prior information on the number of clusters dramatically increase the difficulty of this problem in diarizing spontaneous telephone speech conversations. We propose a simple iterative Mean Shift algorithm based on the cosine distance to perform speaker clustering under these conditions. Two variants of the cosine distance Mean Shift are compared in an exhaustive practical study. We report state of the art results as measured by the Diarization Error Rate and the Number of Detected Speakers on the LDC CallHome telephone corpus.**

***Index Terms*— Speaker diarization, clustering, Mean Shift, cosine distance.**

I. Introduction

SPEAKER diarization consists in splitting an audio stream into homogeneous regions corresponding to speech of participating speakers. As the problem is usually formulated, diarization requires performing two principal steps, namely *segmentation* and *speaker clustering*. The aim of segmentation is to find *speaker change points* in order to form segments (known as speaker turns) that contain speech of a given speaker. The aim of speaker clustering is to link unlabeled segments according to a given metric in order to determine the intrinsic grouping in data. The challenge of speaker clustering increases by virtue of the absence of any prior knowledge about the constituent number of speakers in the stream.

Model selection based on the Bayesian information criterion (BIC) is the most popular method for speaker segmentation [1][2]. BIC can also be used to estimate the number of speakers in a recording and other Bayesian methods have recently been proposed for this purpose [3][4]. Hierarchical Agglomerative Clustering (HAC) is by far the most widespread approach to the speaker clustering problem. Other methods, including hybrid approaches continue to be developed [4] [5].

In this work, we focus on the speaker clustering task rather than speaker segmentation. We propose a clustering method which is capable of estimating the number of speakers participating in a telephone conversation, a challenging problem considering that speaker turns are generally of very short duration [4]. The method in question is the so-called Mean Shift (MS) algorithm. This approach is borrowed from the field of computer vision where it is widely used to detect the number of colors and for image segmentation purposes. The MS algorithm is a nonparametric iterative mode-seeking algorithm introduced by Fukunaga [6]. Despite its first appearance in 1975, MS remained in oblivion except for works such as [7] that aimed to generalize the original version. The MS algorithm reappeared in 2002 with the work of Comaniciu [8] in image processing. Recently, Stafylakis *et al.* [9][10] has shown how to generalize the basic Euclidean space MS algorithm to non-Euclidean manifolds so that objects other than points in Euclidean space can be clustered. This generalized method was applied to the problem of speaker clustering in a context where speaker turns were characterized by multivariate Gaussian distributions.

Our choice of the MS algorithm is mainly motivated by its nonparametric nature. This characteristic offers the major advantage of not having to make assumptions about the shape of data distribution, in contrast to conventional probabilistic diarization methods.

Recently [11], we presented a new extension of the Euclidean Mean Shift that works with a cosine distance metric. This new algorithm was shown to be very effective for speaker clustering in large populations where each speaker was represented by a whole side of a telephone conversation. This work was motivated by the success of cosine similarity matching in the speaker verification field [12][13][14][15].

Cosine distance has also been successfully tested in speaker diarization of the CallHome telephone corpus [16][17]. In this work, firstly, we propose to test the cosine-based MS algorithm on the diarization of multi-speaker 2-wire telephone recordings. (We do *not* assume that the number of participating speakers is given.) Secondly, we compare two clustering mechanisms that exploit the cosine-based MS algorithm with respect to diarization performance (as measured by the number of speakers detected and the standard diarization error metric) and with respect to execution times.

Although diarization on telephone conversations is an important and difficult task, there are not, to our knowledge, any published studies on the use of the MS algorithm to solve this problem. Unlike broadcast news speech, the shortness of

Manuscript received June 21, 2013, revised August 31, 2013 and accepted September 24, 2013. This work was supported by the Natural Science and Engineering Research Council of Canada.

M. Senoussaoui is with Centre de recherche informatique de Montréal (CRIM), Montréal, Qc, H3A 1B9, Canada and with École de technologie supérieure (ÉTS), Montréal, QC, Canada (e-mail: mohammed.senoussaoui@crim.ca)

P. Kenny and T. Stafylakis are with Centre de recherche informatique de Montréal (CRIM), Montréal, QC, H3A 1B9, Canada (e-mail: patrick.kenny@crim.ca; themos.stafylakis@crim.ca)

P. Dumouchel is with École de technologie supérieure (ÉTS), Montréal, QC, Canada (e-mail: pierre.dumouchel@etsmtl.ca)

the speaker turn duration in telephone speech (typically one second) makes the task of properly representing these segments in a feature space more difficult. In order to deal with this problem, we represent each speaker turn by an i-vector (a representation of speech segments by vectors of fixed dimension, independent of segment durations [12]).

I-vector features have been used successfully not only in speaker recognition [12][13][14][15] and speaker diarization and clustering [16][17][21][11] but also in language recognition [22][23]. Although probabilistic classifiers such as Probabilistic Linear Discriminant Analysis have become predominant in applying i-vector methods to speaker recognition, simple cosine distance based classifiers remain competitive [12][13] and we will use this approach in developing the speaker diarization algorithms presented here. (Note that in [24], the authors show that cosine distance provides a better metric than Euclidean distance in GMM-supervector space.) In [16], the authors introduced a diarization system where i-vectors were used to represent speaker turns and cosine distance based k-means clustering was used to associate speaker turns with individual speakers. Tested on two-speaker conversations, this approach outperformed a BIC-based hierarchical agglomerative clustering system by a wide margin. But, in order for k-means clustering to work, the number of speakers in a given conversation needs to be known in advance, so it is not straightforward to extend this approach to the general diarization problem where the number of speakers participating in the conversation needs to be determined (a simple heuristic is presented in [17]). The main contribution of this paper is to show how using the Mean Shift algorithm in place of k-means enables this problem to be dealt with very effectively.

As our test bed we use the CallHome telephone speech corpus (development/test) provided by NIST in the year 2000 speaker recognition evaluation (SRE). This consists of spontaneous telephone conversations involving varying numbers of speakers. The CallHome dataset has been the subject of several studies [17][18][19][20][21].

The rest of this paper is organized as follows. In Section II we first provide some background material on the i-vector feature space that will be used in this work. In Section III, we give some preliminaries on the original version of the Mean Shift algorithm and explain how we include the cosine distance by introducing a simple modification. Two ways of exploiting the MS algorithm for clustering purposes will also be given. In Section IV, we present different methods of normalizing i-vectors for diarization (such normalizations turn out to be very important for our approach). Thereafter, we perform a detailed experimental study and analysis in Sections V and VI before concluding this work in Section VII.

II. I-VECTORS FOR SPEAKER DIARIZATION

The *supervector* representation has been applied with great success to the field of speaker recognition, especially when it was exploited in the well-known generative model named Joint Factor Analysis (JFA) [25]. In high-dimensional *supervector* space, JFA attempts to jointly model speaker and channel variabilities using a large amount of background data. When a relatively small amount of speaker data is available (i.e. during enrolment and test stages), JFA enables effective speaker modeling by suppressing channel variability from the speech signal.

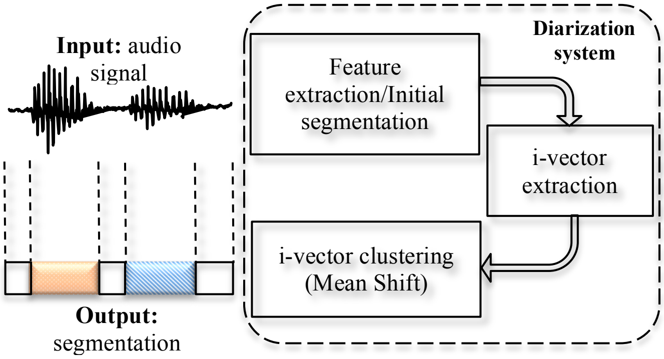

Fig. 1. Skeleton of the Mean shift i-vector diarization system: Segmentation of the speech signal is followed by extracting i-vectors for each segment and then the i-vectors are clustered (using Mean Shift in our case).

