Speaker Diarization Using Divide-and-Conquer

Shih-Sian Cheng[1], *Chun-Han Tseng*[2], *Chia-Ping Chen*[2], *and Hsin-Min Wang*[1]

[1]Institute of Information Science, Academia Sinica, Taipei, Taiwan
{sscheng, whm}@iis.sinica.edu.tw
[2]Dept. of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan
m943040041@student.nsysu.edu.tw, cpchen@mail.cse.nsysu.edu.tw

Abstract

Speaker diarization systems usually consist of two core components: speaker segmentation and speaker clustering. The current state-of-the-art speaker diarization systems usually apply hierarchical agglomerative clustering (HAC) for speaker clustering after segmentation. However, HAC's *quadratic* computational complexity with respect to the number of data samples inevitably limits its application in large-scale data sets. In this paper, we propose a divide-and-conquer (DAC) framework for speaker diarization. It recursively partitions the input speech stream into two sub-streams, performs diarization on them separately, and then combines the diarization results obtained from them using HAC. The results of experiments conducted on RT-02 and RT-03 broadcast news data show that the proposed framework is faster than the conventional segmentation and clustering-based approach while achieving comparable diarization accuracy. Moreover, the proposed framework obtains a higher speedup over the conventional approach on a larger test data set.

Index Terms: speaker diarization, speaker segmentation, speaker clustering, divide-and-conquer

1. Introduction

Speaker diarization, also known as the "who spoke when" task, aims to group together speech segments produced by the same speaker within an audio stream [1]. This technique is a vital processing step for automatic audio transcription/indexing [1] and spoken document retrieval [2]. It has been studied in various data domains, e.g., conversational telephone speech [3], broadcast news data [4, 5], and meeting data [6].

Speaker diarization systems usually consist of two core components, namely *speaker segmentation*, which chops the audio stream into homogeneous segments, and *speaker clustering*, which groups the homogeneous segments into speaker clusters. In the two-stage process, the audio stream is usually over-chopped in order to guarantee the homogeneity of the speech segment, considering that the error derived from the segment impurity will propagate in the clustering process. Various speaker segmentation and clustering approaches have been proposed. One can refer to [7] for a thorough review.

Currently, leading speaker diarization systems usually apply hierarchical agglomerative clustering (HAC) to perform speaker clustering [4, 5] after segmentation. Although HAC has been proven to achieve sound diarization performance, its *quadratic* computational complexity with respect to the number of data samples inevitably limits the application in large-scale data sets [8]. In this paper, we propose a divide-and-conquer (DAC) framework for speaker diarization. The proposed framework recursively divides the input speech stream into two sub-streams, performs diarization on the two sub-streams separately, and then combines the diarization results of the two sub-streams using HAC. Compared to the conventional two-stage (i.e., segmentation followed by clustering) diarization approach, the proposed DAC framework has a lower computational complexity by the algorithmic nature; hence, it is more suitable for large-scale data sets than the conventional two-stage approach. The results of experiments conducted on NIST RT-02 and RT-03 broadcast news data show that the proposed DAC framework is faster than the conventional approach while achieving comparable diarization accuracy.

The remainder of this paper is organized as follows. In Section 2, we briefly review HAC-based speaker clustering. The proposed divide-and-conquer diarization framework and the implementation details are described in Section 3. The experiment results are detailed in Section 4. We then present our conclusions in Section 5.

2. HAC-based speaker clustering

When performing HAC for speaker clustering, each speech segment given by speaker segmentation is considered a cluster initially; then, in each merging step, the two clusters with the smallest distance measurement are merged into a new cluster. The two major aspects of HAC are the computation of the inter-cluster distances and the determination of the number of clusters.

