Implicit Segmentation in Two-Wire Speaker Recognition

Yosef A. Solewicz[1], *Hagai Aronowitz*[2]

[1] Technology Section, Israel National Police, Jerusalem, Israel
[2] IBM Research - Haifa, Haifa 31905, Israel

solewicz@police.gov.il, hagaia@il.ibm.com

Abstract

This paper presents a novel self-contained two-wire speaker recognition framework. The classical approach to two-wire speaker recognition usually requires a preliminary explicit speaker segmentation stage in order to extract audio files for the two hypothesized speakers. We propose an implicit speaker segmentation method implemented at the supervector level of speaker recognition systems. By periodically extracting successive supervectors from the two-wire audio it is possible to further associate them to each of the hypothesized speakers before scoring both streams. We show that the proposed technique leads to recognition performance comparable to standard approaches while requiring substantially less resources.

Index Terms: speaker recognition, speaker diarization, two-wire

1. Introduction

Use of speaker segmentation in two-wire sessions is essential in scenarios such as call-center transactions or network monitoring. Speaker segmentation has being traditionally performed based on a combination of the following steps: speaker change point detection, agglomerative speaker clustering and Viterbi re-segmentation [1]. Recently, more sophisticated segmentation approaches based on Factor analysis have been introduced [2]. These approaches model the speaker in a low dimensional space thus leading to more robust models since less parameters must be estimated.

We focus on the specific task of two-wire speaker recognition, in the sense that the speaker segmentation output is not an objective per se. To date, this problem is tackled simplistically as a two-stage uncoupled process. Initially, a first stage of blind speaker diarization is applied to the two-wire conversation and its output is driven to an independent speaker recognition system. Nevertheless, it was already suggested [2] that this scheme might be suboptimal for two-wire speaker recognition and that the segmentation quality might not be directly correlated with the recognition performance.

This paper continues previous papers addressing the task of two-wire recognition without explicitly using speaker segmentation [3, 4]. It presents a simple approach which integrates an implicit preliminary segmentation stage into a supervector based speaker recognition system. The segmentation is not performed explicitly, although rough speaker segmentation can be obtained if desired. This system and a baseline system are described in Section 2 of this paper. In Section 3, we report experiments performed using the NIST 2005 Speaker Recognition Evaluation. Finally, conclusion and future plans are presented in Section 4.

2. Explicit and implicit segmentation

In this section we initially describe a baseline segmentation system which is conceptually similar to the proposed system, in the sense that both systems rely on PCA (Principal Component Analysis) in the supervector space to perform segmentation. Nevertheless, while the baseline segmentation system is a traditional explicit speaker segmentation system, the proposed system is quite straightforward dispensing with the common segmentation steps.

2.1. Explicit speaker segmentation

Our explicit speaker diarization system is based on unsupervised compensation of intra-speaker within-session variability followed by PCA and is described in detail in [5]. The system assumes that only two speakers exist in a given session and exploits this knowledge by performing clustering into two clusters. The clustering is done using GMM-supervectors extracted for evenly overlapping one second segments. Intra-speaker within-session variability is estimated in an unsupervised manner and removed from the GMM-supervectors using the NAP method. The segmentation is smoothed using Viterbi segmentation, and refined by applying Viterbi re-segmentation using the original frame-based features. The outline of the algorithm is as follows:

1. Compute standard frame-based features without channel normalization.
2. Detect and remove non-speech frames.
3. Estimate a session-dependent UBM.
4. Divide the audio session into 1 second overlapping superframes and estimate a GMM-supervector for each superframe.
5. Estimate and compensate intra-speaker within-session variability from the superframes (details in [5]).
6. Compute the covariance matrix of the compensated supervectors.
7. Apply PCA to find the eigenvector corresponding to the largest eigenvalue of the covariance matrix from step 6.
8. Project each compensated supervector onto the eigenvector found in step 7.
9. The outcome of step 8 is converted to a LLR (log-likelihood ratio) with respect to the two speakers using a linear transformation.
10. Viterbi segmentation is used to convert the superframe-based LLRs into a smoothed segmentation.
11. Perform a few iterations of adaptation and Viterbi re-segmentation in the original feature space.

