Odyssey 2012
The Speaker and Language Recognition Workshop
25-28 June 2012, Singapore

On the use of Agglomerative and Spectral Clustering in Speaker Diarization of Meetings

J. Luque [1,2], J. Hernando[1]

[1] Dept. of Teoria del Senyal i Comunicacions, TALP Research Center, Universitat Politècnica de Catalunya (UPC), Barcelona, (Spain)
[2] Dept. of Matemática Aplicada y Estadística, ETSI Aeronáuticos, Universidad Politécnica de Madrid (UPM), Madrid, (Spain)
{luque,javier}@tsc.upc.edu

Abstract

In this paper, we present a clustering algorithm for speaker diarization based on spectral clustering. State-of-the-art diarization systems are based on agglomerative hierarchical clustering using Bayesian Information Criterion and other statistical metrics among clusters which results in a high computational cost and in a time demanding approach. Our proposal avoids the use of such metrics applying Euclidean distances on the eigenvectors computed from the normalized graph Laplacian. A hybrid system is proposed in which HMM/GMM modelling and Viterbi alignment are still applied, but the BIC for merging and stopping criterion are substituted by a spectral clustering algorithm. Once an initial segmentation is obtained and the clustering alignment is computed using the Viterbi algorithm, the remaining clusters are modeled by stacking the means of the Gaussians in a super vector. In such a space single value decomposition of the associated normalized graph Laplacian is computed. Most similar clusters are merged based on the Euclidean distances in resulting eigenspace. Cluster number estimation is based on analyzing eigenstructure of the similarity matrix by selecting a threshold on the eigenvalues gap. In experiments, this approach has obtained a comparable performance to the traditional AHC+BIC approach on the Rich Transcription conference evaluation data. Although it still relies on Gaussian modelling of clusters and Viterbi alignment, the proposed approach leads to a system which runs several times faster than traditional one.

Index Terms: Speaker diarization, speaker segmentation, speaker clustering, spectral clustering

1. Introduction

Speaker diarization consists in segmenting and labeling an unknown set of speakers in a continuous audio stream trying to answer to the question *who is speaking?* This information is useful in a range of applications such as speaker indexing, information retrieval and speaker adaptation as a pre-processing for the speech content transcription. [1].

Agglomerative hierarchical clustering (AHC) has become one of the most widely applied approach to speaker diarization task. Clusters are represented by parametric probability densities like Gaussian mixture models (GMMs). Hidden Markov Models (HMM) together with Viterbi perform segmentation and clustering of audio data in an iterative bottom-up fashion [2]. In such a framework, Bayesian information criterion (BIC) is one of the most popular metrics to estimate which couple of clusters merge at each agglomerative iteration. BIC is usually also employed as a stopping criterion for the agglomerative process [3]. Metrics like as Generalized likelihood ratio (GLR), Kullback-Leibler (KL) divergence, information change rate (ICR), amongst others, has been also proposed, but all of them with same Achilles' heel, that is, a high computational cost and a performance heavily depending on the choice of the metric [4].

To overcome this drawback, we propose a speaker diarization approach method based on spectral clustering (SC) avoiding the use of computationally demanding statistical metrics like BIC. Spectral clustering refers to a class of techniques which rely on the eigenstructure of a similarity matrix to partition points into disjoint clusters. Points having high similarity are pooled together in the same cluster whereas they evidence a low similarity among other points grouped in different clusters. SC has been successfully applied in blind source separation, separating speech mixtures from a single microphone [5] with no requirement of explicit models for speakers. However, there are a few recent works which use SC to infer speaker clusters specifically in speaker diarization task [6, 7, 8, 9].

Instead of making assumptions on data distribution, SC relies on analyzing the eigenstructure of an affinity matrix [5, 10] which models the similarity among the clusters. Nevertheless, in contrast to classical AHC clustering approaches, such affinity matrix is treated as part of the learning problem. Our proposal is based on a parametric segment representation through a Gaussian super vector (GSV). The GSV vector is composed by stacking just the means of the Gaussians [11]. The classical BIC metric in AHC is replaced by Ng-Jordan-Weiss (NJW) spectral clustering algorithm [5]. In our work, the affinity matrix is built by defining the similarity between segments through the Euclidean distance in the GSV space of segments representation. We employ spectral clustering algorithm with cluster number estimation based on eigenstructure analysis, searching the drop in the magnitude of the eigenvalues as in [7, 9].

Our clustering algorithm still depends on HMM/GMM modelling and Viterbi segmentation as pre and post - processing for spectral clustering. For instance, they are used for obtaining the GSV vector representation per each segment which feed the SC algorithm. In that case, the initial segmentation is computed through a initial partition in homogeneous segments. Such segments are realigned by an HMM/GMM model together with Viterbi decoding up to no variation in segmentation structure is noticed. Finally, it is also applied as a post-processing of spectral clustering results. This approach generates results comparable to AHC+BIC ones but achieves much higher speed than the latter.

Algorithm 1 Agglomerative Hierarchical Clustering (AHC), bottom-up alternative.

