On the Use of Spectral and Iterative Methods for Speaker Diarization

Stephen Shum, Najim Dehak, Jim Glass

MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA, USA
{sshum, najim, glass}@csail.mit.edu

Abstract

This paper extends upon our previous work using i-vectors for speaker diarization. We examine the effectiveness of spectral clustering as an alternative to our previous approach using K-means clustering and adapt a previously-used heuristic to estimate the number of speakers. Additionally, we consider an iterative optimization scheme and experiment with its ability to improve both cluster assignments and segmentation boundaries in an unsupervised manner. Our proposed methods attain results similar to those of a state-of-the-art benchmark set on the multi-speaker CallHome telephone corpus.

Index Terms: speaker diarization, factor analysis, Total Variability, spectral clustering

1. Introduction

Our previous work proposed an approach to speaker diarization that performs speaker clustering directly in a low-dimensional factor analysis-based subspace (i.e. the Total Variability subspace) [1]. We showed that this method is not only simpler than the Variational Bayes-based system formulated in [2], but can also achieve the same state-of-the-art diarization performance on summed-channel telephone data from the 2008 NIST SRE. Such success, however, was limited to the task in which we knew there were exactly two speakers in the given conversation. To solve the diarization problem in general, we must address the setting in which the number of participating speakers is unknown *a priori*.

In this paper, we extend the previous system described in [1] to approach the more general problem in incremental fashion. First, we motivate the use of a spectral clustering algorithm as an alternative to the previous approach involving K-means clustering based on the cosine distance. More importantly, we adapt a heuristic from previous work applying spectral clustering to diarization and use it to determine the number of clusters (i.e. speakers) [3]. Second, we experiment with utilizing the information at different temporal resolutions to evaluate the hypothesis that there exists a symbiotic relationship between clustering and segmentation; that is, better initial segmentations should yield better speaker clusters, and conversely, better speaker clusters should aid in providing cleaner speaker segments.

The rest of this paper is organized as follows. Section 2 provides a brief review of the Total Variability approach as a factor analysis-based front-end for extracting speaker-specific features, and Section 3 outlines the general setup of our diarization system. Section 4 motivates the spectral clustering algorithm and its associated heuristic to estimate the number of speakers. In Section 5, we propose the use of an iterative re-segmentation/clustering algorithm to refine speaker segments and clusters. The results of our experiments are analyzed and explained in Section 6, while Section 7 concludes with a discussion of future work.

2. A Review of Total Variability

To enhance the classical method of modeling speakers using Gaussian Mixture Models (GMMs) [4], recently developed methods apply factor analysis to supervectors - a vector consisting of stacked mean vectors from a GMM - in order to better represent speaker variabilities and compensate for channel (or session) inconsistencies [5]. One such approach is Total Variability, which decomposes a speaker- and session-dependent supervector M as follows:

$$M = m + Tw \tag{1}$$

where m is the speaker- and session-independent supervector commonly taken from a large GMM trained to represent the speaker-independent distribution of acoustic features [4]. T is a rectangular matrix of low rank that defines the Total Variability subspace, and w is a low-dimensional random vector with a standard normal prior distribution $\mathcal{N}(0, I)$ that is referred to as a *total factor vector* or an *i-vector* [5]. For some speech segment s, its associated i-vector w_s can be seen as a low-dimensional summary of the speaker's distribution of acoustic features.

The cosine similarity metric has been applied successfully in the Total Variability subspace to compare two i-vectors [5]. Given any two total factor vectors w_1 and w_2, the cosine similarity score is given as

$$\texttt{cos\_score}(w_1, w_2) = \frac{(w_1)^t (w_2)}{\|w_1\| \cdot \|w_2\|} \tag{2}$$

3. System Setup

We set up our diarization framework to be consistent with our previous work in [1] with just a few modifications. The rest of this section outlines the various parts of the system.

3.1. Segmentation

In order to focus solely on the speaker confusion portion of the Diarization Error Rate (DER) and not be misled by mismatches between the reference speech/non-speech detector and our own (i.e. miss and false alarm errors), we follow the convention of previous works [2, 6] and use the provided reference boundaries to define our initial segmentation. Each segment is restricted to a maximum length of one second, and an i-vector is extracted for each segment. It should be noted that this rather crude initial segmentation may result in segments that contain speech from more than one speaker.

3.2. Clustering

This paper discusses two different clustering algorithms: (a) K-means clustering based on the cosine distance and (b) spectral clustering. Our subsequent results will specify exactly which method is used for each experiment.

