Unsupervised sequential organization for cochannel speech separation

Ke Hu and DeLiang Wang

Department of Computer Science and Engineering
& Center for Cognitive Science
The Ohio State University
Columbus, OH 43210-1277, USA
{huk, dwang}@cse.ohio-state.edu

Abstract

The problem of sequential organization in the cochannel speech situation has previously been studied using speaker-model based methods. A major limitation of these methods is that they require the availability of pretrained speaker models and prior knowledge (or detection) of participating speakers. We propose an unsupervised clustering approach to cochannel speech sequential organization. Given enhanced cepstral features, we search for the optimal assignment of simultaneous speech streams by maximizing the between- and within-cluster scatter matrix ratio penalized by concurrent pitches within individual speakers. A genetic algorithm is employed to speed up the search. Our method does not require trained speaker models, and experiments with both ideal and estimated simultaneous streams show the proposed method outperforms a speaker-model based method in both speech segregation and computational efficiency.

Index Terms: sequential grouping, cochannel speech separation, clustering

1. Introduction

Cochannel speech separation is the task of separating two simultaneous speech signals in a single channel. This is a very challenging task considering the significant amount of speech overlap between two talkers and only one available mixture. Despite the difficulty of this task, humans show remarkable ability to select and follow one speaker under such conditions. Bregman calls this perceptual process auditory scene analysis [1], which takes place in two main stages: segmentation and grouping. Segmentation decomposes an auditory scene into time-frequency (T-F) segments, each of which primarily originates from a single sound source, and grouping selectively aggregates them to form perceptual streams corresponding to sound sources. Grouping itself consists of simultaneous and sequential grouping. Simultaneous grouping organizes T-F segments across frequency to produce simultaneous streams, and sequential grouping organizes segments across time. In this work, we study how to sequentially organize simultaneous streams of two speakers in an unsupervised manner.

Previous research on sequential grouping of cochannel speech uses speaker model based methods. In [2], Shao and Wang extend the traditional speaker identification framework to the cochannel situation and perform sequential organization by maximizing the joint speaker recognition score given all possible groupings and speaker pairs. Their method is further developed in [3] to deal with the situation where only the target speaker model is known. Similarly, a CASA system in [4] employs speaker-dependent (SD) hidden Markov models and searches for the best grouping by coupling segmentation with speech recognition. Related model based methods directly recover individual speech signals [5], [6]. Model-based methods can achieve satisfactory performance when trained models match those of participating speakers. However, this condition is often not met in practice.

On the other hand, unsupervised speaker clustering aims to organize speech contents based on speaker identities in multi-talker environments. For example, in [7], two Gaussian mixture models (GMM), each representing one speaker, are built from two speaker-homogeneous sections of the mixture on the fly and used to label such sections. Sequential grouping resembles speaker clustering except for two major differences. First, simultaneous streams in sequential grouping contain spectrally separated components while the speech sections in speaker clustering consist of whole frames. Second, a simultaneous stream is much shorter than a speech section in speaker clustering. The analysis of Ofoegbu et al. [8] on intra- and inter-speaker distances of voiced speech suggests that a minimum of 5 phones is needed for speaker separability. Short simultaneous streams generally do not contain enough acoustic information for speaker clustering. To verify this, we have directly applied speaker clustering methods for sequential grouping but found unsatisfactory results.

We propose a search based clustering approach for cochannel speech sequential grouping. Unreliable units in simultaneous streams are first reconstructed using a speech prior, and cepstral features are subsequently derived for clustering. We search for two clusters exhibiting the biggest speaker difference, i.e. the trace of the between- and within-cluster scatter matrix ratio. An exhaustive search becomes computationally expensive as the mixture length grows. Thus we employ a genetic algorithm to speed up the search.

In the next section, we describe proposed clustering based sequential organization. Evaluation and comparison are given in Section 3, and we conclude the paper in Section 4.

2. Unsupervised sequential organization

2.1. Early processing

The input mixture is first decomposed into the T-F domain using a 128-channel gammatone filterbank with center frequencies ranging from 50 Hz to 8000 Hz [9]. Gammatone features (GF) are then extracted by downsampling each of the 128-channel outputs to 100 Hz along the time dimension and compressing the magnitude of each downsampled output by a cubic root operation [10].