Require: $\{\mathbf{x}_i\},\ i = 1 \ldots, \hat{n}$: speech segments
$\hat{C}_i,\ i = 1, \ldots, \hat{n}$: initial clusters
Ensure: $C_i,\ i = 1, \ldots, n$: finally remaining clusters
1: $\hat{C}_i \leftarrow \{\mathbf{x}_i\},\ i = 1, \ldots, \hat{n}$
2: **repeat**
3: $i, j \leftarrow \text{argmin}\ d(\hat{C}_k, \hat{C}_l),\ k, l = 1, \ldots, \hat{n}, k \neq l$
4: merge $\hat{C}_i$ and $\hat{C}_j$
5: $\hat{n} \leftarrow \hat{n} - 1$
6: **until** no more extra cluster merging is needed
7: **return** $C_i,\ i = 1, \ldots, n$

2. AHC diarization based on HMM/GMM and BIC

In Figure 1 we depict an overall scheme of our baseline diarization system based on classical AHC. The system was submitted to Rich Transcription (RT) 2007 and 2009 evaluations with minor changes [12, 13]. In this work, only single distant microphone (SDM) condition is taken into account. It performs the diarization on a mono-channel audio stream which is given by NIST.

Since we are interested in algorithm performances related to speaker clustering, no algorithm is applied for speech activity detection. The diarization reference files, provided by NIST, are applied to produce speech activity labels, avoiding the speech insertions – which produce false alarms errors – and speech deletions – which lead to misses – and creating oracle speech detection labels. Pre-processing of the data consists of Wiener filter denoising for each sdm channel. 19 MFCC features are then extracted from the filtered signal.

The baseline diarization system follows the commonly used agglomerative hierarchical clustering (AHC) approach as explained in Algorithm 1. Firstly, speech is broken into short uniform segments and the successive clustering iterations group acoustically similar segments and assign them to speaker clusters. The main steps of the system can be condensed in the following points:

- *Feature extraction and removal of non-speech frames*. At this stage, a clustering initialization is also performed based on an homogeneous partition of the data (Fig. 1 block A).
- *Complexity selection* of the models based on the amount of data per cluster and the cluster complexity ratio (CCR), which fixes the amount of speech per Gaussian. HMM/GMM training and cluster realignment by Viterbi decoding based on maximum likelihood (Fig. 1 block B).
- *Agglomerative hierarchical clustering* based on the Bayesian information criterion (BIC) metric among clusters. The stopping criterion, also based on the BIC, drives the ending point of the algorithm (Fig. 1 block C).

The number of initial clusters is determined automatically depending on the meeting length with minimal and maximal value constraints. In this work, the total amount of clusters was constrained to a minimum and a maximum of 35 and 65 clusters respectively, aiming to avoid overclustering and to reduce the computational cost of the iterative approach. Each initial cluster is modeled by a mixture of Gaussians, fitting the probability distribution of the features by the classical expectation-maximization (EM) algorithm (Fig. 1 block B). The automatic selection of the initial number of clusters (K_{init}) is defined as,

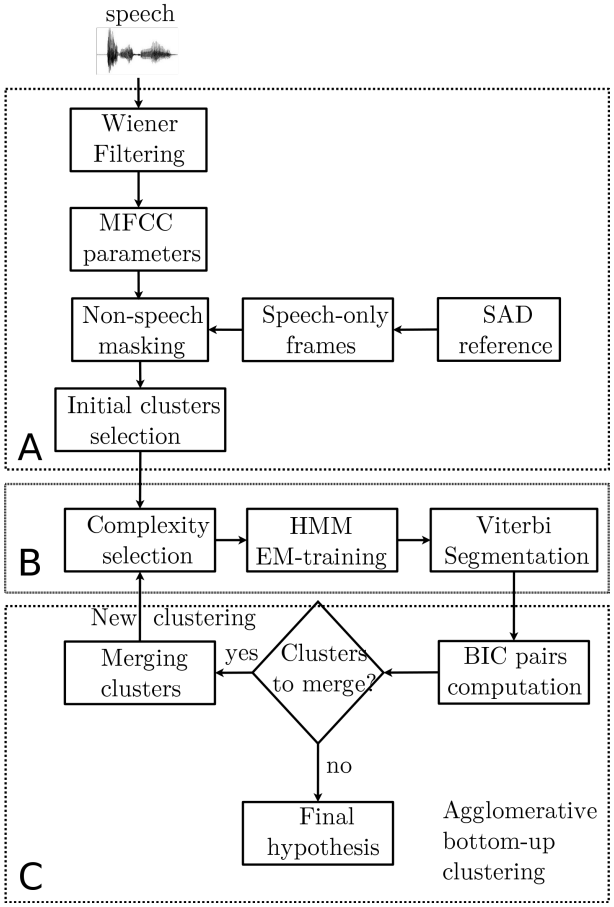

Figure 1: *Speaker diarization scheme based on AHC baseline system with automatic complexity selection.*

$$K_{\text{init}} = \frac{N}{G_{\text{init}}\ R_{\text{CC}}} \tag{1}$$

This expression takes into account the total amount of data available per speaker cluster (N), the number of Gaussian mixtures initially assigned to each speaker cluster (G_{init}) and the cluster complexity ratio (R_{CC}). The R_{CC} is a constant value across all meetings that defines the number of frames per Gaussian. It was fixed to 7 seconds of speech per Gaussian whereas the initial number of Gaussians per model (G_{init}) was set to 5.

