DiarTk : An Open Source Toolkit for Research in Multistream Speaker Diarization and its Application to Meetings Recordings.

Deepu Vijayasenan[1], *Fabio Valente*[2]

[1] Universität des Saarlandes, Saarbrücken, Germany
[2] Idiap Research Institute, 1920, Martigny, Switzerland
deepu.vijayasenan@lsv.uni-saarland.de,fabio.valente@idiap.ch

Abstract

The speaker diarization task consists of inferring "who spoke when" in an audio stream without any prior knowledge and has been object of several NIST international evaluation campaigns is last years. A common trend for improving performances has been the use of several different feature streams as diverse as speaker location features, visual features or noise robust acoustic features. This paper describes an open source toolkit released under GPL license aiming at facilitating research in multistream speaker diarization and reproducing state-of-the-art results. In contrary to other related diarization toolkits, it is explicitly designed to handle an arbitrary number of features with very different statistics while limiting the computational complexity. The release includes a set of scripts to replicate benchmark results on previous NIST evaluations and is intended to provide an easy to use software to study and include novel features into diarization systems.

Index Terms: Open Source toolkit, Speaker Diarization, multistream features, NIST Rich Transcription.

1. Introduction and Motivations

Diarization is the task that consists of annotating temporal regions of audio recordings with labels. The labels represent names of speakers, or their gender, the channel type (narrow bandwidth vs. wide bandwidth), the background environment (quite, noise, music...), or other characteristics present in the signal. In details, speaker diarization is that task that aims at inferring "who spoke when" in an audio file and involves two simultaneous tasks 1) inferring the number of speakers in the audio file 2) associating a label with each temporal segment so that segments uttered by the same speaker belong to the same cluster. This process happens without any type of prior information on the number of speakers in the audio stream nor any information on their identities. Speaker diarization has been originally proposed for segmenting the audio in speaker homogeneous regions in order to perform adaptation during the speech recognition process. More recently, it has been applied into a number of other problems like speaker-based indexing, retrieval and audio structuring.

The most common approaches to diarization of broadcast audio consists in Hidden Markov Models/Gaussian Mixture Models (HMM/GMM), agglomerative clustering and conventional acoustic features like MFCC [1]. Several toolkits are already available for speaker diarization of broadcast data like the Audioseg toolkit [1], the LIUM_SpkDiarization toolkit [2] and the SHoUT toolkit [3].

Beside broadcast news segmentation, in recent years the diarization task has been broadly applied to spontaneous multiparty conversations, also known as "meetings", recorded in specially instrumented rooms (see Figure 1). Recordings are done with far-fields microphone arrays and also information from close caption and overview cameras is available. Since 2005, advances in diarization of spontaneous conversations (or meetings) have been benchmarked into several NIST Rich Transcription (RT) evaluation campaigns [4] and systems have been ranked according to a common metric, the Diarization Error Rate (DER) [5].

Compared to broadcast data, meetings have several additional challenges coming from the far field audio corrupted with noise and reverberation as well as the conversational nature of the speech (very short speaker turns and continuous overlap in between speakers) [2]. An active research field has been the use of multiple sources of information or equivalently, multiple feature streams beside more conventional acoustic features like MFCC. Examples in the literature include features extracted from the microphone array like speaker location [3] or speaker intensity [4], visual features extracted from cameras [5, 6, 7, 8] or noise/reverberation robust acoustic features like the signal modulation spectrum [9]. As the different feature streams often have very different statistics and dimensionality, their use into diarization systems often requires several modifications and fine tuning of parameters in order to became effective. To date, none of the previously mentioned toolkits is explicitly designed to handle more than a single feature stream.

We present here *DiarTK*, a completely open source toolkit aiming at facilitating research in multistream speaker diarization. The toolkit is written in C++ and released under GPL licence. *DiarTK* is designed according to the following wish-list:

- Simplicity of the base code and self-contained modules.
- Able to handle an arbitrary number of feature streams with very different statistics.
- Limit the computational complexity of the diarization system thus being able to perform real-time diarization even with several feature streams.
- Reproduce state-of-the-art results on standard NIST benchmark databases.