A major advance in the area of speaker recognition was the introduction of the low dimensional feature vectors known as i-vectors [12]. We can define an i-vector as the mapping (using a Factor Analysis or a Probabilistic Principal Component Analysis) of a high-dimensional supervector to a low-dimensional space called total variability space (here the word *total* is used to refer to both speaker and channel variabilities). Unlike JFA that proposes to distinguish between speaker and channel effects in the supervector space, i-vector methods seek to model these effects in a low dimensional space where many standard pattern recognition methods can be brought to bear on the problem.

Mathematically, the mapping of a *supervector* **X** to an i-vector **x** is expressed by the following formula:

$$\mathbf{X} = \mathbf{X}_{UBM} + \mathbf{T}\mathbf{x}. \tag{1}$$

where $\mathbf{X}_{UBM}$ is the supervector of the Universal Background Model (UBM) and the rectangular matrix **T** is the so-called Total Variability matrix. More mathematical details of i-vectors and their estimation can be found in [12][25][26].

I-vectors have successfully been deployed in many fields other than speaker recognition [16][17][21][22][23]. Methods successful in one field can often be translated to other fields by identifying the sources of “useful” and “nuisance” variability. Thus in speaker recognition, speaker variability is useful but it counts as nuisance variability in language recognition.

In the diarization problem, the speaker turn (represented by an i-vector in our case) is the fundamental representation unit or what we usually call a *sample* in Pattern Recognition terminology. Moreover, an aggregation of homogenous i-vectors within one conversation represents a cluster (speaker in our case) or what is commonly known as a *class*. Thus, the diarization problem becomes one of clustering i-vectors [11][16][17][21].

III. The Mean Shift Algorithm

The Mean Shift algorithm can be viewed as a clustering algorithm or as a way of finding the modes in a non-parametric distribution. In this section we will present the intuitive idea behind the Mean Shift mode-seeking process as well as the mathematical derivations of this algorithm. Additionally, we present two variants of this algorithm which can be applied for clustering purposes. Finally, the extension of the traditional MS to the cosine-based MS is presented.

A. The intuitive idea behind Mean Shift

The intuitive idea of Mean Shift is quite natural and simple. Starting from a given vector $\mathbf{x}_i$ in a set $S=\{\mathbf{x}_1,\mathbf{x}_2,...,\mathbf{x}_n\}$ of unlabeled data (which are i-vectors in our case) we can reach a stationary point called a *density mode* through the iterative process depicted in Algorithm 1. Note that the Algorithm 1 refers to the original Mean Shift process.

The mathematical convergence proof of the sequence of successive positions $\{\mathbf{y}_i\}_{i=1,2...}$ is found in [6][8].

Algorithm 1 *Mean Shift – Intuition idea*

```
- $i=1$, $\mathbf{y}_i = \mathbf{x}_i$
- Center a window around $\mathbf{y}_i$ // Initialization
repeat
    - $\mu_h(\mathbf{y}_i)$ // estimate the sample mean of data falling
      within the window (i.e. neighborhood of $\mathbf{y}_i$ in
      terms of Euclidean distance)
    - $\mathbf{y}_{i+1} = \mu_h(\mathbf{y}_i)$
    - Move the window from $\mathbf{x}_i$ to $\mathbf{y}_{i+1}$
    - $i = i+1$
until Stabilization // a mode has been found
```

B. Mathematical development

Mean Shift is a member of the Kernel Density Estimation (KDE) family of algorithms (also known as Parzen windowing). Estimating the probability density function of a distribution using a limited sample of data is a fundamental problem in pattern recognition. The standard form of the estimated kernel density function $\hat{f}(\mathbf{x})$ at a randomly selected point x is given by the following formula[1]:

$$\hat{f}(\mathbf{x}) = \frac{1}{nh^d}\sum_{i=1}^{n} k\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right) \tag{2}$$

where $k(\mathbf{x})$ is a kernel function and h is its radial width, referred to as the kernel bandwidth. Ignoring the selection of kernel type, h is the only tunable parameter in the Mean shift algorithm; its role is to smooth the estimated density function. In order to ensure certain properties such as asymptotic unbiasedness and consistency, the kernel and bandwidth h should satisfy some conditions that are discussed in detail in [6].

In general, the purpose of KDE is to estimate the density function but the Mean shift procedure is only concerned with locating the modes of the density function $f(\mathbf{x})$ (and not the values of the density function at these points). To find the modes, the Mean shift algorithm derivation requires calculating the gradient of the density function $f(\mathbf{x})$. The estimate of the gradient of the density function $f(\mathbf{x})$ is given by the gradient of the estimate of the density function $\hat{f}(\mathbf{x})$ as follows [6][8][27][28][29]:

$$\begin{aligned}\hat{\nabla} f(\mathbf{x}) \equiv \nabla\hat{f}(\mathbf{x}) &= \frac{1}{nh^d}\sum_{i=1}^{n}\nabla k\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)\\ &= \frac{2}{nh^{d+2}}\sum_{i=1}^{n}(\mathbf{x}-\mathbf{x}_i)k'\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right).\end{aligned} \tag{3}$$

A simple type of kernel is the Epanechnikov kernel given by the following formula:

$$k(\mathbf{x}) = \begin{cases} 1-\|\mathbf{x}\|^2 & \|\mathbf{x}\|\le 1 \\ 0 & \|\mathbf{x}\|>1 \end{cases}. \tag{4}$$

Let $g(\mathbf{x})$ be the uniform kernel:

$$g(\mathbf{x}) = \begin{cases} 1 & \|\mathbf{x}\|\le 1 \\ 0 & \|\mathbf{x}\|>1 \end{cases}. \tag{5}$$

Note that it satisfies:

$$k'(\mathbf{x}) = -c\ g(\mathbf{x}) \tag{6}$$

where c is a constant and the prime is the derivation operator. Then we can write $\nabla\hat{f}(\mathbf{x})$ as:

$$\begin{aligned}\nabla\hat{f}(\mathbf{x}) &= \frac{2}{nh^{d+2}}\sum_{i=1}^{n}(\mathbf{x}_i-\mathbf{x})g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)\\ &= \frac{2}{nh^{d+2}}\left[\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)\right]\left[\frac{\sum_{i=1}^{n}\mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}-\mathbf{x}\right].\end{aligned} \tag{7}$$

The expression:

$$m_h(\mathbf{x}) = \frac{\sum_{i=1}^{n}\mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}-\mathbf{x} \tag{8}$$

is what we refer to as the Mean Shift vector $m_h(\mathbf{x})$.

Note that the Mean Shift vector $m_h(\mathbf{x})$ is just the difference of the current position (instance vector $\mathbf{x}$) from the next position presented by the weighed sample mean vector of all data. Indeed, the weights in the mean formula are given by the binary outputs (i.e. 0 or 1) of the flat kernel $g(\mathbf{x})$.

[1] Note that for simplicity we ignore some constants in the mathematical derivations.