It follows an iterative bottom-up strategy driven by a loop of BIC estimations and HMM alignments (Fig. 1 block C). In this step the segments which belong to the same speaker are combined in a new model at each iteration. A time constraint as in [2] is also imposed on the duration of the speaker segments through a hierarchical modelling of each state. In that sense, Viterbi decoding decisions are taken based on the estimation of the observation probabilities by accumulating the likelihoods per cluster/state in a 3 seconds window.

We used a modified BIC-based metric [2] to decide most likely-pair of clusters to merge. The segmentation obtained at the output of the block B (see Fig. 1) defines a new set of speaker

clusters/states which will be retrained. Most of the systems based on agglomerative clustering perform just one merge at each BIC iteration, in which they choose to merge those couple of clusters with higher BIC value. Nonetheless, for this work a threshold was applied depending on the standard deviation of the set of BIC value obtained among cluster-pairs. We decide to merge all of those cluster-pairs (i, j) fulfilling:

$$BIC_{ij} > BIC_{\mu} + \frac{3}{2} BIC_{\sigma} \quad (2)$$

where BIC_{ij} is the BIC value between the clusters i and j, BIC_{μ} is the mean of BIC_{ij} for $i \neq j$ and BIC_{σ} the standard deviation for the same set. Therefore, the system might merge more than one pair of clusters per iteration yielding to a speed up in the agglomerative clustering. At each iteration j, the number M_i^j of Gaussian mixtures to model the cluster i is updated by

$$M_i^j = \left\lfloor \left(\frac{N_i^j}{R_{\mathrm{CC}}} \right) + \frac{1}{2} \right\rfloor, \quad (3)$$

where N_i^j is the number of frames belonging to the cluster i. Whenever two segments are merged, a new segment model is also trained pooling all the features from the merged segments and fixing the model complexity according to the R_{CC} value. Such automatic selection of the modelling complexity has demonstrated a successful performance while avoiding the use of the penalty term in the classical BIC metric [14]. This procedure is iterated until the stopping criterion is reached. It is met whenever all the remaining set of BIC cluster-pairs show negative values, meaning that no suitable candidates are found to merge and consequently the algorithm ends.

Finally, at the last iteration and once the stopping criterion is met, each remaining state represents a different speaker. A more detailed description of the system can be found in [13].

3. Diarization based on spectral clustering in Gaussian space

Despite of the good results achieved by popular AHC systems, an important drawback arises in the case of long duration audio documents. AHC approach is a highly time consuming approach. The processing time for audio recordings depends directly on the number of initial segments taken into account. For instance, augmenting the initial number of segments in long audio documents considerably increases the size of the BIC comparison matrix and, therefore, the total time processing of the iterative approach. Reducing the number of initial segments drastically makes smaller such time but at the expense of the speaker detection accuracy due to the initial cluster impurity. So there exists a tradeoff between computational cost and detection performance in AHC based systems. To overcome such drawback we propose a clustering approach based on spectral clustering that, despite of its computing time is still dependent on the number of initial segments, it avoids statistical metrics to build the similarity matrix yielding to a faster algorithm than AHC+BIC one.

In Figure 2 we draw the scheme of the proposed system based on spectral clustering. As in the AHC approach, we keep as prior modules, the oracle Speech/Non-Speech detection module and a Wiener filtering implementation from the QIO front-end. Cluster initialization is still based on an homogeneous splitting of data but, in contrast to AHC approach, no automatic selection of number of clusters is performed. Number of initial cluster is tuned with development data.

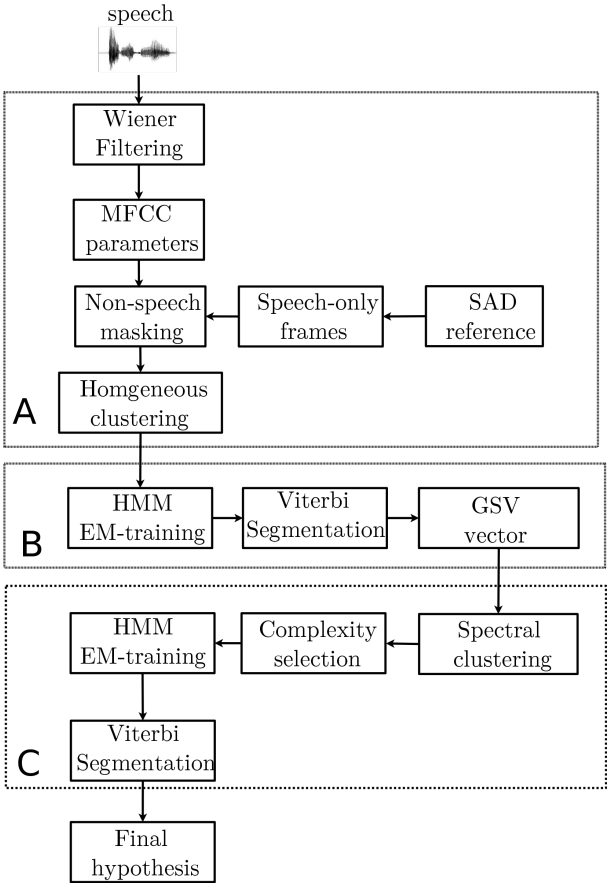

Figure 2: *Speaker diarization scheme based on spectral clustering with Viterbi HMM/GMM initialization and clustering refinement.*