The remainder of the paper is organized as follows, section 2 presents briefly the diarization method implemented in

[1] http:audioseg.gforge.inria.fr
[2] http:lium3.univ-lemans.frdiarization
[3] http:shout-toolkit.sourceforge.net
[4] http://www.itl.nist.gov/iad/mig/tests/rt/
[5] http://www.itl.nist.gov/iad/mig/tests/rt/2005-spring/rt05s-meeting-eval-plan-V1.pdf

Figure 1: Example of meeting recording environment equipped with microphone array, close caption cameras and overview cameras. A different set of features (acoustic, location or visual features) can be extracted from each of the sensor and used to enhance the diarization process.

the toolkit, section 3 describes the various modules and the processing chain, while section 4 presents recipes, benchmark on the rich transcription datasets and analysis of computational performance.

2. Diarization method

The diarization chain underlying *DiarTk* is similar to other diarization systems [1, 2] and consists of:

1 An initial segmentation step in which the audio stream is segmented into homogeneous regions.
2 An agglomerative clustering step in which segments belonging to the same speaker are clustered together.
3 A Viterbi realignment step in which speaker boundaries are refined thus producing the final diarization output.

In conventional system, those steps are performed using parametric models, i.e., a Gaussian Mixture Model (GMM). Whenever several feature streams are used a separate GMM for each stream is estimated for each of them [2]. On the other hand, *DiarTk* makes use of non-parametric clustering and realignment based on the agglomerative Information Bottleneck principle [10] thus avoiding explicit GMM speaker modeling.

In order to achieve this, let us consider a set of speech segments $X = \{x_1, \ldots, x_T\}$ obtained from segmentation of an input audio stream, to be clustered into set of clusters $C = \{c_1, \ldots, c_K\}$. A space of relevance variables Y that contain relevant information about the problem is constructed and each segment X is mapped into Y thus obtaining $p(Y|X)$. The variable Y are defined as the components of a *single background GMM* estimated on the entire recording. According to IB principle the optimal clustering compresses the input variables while preserving as much mutual information as possible about the relevance variables Y [11]. This corresponds to the minimization of:

$$\mathcal{F} = I(X, C) - \beta I(C, Y) \tag{1}$$

Where β is a Lagrange multiplier. The clustering operates using probabilities $p(y|x)$ that are obtained using Bayes' rule. The optimization of the objective function (1) can be done in a greedy fashion using the agglomerative Information Bottleneck method [12].

The algorithm is initialized with the trivial clustering of each point considered as a separate cluster ($|X|$ clusters). At each step of the algorithm a cluster merge is performed such that the information loss with respect to the relevance variables is minimum. The loss of mutual information can be obtained in close form. The optimal number of clusters are selected based on a threshold on the Normalized Mutual Information (NMI) (for details see [10]). The complete algorithm in case of a single feature stream, i.e. MFCC, is summarized as follows :

1 Feature extraction (MFCC) from the audio.
2 Speech/non-speech segmentation and rejection of non-speech frames.
3 Segmentation of speech in chucks (uniform or not), i.e., extraction of variable X.
4 Estimation of a background GMM model with a shared diagonal matrix, i.e., definition of the set Y.
5 Estimation of conditional distribution $p(Y|X)$ for each segment X.
6 aIB clustering and model selection to determine the speaker clusters (Diarization output)
7 Realignment of the speaker boundaries using a Viterbi realignment step.

The main advantage of this approach is that the entire diarization system (IB clustering, feature combination and realignment) works in the space of relevance variables thus, as the method does not estimate a GMM for each speaker model, the computational complexity appears limited respect to HMM/GMM systems [10].