For simplicity, let us denote the uniform kernel with bandwidth h by $g(\mathbf{x},\mathbf{x}_i,h)$ so that:

$$g(\mathbf{x},\mathbf{x}_i,h)=\begin{cases}1 & \|\mathbf{x}-\mathbf{x}_i\|^2 \le h^2\\ 0 & \|\mathbf{x}-\mathbf{x}_i\|^2 > h^2\end{cases}. \tag{9}$$

In other words, $g(\mathbf{x},\mathbf{x}_i,h)$ selects a subset $S_h(\mathbf{x})$ of n_x samples (by analogy with Parzen windows we refer to this subset as a window) in which the Euclidean pairwise distances with $\mathbf{x}$ are less or equal to the threshold (bandwidth) h:

$$S_h(\mathbf{x}) \equiv \{\mathbf{x}_i : \|\mathbf{x}_i-\mathbf{x}\| \le h\}. \tag{10}$$

Therefore, we can rewrite the Mean Shift vector as:

$$m_h(\mathbf{x}) = \mu_h(\mathbf{x}) - \mathbf{x} \tag{11}$$

where $\mu_h(\mathbf{x})$ is the sample mean of the n_x samples of $S_h(\mathbf{x})$:

$$\mu_h(\mathbf{x}) = \frac{1}{n_x}\sum_{\mathbf{x}_i\in S_h(\mathbf{x})} \mathbf{x}_i . \tag{12}$$

The iterative processing of calculating the sample mean followed by data shifting (which produces the sequence $\{\mathbf{y}_i\}_{i=1,2...}$ referred to in Algorithm1) converges to a mode of the data distribution.

C. Mean Shift for speaker clustering

The Mean Shift algorithm can be exploited to deal with the problem of speaker clustering in the case where the number of clusters (speakers in our case) is unknown, as well as other problems such as the segmentation steps involved in image processing and object tracking [8]. In the following subsections, we present two clustering mechanisms based on the MS algorithm, namely, the *Full* and the *Selective* clustering strategies.

Full strategy

One may apply the iterative Mean Shift procedure at each data instance. In general, some of the MS processes will converge to the same density mode. The number of density modes (after pruning) represents the number of detected clusters and instances that converge to the same mode are deemed to belong to the same cluster (we call these points the *basin of attraction* of the mode). In this work we refer to this approach as *Full* strategy.

Selective strategy

Unlike the *Full* Mean Shift clustering strategy, we can adapt this strategy to run the MS process on a subset of data only. The idea is to keep track of the number of visits to each data point that occurs during the evolution of a Mean Shift process. After the convergence of the first Mean Shift process the samples that have been visited are assigned to the first cluster. We then run a second process starting from one of the unvisited samples and create a second cluster. We continue to run MS processes one after another until we have no unvisited data samples. Some of the samples may be allocated to more than one cluster by this procedure then majority voting is needed to reconcile these conflicts.

Note that the computational complexity depends on the number of samples in the *Full* strategy and it depends only on the number of clusters in the case of the *Selective* strategy. A MATLAB implementation of the *Selective* strategy can be found online.[2]

In this work the experimental results of the *Full* and *Selective* clustering strategies are compared in Section VI.

D. Mean Shift- based on cosine distance

The success of the cosine distance in speaker recognition is well known [12][13][14][15]. A rationale for using cosine distance instead of Euclidean distance can be supplied by postulating a normal distribution for the speaker population (as in PLDA [30]). Suppose we are given a pair of i-vectors and we wish to test the hypothesis that they belong to the same speaker cluster against the hypothesis that they belong to different clusters. Because most of the population mass is concentrated in the neighborhood of the origin, speakers in this region are in danger of being confused with each other. In the case of a pair of i-vectors which are close to the origin, the same speaker hypothesis will only be accepted if the i-vectors are relatively close together. On the other hand, if the i-vectors are far from the origin, they can be relatively far apart from each other without invalidating the same speaker hypothesis. Hence, in order to incorporate this prior knowledge regarding the distribution of the speaker means into the MS algorithm, we may either use (a) the Euclidean distance and a variable bandwidth *that increases with the distance from the origin* or (b) fixed bandwidth and the cosine similarity. The latter approach is evidently preferable.

The cosine distance between two vectors $\mathbf{x}$ and $\mathbf{y}$ is given by:

$$D(\mathbf{x},\mathbf{y}) = 1-\left(\frac{\mathbf{x}\cdot\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}\right). \tag{13}$$

The original Mean Shift algorithm based on a flat kernel relies on the Euclidean distance to find points falling within the window as shown in (10). In [11] we proposed the use of the cosine metric instead of the Euclidean one to build a new version of the Mean shift algorithm. Only one modification needs be introduced in (10); we set

$$S_h(\mathbf{x}) \equiv \{\mathbf{x}_i : D(\mathbf{x},\mathbf{x}_i) \le h\} \tag{14}$$

where $D(\mathbf{x},\mathbf{x}_i)$ is the cosine distance between $\mathbf{x}_i$ and $\mathbf{x}$ given by the formula (13). This corresponds to redefining the uniform kernel as:

[2] http://www.mathworks.com/matlabcentral/fileexchange/authors/22444

$$g\left(\mathbf{x},\mathbf{x}_i,h\right)=\begin{cases}1 & D(\mathbf{x},\mathbf{x}_i)\le h\\ 0 & D(\mathbf{x},\mathbf{x}_i)>h\end{cases}. \qquad (15)$$

E. Conversation-dependent bandwidth

It is known from the literature [31] that one of the practical limitations of Mean Shift algorithm is the need to fix the bandwidth h. Using a fixed bandwidth is not generally appropriate, as the local structure of samples can change the data that needs to be clustered. We have found that varying the bandwidth from one conversation to another turns out to be useful in diarization based on Mean Shift algorithm. In order to deal with the disparity caused by the variable duration of conversations, we adopt a version of the variable bandwidth scheme proposed in [10]. This is designed to smooth the density estimator (2) in the case of short conversations where the number of segments to be clustered is small.

The variable bandwidth is controlled by two parameters τ and the fixed bandwidth h. For a conversation c, the conversation-dependent bandwidth $\tilde{h}^{(c)}$ is given by

$$\tilde{h}^{(c)}=1-\left(\frac{n^{(c)}\tau(1-h)}{n^{(c)}\tau+(1-h)}\right) \qquad (16)$$

where $n^{(c)}$ is the number of segments in the conversation. Note that $\tilde{h}^{(c)}\ge h$ with equality if $n^{(c)}$ is very large.

F. Cluster pruning

An artifact of the Mean Shift algorithm is that there is nothing to prevent it from producing clusters with very small numbers of segments. To counter this tendency, we simply prune clusters containing a small number of samples (less than or equal to a constant p) by merging them with their nearest neighbors.

IV. I-VECTOR NORMALIZAION FOR DIARIZATION

By design, i-vectors are intended to represent a wide range of speech variabilities. Hence, raw i-vectors need to be normalized in ways which vary from one application to another. Based on the above definitions of *class* and *sample* in relation to our problem (see Section II), we will present in the following sections some methods to normalize i-vectors which are suitable for speaker diarization.

A. Principal components analysis (PCA)

In [16] it was shown that projecting i-vectors onto the conversation-dependent PCA axes with high variance helps to compensate for intra-session variability. (A further weighting with the square root of the corresponding eigenvalues was also applied to these axes in order to emphasize their importance.)

The authors of [16] recommend choosing the PCA dimensionality so as to retain 50% of the data variance. We will denote this quantity by r. Ideally each retained PCA axis represents the variability due to a single speaker in the conversation.

Note that that this type of PCA is "local" in the sense that

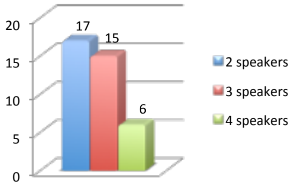

Fig. 2 CallHome development data set broken down by categories representing the number of participating speakers in conversations.

analysis is done on a file-by-file basis. Thus has the advantage that no background data is required to implement it.

B. Within Class Covariance Normalization (WCCN)

Normalizing data variances using a Within Class Covariance matrix has become common practice in the Speaker Recognition field [12][13][15]. The idea behind this normalization is to penalize axes with high intra-class variance by rotating data using a decomposition of the inverse of the Within Class Covariance matrix.