2.2. Implicit speaker segmentation

In this subsection, we describe the proposed integrated two-wire segmentation and recognition system referred to as the implicit diarization-based system. In short, this method is based on parameterization of a test conversation into successive time-windowed supervectors instead of the usual single supervector. These supervectors are projected onto the principal component axis describing their highest variance. The projection values are finally used to group the supervectors into two clusters, which are scored against the hypothesized speaker model. The final score is the maximum between the averaged score of both supervector clusters. The whole process is schematically summarized in Figure 1.

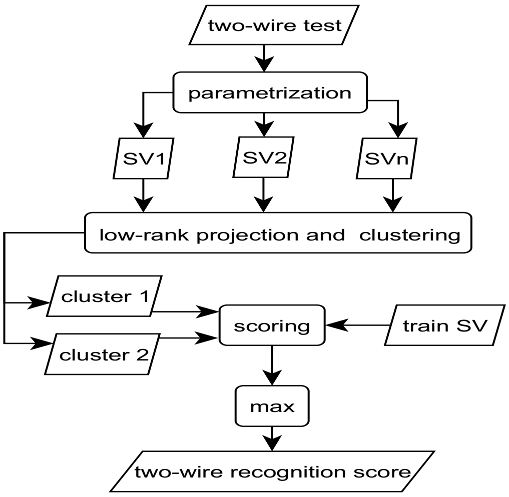

Figure 1: *Implicit two-wire recognition scheme.*

In more detail, we segment an incoming two-wire test session into overlapping time windows (superframes) of typically a few seconds in length. Supervectors are computed for each superframe, using a lower than usual relevance factor for the MAP process, due to the relatively low amount of data available for adaptation. PCA is then used on the computed supervectors to find the fundamental eigenvector defining the direction of maximum data variance. We project all supervectors on the main eigenvector axis. Ideally, we expect to obtain two projection clusters at both axis extremities, corresponding to the two conversation sides. In practice, less defined projections occur and should be assigned to each of the speakers or possibly discarded. It is worth to note that this method is fairly transparent to the recognition process. It requires that the original testing supervector be extracted in time slices and classified before scoring. These steps can be efficiently implemented without adding excessive computational load to the existing recognition system.

2.3. Low-rank projection and clustering

The key stage in the segmentation process described in the previous subsection is classifying the individual supeframes according to speaker identity. Our task is to classify relatively few high-dimensional vectors enclosed in a very low-rank subspace spanned by the two speakers. The complexity involved can be efficiently reduced by projecting the successive supervectors along the axis containing most of the data-variability. Intuitively, projected supervectors associated to each of the speakers will be positioned apart along this axis. This projection actually shapes a one-dimensional function in time which can then be used to assign each supervector to the proper speaker.

In more detail, the low-rank projection operation is performed through PCA. The succession of supervectors is projected onto the fundamental eigenvector to obtain a series of projection coefficients in time denoted by $\{X_i\}$. Although more sophisticated schemes like bi-Gaussian modeling could be applied for classification, in this implementation the respective supervectors are assigned to either of the speakers by means of a simple threshold test as follows:

$$X_i \gtrless (\min(X)+\max(X))/2. \qquad (1)$$

We found it useful to introduce a noise floor around the threshold and discard supervectors whose projections fall below this tolerance. These supervectors presumably contain noise or a balanced amount of speech from both speakers. In our experiments we eliminate supervectors having a projection with absolute value below 5% of the range of $\{X_i\}$. For illustration, an example of a projection function can be seen in Figure 2.