Furthermore whenever several feature streams are available the only extra computational load comes in the estimation of the relevance variable space leaving the cost of the clustering and the realignments unchanged. In fact let us designate with W_i a set of different feature streams. The combination can be performed in the relevance variable space, i.e, using the posterior probabilities $p(y|x)$. For each feature stream W_i, a background GMM M_i is estimated, and a posterior distribution $p(y|M_i, x)$ calculated. The combined distribution is then calculated as:

$$p(y|x) = \sum_i p(y|M_i, x) W_i \tag{2}$$

Where W_i is the feature weight corresponding to i^{th} feature stream ($\sum_i W_i = 1$). Another advantage comes from the fact $p(y|M_i, x)$ are posterior distributions, thus in the range $[0, 1]$, making the combination more robust to different dimensionality or different statistics of the features [13, 14]. This allow the use of very different statistics like those coming from acoustic and location information. Also the realignment can happen using the relevance posterior variables computing the optimal speaker segmentation as:

$$\mathbf{c}_{opt} = \arg\min_{\mathbf{c}} \sum_t KL(p(Y|x_t)||p(Y|c_t)) - \log(a_{c_t c_{t+1}}) \tag{3}$$

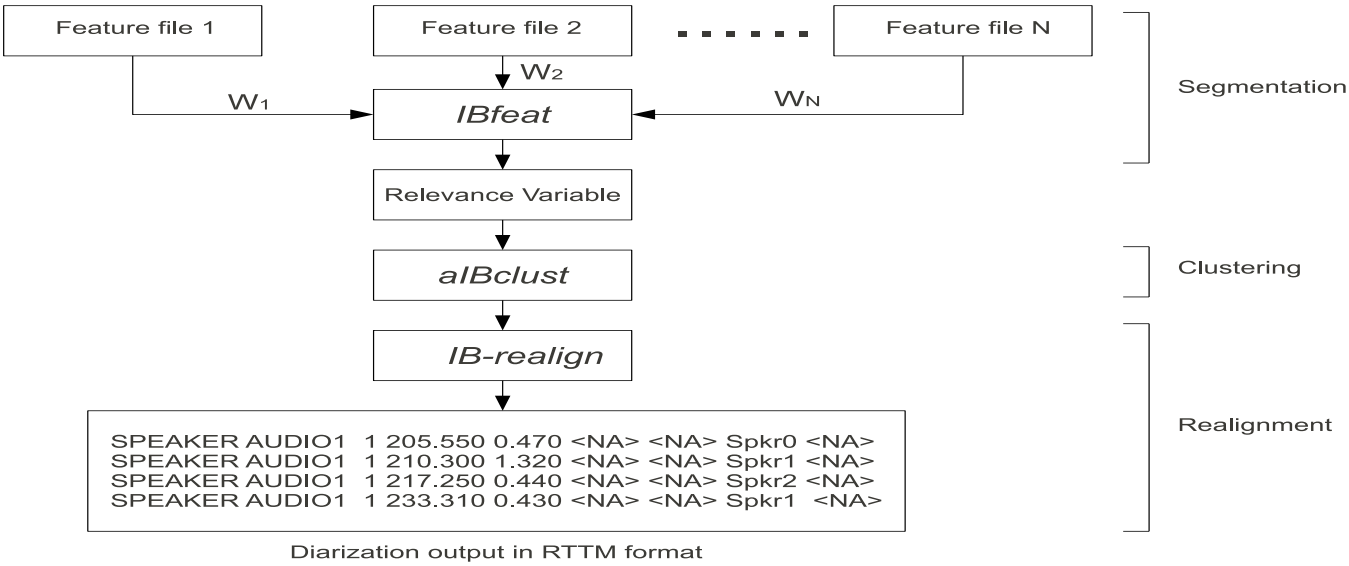

Figure 2: Structure of the diarization chain using the open source modules IBfeat, aIBclust and IBrealign which performs respectively segmentation, clustering and realignment steps.

Where $p(Y|x_t)$ is the posterior distribution of the relevance variables at a given speech segment x_t and $p(Y|c_t)$ is the posterior distribution of relevance variables in a given cluster c_t (see [13]).

3. The Diarization Modules

DiarTk consists of three main modules an audio segmentation tool (IBfeat), an agglomerative clustering tool (aIBclust) and a Viterbi realignment tool (IBrealign). The processing chain is depicted in Figure 2 and briefly described in the following:

1 The initial segmentation and relevance variable estimation are performed by the *aIBfeat* tool which takes as input a list of feature files in standard HTK format associated with a weight. The tool can handle an arbitrary number of different feature files irrespectively of their dimensions and statistics simply by running:

```
aibfeat
--mfcc mfcc.fea 0.5
--tdoa tdoa.fea 0.2
--other fea1.fea 0.2
--other fea2.fea 0.1
```

aIBfeat performs the initial segmentation in homogeneous speech regions, estimates the background GMMs (one per each feature stream) and computes the final relevance variable distributions $p(Y|X)$ as weighted sum of individual distributions $\sum_i W_i\, p(Y|X_i)$ as described in section 2.