C. Between Class Covariance Normalization (BCCN)

By analogy with the WCCN approach, we propose a new normalization method based on the maximization of the directions of between class variance by normalizing the i-vectors with the decomposition of the between class covariance matrix B. The between class covariance matrix is given by the following formula:

$$\mathbf{B}=\frac{1}{n}\sum_{i=1}^{I}n_i(\overline{\mathbf{x}}_i-\overline{\mathbf{x}})(\overline{\mathbf{x}}_i-\overline{\mathbf{x}})^t \qquad (17)$$

where the sum ranges over I conversation sides in a background training set, $\overline{\mathbf{x}}_i=\frac{1}{n_k}\sum_{j=1}^{n_i}\mathbf{x}_j^i$ is the sample mean of speaker turns within the conversation side i and $\overline{\mathbf{x}}$ is the sample mean of all i-vectors.

V. IMPLEMENTATION DETAILS

A. CallHome data

We use the CallHome dataset distributed by NIST during the year 2000 speaker recognition evaluation [18]. CallHome is a multi-lingual (6 languages) dataset of multi-speaker telephone recordings of 1 to 10 minutes duration. Fig. 2 depicts the development part of the dataset which contains 38 conversations, broken down by the number of speakers (2 to 4

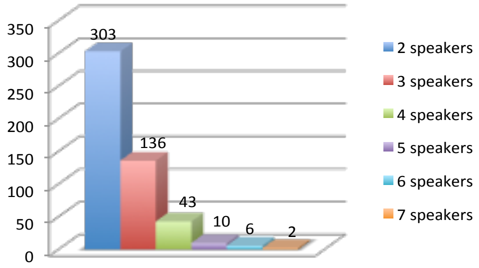

Fig. 3 CallHome test set broken down by categories representing the number of participating speakers in conversations.

speakers). The CallHome test set contains 500 conversations, broken down by the number of speakers in Fig. 3. Note that the number of speakers ranges from 2 to 4 in the development set and from 2 to 7 in the test set, so that there is a danger of over-tuning on the development set. For our purposes the development set serves to decide which types of i-vector normalization to use, to fix the bandwidth parameter h in (15), (16) and to determine a strategy for pruning sparsely populated clusters. Because there is essentially only one scalar parameter to be tuned, our approach is not at risk for over-tuning on the development set.

B. Feature extraction

1) Speech parameterization

Every 10ms, 20 Mel Frequency Cepstral Coefficients (MFCC) are extracted from a 25 ms hamming window (19 MFC Coefficients + energy). As is traditional in diarization, no feature normalization is applied.

2) Universal background model

We use a gender-independent UBM containing 512 Gaussians. This UBM is trained with the LDC releases of Switchboard II, Phases 2 and 3; Switchboard Cellular, Parts 1 and 2; and NIST 2004–2005 SRE (telephone speech only).

3) I-vector extractor

We use a gender-independent i-vector extractor of dimension 100, trained on the same data as UBM together with data from the Fisher corpus.

C. I-vector normalization

Among the normalization methods presented in Section IV, only the within and the between class covariance matrices need background data to be estimated. In order to estimate them we used telephone speech (whole conversation sides) from the 2004 and 2005 NIST speaker recognition evaluations.

D. Initial segmentation

The focus in this work is speaker clustering rather than segmentation. Following the authors of [4] [16], we uniformly segmented speech intervals found by a voice activity detector into segments of about one second of duration. This naïve approach to speaker turn segmentation is traditional in diarizing telephone speech (where speaker turns tend to be very short) and Viterbi re-segmentation is generally applied in subsequent processing. Note that the results presented in [16] show that using reference silence detector offers no significant improvement in comparison to their own speech detector.

E. Evaluation protocol

In order to evaluate the performances of different systems we use the NIST Diarization Error Rate (DER) as the principal measure system performance. Using the NIST scoring script *md-eval-v21.pl*[3] we evaluate the DER of the concatenated "*.rttm*" files produced for all conversations in the development and test sets. As is traditional in speaker diarization of telephone speech, we ignore overlapping speech segments and we tolerate errors less than 250 ms in locating segment boundaries.

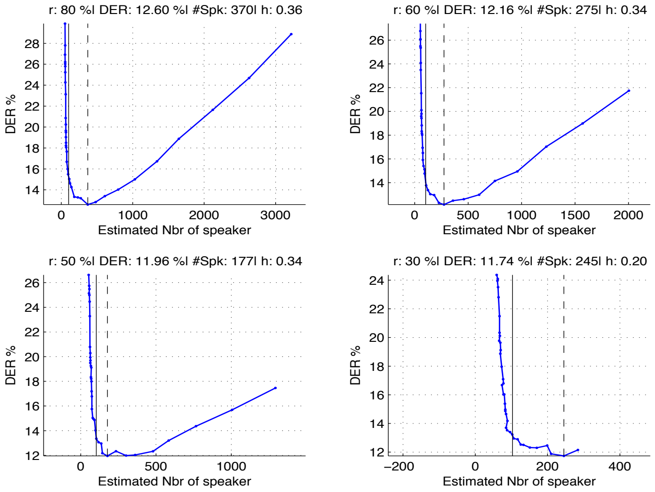

Fig. 4 Results on the development set obtained with PCA i-vector normalization: ***Full*** Mean Shift performances (DER/Number of estimated speakers). The minimum of DER, the corresponding bandwidth (h) and the number of detected speakers (#Spk) are also given for each PCA reduction factor r = 80, 60, 50 and 30.

In addition to DER, the Number of Detected Speakers (NDS) and its average calculated over all files (ANDS) are also useful performance evaluation metrics in the context of clustering with unknown numbers of speakers. We adopt a graphical illustration of DER vs. NDS to represent systems' behaviors (Figs. 4 and 5). These graphs are obtained by sweeping out the bandwidth parameter h. On these graphs, the actual number of speakers is given by the vertical solid line and the estimated number is given by the dashed line.

VI. Results and discussions

In this section we provide a detailed study of the effect of the i-vector normalization methods described in Section VI-C.

A. Parameter tuning on the development set

In order to establish a benchmark we first ran the two

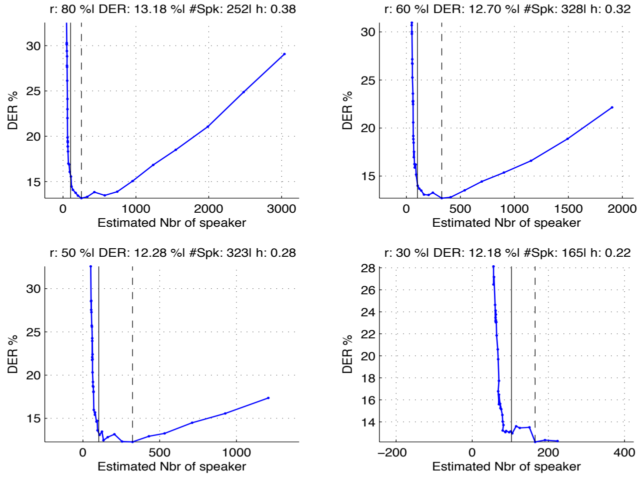

Fig. 5 Results on the development set obtained with PCA i-vector normalization: ***Selective*** Mean Shift performances (DER/Number of estimated speakers). The minimum of DER, the corresponding bandwidth (h) and the number of detected speakers (#Spk) are also given for each PCA reduction factor r = 80, 60, 50 and 30.

[3] http://www.itl.nist.gov/iad/mig/tests/rt/2006-spring/code/md-eval-v21.pl

versions of Mean Shift with PCA normalization of i-vectors. Each graph in Figs. 4 and 5 corresponds to a percentage of retained eigenvalues (r = 80, 60, 50 and 30 respectively).

