Speaker Diarization Based on Locally Linear Embedding

Ori Shahar, Lee Twito, Nurit Spingarn and Israel Cohen
Andrew and Erna Viterbi Faculty of Electrical Engineering
Technion - Israel Institute of Technology
Technion City, Haifa 32000

***Abstract*—Speaker diarization is a significant part of many applications in today's fast growing user-end software and technologies. In the last decade, speaker diarization has attracted significant research effort, however most of the speaker diarization methods deeply rely on statistical models which become unreliable in case of short utterances diarization and in noisy conditions. In this paper, we introduce a speaker diarization system which is based on the Locally Linear Embedding (LLE). The LLE enables to extract the inherent structure of the data and thus provide better clustering. Experimental results show error rates lower than 10% and improved stability in comparison with a conventional speaker diarization method.**

Keywords*—Speaker diarization, speaker clustering, noisy environment, LLE.

I. Introduction

Speaker diarization is defined as the task of tagging different speakers within a given conversation. It is an integral part of many applications involving speech processing, such as speech tagging, speaker recognition/verification and automated transcription. Various methods were developed for dealing with the speaker diarization problem, many of them are reviewed in [1], [2].

There are many considerations which are taken into account when design a speaker diarization algorithm. Variants designed for online processing will naturally have a shorter running time than offline applications, though it may be accompanied by higher miss-detection and false alarm rates. Assumptions about the received conversation, such as the number of speakers in the conversation, overlapping speakers, intensity of background noises and the duration between speaker changes might also significantly affect the algorithm planning. State-of-the-art speaker diarization systems are comprised of two main phases: Speaker change point detection (CPD) and clustering. A classical CPD algorithm is based on the Bayesian Information Criterion (BIC) [3]. This method formulates the speaker change point detection as a hypothesis testing problem when it uses the likelihood of two hypothesis: A given time point is a speaker change point or not. This method has fairly satisfying results considering misdetection and false alarm rates [3]. However, this method has significant performance degradation in case of rapidly speaker change points and noisy environment.

As for clustering, many speaker diarization algorithms utilize the Bottom-Up (BU) clustering [1]. The BU clustering is an iterative algorithm in which at each iteration, the closet clusters are merged and the models of each cluster is updated until a stopping criterion is met. Since this method is inherently based on statistical models it may produce incorrect results in case of merging a segment that does not belong to the chosen cluster. It decreases the reliability of the models such that more segments of the wrong speaker can be added to the wrong cluster. Furthermore, clustering of short segments (shorter than 3 seconds) becomes difficult since the models which are obtained by the BU algorithm are not reliable enough. This phenomena is very likely to happen for quick speaker changes or false alarms during the CPD and leads to incorrect clustering process.

In this paper, we propose to represent the speech segments by a low-dimensional vector based on the Locally Linear Embedding (LLE) algorithm which is known for its ability to deal with nonlinear problems in high-dimensional space [4]–[6]. LLE works successfully in situations where the data in the high dimension lies on a well-constructed complex manifold. Since speech can be represented well in a low dimensional space [7], it is reasonable to represent the speech segments in that way, which facilitates better exploration, visualization and clustering of the data. LLE has the ability to preserve the neighboring relations of points in the high-dimensional space when assembling them to the low-dimensional space. This is the key for its success in our usage, as opposed to linear algorithms such as PCA, which are relevant for a linear and convex manifold- not the case of speech data.

After projecting the data to the low-dimension space, each segment is represented as a point (a low-dimensional vector), and a classical clustering algorithm such as k-means can be applied. Due to the lack of correlation between points, this method tolerates mistakes, and one segment clustering mistake would not affect the other segments, unlike BU algorithm where one mistake can lead to a failure in the whole process. As we tested the LLE clustering we deduced that it manages to deal with false alarms in the CPD very well, yet miss-detections cause higher error percentages in the final result. The miss-detection rate of the CPD algorithm had to be reduced in order for us to use it. Our solution is an iterative version of the BIC CPD algorithm, which provides a lower miss-detection rate.

II. Problem Formulation

The goal of speaker diarization algorithm is tagging a human conversation to its sources, i.e., answering the question "Who spoke when?". A successful diarization algorithm will

978-1-5090-2152-9/16/$31.00

produce correct tagging: Segmentation and Clustering. Our problem assumptions are: background noise (recorded by a simple phone as so passing through a quality defecting filter), manually noised recordings and clean speech recordings as the baseline results quality check. In addition, we assume no overlapping speakers at any part of the conversation. The proposed algorithm is comprised of two main phases: Speaker change point detecting and clustering based on LLE algorithm. The conversations we deal with in this paper are both synthetic and original, they include fast speaker changes and are both short and long (from 1 minute up to 10 minutes), under the assumption of no overlapping.