An early well-recognized distance measure is the uni-Gaussian ΔBIC computed by [4, 9]:

$$\Delta BIC(\mathcal{X},\mathcal{Y}) = \frac{n_z}{2}\log|\hat{\mathbf{\Sigma}}_z| - \frac{n_x}{2}\log|\hat{\mathbf{\Sigma}}_x| - \frac{n_y}{2}\log|\hat{\mathbf{\Sigma}}_y| \\ -\frac{1}{2}\lambda(d+\frac{1}{2}d(d+1))\log n_z, \quad (1)$$

where $\mathcal{X}$ and $\mathcal{Y}$ represent two clusters to be checked; $\mathcal{Z}$=$\mathcal{X}\cup\mathcal{Y}$; n_x, n_y, and n_z are the numbers of data samples of $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$, respectively; $\hat{\mathbf{\Sigma}}_x$, $\hat{\mathbf{\Sigma}}_y$, and $\hat{\mathbf{\Sigma}}_z$ are the sample covariance matrices of $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$, respectively; and d is the dimension of the feature vector of a data sample. ΔBIC is the difference between two evaluation scores based on the Bayesian Information Criterion (BIC) [10]: 1) the union of the feature vectors of the two clusters forms a Gaussian distribution in the feature space, and 2) the feature vectors of each cluster form a distinct Gaussian distribution. According to the BIC theory, the penalty factor λ in Eq. (1) is 1; however, in practical clustering tasks, it is usually adjusted to allow a tradeoff between error types.

This work was supported in part by the National Science Council, Taiwan, under Grant: NSC96-2628-E-001-024-MY3.

When $\lambda = 0$, ΔBIC is equivalent to the generalized likelihood ratio (GLR) [11]. ΔBIC can also be used as a stopping criterion in such a way that the merging process is stopped when the smallest ΔBIC value among all cluster pairs is larger than zero; this is called the local BIC-based approach [4].

Recently, a more sophisticated distance measure is derived from the Gaussian mixture model (GMM). For example, in [12], Ajmera and Wooters proposed a threshold-free variant of ΔBIC, where each cluster is modeled with a GMM. Motivated by the GMM-UBM method widely used in the speaker recognition task [13], Barras *et al.* [4] proposed a GMM-based cross likelihood ratio (CLR) as follows:

$$CLR_{GMM}(\mathcal{X}, \mathcal{Y}) = \frac{1}{n_x} \log \frac{p(\mathcal{X}|M_y)}{p(\mathcal{X}|B)} + \frac{1}{n_y} \log \frac{p(\mathcal{Y}|M_x)}{p(\mathcal{Y}|B)}, \tag{2}$$

where M_x and M_y are the GMMs for $\mathcal{X}$ and $\mathcal{Y}$ estimated via maximum *a posteriori* (MAP) adaptation from the universal background model (UBM) B. CLR_{GMM} reveals the similarity between $\mathcal{X}$ and $\mathcal{Y}$. Therefore, when applying this measure in HAC, the two clusters with the largest CLR_{GMM} value are merged. The merging process is stopped when the largest CLR_{GMM} value among all cluster pairs is smaller than a predefined threshold δ_{CLR}.

Rather than using a single distance measure during the merging process of HAC, some leading speaker diarization systems apply multiple measures in HAC [1, 4]. For example, in [4], the authors first applied ΔBIC in the initial clustering process, and then applied CLR_{GMM} in the second clustering process. This approach is to stop the initial clustering stage early, and use the results to seed a second clustering stage with more initial data per cluster. This second stage can therefore estimate more complex models for the clusters. In [4], the initial stage is called BIC clustering, while the second stage is called speaker identification (SID) clustering.

3. DAC-based speaker diarization

The proposed divide-and-conquer framework for speaker diarization (DACDiar) is shown in Algorithm 1.

As shown in Figure 1 (a), the input audio stream is first passed through a speech activity detection (SAD) component to filter out non-speech data. As with the SAD method in [4], the GMMs for speech, noisy speech, speech over music, pure music, and silence/noise are trained beforehand. Then, the speech, noisy speech, and speech over music segments are extracted from the input stream with Viterbi decoding. After SAD, the speech clips are concatenated into the speech stream W. Then, a distance curve is obtained by evaluating the GLR values of two adjacent windows that slide along W. The time index t associated with the peak that has the largest GLR value within the interval $[t - pRange, t + pRange]$ is considered a divide-point. In this way, the size of a resulting speech segment is at least $pRange$ seconds. In this example, all the peaks except S are divide-points.