In Figs. 4 and 5 we observe that although the results for the two strategies with r = 30% are slightly better than those with r = 50%, the graphs are irregular in the former case so that taking r = 50% (as in [16]) seems to be the better course. Note that the optimal DER for all configurations is reached with an overestimation of the number of speakers. Fortunately overestimation is preferable to underestimation, as it can be remedied by pruning sparsely populated clusters.

Impact of length normalization

We began by testing the effect of length normalization of raw i-vectors before applying PCA. Surprisingly, this simple operation improves the DER by 2% absolute (row 3 - Len.n - in Table 1). With length normalization and r = 50%, DER decreases from 11.9% (see Fig. 4) to 10% (*Full* strategy) and from 12.2% (see Fig. 5) to 10.2% (*Selective* strategy). Furthermore, the number of detected speakers (NDS) in the case of *Selective* strategy decreases from 323 to 281, thus approaching the actual value of 103. However, in the case of *Full* strategy the detected NDS increases form 177 to 316.

Table 1
Results on the developemnt test set illustrating the effect of different normalization methods (DER is the diarization error rate, NDS the number of detected speakers, *h* the bandwidth and *p* the prunining pararmeter) The actual number of speakers is 103.

Norm method	*Full* MS				*Selective* MS			
	DER (%)	NDS	*h*	*p*	DER (%)	NDS	*h*	*p*
Len. n.	10.0	316	0.34	0	10.2	281	0.34	0
WCC	11.7	320	0.30		11.7	343	0.28	
BCC	7.6	285	0.26		7.7	189	0.28	
Var. *h*	7.5	300	0.22		7.6	203	0.24	
Prun.	8.3	109	**0.32**	1	7.5	111	0.24	3

Impact of within class covariance normalization

In this experiment we first normalize i-vectors using the Cholesky decomposition of the inverse of the WCC matrix, and follow this with length normalization and PCA projection. As we see in row 4 of Table 1 WCC normalization causes performance degradation. The DER increases from 10% to 11.7% in the *Full* case and from 10.2% to 11.7% in the *Selective* case compared to both previous normalization methods, namely PCA and length normalization. These results were not in line with our expectations derived from our experience in speaker recognition; they may be due to an interaction between the PCA and WCC normalizations.

Impact of between class covariance normalization

We proceeded in a similar way to WCC normalization. We project data using the Cholesky decomposition of the BCC matrix followed by length normalization and PCA projection. In row 5 of Table 1 we notice a remarkable twofold improvement compared with row 2. On the one hand, we obtain a DER decrease from 10% to 7.6% for the *Full* strategy case and from 10.2% to 7.7% for the *Selective* case. On the other hand, we detect a number of speakers much nearer to the actual value of 103, particularly in the *Selective* case (189 speakers).

Conversation-dependent bandwidth Mean Shift

We applied the variable bandwidth scheme given in formula (16) to the previous BCC normalization system. In row 6 of Table 1, we observe a slight improvement in DER for both strategies.

Cluster pruning

Although we succeed in reducing the DER from ~12% to ~7% for both strategies, the estimated number of speakers corresponding to the minimum of DER is still higher than the actual value. As discussed in Section III-F we prune clusters containing a small number of samples (less than or equal to a constant p) in order to counter this tendency. The corresponding results appear in the last row of Table 1. We observe that for the *Full* strategy, merging clusters having one instance (p = 1) reduces the estimated number of speakers from 300 to 109 while the DER slightly increases from 7.5% to 8.3%. For the *Selective* strategy, with p = 3 we get a nice improvement regarding Number of Detected Speakers (111 speakers instead of 203) while the DER is essentially unaffected, decreasing from 7.6% to 7.5%.

B. *Results on the test set*

As we explained when discussing the evaluation protocol (Section V-E), we now present the results obtained on the test set by using parameters (bandwidth and the pruning factor p) tuned on the development set. Table 2 presents the most important results. The term "Fix. h" in row 3 of Table 2 refers to the best system using fixed bandwidth presented in row 5 of Table 1 (BCC). In this system we used respectively BCC, length normalization followed by length normalization and PCA projection with r = 50% as optimized on the development set. In row 4 of table 2 (Var. h), the system is exactly the same as the previous one (Fix. h system) but with a variable bandwidth. Finally, the last row of Table 2 (Prun.) shows the impact of clusters pruning on the variable bandwidth system (Var. h).

Table 2
Results on test data set using optimal parameters estimated on the development set. The total actual number of speakers is 1283.

Norm method	*Full* MS				*Selective* MS			
	DER (%)	NDS	*h*	*p*	DER (%)	NDS	*h*	*p*
Fix. *h*	14.3	3456	0.26	0	13.9	3089	0.28	0
Var. *h*	12.7	2550	0.22	0	12.6	2310	0.24	0
Prun.	**12.4**	**1361**	**0.32**	1	14.3	1501	0.24	3

From the results in Table 2 we observe the usefulness of the variable bandwidth in reducing the DER (from 14.3% to 12.7% in the *Full* MS case and from 13.9% to 12.6% in the *Selective* case). Observe also that the number of detected speakers (NDS) is reduced from 3456 to 2550 in the *Full* MS strategy and from 3089 to 2310 in the *Selective* case. Finally, in the test set, cluster pruning leads to a degradation of the

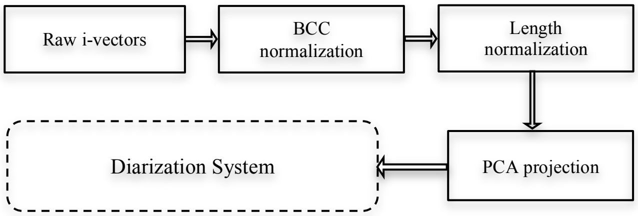

Fig. 6 The best protocol of i-vector normalization for the MS diarization systems.

DER from 12.6% to 14.3% using the *Selective* strategy, in contrast to what is observed on the development set where the DER showed a slight improvement. However, cluster pruning on the test set for the *Full* case is surprisingly helpful to the point that the DER (12.4%) coincides perfectly with the one optimized on the development set (see the bandwidth h in row 7 of Table 1). Given that the above results are obtained using the parameters tuned on an independent dataset, this confirms the generalization capability of the cosine-based Mean Shift for both clustering strategies.

Among the publications reporting results on the CallHome dataset [17][18][19][20][21], only Vaquero's thesis presents results based on the total DER calculated over all files [21]. He uses speaker factors rather than i-vectors to represent speech segments and he used a multistage system based principally on Hierarchical Agglomerative Clustering (HAC), k-means and Viterbi segmentation. He also estimated tunable parameters on an independent development set (consisting solely of two-speaker recordings). However, he was constrained to provide the actual number of speakers as stopping criterion for HAC in order to achieve a total DER of 13.7% on the test CallHome set. Without this constraint, the performance was 19.8%. Compared to his results, we were able to achieve a 37% relative improvement in the total DER (see Table 2).

In summary, we presented some results on development and test sets from which we can draw the following conclusions:

- Length normalization of the raw i-vectors before PCA projection helps in reducing DER.
- PCA with r = 50% offers the best configuration.
- WCC normalization degrades performance.
- BCC normalization, followed by length normalization and PCA, helps to decrease both DER and NDS.
- Variable bandwidth combined with cluster pruning (p = 1) applied after length normalization, PCA projection and BCC normalization help in reducing DER and NDS in the *Full* case.
- Both strategies, namely *Full* and *Selective*, perform equivalently well on development and test sets.

In Fig. 6 we depict the best i-vectors normalization protocol that we adopt in this study.