3.1. Segments representation

The core of the proposed system is shown in blocks B and C of Figure 2. Before spectral clustering was carried out, initial segments are modeled by a mixture of Gaussians with fixed complexity, that is, number of Gaussians is independent of the duration of the segment. Following, a Viterbi decoding is performed by means an ergodic HMM. Once initial segmentation is stabilized, the segments presents a great variety of durations. To overcome this drawback, a Gaussian super vector (GSV) modelling is proposed [11]. Furthermore, segments lesser than 3 seconds are discarded in order to ensure statistical significance in Gaussian parameter estimation. Such segment discarding is motivated by characteristics of our data. The estimated probability density for a speech segment is assumed to represent speaker characteristics. However for conversational speech recordings, plenty of short utterances and changes in speaker turns, the density estimation by means GMM will be strongly biased by their phonemic variations. In any case, initial discarded segments will be assigned to discovered clusters by the SC through Viterbi alignment in last step of the approach, see block C in Figure 2.

Only the means of the Gaussians μ_{ik} are stacked in a vector to build the GSV. The μ_{ik} means are normalized through the corresponding variance σ_{ik} and weight of the Gaussian as follows:

$$GSV_{ik} = \sqrt{w_{ik}}\ \Sigma_{ik}^{(-1/2)}\mu_{ik}, \quad k = 1, \ldots, D\ ,\ i = 1, \ldots, M \tag{4}$$

where w stands for the weight of the Gaussian, Σ is the corresponding variance and μ represents the mean of the Gaussian. Indexes i and k stand for the number of Gaussian in the mixture model and the Gaussian dimension respectively. Therefore, stacking normalized Gaussians' means in a vector leads to a length of the GSV vector equals to the number of Gaussian M employed to model i-th segment (which is always the same for all segments) times the number of dimensions D.

Other segment representation has been proposed for spectral clustering in diarization task. In [7] GMM parameters adapted from a UBM, trained on the whole audio data, are employed as representation for speech segments whereas KL distance is used for building the affinity matrix. In [9], author employed a non-parametric representation of speech segments based on Vector Quantization (VQ) in which the VQ codebook is created from the audio recording and utterances are represented as a vector of frequencies in VQ space. The affinity matrix is constructed by means cosine similarity distance. In our approach we have decided to apply Gaussian super vector model due his excellent results in speaker verification tasks and its robustness against trials involving segments of different duration [11]. In addition, no statistical measure as KL is proposed to construct the affinity matrix but Euclidean distance is computed in GSV space, consequently saving in computational time.

3.2. Spectral Clustering

Once a initial segmentation and a segment representation is computed, a speaker clustering is performed to join those segments which belong to same speaker. We use a modification of the Ng-Jordan-Weiss (NJW) algorithm [15] and a modified implementation in C++ programming language taken from [16], which we first briefly review. Given a set of speech segments $S = \{s_1, \ldots, s_n\}$ represented by n points $X = \{x_1, \ldots, x_D\}$, in this work the GSV vector, that we want to cluster into k subsets:

- Form a similarity graph defined by the affinity matrix $A \in \mathbb{R}^{n \times n}$ where $A_{ij} = \exp(\frac{d^2(s_i, s_j)}{\sigma^2})$ if $i = j$, and $A_{ii} = 0$, where $d(s_i, s_j)$ is distance function and σ^2 is a scaling parameter.
- Define D to be the diagonal matrix whose (i, i)-element is the sum of A's i-th row, and construct the normalized symmetric graph Laplacian matrix $L = D^{1/2}AD^{1/2}$.
- Select the number of clusters k.
- Find $\{u_1, u_2, \ldots, u_k\}$, the k largest eigenvectors of L, and form the matrix $U = \{u_1, u_2, \ldots, u_k\} \in \mathbb{R}^{n \times k}$.
- Re-normalize the rows of U to have unit length yielding $Y \in \mathbb{R}^{n \times k}$, such that $Y_{ij} = U_{ij}/(\sum_j U_{ij}^2)^{1/2}$.
- Cluster the points Y_{ij} with k-means algorithm into clusters $C_1, \ldots, C_k$.

The main idea behind spectral clustering algorithm relies on changing the representation of data points x_i in $y_i \in \mathbb{R}^k$, that is, mapping x_i into a space where the simple k-means clustering algorithm has no difficulty to detect clusters. Nevertheless, such a situation only occurs in an ideal case whether data is enough clean and consequently no overlap among different classes takes place.