The *Check termination* stage: If there is no divide-point within W, DACDiar returns Cls that contains only one speech segment (i.e., one cluster), W, which is the initial unit for diarization.

The *Divide* stage: In this stage, the speech stream is partitioned into two sub-streams at the time index of the divide-point with the largest GLR value. Then, the sub-streams together with their corresponding divide-point set and GLR set are input to DACDiar in the *Solve sub-instances* stage.

The *Combine* stage: Suppose that the clustering results obtained from the *Solve sub-instances* stage, Cl_{W_1} and Cl_{W_2}, contain m and n clusters, respectively, this stage performs HAC on the $m+n$ clusters because there may be clusters from Cl_{W_1} and Cl_{W_2} that are produced by the same speaker.

3.1. Implementations

As discussed in Section 2, various clustering criteria can be applied in HAC-based clustering for combining Cl_{W_1} and Cl_{W_2} in the *Combine* stage. We implemented the proposed DACDiar framework in three ways.

a) DACDiar_BIC: Cl_{W_1} and Cl_{W_2} are combined using HAC with ΔBIC as the inter-cluster distance measure and stopping criterion (BIC clustering).

b) DACDiar_SID: Cl_{W_1} and Cl_{W_2} are combined using HAC with CLR_{GMM} as the inter-cluster distance measure and δ_{CLR} as the pre-defined threshold (SID clustering).

c) DACDiar_BIC_SID: In this approach, we implemented the *Combine* stage as,

```
//Combine
if ($|Cl_{W_1}| > 1$ and $|Cl_{W_2}| > 1$)
    $Cls \leftarrow$perform SID clustering on $Cl_{W_1} \cup Cl_{W_2}$;
else
    $Cls \leftarrow$perform BIC clustering on $Cl_{W_1} \cup Cl_{W_2}$;
```

where $|\cdot|$ denotes the size of a set. $|Cl_{W_i}| > 1$ $(i = 1, 2)$ indicates that the smallest ΔBIC value among all cluster pairs extracted from Cl_{W_i} is larger than zero.

We use the recursive tree example in Figure 1 (b) to explain the clustering process of DACDiar_BIC_SID. In the figure, each tree node corresponds to a divide-point in W; the number inside the node indicates the order of the division, while the number below the node indicates the order in which clustering (*Combine*) is performed. In this example, BIC clustering is performed in order for nodes 3, 5, 6, 4, 2, 8, and 7, because, for these nodes, at least one of their child nodes contains only one cluster; then, SID clustering is performed for node 1, because both of its child nodes contain more than one cluster.

For DACDiar_BIC (DACDiar_SID), however, BIC clustering (SID clustering) is performed for each tree node.

4. Experiments

Our experiments were conducted on NIST RT-02 and RT-03 broadcast news data. RT-02, which consisted of six 10-minute news shows, was used as the development set (DEV); while RT-03, which consisted of six 30-minute shows from channels ABC, CNN, NBC, PRI, MNB, and VOA, respectively, was used as the evaluation set (EVAL).

For acoustic feature extraction, 12 Mel frequency cepstrum coefficients (MFCCs) and the energy were used for producing the GLR distance curve in Figure 1 (a) and for BIC clustering; while 15 MFCCs plus delta coefficients and delta energy with feature warping normalization [4] were used for SID clustering. The 1998 DARPA/NIST HUB-4 broadcast news evaluation test data was used to train the UBM for SID clustering, each containing 128 mixture Gaussians, and the GMMs for speech, noisy speech, speech over music, pure music, and silence/noise for SAD, each containing 64 mixture Gaussians.

For the performance evaluation, we used the diarization evaluation tool (md-eval-v21.pl) released by NIST [14] to evaluate the diarization error rate.