2 The agglomerative clustering is performed by the *aIBclust* tool. The speech segments X associated with the relevance variable distributions $p(Y|X)$ are then clustered together into C clusters, according to an agglomerative clustering procedure until a stopping criterion is met. It is important to notice that the dimensions of $p(Y|X)$ does not depends on the number of feature streams. As consequence, the complexity of the clustering stays unchanged regardless of the number of streams.

3- The Viterbi Realignment is performed by the *IBrealign* tool which takes as input the partitions obtained from the agglomerative clustering and performs a realignment of the speaker boundaries. The realignment as well depends only on the distribution of $p(Y|X)$ regardless of the number of feature streams. The output is provided in RTTM format [6] ready to be scored by the NIST evaluation modules [7] in order to obtain Diarization Error Rate scores.

4. Benchmarks and Recipes

The toolkit is provided with a set of scripts in order to reproduce benchmarks on standard diarization databases like the NIST Rich Transcription data from the 2006, 2007 and 2009 evaluation campaigns (note there was no evaluation in 2008). The datasets comprises 34 spontaneous conversations, i.e., meetings recorded in 7 different meeting rooms. Recordings are done with far-field microphone arrays. Results are reported according to the Diarization Error Rate which is composed of two parts: a speech/non-speech error and a speaker error.

To test the capabilities to integrate several sources of information, the toolkit is benchmarked with conventional acoustic features, i.e., MFCC coefficients (see Table 1), as well as other features like the Time Delay of Arrivals (TDOA) extracted from microphone arrays [3] (see Table 2) and the Frequency Domain Linear Prediction features (FDLP)/Modulation Spectrum (MS) of the signal (see Table 3). The benchmark reveals that multi-stream modeling consistently reduces the speaker error and results are competitive with those obtained during the latest evaluation campaigns [2] thus producing state-of-the-art diarization results.

In case of HMM/GMM modeling, the use of several feature streams significantly increase the computational complexity. In

[6] http://www.itl.nist.gov/iad/mig/tests/rt/2003-fall/docs/RTTM-format-v13.pdf

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

MFCC features			
Dataset	Sp./non-Sp. Error	Speak. Error	Diar. Error
RT06	6.60	15.60	22.25
RT07	3.70	11.30	15.03
RT09	12.70	21.30	33.98

Table 1: Benchmark results (NIST Speech/non-Speech Error, Speaker Error and Diarization Error) on Rich Transcription data for Diarization using MFCC features.

MFCC and TDOA features			
Dataset	Sp./non-Sp. Error	Speak. Error	Diar. Error
RT06	6.60	10.00	16.57
RT07	3.70	13.10	16.84
RT09	12.70	17.50	30.22

Table 2: Benchmark results (NIST Speech/non-Speech Error, Speaker Error and Diarization Error) on Rich Transcription data for Diarization using MFCC and TDOA features.

MFCC, TDOA, Modulation Spectrum and FDLP			
Dataset	Sp./non-Sp. Error	Speak. Error	Diar. Error
RT06	6.60	6.60	13.20
RT07	3.70	5.80	9.54
RT09	12.70	9.50	22.22

Table 3: Benchmark results (NIST Speech/non-Speech Error, Speaker Error and Diarization Error) on Rich Transcription data for Diarization using MFCC, TDOA, FDLP and Modulation Spectrum features.

DiarTk, the only increase in complexity happens whenever the distributions $P(Y|X)$ are estimated using the *IBfeat*tool leaving the complexity of *aIBclust* and *IBrealign* unchanged. Table 4 reports Real Time factors for in case of one, two and four feature streams for each of the modules. Process are run on a Dual Core Intel(R) CPU 6700 2.66GHz machine. It can be noticed that only the *aibfeat* modules result in a significant running time increase while, clustering (IBclust) and realignment (IBrealign) have similar RT factors regardless of number of features.