In order to form the affinity matrix, it is required to define a similarity function d on the data and a scaling parameter σ. In this work, the Euclidean distance among GSV vectors has been employed, fulfilling distance requirements such as: be non-negative, be low for similar segments and high otherwise. Euclidean distance has clearly an intuitive sense in GSV space, giving an idea of how far are Gaussian mixtures among different segments. In addition, all distances amongst segment-pairs has been considered leading to a fully connected graph. The scaling parameter σ is some kind of measure of when two points should be considered similar and controls how rapidly the affinity matrix A_{ij} falls off with the distance between s_i and s_j segments. As the work presented in [7], we calculate a scaling parameter depending on the pair of segments (s_i, s_j) involved in distance computation, by considering the second order statistics of distances to all other data segments as follows,

$$\sigma_{ij} = \sqrt{Var(d(s_i, s_n))Var(d(s_j, s_m))}, \quad \text{with } n \neq i,\ m \neq j \tag{5}$$

where $Var(\cdot)$ computes the variance and $d(s_i, s_n)$ are distances from segment s_i to all other segments. In contrast to [7] we do not include the scalar parameter β in computation of σ_{ij}.

As part of the diarization task, the number of clusters has to be estimated automatically. In model-based clustering approaches, such decision is usually based on the likelihood performed from data as in the previous AHC system. In this work, number of clusters is estimated by analyzing the magnitude of the eigenvalues of the normalized Laplacian matrix L as in [7, 9]. It is known as eigengap heuristic, where the objective is to select k clusters as the number of k maximum eigenvalues of the Laplacian L matrix,

$$\gamma_k = |\lambda_k - \lambda_{k+1}| > \Theta, \tag{6}$$

where γ_k is the eigengap between two consecutive eigenvalues $\{\lambda_k, \lambda_{k+1}\}$ and Θ is a threshold we tune with development data. There exists different explanations to the use of such criterion, as those from perturbation theory or geometric graph invariants, due to the fact that similarity information can be compacted with just first eigenvalues/eigenvectors of the Laplacian matrix L [5, 10].

Finally, in the last step of the SC approach and once we have selected the number of k clusters, a k-means algorithm is employed to link up segments in clusters into the new space representation, $y_i \in \mathbb{R}^k$

3.3. Clustering refinement

As we can see in block C of Figure 2, the resulting clustering obtained by SC feeds a last HMM alignment step. In contrast to the initialization step, a complexity selection as in AHC system is employed, and the newly clusters are modeled by an HMM/GMM. Several Viterbi alignments are performed until no variation in the segmentation is perceived and a final clustering hypothesis is obtained.

RT data	#Shows	SNR (dB)	#Speakers	Eval time	Speech time	Speaker time	Overlap time
Development RT06s	9	19.72	5.11 (9/4)	1109.33	989.07	1305.50	232.30
Development RT07s	8	22.94	4.37 (6/4)	1316.84	1048.322	1191.186	154.84
Development RT06-RT07s	17	21.24	4.76 (9/4)	1206.98	1016.95	1251.70	195.85
Evaluation RT09s	7	24.11	5.43 (11/4)	1549.47	1262.30	1484.69	194.00

Table 1: *NIST Rich Transcription official conference evaluation data from RT06s, RT07s and RT09s. From left to right, columns denote: Number of shows in the data set, Signal to Noise Ratio (SNR) in dBs, mean number of speakers involved (with maximum and minimum inside the parentheses), mean total time evaluated with and without non-speech segments, mean speaker time counting the overlap speech as often as the numbers of overlapped speakers and the mean speech time corresponding to any kind of speaker overlap. All time columns are expressed in seconds.*

4. Experiments and results

In order to assess the proposed spectral clustering approach, it is compared to classical AHC system with BIC metric and complexity selection. The performance of the speaker diarization was evaluated by means of the diarization error rate (DER) as defined by NIST [12]. The DER is a time-weighted metric composed of the sum of missed speaker time, false alarms and speaker error time. Due the use of oracle speech activity detection references, both missed speaker time and false alarms time is not taking into account. Such errors are 0% in a single speaker scoring metric, which does not consider more than one reference at the same time. Anyway, speaker overlap should be considered since shows exhibiting a high percentage of speaker overlap traduces in a hard challenge for diarization approaches which not handle this issue directly, e.g, given more than one label at the speaker overlap regions. Since neither of the approaches presented in this paper gives more than one speaker label at the same time, we will restrain our experiments to the speaker error produced by just one speaker. That is, considering such overlap regions as uttered by a single speaker we removed the speaker error produced by the overlap references.

In conclusion, we are not considering DER degradation due miss speaker time produced by overlap and just one-speaker time is taking into account to compute the DER. In addition, as usually in NIST evaluations, a collar of 0.25 seconds is applied in the scoring tool [1], that is, there is a non-score zone around reference segment boundaries where the clustering output is not evaluated which is within ± 0.25 seconds.

4.1. Rich Transcription Data

NIST Rich Transcription (RT) data consists of excerpts from multi-party meetings in English collected at eight different sites at various time periods. From each meeting only an portion of 20 minutes is evaluated. The number of microphones available for each recording ranges from 1 to 16 but we will only focus on the single reference channel given by NIST, which is known as SDM condition. Evaluation conference data from RT 2006 and 2007 has been used to perform algorithm development whereas conference data from RT09s has been used to assess the performance of the algorithms.

Table 1 gives a brief summary of RT data characteristics regarding number of speakers, SNR [2] , evaluation time and speaker overlapped time.