Algorithm 1 $Cls \leftarrow$DACDiar(W, DP_{set}, GLR_{set})

```
Require: $W$: the speech stream;
    $DP_{set} = \{DP_1, \ldots, DP_N\}$: the divide-points in $W$;
    $GLR_{set} = \{GLR_1, \ldots, GLR_N\}$: $GLR_i$ denotes the
    GLR value at $DP_i$ for $i = 1, 2, \cdots, N$;
Ensure: $Cls$: the set of output clusters
  1) //Check termination
    if ($DP_{set}$ is empty)
        $Cls \leftarrow W$;
        return;
  2) //Divide
    search in $DP_{set}$ and let $DP_k$ be the divide-point whose
    GLR value is the largest in $GLR_{set}$;
    let $\hat{t}$ be the time index of $DP_k$;
    divide $W$ into two sub-streams, $W_1$ and $W_2$, at $\hat{t}$;
    divide $DP_{set}$ into two sub-sets,
    $DP_{set1} = \{DP_1, \ldots, DP_{k-1}\}$ and
    $DP_{set2} = \{DP_{k+1}, \ldots, DP_N\}$;
    divide $GLR_{set}$ into two sub-sets,
    $GLR_{set1} = \{GLR_1, \ldots, GLR_{k-1}\}$ and
    $GLR_{set2} = \{GLR_{k+1}, \ldots, GLR_N\}$;
  3) //Solve sub-instances
    $Cl_{W_1} \leftarrow DACDiar(W_1, DP_{set1}, GLR_{set1})$;
    $Cl_{W_2} \leftarrow DACDiar(W_2, DP_{set2}, GLR_{set2})$;
  4) //Combine
    $Cls \leftarrow$ perform clustering on $Cl_{W_1} \cup Cl_{W_2}$;
```

4.1. Experiment results

We used three HAC-based speaker clustering methods to implement the conventional two-stage (segmentation followed by clustering) diarization approach: 1) BIC clustering (HAC_BIC), 2) SID clustering (HAC_SID), and 3) BIC clustering followed by SID clustering (HAC_BIC_SID). They respectively correspond to the baseline systems of the three implementations of the proposed DAC framework, namely DACDiar_BIC, DACDiar_SID, and DACDiar_BIC_SID. All the baseline systems used the same speech segmentation method, as shown in Figure 1(a), as the proposed DAC-based approaches. Therefore, we can fairly evaluate the advantage of integrating HAC into a DAC framework.

First, we conducted experiments on DEV using HAC_BIC to evaluate the proper setting for the size of the sliding window and $pRange$ used in speech segmentation (*cf.* Figure 1(a)). We found that it was appropriate to set both at three seconds. In the following experiments, this setting was applied to all approaches.

Table 1 shows the diarization error rates (DERs) of all the approaches investigated in this paper. From the table, we observe that the proposed approaches obtain similar DERs compared to their respective baselines, except for DACDiar_BIC (vs. HAC_BIC) on the DEV set. Comparing the DERs of DACDiar_SID and HAC_SID to those of DACDiar_BIC and HAC_BIC, it is clear that the approaches that use CLR_{GMM} defined in Eq. (2) as the distance measure in HAC substantially outperform the approaches that apply ΔBIC defined in Eq. (1). For example, compared to DACDiar_BIC, DACDiar_SID achieves a 44.19% relative DER reduction (11.91% vs. 21.34%) on DEV, and a 10.99% relative DER reduction (19.69% vs. 22.12%) on EVAL. Moreover, applying multiple distance measures in speaker clustering improves the diarization accuracy. From Table 1, we observe that DACDiar_BIC_SID (HAC_BIC_SID) outperforms DACDiar_SID (HAC_SID) on both DEV and EVAL.