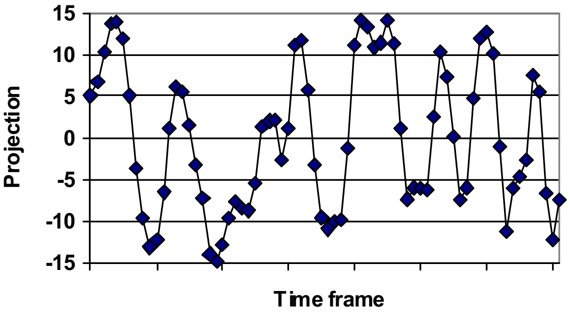

Figure 2: *An example of the projection function.*

After classification, the supervectors are scored against the hypothesized target speaker. The individual scores belonging to each cluster are averaged to obtain a unique score for each speaker. The maximum of both scores is the ultimate recognition score.

A soft segmentation scheme was also evaluated. In this mode, we do not perform a hard classification of the testing supervectors, rather the projection value X_i is used as a confidence value for each supervector regarding which speaker it belongs to. The final score for each speaker is a weighted sum of all supervector scores, weighted by the confidence that a specific supervector belongs to him. In fact, this approach performed poorer than the hard segmentation approach.

3. Experiments

3.1. System set-up and protocols

The speaker recognition system used in all the reported experiments is a GMM-NAP-SVM supervector system quite similar to the one described in [6]. The GMM subsystem is based on [7] and the number of Gaussians is 512. Two-wire NAP [4] is used instead of standard NAP. Each supervector is normalized by the corresponding standard deviation of the UBM Gaussians. We use LIBSVM [8] to implement the SVM subsystem, which renders the actual recognition score, with no further normalizations applied. The front end is based on Melfrequency cepstrum coefficients (MFCC). An energy-based voice activity detector is used to locate and remove non-speech frames. The final feature set consists of 13 cepstral coefficients augmented by 13 delta cepstral coefficients extracted every 10ms using a 25ms window. Finally, standard feature warping [9] is applied.

The experiments reported in this paper consist of the male subset of the NIST-2005 SRE protocol [10]. There are overall 274 target speakers and 965 conversation tests, producing 951 target and around 8000 impostor scores. Moreover, since this evaluation is designed for four-wire sessions, in order to obtain two-wire testing sessions for our experiments, we artificially sum the two sides of the testing conversations in the original protocol. (In fact, the speaker present in the second conversation side might be eventually the same speaker present in the specific trial model labeled as an impostor trial). Therefore, in these cases, original four-wire impostor trials mistakenly turn out to be two-wire target trials. Since the evaluation key files lack this piece of information, we drop a few trials for which the corresponding additional side attain a higher recognition score than the maximum score obtained using the original four-wire impostor trials.)

The NIST-2004 and 2006 SREs corpora [10] were used as development datasets. Specifically, SRE'04 was used for training the UBM (240 conversations), SVM background modeling (200 conversations) and for inter-session variability modeling (124 male speakers, 1457 conversations in total), while the SRE'06 dataset was used for speaker space modeling (350 conversations).

3.2. Segmentation resolution

There are a few parameters to be optimized in the proposed segmentation scheme. Initially, we should define the resolution of the implicit segmentation process, determined by the length and overlap ratio of the superframes from which the supervectors are estimated. In addition, decreasing the length of the excerpts demands a more aggressive relevance factor used in MAP. We recall that our ultimate goal is to improve recognition performance rather than obtaining fully segmented conversations. Therefore, we wish to increase as much as possible the length of the superframes in order to decrease computational load, while not excessively degrading the recognition performance. In addition, long adjacent windows being relatively uncorrelated, alleviates the needs of Viterbi segmentation Although extensive parameter optimization was not performed, we observed that superframe lengths of a few seconds attain good recognition results given relevance factors around 0.5. For illustration, Table 1 depicts speaker recognition performance for different superframe lengths with a 50% overlap and using a relevance factor of 0.5. Throughout the rest of the paper we use superframes of length 5 seconds with 50% overlap and a relevance factor of 0.5.