5. Discussion

This paper introduces DiarTk, a C++ open source toolkit for multistream speaker diarization released under GPL license[8].

In contrary to other diarization toolkit, DiarTk is explicitly designed for handling an arbitrary number of feature streams and it is expected to facilitate research and tests in novel feature types (visual information, location information) for diarizing multi-modal recordings while keeping limited the computational complexity. As the feature combination happens in a space of normalized relevance variables, the toolkit can handle easily very different statistics as acoustic features, location features or even visual features.

Furthermore the toolkit is provided with a set of recipes scripts to reproduce state-of-the-art results on the NIST Rich Transcription datasets.[9]

[8]The toolkit can be downloaded at http://www.idiap.ch/scientific-research/resources/speaker-diarization-toolkit.

[9]Acknowledgments: the authors would like to thank all the colleagues in the IM2 and AMI project for their help in setting the RT system. This work was supported by the Swiss National Center of Competence IM2, the EU Seventh Framework Program ([FP7/2007-2013] under grant agreement n [213850] in the SCALE project and in the EU Network of Excellence SSPNet.

Real Time Factors (RT)				
Features	IBfeat	aIBclust	IBrealign	Total
MFCC	0.06	0.03	0.02	0.11
+TDOA	0.12	0.03	0.03	0.18
+MS+FDLP	0.25	0.03	0.04	0.32

Table 4: Real Time factors obtained on a Intel(R) Core(TM)2 CPU 6700 2.66GHz machine for the three modules (IBfeat,aIBclust,IBrealign) whenever one,two or four different feature streams are used. The only part with a significant increase in complexity is the IBfeat.

6. References

[1] S. E. Tranter and D. A. Reynolds, "An overview of automatic speaker diarization systems," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 14, no. 5, 2006.

[2] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research," *IEEE Transactions On Acoustics Speech and Language Processing (TASLP), special issue on New Frontiers in Rich Transcription*, vol. 2, 2012.

[3] X. A. Jose M. Pardo and C. Wooters, "Speaker diarization for multiple-distant-microphone meetings using several sources of information," *IEEE Transactions on Computers*, vol. 56, no. 9, September 2007.

[4] R. Barra-Chicote, J. M. Pardo, J. Ferreiros, and J. M. Montero, "Speaker diarization based on intensity channel contribution," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 19, no. 4, 2011.

[5] A. K. Noulas, G. Englebienne, and B. J. A. Krose, "Multimodal speaker diarization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 79–93, 2012.

[6] G. Garau and H. Bourlard, "Using audio and visual cues for speaker diarisation initialisation," in *ICASSP*, 2010, pp. 4942–4945.

[7] M. Knox and G. Friedland, "Multimodal speaker diarization using oriented optical flow histograms," *Proceedings of Interspeech*, 2011.

[8] J. Schmalenstroeer and R. Haeb-Umbach, "Online diarization of streaming audio-visual data for smart environments," *J. Sel. Topics Signal Processing*, vol. 4, no. 5, pp. 845–856, 2010.

[9] O. Vinyals and G. Friedland, "Modulation spectrogram features for speaker diarization," *Proceedings of the 9th International Conference of the ISCA*, 2008.

[10] D. Vijayasenan, F. Valente, and H. Bourlard, "**An Information Theoretic Approach to Speaker Diarization of Meeting Data**," *IEEE Transactions on Audio Speech and Language Processing*, vol. 17, no. 7, pp. 1382–1393, September 2009.

[11] N. Tishby, F. Pereira, and W. Bialek, "The information bottleneck method," in *NEC Research Institute TR*, 1998.

[12] N. Slonim, N. Friedman, and N. Tishby, "Agglomerative information bottleneck," in *Proceedings of Advances in Neural Information Processing Systems*. MIT Press, 1999, pp. 617–623.

[13] D. Vijayasenan, F. Valente, and H. Bourlard, "An information theoretic combination of mfcc and tdoa features for speaker diarization," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 2, 2011.

[14] ——, "Multistream speaker diarization of meetings recordings beyond mfcc and tdoa features," *Speech Communication*, vol. 54, no. 1, Jan 2012.