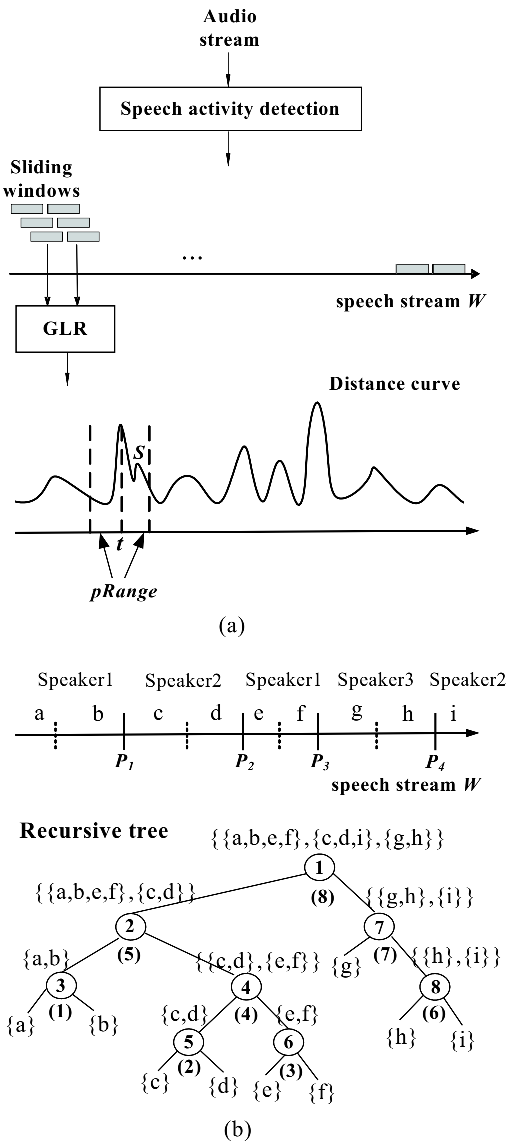

Figure 1: (a) The process for extracting the speech stream W from an audio stream and creating divide-points for the speech stream. In this example, all the peaks except S are divide-points. (b) An example of recursive tree illustration of DACDiar based on the divide-points in (a). Here, it is assumed that the diarization process is perfect, i.e., no segmentation and clustering errors occur. P_1, P_2, P_3, and P_4 are the speaker change points.

Table 2 shows the speeds of all approaches in terms of real-time factor, xRT=T_s/T_d, where T_s is the system runtime of the clustering module and T_d denotes the time duration of the test data set. Since all approaches share the same speech segmentation strategy, the runtime of segmentation is excluded in the evaluation. The average numbers of speech segments obtained by fixed-size sliding window segmentation on DEV and EVAL are 69.5 and 165.5, respectively. Therefore, we may think that the size of EVAL is about 2.38 times that of DEV. From Table 2, several observations can be drawn. 1) The speedup obtained by

Table 1: The speaker diarization error rates (DERs) of different approaches.

Approach	λ/δ_{CLR}	Data set	DER (%)
HAC_BIC	$\lambda = 5.8$	DEV	17.56
		EVAL	21.27
DACDiar_BIC	$\lambda = 4.6$	DEV	21.34
		EVAL	22.12
HAC_SID	$\delta_{CLR} = 0.3$	DEV	12.06
		EVAL	18.67
DACDiar_SID	$\delta_{CLR} = 0.3$	DEV	11.91
		EVAL	19.69
HAC_BIC_SID	$\lambda = 1.4$	DEV	10.9
	$\delta_{CLR} = 0.3$	EVAL	17.15
DACDiar_BIC_SID	$\lambda = 1.7$	DEV	10.82
	$\delta_{CLR} = 0.3$	EVAL	17.45

Table 2: The real-time factor, xRT, of different approaches.

Approach	Data set	xRT	Speedup of DAC over HAC
HAC_BIC	DEV	0.04	-
	EVAL	0.1	-
DACDiar_BIC	DEV	0.003	13.33
	EVAL	0.005	20
HAC_SID	DEV	0.86	-
	EVAL	1.6	-
DACDiar_SID	DEV	0.29	2.97
	EVAL	0.39	4.1
HAC_BIC_SID	DEV	0.35	-
	EVAL	0.61	-
DACDiar_BIC_SID	DEV	0.21	1.67
	EVAL	0.28	2.18