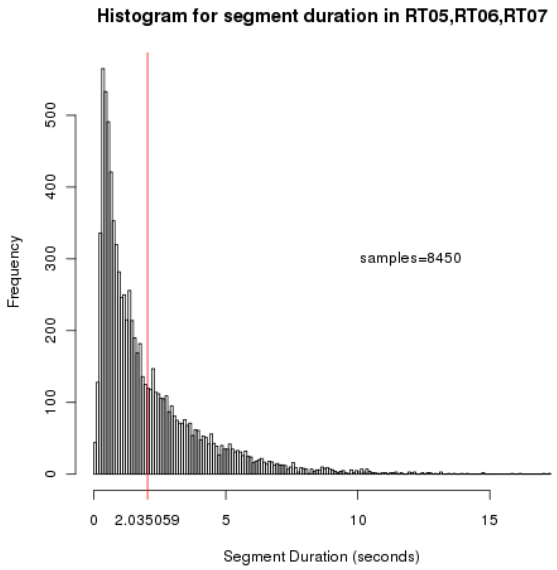

Figure 3: *Histogram for segment durations in RT'05, RT'06 and RT'07 data. The tick marked as the red vertical line stands for the mean duration of a speaker segment.*

4.2. Tuning system parameters

Some parameters are tuned through a set of experiments on the development data, such as: The minimum duration turn per speaker, the initial number of segments, its GMM model complexity and finally the threshold Θ as maximum eigengap. In Figure 3 we depict the histogram for the segment duration in NIST RT data for the evaluations in 2005, 2006 and 2007. It takes into account any speaker segment in the evaluation time, that is, all consecutive speech from the same speaker without silences greater than 0.5 seconds. Speaker overlapped segments are also considered to draw the picture yielding to a total of 8450 samples. As we can see at the red line in the histogram, the mean duration of the segments is around 2 seconds. The minimum duration constrain for HMM/Viterbi alignment is set to such value in both SC and HMM+BIC implementations.

[1] Tool evaluation command looks like "./md-eval-v21.pl -1 -nafc -c 0.25 -s output -r reference". Evaluation tool from NIST can be directly downloaded from ftp://jaguar.ncsl.nist.gov/pub/sctk-2.4.0-20091110-0958.tar.bz2

[2] The NIST Speech Quality Assurance (SPQA) package has been used for calculating the SNR for speech. It can be downloaded from `http://www.itl.nist.gov/iad/mig//tools/spqa_23sphere25tarZ.htm`

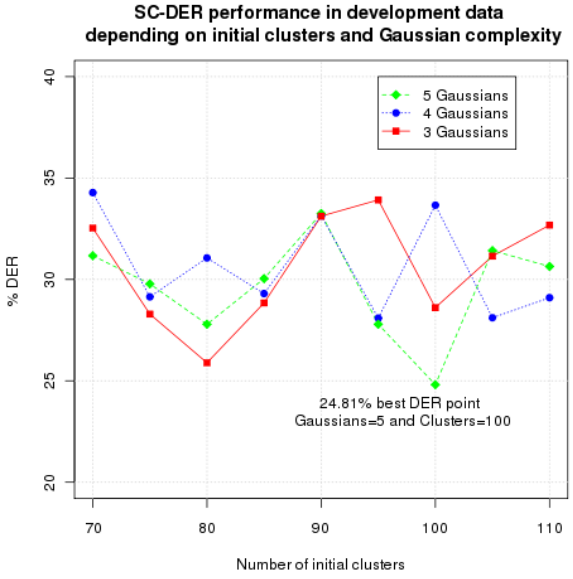

Figure 4: *Spectral clustering performance in terms of % DER depending on the initial number of clusters and the Gaussian complexity for building the GSV vectors.*

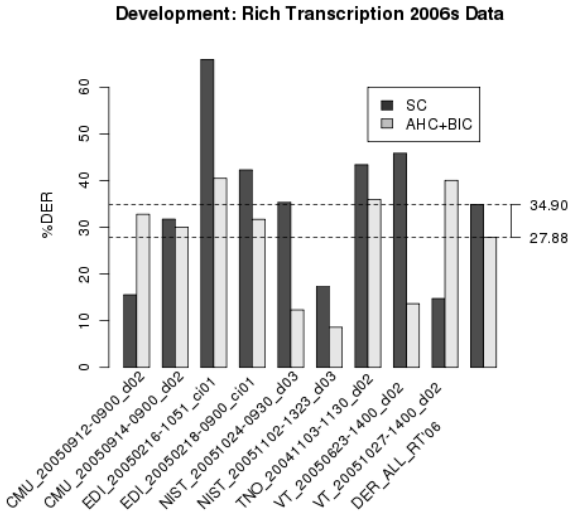

Figure 5: *Development results on Rich Transcription 2006s data.*

The initial number of segments and the number of initial Gaussians per segment has been also tuned using the development data sets. Figure 4 presents impact on diarization error rate for the spectral clustering algorithm for different GMM model complexities: 3,4 and 5 Gaussians respectively; and for a number of initial segments ranging from 70 to 110 segments. The DER curves are obtained on the development data RT'06 and RT'07. The lowest DER is reached by using 100 initial clusters and employing GMM models composed by 5 Gaussians. These values are selected in the SC approach applied to the RT'09 evaluation data.

Finally the threshold Θ, which is used to select the number of clusters, is also tuned based on %DER performance in development data. Thus the first maximum γ_k eigengap is fixed to 0.001.