C. Results broken-down by the number speakers

In order to compare our results with those of [17][18][19][20] we need to adopt the same convention for presenting diarization results. As mentioned in Section V-E which describes the evaluation protocol, these works present results broken down by the number of participating speakers.

TABLE 3
***FULL* MEAN SHIFT** RESULTS ON TEST-SET DEPICTED AS A FUNCTION OF THE NUMBER OF PATRICIPATING SPEAKERS.

	Speakers number	2	3	4	5	6	7	h / p
Dev. Param.	DER(%)	11.9	13.5	15.6	22.9	29.5	28.4	**Fix. *h* 0.26 / 0**
	ANDS	6.5	7.2	7.9	8.5	9.6	11.0	
	DER(%)	7.8	12.5	16.2	23.1	30.4	28.6	Var. h 0.22 / 0
	ANDS	4.0	5.5	8.4	9.3	14.3	20.0	
Test param.	DER(%)	11.9	13.5	15.6	22.9	29.5	28.4	**Fix. *h* 0.26 / 0**
	ANDS	6.5	7.2	7.9	8.5	9.6	11.0	
	DER(%)	8.1	12.5	15.5	23.2	27.5	29.0	Var. h 0.20 / 0
	ANDS	4.2	6.1	10.5	11.9	14.6	24.0	

Indeed, the official development set consisted of conversations with just 2 to 4 speakers so it is hard to avoid tuning on the test set if one wishes to optimize performance on conversations with large numbers of speakers. In Tables 3 and 4 we present results broken down by the number of speakers on the test set for the *Full* and *Selective* Mean shift algorithms, with two tunings, one on the development (rows 2--5) and the other on the test set (rows 6--9). Recall that the tunable parameters are the nature of the bandwidth (i.e. fixed or variable), its value (i.e. h) and the pruning factor p. It is apparent from the tables that all of the Mean Shift implementations generalize well from the development set to the test set.

From Table 3 we observe firstly that the *Full* MS implementation does not need any final cluster pruning (i.e. p = 0) when we optimize taking account of the number of participating speakers (see last column of Table 3). Second, estimating the number of speakers works better with a fixed bandwidth (see rows 3 and 7 of Table 3) and the DERs are almost comparable to those obtained with a conversation-dependent bandwidth (i.e. Var. h). Generally speaking, variable bandwidth helps in reducing DER for recordings having small number of speakers (2, 3, 4 speakers). Finally, the most important observation from Table 3 is the high generalization capability of the *Full* MS especially in the fixed bandwidth case. (Comparing rows 2 and 3 with rows 6 and 7 of Table 3 we see that the optimal parameters for the test set are the same as those for the development set.)

TABLE 4
***SELECTIVE* MEAN SHIFT** RESULTS ON TEST DATA SET DEPICTED AS A FUNCTION OF THE NUMBER OF PATRICIPATING SPEAKERS.

	Speakers number	2	3	4	5	6	7	h / p
Dev. Param.	DER(%)	9.9	12.6	15.5	22.6	29.3	29.9	Fix. 0.28 / 3
	ANDS	2.8	3.2	3.3	3.8	4.0	6.0	
	DER(%)	7.2	13.1	15.6	22.8	29.0	27.7	Var. 0.24 / 3
	ANDS	2.3	2.9	3.4	3.5	4.1	5.5	
Test param.	DER(%)	10.8	12.8	15.6	21.7	26.7	27.3	Fix. 0.26 / 3
	ANDS	2.9	3.3	3.7	3.9	4.3	6.5	
	DER(%)	8.1	12.6	15.9	22.2	26.1	27.6	Var. 0.18 / 3
	ANDS	2.4	3.3	4.8	5.2	5.8	10.0	

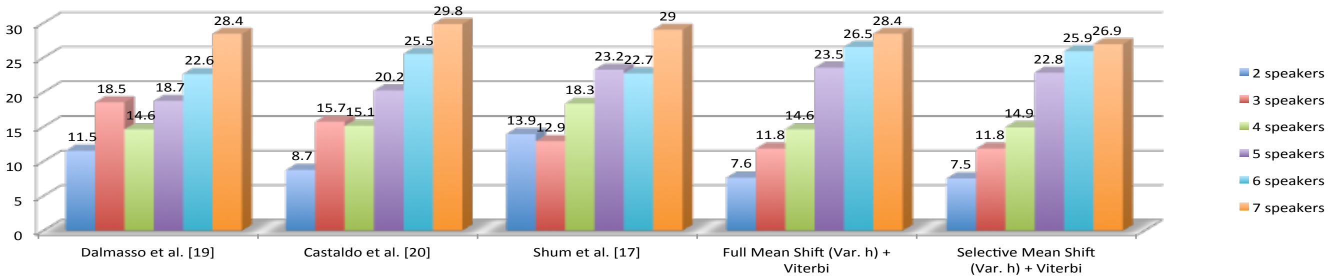

Fig. 7 Comparison of *Full* and *Selective* Mean Shift clustering algorithms with state-of-the-art results based on DER (%) for each category of CallHome test set recordings having same number of speakers.

From the results depicted in Table 4 we observe that the final cluster pruning is necessary in the *Selective* MS case. Compared to *Full* MS results in Table 3, we observe that DERs are similar but the *Selective* strategy outperforms the *Full* one regarding the average number of detected speakers (ANDS). The combination of the variable bandwidth with the final cluster pruning ($p = 3$) enables us to get the best results, both for DER and Average Number of Detected Speakers (see rows 4 and 5 and rows 8 and 9 in table 4). The ANDS values are in fact very close to the actual numbers (row 6 vs. row 1) with a slight overestimation, except in the 6-speaker files case where there is a slight underestimation (5.8).

Finally, we observe that the *Full* strategy generalizes better than the *Selective* one in the sense that we were able to reach the best performance on the test using development tunable parameters.

Viterbi re-segmentation

Refining segment boundaries between speaker turns using Viterbi re-segmentation is a standard procedure for improving diarization system performance. Results reported in Table 5 show its effectiveness when combined with the Mean Shift algorithms. Note that the results without Viterbi re-segmentation (gray entries in table 5) are the results presented in the 6th and 8th rows in tables 3 and 4.

TABLE 5
IMPACT OF VITERBI RE-SEGMENTATION ON THE TEST-SET RESULTS (USING PARAMETRES ESTIMATED ON TEST DATA) DEPICTED AS A FUNCTION OF THE NUMBER OF PATRICIPATING SPEAKERS AND MEASURED WITH DER(%).

Full MS	Fix. *h*	- Viterbi	11.9	13.5	15.6	22.9	29.5	28.4
		+ Viterbi	11.2	12.3	14.5	22.8	27.3	27.4
	Var. *h*	- Viterbi	8.1	12.5	15.5	23.2	27.5	29.0
		+ Viterbi	7.6	11.8	14.6	23.5	26.5	28.4
Selective MS	Fix. *h*	- Viterbi	10.8	12.8	15.6	21.7	26.7	27.3
		+ Viterbi	10.1	11.6	14.3	22.0	25.9	26.7
	Var. *h*	- Viterbi	8.1	12.6	15.9	22.2	26.1	27.6
		+ Viterbi	7.5	11.8	14.9	22.8	25.9	26.9

Comparison with existing state-of-the-art results

We conclude this section with a comparison between our results and those obtained by other authors on the Call Home data although there are several factors which make back-to-back comparisons difficult. Contrary to [21] and our work, the authors of [17][19][20] did not use a development set independent of the test set for parameter tuning. Furthermore in [19] and [20], the authors assumed prior hypotheses about the maximum number of speakers within a slice of speech, and in [19] estimating the number of speakers was done separately from speaker clustering.