Superframe length (sec.)	DCF (x10^{-4})	EER (%)
2	258	6.32
3	228	6.48
5	219	6.05
10	259	7.12

Table 1: *Performance of the proposed speaker recognition system based on implicit speaker diarization for different superframe lengths.*

3.3. Results

In this subsection, we report experiments evaluating both the proposed integrated speaker recognition system based on implicit speaker diarization and the baseline speaker recognition system based on explicit speaker diarization described in Section 2. Throughout our experiments training is done using 4-wire training session. We evaluate several testing conditions. In particular, 4-wire (only the side of interest), 2-wire using a manual segmentation, 2-wire without speaker diarization, 2-wire using either explicit or implicit speaker diarization, and a fusion system which simply averages the output scores of the explicit and implicit segmentation systems.

DCF and EER results for each testing condition are presented in Table 2. It can be observed that both the explicit and implicit segmentation techniques lead to similar recognition performance and seem to be complementary as suggested by the improved fused performance.

Condition	DCF (x10^{-4})	EER (%)
4-Wire	156	4.68
Manual seg.	183	5.41
No seg.	307	9.75
Explicit seg.	209	6.15
Implicit seg.	219	6.05
Fused seg.	192	5.51

Table 2: *Performance for different testing conditions. Training is done using 4-wire data*

Deeper understanding can be achieved by analyzing the recognition performance as a function of the explicit segmentation accuracy of each conversation. For this purpose, we split the testing conversations into accurately and inaccurately segmented conversations and then evaluate recognition performance for each class. The averaged segmentation accuracy of both conversation sides processed by the explicit segmentation system was used for classifying each testing conversation. Conversations with segmentation accuracy greater than 90% were labeled as accurately segmented and the remaining conversations were labeled as inaccurately segmented. Nearly 90% of the conversations are labeled as accurately segmented. Note that inaccurately segmented conversations usually contain a dominant speaker.

Recognition performances for both classes are depicted in Table 3. In particular, the results suggest that while the explicit diarization approach might be slightly more efficient in cases where the conversations are accurately segmented, the implicit approach seems to be more robust when explicit diarization fails.

Condition	DCF (x10^{-4})	EER (%)
4-Wire	148 / 223	4.42 / 4.18
Manual seg.	176 / 225	5.05 / 7.48
No seg.	289 / 447	9.01 / 14.43
Explicit seg.	186 / 389	5.06 / 12.09
Implicit seg.	204 / 328	5.71 / 10.19
Fused seg.	175 / 319	5.04 / 9.12

Table 3: *Performance for different testing conditions for accurately / inaccurately segmented test sessions*

3.4. Towards on-line speaker diarization

The explicit speaker diarization algorithm consists of the following attributes that make it hard to run online:

1. Session dependent UBM estimation.
2. Estimation and compensation of intra-session intra-speaker variability.
3. Application of PCA for the covariance matrix of the compensated supervectors.
4. Iterative Viterbi re-segmentation.

On the other hand, the implicit diarization approach entirely relies on the third component (PCA on segments' supervectors). Proper estimation of the fundamental eigenvector spanning both speakers is the primal factor affecting performance in this method.

In this section, we investigate the feasibility of applying PCA on relatively short prefixes of the audio sessions. Therefore, we consider using just a fraction of all the available supervectors for estimating a unique fundamental eigenvector for the whole conversation. In practice, we store the initial supervectors extracted over the conversation for PCA computation and keep the estimated fundamental eigenvector during the whole conversation. Furthermore, in order to estimate conversation dependent thresholds using (1), the range of the superframe projections must be known in advance which complicates latency requirements. We therefore simplify the classification decision by using a fixed threshold for superframe classification. In particular, we fixed the threshold decision to zero and additionally discarded the above mentioned noise floor which is also a function of the projection function range.

Recognition performance as a function of distinct time delays is presented in Figure 3. The time delay is actually reflected by the amount of initial supervectors stored for PCA computation. The worst recognition performance (leftmost point in the graph) is obtained for window size equal to zero (no PCA and no implicit segmentation), and the best performance (rightmost point in the graph) is defined by using all the conversation (170 seconds, in average) for PCA estimation.