4.3. Results

Figures 5 and 6 display the results per each show obtained on RT'06 and RT'07 conference data respectively. In both data sets, the DER errors produced by the SC-based implementation are only slightly worse than those obtained by the AHC+BIC approach. In general, AHC system obtains a better performance for both development and evaluation data sets. Nevertheless and depending on the development subset, SC outperforms the results obtained by classical AHC+BIC.

As we can see in the RT'07 data, SC obtains a 14.54% DER outperforming the AHC+BIC system with a 17.73% DER. Nonetheless, the same does not happen in RT'06 data in which SC performance fall off compared to the AHC approach. In overall, in the Figure 7 we report the DER per show and the total error computed on the development data in which AHC+BIC obtains slightly lower DER results.

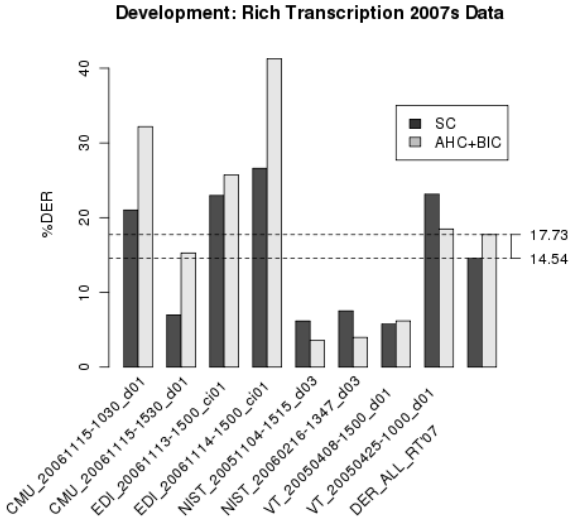

Figure 6: *Development results on Rich Transcription 2007s data.*

Finally, the Figure 8 shows the generalization of the results to unseen data in the evaluation data set. As in the case of development experiments, the AHC+BIC approach outperforms slightly the SC-based one, 25.19% compared to 27.52% DER.

Table 2 summarizes the results performed by both approaches on the different data sets. DER error rates and the associated standard deviation (σ) per set are also reported. It is worth to mention the lowest deviation (σ) observed in the SC results compared to the AHC approach. The SC implementation seems to perform more robustly across different shows than AHC does, specifically in RT'09 and RT'07 data sets. In

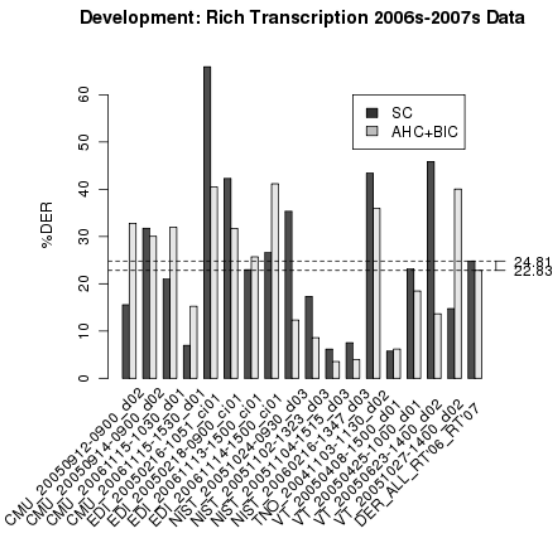

Figure 7: *Development results on Rich Transcription 2006s and 2007s data.*

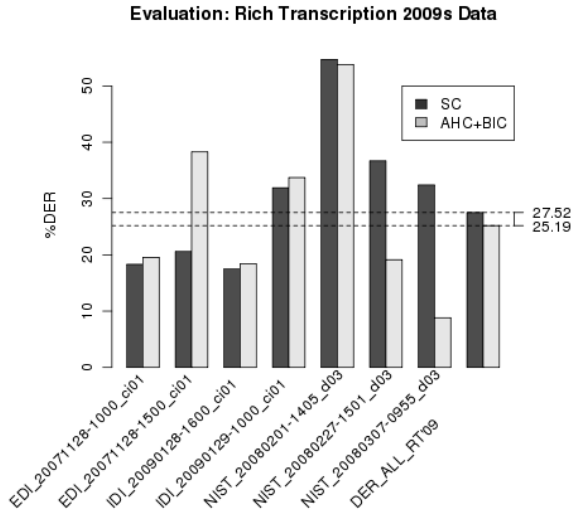

Figure 8: *Evaluations results on Rich Transcription 2009s data.*

addition and aiming to verify that the SC clustering provides a significant reduction in terms of complexity, we report in Table 2 computational relative time on the different RT evaluation data sets for AHC+BIC and SC approaches. Feature extraction processing is common for both methods and it was not taken into account for measuring time consumption. Processes were run on a Intel(R) Xeon(R) CPU E5540 2.53GHz machine. Experiments conclude that SC based clustering runs around 3 times faster than the AHC+BIC system.