III. Proposed Algorithm

The baseline diarization system consists of three main stages: Speech detection, speaker change point detection and speaker clustering. The baseline system is based on the traditional approach of speaker segmentation which uses the BIC technique followed by agglomerative speaker clustering. In this Section, we briefly discuss each of these stages.

A. Change Point Detection

A common speaker change point detector is the BIC CPD algorithm, it decides on speaker change point existence or absence by a BIC decision rule. More specifically, let H_1 and H_0 be the hypothesis of speaker change point absence or presence at time t_j, respectively. Let L_0 and L_1 be the likelihoods of the observations given hypothesis H_0 and H_1, respectively, as follows:

$$L_0 = \sum_{i=1}^{N_x} \log P(\boldsymbol{x}_i|\theta_z) + \sum_{i=1}^{N_y} \log P(\boldsymbol{y}_i|\theta_z) \quad (1)$$

$$L_1 = \sum_{i=1}^{N_x} \log P(\boldsymbol{x}_i|\theta_x) + \sum_{i=1}^{N_y} \log P(\boldsymbol{y}_i|\theta_y) \quad (2)$$

where N_x and N_y are the total number of feature vectors of segments X and Y, respectively. θ_x, θ_y, θ_z are the models parameters of the probability density functions (PDF) which represent the segments X, Y and Z, respectively. The BIC dissimilarity is estimated by:

$$S = L_1 - L_0 - P\frac{\lambda}{2}\log N_z \quad (3)$$

where λ is a penalty factor which depends on $\log N_z$. An existence of speaker change point at t_j is decided if $S > 0$ and vice versa. For further discussion see [8].

Basing on shown results, the miss-detection rate, i.e., the amount of real change points missed by the basic CPD algorithm is approximately 10%-20%. Although this rate is considered quite low for many speaker analysis usages, for diarization purpose it might be high. The clustering algorithm uses the change point vector as a starting point. False speaker change points (false-alarms) are recognized well by it and then dismissed, but its performance degrades significantly in case of miss-detections. In this paper we introduce a new method for speaker change point detection which aims to reduce dramatically the number miss-detections. To explain the change done to the baseline CPD algorithm we will view it as follows: instead of deciding whether there is a change point between two segments or not, we set a change point between every p samples and then review them one after the other and decide whether it is a real change point or a not. The ones that are suspected as false change points are removed. The miss-detections occur since not all the correct change points get a high enough value of d to be recognized by the algorithm, but all of them do have a d value larger than most of the false change points. This means we find most of the false change points but in the process discharge also some of the real ones (around 20%-30% miss-detection and false alarm rates).

In order to deal with miss-detections, we dismiss change points with greater caution- since the most negative d values always belong to false change points, we delete the most negative d valued change point in a certain iteration of the original algorithm, then update the change points vector and repeat with the new version of the vector for another iteration. This process continues until the most negative d value achieved is above a certain threshold. When setting the threshold value the following tradeoff comes into consideration: decreasing the rate of false-alarms while keeping miss-detection rate as close to zero as possible. This approach has the inequality feature we required between the miss-detections and false alarms, since only one point (or few points) is deleted from the change point vector in each iteration, and so the chance for deleting the real points decreases and miss-detection rate drops drastically. Furthermore, the algorithm updates the change point vector from one iteration to the next and by that extending some of the segments and makes their MFCC gaussian model stronger and by that increasing the chance of enlarging them in the following iterations by deleting false change points neighboring the larger segment.

B. Clustering

In the proposed algorithm, we utilize the LLE algorithm which is a dimensional reduction algorithm and known for its ability to deal with nonlinear problems in high dimensional space. It transforms the high-dimensional data set to a low-dimensional space. LLE thrives in cases where the data set lays in a manifold with a complex structure. The algorithm discovers the global internal coordinates of the manifold and preserves the neighborhood relations between the points while transforming them to low dimensional space. The data which is being processed by LLE after the change point locations are achieved, is indeed placed on a high-dim manifold, as explained in the following paragraph. Assuming the CPD algorithm produces n change-points which induce $n + 1$ segments. Each segment contains samples of speech from a single person, as described in the CPD part. In order to represent those segments we utilize the Gaussian Mixture Model (GMM) method. Thanks to the fact that short segments might introduce unreliable models, we perform mean adaptation by using the Universal Background Model (UBM) [9]. Choosing a UBM with 32 Gaussian lead to 32×19-dimensional vector that represents each segment features. UBM provides a good starting point for GMM modeling. When trying to use default GMM algorithms with a random starting point, segments of 3 seconds will not consist of enough data in order to create an accurate model for them, and will only build $1 - 2$ Gaussian components describing them. With UBM we ensure

the starting point is related to the MFCC feature space and even short segments are decently modeled.
The last process is concatenating those 32 GMM means components into a high-dimensional vector which also called a supervector [10]. By repeating this process for all of the segments, $n+1$ points in 608-dimensional space are created representing the whole signal.