3.3. Re-segmentation

We use the exact same re-segmentation algorithm discussed in both [1, 2] to refine our initial segmentation boundaries. At the acoustic feature level, this stage initializes a 32-mixture GMM for each of the $K+1$ clusters (Speakers $\{S_1, ..., S_K\}$ and non-speech NS) defined by the previous clustering. Posterior probabilities for each cluster are then calculated given each feature vector x_t (i.e. $P(S_1|x_t), ..., P(S_K|x_t), P(NS|x_t)$) and pooled across the entire conversation, providing a set of Baum-Welch statistics from which we can re-estimate each respective speaker's GMM. In order to prevent this unsupervised procedure from going out of control, the non-speech GMM is never re-trained. In the Viterbi stage, each frame is assigned to the speaker/non-speech model with the highest posterior probability. This algorithm runs until convergence but is capped at 20 Viterbi iterations, each of which involves 5 iterations of Baum-Welch re-estimation.

3.4. Final Pass Refinements

As in [1], we can further refine the diarization output by extracting a single i-vector for each respective speaker using the (newly-defined) segmentation assignments. The i-vector corresponding to each segment (also newly extracted) is then re-assigned to the speaker whose i-vector is closer in cosine similarity. We iterate this procedure until convergence - when the segment assignments no longer change. This can be seen as another pass of K-means clustering, where the "means" are computed according to the process of i-vector estimation.

4. Towards Spectral Clustering

Spectral clustering has the ability to handle complex and unknown cluster shapes where other commonly-used methods such as K-means and mixture-modeling may fail. Rather than estimating some explicit model of data distribution, this technique relies on analyzing the eigen-structure of an affinity matrix.

4.1. The Algorithm

Below, we outline a slightly modified version of the Ng-Jordan-Weiss spectral algorithm [7]:

0. Assume we are given n i-vectors $\{w_1, ..., w_n\}$ (each corresponding to a speech segment $\approx$1sec in length) that we want to cluster into K subsets.
1. Form the affinity matrix $A \in \mathbb{R}^{n \times n}$, where $A_{ij} = \exp\left(-d\left(w_i, w_j\right)^2 / \sigma^2\right)$ when $i \neq j$ and $A_{ii} = 0$. Here, $d\left(w_i, w_j\right) = 1 - \texttt{cos\_score}\left(w_i, w_j\right)$, where `cos_score` is given by (2).
2. Define D to be the diagonal matrix whose (i,i)-element is the sum of A's i-th row, and construct the matrix $L = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$.
3. Find $x_1, ..., x_K$, the K largest eigenvectors of L and form the matrix $X = [x_1, ..., x_K] \in \mathbb{R}^{n \times K}$.
4. Create the matrix Y from X by re-normalizing each of X's rows to have unit length (i.e. $Y_{ij} = X_{ij} / (\sum_j X_{ij}^2)^{1/2}$).
5. Treating each row of Y as a point in $\mathbb{R}^K$, cluster them into K clusters via K-means.
6. Assign the original i-vector w_i to cluster k if and only if row i of Y is assigned to cluster k.

Instead of the Euclidean distance that is specified in Step 1 of [7], the cosine distance is more appropriately suited for our i-vector data. Picking a scaling factor σ^2 is easy in our case because a cosine distance can be no larger than 2. As such, we simply pick a σ^2 such that $\exp\left(-2^2/\sigma^2\right) \leq \epsilon$ for some reasonable value of ϵ. In our experiments, we simply set $\sigma^2 = 0.5$.

A more detailed analysis of the algorithm is presented in [7]; we briefly provide some intuition as to how it works. In the ideal case of K clusters where all points in different clusters are infinitely far apart, we would have an affinity matrix $\tilde{A}$ that is block-diagonal. Each of the K blocks, $\tilde{A}_k$, is a matrix of "intra-cluster" affinities for cluster k. Subsequently, $\tilde{L}$ will have the same structure, so its eigenvalues and eigenvectors will be the union of the eigenvalues and eigenvectors of its blocks (the latter padded appropriately with zeros) [7]. Furthermore, this will result in K eigenvalues equal to 1 with each of their corresponding eigenvectors spanning one of the K distinct blocks in $\tilde{L}$. We generate $\tilde{X}$ from these K eigenvectors of $\tilde{L}$ and subsequently obtain $\tilde{Y}$ by normalizing the rows of $\tilde{X}$. Now, the i-th row of $\tilde{Y}$ (corresponding to the i-th data point or i-vector) is 1 for the k-th column if segment i belongs in cluster k and 0 everywhere else. As such, each of the clusters will be located at orthogonal locations on the K-dimensional hypersphere.