	AHC+BIC %DER / σ	SC %DER / σ	xfaster
RT06	27.88% / 12.38	34.90% / 16.98	3.02x
RT07	17.73% / 13.90	14.54% / 9.13	2.43x
RT06+RT07	22.83% / 13.51	24.81% / 16.82	2.67x
RT'09	25.19% / 15.33	27.52% / 13.19	3.24x

Table 2: *DER results and standard deviation (σ) per set on Rich Transcription 2006, 2007 and 2009 conference data and number of times that SC implementation is faster than classical AHC+BIC.*

5. Discussion and Conclusion

We compare SC based speaker diarization system with a baseline system based on agglomerative clustering with HMM/GMM modelling and automatic Gaussian complexity selection. The main advantage of spectral clustering is that it does not build any statistical metric for deciding if two clusters should be merged. This avoids explicit BIC or KL computation at each merging step, by employing a Euclidean distance among super vector representation of clusters, thus signicantly reduces the complexity of the clustering algorithm. Experiments are performed on RT'06, RT'07 and RT'09 conference evaluation data and results are provided in terms of diarization error rate and using an oracle speech detector.

A set of experiments are performed on a development set in order to chose optimal parameters for SC-based system. In the second set of experiments, the results are generalized to a "blind" data set. In this case the SC system has a drop in performances by 2% w.r.t baseline AHC+BIC approach. To summarize, the spectral clustering based algorithm along with Viterbi realignment is found to achieve DER results slightly worse than the conventional AHC+BIC system but with reduced computation. Results presented also display a great variance among different shows as well as between evaluation data sets. It may be due to data characteristics, e.g., number of speakers involved, room setups, SNR levels and speech overlap segments. For instance, the worst performance is reported in RT'06 data which exhibits both lower SNR and higher overlap time than the other databases used, see Table 1.

Finally, this work on spectral clustering theory is based on a series of assumptions that will be further investigated in future works. For instance, the similarity matrix is built based on Euclidean distances among Gaussian super vector representation of clusters obtained without MAP adaptation of an universal background model. In addition, we employ a full connected graph weighted by a scalar parameter for building the affinity matrix. These steps can be improved by means a more robust cluster representation and a best adapted metric distance among them, or by a most suitable scalar parameter to improve the merging step. Furthermore, we used an initial uniform segmentation in blocks of fixed duration. In this case, the system may be improved through the use of a speaker change detection algorithms to obtain "pure" initial segments containing a single speaker. Further research comparing implementation with multiple channels and different sets of features will also be addressed in future works.

Acknowledgements. We acknowledge financial support by the MEC and Comunidad de Madrid (Spain) through Project Nos. FIS2009-13690 and S2009ESP-1691 and by Spanish project SARAI (TEC2010-21040-C02-01).

References

[1] D.A. Reynolds and P. Torres-Carrasquillo, "Approaches and Applications of Audio Diarization," in *IEEE International Conference on Acoustics Speech and Signal Processing, ICASSP*, 2005, vol. 5.

[2] J. Ajmera and C. Wooters, "A robust speaker clustering algorithm," in *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU*, 2003.

[3] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research," *IEEE Transactions on Audio, Speech and Language Processing*, 2011.

[4] K.J. Han, S. Kim, and S.S. Narayanan, "Strategies to improve the robustness of agglomerative hierarchical clustering under data source variation for speaker diarization," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 8, pp. 1590 –1601, nov. 2008.

[5] Joseph Keshet and Samy Bengio, *Automatic Speech and Speaker Recognition: Large Margin and Kernel Methods*, John Wiley & Sons, 2008.

[6] Daniel P. W. Ellis and Jerry C. Liu, "Speaker turn segmentation based on between-channel differences," in *at NIST Meeting Recognition Workshop at ICASSP 2004*, 2004.

[7] Huazhong Ning, Ming Liu, Hao Tang, and Thomas S. Huang, "A spectral clustering approach to speaker diarization.," in *INTERSPEECH'06*, 2006, pp. –1–1.

[8] Huazhong Ning, Wei Xu, Yun Chi, Yihong Gong, and Thomas S. Huang, "Incremental spectral clustering by efficiently updating the eigen-system," *Pattern Recognition*, vol. 43, no. 1, pp. 113 – 127, 2010.

[9] K. Iso, "Speaker clustering using vector quantization and spectral clustering," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, march 2010, pp. 4986 –4989.

[10] Ulrike Von Luxburg, Mikhail Belkin, Olivier Bousquet, and Pertinence, "A tutorial on spectral clustering," *Stat. Comput*, 2007.

[11] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using gmm supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13(5), pp. 308–311, 2006.

[12] J. Fiscus and et al., "The rich transcription evaluation project," `http://www.nist.gov/speech/tests/rt/`, 2002-2007.

[13] J. Luque and J. Hernando, "Robust speaker identification for meetings: Upc clear-07 meeting room evaluation system," in *Lecture Notes on Computer Science, LNCS*, vol. 4625. Springer-Verlag, 2008.

[14] X. Anguera, C. Wooters, and J. Hernando, "Robust speaker diarization for meetings: ICSI RT06s evaluation system," in *International Conference on Spoken Language Processing, ICSLP*, 2006.

[15] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in neural information processing systems*. 2001, pp. 849–856, MIT Press.

[16] Wen-Yen Chen, Yangqiu Song, Hongjie Bai, Chih-Jen Lin, and Edward Y. Chang, "Parallel spectral clustering in distributed systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 568–586, 2011.