integrating HAC into the DAC framework increases when the size of test data increases; e.g., the speedup of DACDiar_BIC over HAC_BIC increases from 13.33 on DEV to 20 on EVAL. 2) The time cost of SID clustering is much higher than that of BIC clustering; e.g., xRT of DACDiar_SID (HAC_SID) on DEV is 0.29 (0.86), whereas xRT of DACDiar_BIC (HAC_BIC) is 0.003 (0.04). This is because the complexity of calculating CLR_{GMM} is much higher than the complexity of calculating ΔBIC. 3) DACDiar_BIC_SID (HAC_BIC_SID) is faster than DACDiar_SID (HAC_SID). This is because the former applies BIC clustering initially, which is faster than SID clustering; while the latter applies SID clustering throughout the clustering process. It is worth mentioning that the time cost of SID clustering is in general proportional to the number of speech segments to be clustered and the current number of clusters to be checked for merging. This is why HAC_SID is the slowest among all approaches tested in this paper and the speedup of DACDiar_BIC_SID over HAC_BIC_SID is smaller than that of DACDiar_SID over HAC_SID.

5. Conclusions

We have proposed a divide-and-conquer framework for speaker diarization. The proposed DAC framework recursively partitions the input speech stream into two sub-streams, performs diarization on them separately, and then combines the diarization results obtained from them using HAC. By integrating HAC into a DAC framework, the diarization process can be more efficient. We have implemented the proposed DAC framework using three clustering methods, namely BIC clustering (HAC_BIC), SID clustering (HAC_SID), and BIC clustering followed by SID clustering (HAC_BIC_SID). Our experiment results show that the proposed approaches achieve comparable diarization accuracy to their associated conventional approaches and a higher speedup is obtained when testing on a larger data set.

Like many existing speaker diarization systems, the diarization error rates of the proposed systems may be reduced by integrating other processing steps/components into the diarization procedure. For example, as in [5], we can use Viterbi resegmentation as a post-processing step. We may also integrate gender/bandwith classification [5] into the DAC framework.

6. References

[1] S. E. Tranter and D. A. Reynolds, "An overview of automatic speaker diarization systems," *IEEE Trans. Audio, Speech and Language Processing*, vol. 14, no. 5, pp. 1557–1565, 2006.

[2] H. M. Wang, S. S. Cheng, and Y. C. Chen, "The SoVideo Mandarin Chinese broadcast news retrieval system," *International Journal of Speech Technology*, vol. 7, no. 2-3, pp. 189–202, 2004.

[3] X. Zhong, M. Clements, and S. Lim, "Acoustic change detection and segment clustering of two-way telephone conversation," in *Proc. Eur. Conf. Speech Communication and Technology*, Geneva, Switzerland, Sep. 2003, pp. 2925–2928.

[4] C. Barras, X. Zhu, S. Meignier, and J.-L. Gauvain, "Multistage speaker diarization of broadcast news," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1505–1512, 2006.

[5] S. Meignier, D. Moraru, C. Fredouille, J.-F. Bonastre, and L. Besacier, "Step-by-step and integrated approaches in broadcast news speaker diarization," *Computer Speech & Language*, vol. 20, pp. 303–330, 2006.

[6] J. M. Pardo, X. Anguera, and C. Wooters, "Speaker diarization for multiple-distant-microphone meetings using several sources of information," *IEEE Trans. Computers*, vol. 56, no. 9, pp. 1212–1224, 2007.

[7] M. Kotti, V. Moschou, and C. Kotropoulos, "Speaker segmentation and clustering," *Signal Processing*, vol. 88, pp. 1091–1124, 2008.

[8] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Trans. Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[9] S. Chen and P. S. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the Bayesian information criterion," in *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, VA, Feb. 1998, pp. 127–132.

[10] G. Schwarz, "Estimation the dimension of a model," *Annals of Statistics*, vol. 6, pp. 461–464, 1978.

[11] P. Delacourt and C. J. Wellekens, "DISTBIC: A speaker-based segmentation for audio data indexing," *Speech Communication*, vol. 32, no. 1, pp. 111–126, 2000.

[12] J. Ajmera and C.Wooters, "A robust speaker clustering algorithm," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, US Virgin Islands, USA, Dec. 2003, pp. 411–416.

[13] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, pp. 19–41, 2000.

[14] NIST, *Rich Transcription Spring 2006 Evaluation*, http://www.nist.gov/speech/tests/rt/2006-spring/index.html.