We compare graphically in Fig. 7 the results (as measured by DER) of our best configurations (with Viterbi re-segmentation) of the *Full* and *Selective* strategies (i.e. *Full* and *Selective* systems presented in the 4th and 8th rows of Table 5 respectively) with those in [19][20][17]. It is evident that our results as measured by DER are in line with the state-of-the-art. (To be clear, since our results were taken from Tables 3 and 4, there was some tuning on the test set as in Dalmasso *et al.* [19], Castaldo *et al.* [20], and Shum *et al.* [17])

Furthermore, the comparison based on the average number of detected speakers is not possible except in the case of Dalmasso *et al.* [19]. In Table 6 we compare our best results from Table 4 using this criterion with those of [19]. The results are similar although the Mean Shift algorithm tends to overestimate the speaker number.

TABLE 6
COMPARISON WITH DALMASSO RESULTS BASED ON THE AVRAGE OF THE NUMBER OF DETECTED SPEAKERS (ANDS).

Actual Number of speakers	2	3	4	5	6	7
Dalmasso *et al.* [19]	1.9	2.3	3.3	4.4	4.8	6.5
Selective MS	2.4	3.3	4.8	5.2	5.8	10.0

D. *Time complexity*

Time complexity is not a major concern in this study but Fig. 8 illustrates the difference between the *Full* and the *Selective* strategies in this regard. The average time for the *Full* case is 0.0934 seconds per file vs. 0.0091 seconds for the *Selective* case.

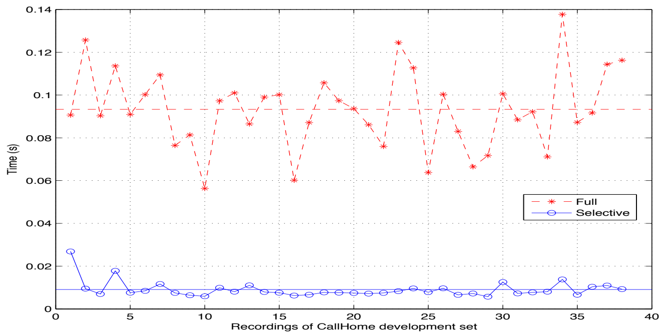

Fig. 8 Time complexities of the *Full* and *Selective* strategies calculated (in seconds) on each conversation of the development set. The horizontal lines indicate the processing times averaged over all files.

VII. Conclusion

This paper provides a detailed study of the application of the non-parametric Mean Shift algorithm to the problem of speaker clustering in diarizing telephone speech conversations using two variants of the basic clustering algorithm (the *Full* and *Selective* versions). We have supplied (in the Appendix) a convergence proof which justifies our extension of the Mean Shift algorithm from the Euclidean distance metric to the cosine distance metric. We have shown how, together with an i-vector representation of speaker turns, this simple approach to the speaker clustering problem can handle several difficult problems --- short speaker turns, varying numbers of speakers and varying conversation durations.

With a single pass clustering strategy (that is, without Viterbi re-segmentation) we were able to achieve a 37% relative improvement as measured by global diarization error rate on the Call Home data (using as a benchmark [21], the only other study that evaluates performance in this way). We have seen how our results using other metrics are similar to the state-of-the art as reported by other authors [16][17][19][20].

We have seen that refining speaker boundaries with Viterbi re-segmentation is also helpful. Using segment boundaries obtained in this way could serve as a good initialization for a second pass of Mean Shift clustering. An interesting complication that would arise in exploring this avenue is that speaker turns would be of much more variable duration than in the first pass (based on the uniform segmentation described in Section V.D). Since the uncertainty entailed in estimating an i-vector in the case of short speaker turns than in the case of long speaker turns, this suggests that taking account of this uncertainty as in [32] would be helpful.

Appendix

In this appendix we present the mathematical convergence proof of the cosine distance-based Mean Shift. Indeed, this proof is very similar the one of theorem 1 presented in [8].

Theorem 1 [8]: *if the kernel k has a convex and monotonically decreasing profile, the sequence* $\{\hat{f}_i\}_{i=1,2...}$ *converges, and is monotonically increasing.*

Let us suppose that all vectors in our dataset are constrained to live in the unit sphere by normalizing their Euclidean-norm during MS convergence process.

$$\hat{f}_{j+1} - \hat{f}_j = c\sum_{j=1}^{n}\left[k\left(\frac{1-\mathbf{y}_{i+1}\cdot\mathbf{x}_j}{h}\right) - k\left(\frac{1-\mathbf{y}_i\cdot\mathbf{y}_i}{h}\right)\right].$$

Due to the convexity of the profile:

$$k(x_2) - k(x_1) \ge k'(x_1)(x_2 - x_1)$$

and since $g(x) = -k(x)$ from (6) than:

$$k(x_2) - k(x_1) \ge g(x_1)(x_1 - x_2)$$

we obtain:

$$\begin{aligned}\hat{f}_{i+1} - \hat{f}_i &\ge c\sum_{j=1}^{n} g\left(\frac{1-\mathbf{y}_i\cdot\mathbf{x}_j}{h}\right)\left[\frac{(1-\mathbf{y}_i\cdot\mathbf{x}_j)-(1-\mathbf{y}_{i+1}\cdot\mathbf{x}_j)}{h}\right]\\ &= c\sum_{j=1}^{n} g\left(\frac{1-\mathbf{y}_i\cdot\mathbf{x}_j}{h}\right)\mathbf{x}_j\left(\frac{\mathbf{y}_{i+1}-\mathbf{y}_i}{h}\right)\end{aligned}$$

we know from (8) and (11) that the (i+1)th position ($\mathbf{y}_{i+1}$) is equal to the weighted mean vector, so

$$\sum_{j=1}^{n} g\left(\frac{1-\mathbf{y}_i\cdot\mathbf{x}_j}{h}\right)\mathbf{y}_{i+1} = \sum_{j=1}^{n} g\left(\frac{1-\mathbf{y}_i\cdot\mathbf{x}_j}{h}\right)\mathbf{x}_j\ .$$

Thus:

$$\begin{aligned}\hat{f}_{i+1} - \hat{f}_i &\ge c\sum_{j=1}^{n} g\left(\frac{1-\mathbf{y}_i\cdot\mathbf{x}_j}{h}\right)\mathbf{y}_{i+1}\left(\frac{\mathbf{y}_{i+1}-\mathbf{y}_i}{h}\right)\\ &= c\sum_{j=1}^{n} g\left(\frac{1-\mathbf{y}_i\cdot\mathbf{x}_j}{h}\right)\left(\frac{1-\mathbf{y}_{i+1}\cdot\mathbf{y}_i}{h}\right) \ge 0\end{aligned}$$

with equality iff $\mathbf{y}_{i+1} = \mathbf{y}_i$.

The sequence $\{\hat{f}_i\}_{i=1,2...}$ is bounded and monotonically increasing, and so is convergent. This argument does not show that $\{\mathbf{y}_i\}_{i=1,2...}$ is convergent (it may be possible to construct pathological examples in which $\{\hat{f}_i\}_{i=1,2...}$ converges but $\{\mathbf{y}_i\}_{i=1,2...}$ does not) but it establishes convergence of the Mean Shift algorithm in the same sense as convergence of the EM algorithm is demonstrated in [33].

Acknowledgment

First we would like to thank the editor as well as reviewers for their helpful comments. Then, we would like to thank Stephen Shum and Najim Dehak from MIT for their useful discussions and feedback and also for sharing their initial segmentation with us. We would like also to thank our colleague Vishwa Gupta for his help with Viterbi re-segmentation software and our colleague Pierre Ouellet for his help with other software.

References

[1] G. Schwarz, "Estimating the dimension of a model," Ann. Statist. 6, 461-464. (1978).

[2] S. S. Chen and P. Gopalakrishnan, "Clustering via the bayesian information criterion with applications in speech recognition," in ICASSP'98, vol. 2, Seattle, USA, 1998, pp. 645–648.