4.2. Estimating the Number of Clusters

To estimate the number of clusters $\hat{K}$, we adapt from the work in [3], which demonstrated the usefulness of a simple heuristic. It was seen experimentally that the n sorted eigenvalues of L, say $\Lambda = \{\lambda_1, ..., \lambda_n\}$, exhibit exponential decay and that the number of speakers in a conversation correspond consistently to when the gradient of these eigenvalues exceeds some threshold θ. As such, to determine the number of clusters, we can fit a smooth exponential $\exp(-\alpha k)$ to Λ, where $k = 1, ..., n$ and $\alpha \in [0.1, 10]$. We then take $\hat{K}$ to be the smallest value whose derivative $-\alpha \exp(-\alpha \hat{K}) \geq \theta$.

5. Iterative Optimization

The use of factor analysis for speaker diarization allows us to take advantage of multiple levels of speaker information. I-vectors are designed to provide information specific to speaker (and channel) identity, which is important for clustering; however, the effectiveness of an i-vector is proportional to the length of the speech segment from which it is extracted, thus it is not as well-suited for issues requiring finer temporal resolution (e.g. speaker change detection). By contrast, lower-level acoustic features such as MFCCs are not quite as good for discerning speaker identities, but can provide sufficient temporal resolution to witness local speaker changes and segment boundaries.

We can evaluate the validity of this idea by formulating an algorithm that optimizes both segmentation boundaries and segment cluster assignments in iterative fashion. More specifically, we can alternate between clustering (done at the i-vector level) as described in Section 3.2 and applying the re-segmentation method (done at the acoustic feature level) as described in Section 3.3 until successive diarization hypotheses "converge." To understand this notion of convergence, let us approximate a

"distance" between two diarization hypotheses D_1 and D_2 by running it through a diarization evaluation script as provided by NIST [8]. Then we can define a "convergence" to be when this error rate (i.e. DER(D_1, D_2)) is below some threshold γ. In our experiments, we set $\gamma = 8\%$ and allow a maximum of 20 total iterations.

It should be noted that the re-segmentation output from Section 3.3 includes both segment boundaries and corresponding cluster assignments. During this iterative optimization process, however, the assignment labels from the re-segmentation output are not input to the clustering stage - only the segment boundaries are considered. Lastly, the number of speakers is also re-estimated at the start of each clustering stage.

6. Experiments

In order to use the same telephone-based Total Variability framework from before and utilize the state-of-the-art results from [6] as a benchmark for comparison, we evaluate our system on the 2000 NIST SRE subset of the multilingual CallHome data, a corpus of multi-speaker telephone conversations. This amounts to 500 recordings, each 2-5 minutes in length, containing between two and seven participants [9]. Furthermore, we break down our results to show diarization performance on conversations involving the different numbers of speakers.

We obtain our i-vectors using the same Total Variability matrix T of rank 100 that achieved the best results in [1]. This matrix was trained from a gender-independent UBM of 1024 Gaussians built solely on 20-dimensional MFCC feature vectors without derivatives.

6.1. First Pass Comparisons

We begin by comparing the effectivenes of K-means and spectral clustering when the number of speakers is given. The plot at the top of Figure 1 shows that K-means clustering outperforms spectral clustering on conversations involving any number of speakers. This is actually not too surprising; one of the motivations behind spectral clustering is to address the inability of standard K-means to separate clusters that are not linearly distinguishable in the input space [7]. In our case, however, we have always been operating on the assumptions that i-vectors live on the unit hypersphere and that the cosine distance between two respective i-vectors is a valid measure of their distance. Thus, as confirmed by the plot at the top of Figure 1, there is no reason why the K-means algorithm based on the cosine distance would not provide a result that is at least as good as that of spectral clustering.

It becomes quite clear, then, that what we really want out of the spectral algorithm is an estimate of the number of speakers based on the eigenvalues of the normalized affinity matrix L. As such, we develop a hybrid approach that combines the respective advantages of both methods: the normalized affinity matrix from spectral clustering provides an estimate of the number of speakers $\hat{K}$, while the K-means algorithm does the actual clustering based on the cosine distance. The plot at the bottom of Figure 1 verifies this claim. For each conversation, the number of speakers $\hat{K}$ is estimated via the method detailed in Section 4.2, which is then used as an input to both the K-means and spectral clustering algorithms. In the DER sense,[1] K-means

[1]Unfortunately, due to the restricted length of this paper, we are unable to show how accurately this method can estimate the number of speakers. Instead, we resort to using DER as an indicator of speaker estimation accuracy.