Thanks to the fact that speech can be well represented in a low dimensional space [7], utilizing dimensionality reeducation algorithm is a reasonable choice. In this paper we use the LLE algorihtm which fits the task of dimension reduction for human speech data due to the fact that points (which represent segments) from the same speaker, will be scattered in a shape with a complex manifold. LLE projects the data to lower dimension based on neighborhood-preserving, which means neighboring points on the high-dim space would remain neighbors in the low-dim space. The neighbors of a point belong, most likely, to the same speaker and the loss of the data in the projection is minimal. In addition, the LLE are based on minimization of reconstruction error, assuming each data point is reconstructed by its neighbors, i.e., taking into consideration also the globally structure of the nonlinear manifolds (think globally, fit locally [4]). The transformation to a lower dimension allows to cluster the points using straightforward clustering methods, such as k-means.

IV. Performance Evaluation

We divide the performance evaluation of our algorithm into two parts: Performance evaluation of the iterative change point detection algorithm and performance evaluation of the proposed clustering method based on LLE. The results of LLE are of course dependent upon the quality of CPD results, and the subject will be discussed afterwards. The algorithms were tested on concatenated recordings from FESTVOX database [11] and on simple recordings done by a cellular phone microphone and an 8 kHz recording application. The recordings of both kinds were as general as possible, containing different languages, accents, both male/female speakers, different conversations lengths and different speaker change point rates ranging from $2-3$ seconds to 30 seconds, which turned to have the most critical influence on the quality of results. FESTVOX recordings were clean 16 kHz recordings, while our recordings had background noises and significant phone filter added to them, which naturally decreased their SNR.

A. Iterative CPD Performance Evaluation

The first phase in many speech processing methods is removing the silent parts of the conversation, since it does not consist an informative data. In this work, we utilized a conventional Voice Activity Detection (VAD) algorithm proposed by Sohn et al. [12]. The Iterative CPD was researched under two tested parameters: Number of change points dismissed at each iteration (n) and the value of algorithm stopping threshold. The value of n did not change drastically, varying in the range of 1-3, mostly being 1 since it is the only option that potentially could achieve perfect result of no miss-detection and no false alarms. The main reason of increasing the parameter n is in order to drop the running time of the algorithm. The threshold value is much harder to define. Testing provided the following values: For recordings consists of only long segments (about 30 seconds each) the value of d was very high, more than 1000, but for short segments $(2-3$ seconds), the value of d was in the range $50-100$. These numbers are purely experimental, but agree with the way d is calculated, where its value gets larger as the segment gets longer.

We will divide the iterative CPD result into four groups: Segments which are longer than 10 seconds, segments which are shorter than 10 seconds crossing with slow changes or fast changes respectively. The reason for this division is that segment length seems to be the only factor affecting the results for CPD. Different languages, accents, age and gender do not change the results. We examine the algorithm under two types of files, the "FESTVOX" which is a public database of clean recordings, and "RECORDED" which are some home-made recordings made for testing the system. Results for CPD will include miss-detection (MD) and false alarm (FA) percentages averaged on the whole data type.

TABLE I: Iterative Change Point Detection Performance.

Database	Changes Rate	MD [%]	FA [%]
FESTVOX	slow	0	19.9
Recorded	slow	1.6	18.7
FESTVOX	fast	2.35	17.75
Recorded	fast	0	34.6

Considering the "Recorded" group results, out of 32 different tests, only 2 miss-detections occurred, both on the same recording and both were easily restored when changing the threshold d. False change points percentage varied between each recording, most were dismissible with a wide range of d values, yet included in result statistics. "FESTVOX" testing showed a similar behavior when miss-detection rate dropped to negligible percentile and FA rate remained around 20%. Note that 87% of the "FESTVOX" slow recordings were perfectly segmented by the algorithm, yet other test with different n and d parameters produced FAs and were included in the results. The fast changing recordings included segments under 5 seconds long, still with MD/FA rates up to par.