[3] F. Valente, "Variational Bayesian methods for audio indexing," Ph.D. dissertation, Eurecom, Sep 2005.

[4] P. Kenny, D. Reynolds and F. Castaldo, "Diarization of Telephone Conversations using Factor Analysis," Selected Topics in Signal Processing, IEEE Journal of, vol.4, no.6, pp.1059-1070, Dec. 2010.

[5] Margarita Kotti, Vassiliki Moschou, Constantine Kotropoulos, "Speaker segmentation and clustering," Signal Processing, Volume 88, Issue 5, May 2008, Pages 1091-1124, ISSN 0165-1684, 10.1016/j.sigpro.2007.11.017.

[6] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," IEEE Trans. on Information Theory, vol. 21, no. 1, pp. 32–40, January 1975.

[7] Y. Cheng, “Mean Shift, Mode Seeking, and Clustering”, IEEE Trans. PAMI, vol. 17, no. 8, pp. 790-799, 1995.

[8] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 603 – 619, May 2002.

[9] T. Stafylakis, V. Katsouros, and G. Carayannis, “Speaker clustering via the mean shift algorithm,” in Odyssey 2010: The Speaker and Language Recognition Workshop - Odyssey-10, Brno, Czech Republic, June 2010.

[10] T. Stafylakis, V. Katsouros, P. Kenny, and P. Dumouchel, “Mean Shift Algorithm for Exponential Families with Applications to Speaker Clustering,” Proc. Odyssey Speaker and Language Recognition Workshop, Singapore, June 2012.

[11] M. Senoussaoui, P. Kenny, P. Dumouchel and T. Stafylakis, “Efficient Iterative Mean Shift based Cosine Dissimilarity for Multi-Recording Speaker Clustering,” in Proceedings of ICASSP, 2013.

[12] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” IEEE Transactions on Audio, Speech, and Language Processing, Vol. 19, No. 4, May 2011, pp. 788-798.

[13] N. Dehak, R. Dehak, J. Glass, D. Reynolds, and P. Kenny, "Cosine Similarity Scoring without Score Normalization Techniques," Proc. IEEE Odyssey Workshop, Brno, Czech Republic, June 2010.

[14] N. Dehak, Z. Karam, D. Reynolds, R. Dehak, W. Campbell, and J. Glass, "A Channel-Blind System for Speaker Verification," Proc. ICASSP, pp. 4536-4539, Prague, Czech Republic, May 2011.

[15] M. Senoussaoui, P. Kenny, N. Dehak and P. Dumouchel, “An i-vector Extractor Suitable for Speaker Recognition with both Microphone and Telephone Speech,” in Proc Odyssey Speaker and Language Recognition Workshop, Brno, Czech Republic, June 2010.

[16] S. Shum, N. Dehak, E. Chuangsuwanich, D. Reynolds, and J. Glass, "Exploiting Intra-Conversation Variability for Speaker Diarization," Proc. Interspeech, pp. 945-948, Florence, Italy, August 2011.

[17] S. Shum, N. Dehak, and J. Glass, "On the Use of Spectral and Iterative Methods for Speaker Diarization," Proc. Interspeech, Portland, Oregon, September 2012.

[18] A. Martin and M. Przybocki, “Speaker recognition in a multi-speaker environment,” in Proceedings of Eurospeech, 2001.

[19] E. Dalmasso, P. Laface, D. Colibro, C. Vair, “Unsupervised Segmentation and Verification of Multi-Speaker Conversational Speech,” Proc. Interspeech 2005.

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

[21] C. Vaquero Avilés-Casco, “Robust Diarization For Speaker Characterization (Diarizacion Robusta Para Caracterizacion De Locutores),” Ph.D. dissertation, Zaragoza University, 2011.

[22] N. Dehak, P. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language Recognition via Ivectors and Dimensionality Reduction," Proc. Interspeech, pp. 857-860, Florence, Italy, August 2011.

[23] D. Martinez, Oldricht Plchot, Lukas Burget, Ondrej Glembek and Pavel Matejka, “Language Recognition in iVectors Space,” Proceedings of Interspeech, Florence, Italy, August 2011.

[24] H. Tang, S.M. Chu, M. Hasegawa-Johnson and T.S. Huang, “Partially Supervised Speaker Clustering,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol.34, no.5, pp.959, 971, May 2012.

[25] P. Kenny, “Joint factor analysis of speaker and session variability: theory and algorithms”. Technical report CRIM-06/08-14, 2006.

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

[27] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking”. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(5), 564–577.

[28] B. Georgescu, I. Shimshoni, and P. Meer, “Mean shift based clustering in high dimensions: A texture classification example,” in Proceedings of International Conference on Computer Vision (pp. 456–463).

[29] U. Ozertem, D. Erdogmus, R. Jenssen, “Mean shift spectral clustering”. Pattern Recognition, Volume 41, Issue 6, June 2008, Pages 1924-1938.

[30] D. Garcia-Romero, “Analysis of i-vector length normalization in Gaussian-PLDA speaker recognition systems,” in Proceedings of Interspeech, Florence, Italy, Aug. 2011.

[31] D. Comaniciu, V. Ramesh, and P. Meer, “The Variable Bandwidth Mean Shift and Data-Driven Scale Selection,” Proc Eighth Int'l Conf. Computer Vision, vol. I, pp. 438-445, July 2001.

[32] P. Kenny, T. Stafylakis, P. Ouellet, J. Alam, and P. Dumouchel, “PLDA for Speaker Verification with Utterances of Arbitrary Duration,” In Proceeding of ICASSP, Vancouver, Canada, May 2013.

[33] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” Journal of the Royal Statistical Society, Series B (Methodological), vol. 39, no. 1, pp. 1–38, 1977.

M. Senoussaoui received the Engineer degree in Artificial Intelligence in 2005 and Magister (Masters) degree in 2007 from Université des Sciences et de la Technologie d’Oran, Algeria. Currently hi is a PhD student in the École de technologie supérieure (ÉTS) of Université du Québec, Canada and also with Centre de recherche informatique de Montréal (CRIM), Canada. His research interests are concentrated to the application of Pattern Recognition and Machine learning methods to the speaker verification and Diarization problems.

P. Kenny received the BA degree in Mathematics from Trinity College, Dublin and the MSc and PhD degrees, also in Mathematics, from McGill University. He was a professor of Electrical Engineering at INRS-Telecommunications in Montreal from 1990 to1995 when he started up a company (Spoken Word Technologies) to spin off INRSs speech recognition technology. He joined CRIM in 1998 where he now holds the position of principal research scientist. His current research interests are in text-dependent and text-independent speaker recognition with particular emphasis on Bayesian methods such as Joint Factor Analysis and Probabilistic Linear Discriminant Analysis.

T. Stafylakis received the Diploma degree in electrical and computer engineering from the National Technical University of Athens (NTUA), Athens, Greece, and the M.Sc. degree in communication and signal processing from Imperial College London, London, U.K., in 2004 and 2005, respectively. He received his Ph.D. from NTUA on speaker diarization, while working for the Institute for Language and Speech Processing, Athens as research assistant. Since 2011, he is a post-doc researcher at CRIM and ETS, under the supervision of Patrick Kenny and Pierre Dumouchel, respectively. His current interests are speaker recognition and diarization, Bayesian modeling and multimedia signal analysis.

P. Dumouchel received B.Eng. (McGill University), M.Sc. (INRS-Télécommunications), PhD (INRS-Télécommunications), has over 25 years of experience in the field of speech recognition, speaker recognition and emotion detection. Pierre is Chairman and Professor at the Software Engineering and IT Department at École de technologie supérieure (ETS) of Université du Québec, Canada.