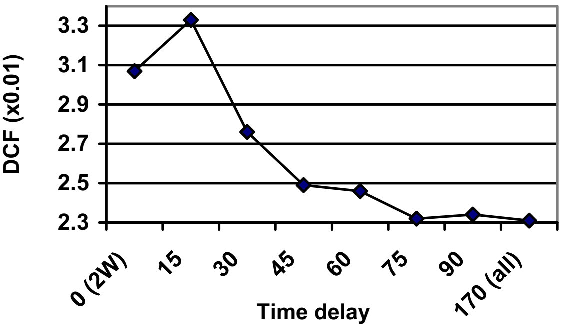

Figure 3: *Recognition performance as a function of window size for PCA estimation.*

This experiment suggests that roughly 45 seconds of two-wire speech is enough for a reasonable PCA estimation considering our simplified real-time implementation. By comparing the performance between the original approach and the real-time version using all supervectors, we also observe that the use of a fixed threshold and absence of the noise floor causes a modest drop in performance.

In this latency consideration we addressed the implicit segmentation module functionality. In fact, the length of the audio excerpt used for testing has also influence on recognition performance and should be addressed in a comprehensive latency analysis.

4. Conclusions and future work

In this paper we presented a simple method for performing implicit speaker segmentation for two-wire speaker recognition tasks. The method does not pursue optimal and explicit speaker segmentation and therefore is simpler than the explicit approach. Apparently, it can be implemented for online applications with delays of about 75 seconds with little loss in recognition performance. The efficiency of the proposed framework is better than that of the standard procedure for two-wire speaker recognition, which is the application of explicit speaker segmentation followed by speaker recognition. In particular, this result confirms the claim that optimal speaker segmentation is not mandatory for efficient two-wire speaker recognition. Moreover, we showed that the novel implicit and the standard explicit segmentation schemes seem to complement each other towards optimal two-wire recognition performance.

In future research we plan to reduce the required latency by exploring methods for improved estimation of the fundamental eigenvector with limited available data. In particular, we plan to incorporate the use of prior information in the estimation process and consider the use of recursive PCA techniques. In addition, we hope to extend this method to conversations with more than two speakers.

5. Acknowledgements

The authors would like to thank the anonymous referees for helpful comments.

6. References

[1] S. Tranter and D. Reynolds, "An overview of automatic speaker diarization systems," IEEE Trans. Audio, Speech, Lang. Process., vol. 14, no. 5, pp. 1557–1565, Sept. 2006.

[2] D. Reynolds, P. Kenny, F. Castaldo. "A study of new approaches to speaker diarization," in *Proc. Interspeech*, 1047-1050, 2009.

[3] H. Aronowitz and Y. Solewicz, "Speaker recognition in two wire test sessions," in *Proc. Interspeech*, pp. 865–868, 2008.

[4] Y. Solewicz and H. Aronowitz, "Two-wire Nuisance Attribute Projection," *Proc. Interspeech*, 928-931, 2009.

[5] H. Aronowitz, "Unsupervised Compensation of Intra-Session Intra-Speaker Variability for Speaker Diarization", in Proc. *Odyssey*, 2010.

[6] N. Brummer, L. Burget, J. Honza Cernocky, O. Glembek, F. Grezl, M. Karafiat, D. A. van Leeuwen, P. Matejka, P. Schwarz, A. Strasheim,, "Fusion of Heterogeneous Speaker Recognition Systems in the STBU Submission for the NIST Speaker Recognition Evaluation 2006", in IEEE Trans. on Audio, Speech & Language Processing, September 2007.

[7] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models", in Digital Signal Processing, Vol. 10, No.1-3, pp. 19-41, 2000.

[8] C. C. Chang and C. J. Lin, LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[9] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in Proc. *ISCA Odyssey Workshop*, 2001, pp. 213-218.

[10] Available online: http://www.ist.gov/speech/tests/sre/.