B. Clustering Performance Evaluation

The next phase of performance evaluation is investigating the clustering performance while using the LLE. We assumed the segmentation phase was provided by the proposed CPD algorithm. The parameter that was changed during LLE testing is the number of chosen neighbors which is required by the LLE algorithm. As expected, increasing speaker change points required higher number of neighbors. Similarly to the CPD result section, this section will be divided into the four groups mentioned earlier. Each clustering result uses the corresponding result of CPD, including provided miss-detections and false alarms. Result quality was calculated by the Average Cluster Purity method [13]:

Considering the "Recorded" recordings results, 9 of 16 slow changes tests were perfectly clustered, while the fast changes recordings provided only 4 perfect clusters out of 16 tests. Conversation length was a major factor when dealing with fast changes. A short conversation (1 minute length) with

TABLE II: Clustering by Locally Linear Embedding Performance.

Database	Changes Rate	ACP [%]
FESTVOX	slow	4.21
Recorded	slow	7.865
FESTVOX	fast	13.09
Recorded	fast	14.94

changes faster than 10 seconds provided a much higher ACP error than the average conversation, due to the dull UBM and segments. Short recordings also harmed the slow change points, but for a different reason: The LLE works well when the feature space is well structured by a lot of data, but when dealing with a short recording and long segments, $5-6$ segments is a common sight, thereby LLE tends to produce higher ACP error rates. "FESTVOX" testing also showed similar behavior, for recordings longer than 1.5 minutes the diarization process mostly produced 0% ACP error regardless of segment length (with exceptions only for fast speaker changes).

Considering the correlation between LLE performance and the quality of CPD result, we reinforced the claim that LLE dismisses false alarms well: Testing LLE once with a perfect CPD results and once with a considerable FA percentage the result mostly experienced only a small deterioration. Some fast change recording were even clustered better with false alarms in CPD product, than without them, probably due to the division of ill modeled segments which were partly tagged correctly instead of the whole segment being tagged wrongly.

V. Conclusion

We have proposed a new speaker diarization system based on Locally Linear Embedding of the speaker data onto a low dimensional manifold. The results show good performance on both professional and amateur everyday recordings, providing perfect results throughout when dealing with speaker changes above 10 seconds. The remarkable results were those for fast speaker changes, where only few errors accrued, keeping a high ACP. LLE worked well regardless of speaker languages, accents, gender and age did not affect the algorithm's performance, where many other speaker diarization methods fail. The iterative CPD produced the exact result it was designed for - few to none miss-detections, and some false-alarms. According to the threshold of d, the result could be accompanied by a larger rate of false alarms. Although, as mentioned, the LLE dimension reduction and clustering using UBM dealt with them successfully. Another side effect added was a longer running time - due to the iterative nature of the algorithm, compared to the original single-run based algorithm, the running time was extended to the point where it is not possible to use it for online change point detection but only off-line, which was its purpose in the first place.

References

[1] X. Anguera Miro, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 2, pp. 356–370, 2012.

[2] S. Meignier, D. Moraru, C. Fredouille, J.-F. Bonastre, and L. Besacier, "Step-by-step and integrated approaches in broadcast news speaker diarization," *Computer Speech & Language*, vol. 20, no. 2, pp. 303–330, 2006.

[3] M. Kotti, E. Benetos, C. Kotropoulos, and L. G. Martins, "Speaker change detection using bic: A comparison on two datasets," in *Proc. 2006 IEEE Int. Symp. Communications, Control, and Signal Processing*, 2006.

[4] L. K. Saul and S. T. Roweis, "Think globally, fit locally: unsupervised learning of low dimensional manifolds," *The Journal of Machine Learning Research*, vol. 4, pp. 119–155, 2003.

[5] S. Yan, D. Xu, B. Zhang, and H.-J. Zhang, "Graph embedding: A general framework for dimensionality reduction," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 830–837.

[6] Y. Wang, X. Huang, C.-S. Lee, S. Zhang, Z. Li, D. Samaras, D. Metaxas, A. Elgammal, and P. Huang, "High resolution acquisition, learning and transfer of dynamic 3-d facial expressions," in *Computer Graphics Forum*, vol. 23, no. 3. Wiley Online Library, 2004, pp. 677–686.

[7] A. Jansen and P. Niyogi, "A geometric perspective on speech sounds," *University of Chicago, Tech. Rep*, 2005.

[8] S. Chen and P. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the bayesian information criterion," in *Proc. DARPA Broadcast News Transcription and Understanding Workshop*. Virginia, USA, 1998, p. 8.

[9] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1, pp. 19–41, 2000.

[10] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using gmm supervectors for speaker verification," *Signal Processing Letters, IEEE*, vol. 13, no. 5, pp. 308–311, 2006.

[11] "[online]. available: http://www.http://festvox.org/."

[12] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *Signal Processing Letters, IEEE*, vol. 6, no. 1, pp. 1–3, 1999.

[13] F. Valente and C. Wellekens, "Variational bayesian speaker clustering," in *ODYSSEY04-The Speaker and Language Recognition Workshop*, 2004.