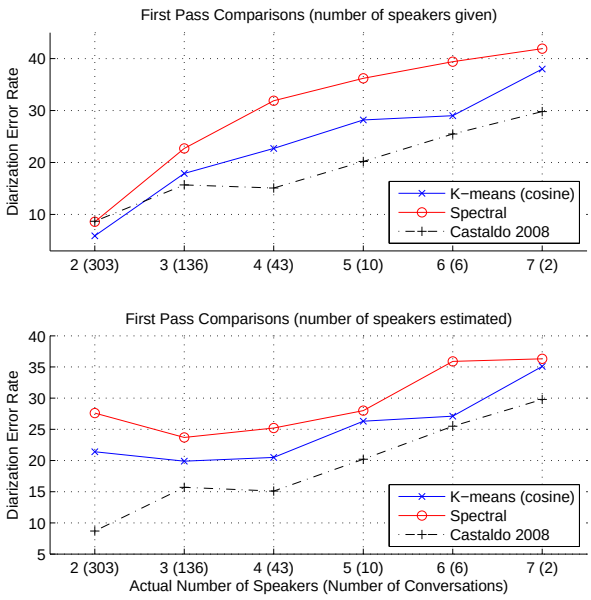

Figure 1: *Results obtained using first-pass clustering where the number of speakers was given a priori (Top) or estimated as explained in Section 4.2 (Bottom).*

clustering outperforms the spectral method once again. From now on, we exclusively use K-means as our clustering method.

6.2. Iterative Optimization

Figure 2 shows the result of evaluating the proposition explained in Section 5. Using K-means clustering, we consider the cases in which the number of speakers is given (Top) and estimated (Bottom). Unfortunately, the results show no obvious trends for either scenario. Upon a more detailed analysis, we saw that the iterative optimization does improve the DER, but only for conversations that already had reasonably good diarization hypotheses (e.g. calls containing two or three speakers). For the conversations whose initial hypotheses were rather poor, however, the unsupervised nature of this technique leads to somewhat unpredictable outcomes. It looks as though a more in-depth study of this approach is warranted.

6.3. Final System

Lastly, we integrate a final pass of refinements (Section 3.4) to obtain our ultimate diarization result, as shown in Figure 3. The configuration of our best-performing system depends on the amount of information provided at the beginning. When the number of speakers is given *a priori*, the best results are obtained without the use of iterative optimization. However, when the number of speakers needs to be estimated, the best results are obtained with the iterative optimization step. It seems as though the repeated iterations give the system more opportunity to re-estimate the number of speakers using (hopefully) cleaner and more refined speech segments.

Whether intentional or incidental, the system in [6] was designed to take advantage of the structure of telephone conversations. In particular, most speaker turns over the telephone involve no more than two participants at any given time. The

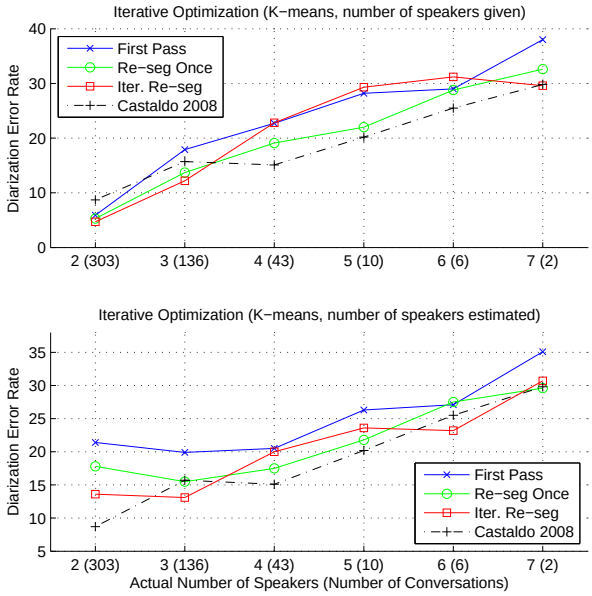

Figure 2: *Plots comparing the results obtained by running re-segmentation (Section 3.3) a single time (green) and running the iterative re-segmentation/clustering method from Section 5 (red).*

system in [6] processes these calls in causal fashion, working on 60-second slices and assuming that each slice contains no more than three speakers. Given the nature of the data, this makes sense; barring the rare use of speakerphones, only during very infrequent "hand-offs" would a third speaker even exist in any particular slice of the conversation.

By contrast, our algorithm sees and processes an entire utterance at once and performs clustering without any regard to the potentially restrictive temporal dynamics of a telephone conversation (i.e. "bag of i-vectors"). This method may be a slightly more general approach; however, it is prone to missing speakers that, say, only participate in a very short snippet of the conversation. One conceivable way to improve our system in this regard might be to run an initial clustering on shorter, say 60-second, slices of conversation before clustering on the entire utterance.

7. Conclusions

Our experiments evaluated the effectiveness of a spectral algorithm for both clustering and estimating the number of speakers, as well as a method for iteratively optimizing re-segmentation and clustering. The system that performed best uses the normalized affinity matrix from the spectral algorithm to estimate the number of speakers before clustering with K-means based on the cosine distance. Segment boundaries and cluster assignments are iteratively optimized until convergence; and lastly, we run final pass refinements to obtain our diarization output.

The methods described in this paper approach the benchmark results set by [6] and provide many avenues for future work. For one, it may be fruitful to consider extending from K-means to a more probabilistic approach to speaker clustering, such as with a GMM or even a mixture of the von Mises-Fisher distribution, which lies on the unit hypersphere [10]. We also

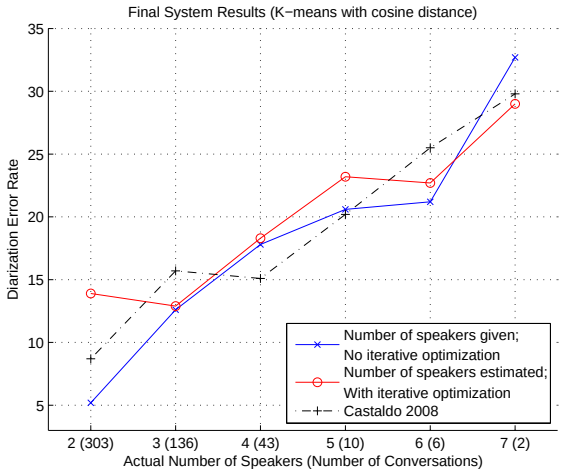

Figure 3: *Final results obtained using K-means (cosine) clustering and a stage of final pass refinements.*

alluded previously to the notion of using i-vectors to model the temporal dynamics of a conversation, such as with a Hidden Markov Model; to that end, a viable next step would be to use i-vectors as feature inputs to the models proposed in [11, 12].

Acknowledgments We would like to thank Ekapol Chuangsuwanich and David Harwath for their helpful discussions that led to this paper, Reda Dehak for the continued use of his code, and Douglas Reynolds for his re-segmentation code, continued advice, and support.

8. References

[1] S. Shum, N. Dehak, E. Chuangsuwanich, D. Reynolds, and J. Glass, "Exploiting intra-conversation variability for speaker diarization," in *Proceedings of Interspeech*, 2011.

[2] P. Kenny, D. Reynolds, and F. Castaldo, "Diarization of telephone conversations using factor analysis," *IEEE Journal of Selected Topics in Signal Processing*, Dec. 2010.

[3] H. Ning, M. Liu, H. Tang, and T. Huang, "A spectral clustering approach to speaker diarization," in *Proceedings of ICSLP*, 2006.

[4] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker verification using adapted gaussian mixture models," *Dig. Sig. Proc.*, 2000.

[5] 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*, July 2010.

[6] 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.

[7] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proceedings of NIPS*, 2001.

[8] NIST, "Diarization error rate (der) scoring code," 2006, www.nist.gov/speech/tests/rt/2006-spring/code/md-eval-v21.pl.

[9] A. Martin and M. Przybocki, "Speaker recognition in a multi-speaker environment," in *Proceedings of Eurospeech*, 2001.

[10] A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra, "Clustering on the unit hypersphere using von mises-fisher distributions," *Journal of Machine Learning Research*, 2005.

[11] E. Fox, E. Sudderth, M. Jordan, and A. Willsky, "An HDP-HMM for systems with state persistence," in *Proceedings of the International Conference on Machine Learning*, 2008.

[12] M. Johnson and A. Willsky, "The hierarchical dirichlet process hidden semi-markov model," in *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2010.