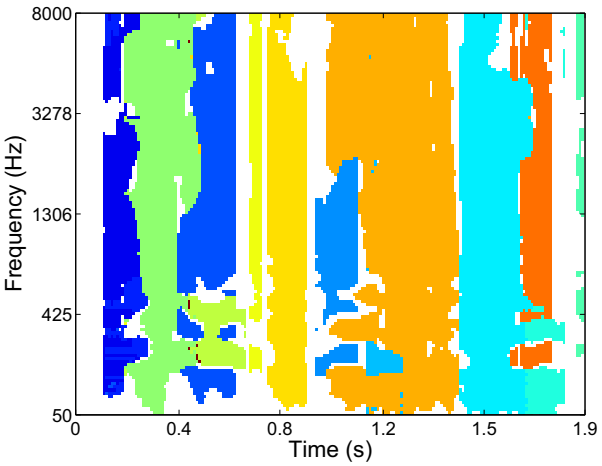

Figure 1: *An example of simultaneous streams estimated using the tandem algorithm.*

To extract simultaneous streams, we employ a recently developed tandem algorithm [11]. This algorithm performs segmentation based on cross-channel correlations, and forms initial pitch contours and corresponding simultaneous streams using harmonicity and temporal continuity. Given the initial estimates, the algorithm re-estimates pitch contours and their corresponding simultaneous streams jointly and iteratively. The resulting simultaneous streams are represented by binary masks, which are estimates of the ideal binary mask (IBM) [12]. In the IBM, 1 indicates target dominance and 0 interference dominance. The tandem algorithm is shown to significantly improve the SNR of segregated speech under various noise conditions. An example of estimated simultaneous streams is shown in Fig. 1 for a cochannel speech mixture. Each colored region represents one simultaneous stream.

2.2. Feature reconstruction

Before clustering, we derive gammatone frequency cepstral coefficients (GFCC) for simultaneous streams [10]. First, unreliable GF units (those labeled as 0 in a binary mask) are reconstructed using a speech prior and the reconstructed GFs are converted to GFCCs through discrete cosine transform.

We denote a GF feature vector derived from one frame of the mixture as $\mathbf{X}$. According to the estimated binary mask, a GF vector can be partitioned into reliable units $\mathbf{X}_r$, and unreliable ones $\mathbf{X}_u$. To enhance the GF vector, we use a GMM speech prior $p(\mathbf{X})$ to reconstruct the unreliable units as the mean conditioned on the reliable units

$$\hat{\mathbf{X}}_u = \sum_{k=1}^{K} p(k|\mathbf{X}_r)\mu_{u,k}, \tag{1}$$

where K is the number of Gaussians, k is the Gaussian index, and $\mu_{u,k}$ refers to the mean vector of the uth unreliable unit in the kth Gaussian of the speech prior. The reliable units are retained in the reconstruction. We calculate the posterior probability of the kth Gaussian given the reliable GF units as

$$p(k|\mathbf{X}_r) = \frac{p(k)p(\mathbf{X}_r|k)}{\sum_{k=1}^{K} p(k)p(\mathbf{X}_r|k)}. \tag{2}$$

where $p(k)$ represents the prior of the kth Gaussian. As in [10], diagonal covariance matrices are assumed in our GMM model.

Then for every frame, the reconstructed GFs are transformed into GFCCs. Our GMM model is trained using premixed uttrances from speakers other than target and interfering speakers; one can expect better reconstruction performance with matched speaker models.

2.3. Objective function

We formulate sequential organization as a problem of unsupervised clustering: simultaneous streams will be clustered into two speakers. Different clustering possibilities are evaluated using an objective function, and the one with the highest score is chosen as the result.

Clustering aims to find a partition of data so that objects in the same cluster are similar while those in different clusters are far apart [13]. Given different speakers, one objective function for cochannel speech separation would be to measure the acoustic difference of two simultaneous stream groups. Given a binary label vector $\mathbf{g}$, we thus pool the GFCC features of simultaneous streams in each group and measure the group difference by calculating the trace of the matrix from the product of the between-cluster and within-cluster matrices

$$O(\mathbf{g}) = \mathrm{tr}(\mathbf{S}_W^{-1}(\mathbf{g})\mathbf{S}_B(\mathbf{g})) \tag{3}$$

where $\mathbf{S}_W(\mathbf{g})$ and $\mathbf{S}_B(\mathbf{g})$ are the within-cluster scatter matrix and between-cluster scatter matrix with respect to $\mathbf{g}$, respectively. The trace operation amounts to measuring the ratio of the between- and within-cluster scatter matrices along the eigenvector dimensions [13].

While maximizing (3), two simultaneous streams with overlapping pitch contours should not be assigned to the same speaker cluster. For any clustering $\mathbf{g}$ with a total of m overlapping frames in the two individual speaker clusters, we penalize this clustering by

$$P(\mathbf{g}) = 1/(1 + e^{a(m_{\mathbf{g}} - b)}), \quad a < 0 \text{ and } b \geq 0 \tag{4}$$

where $m_{\mathbf{g}}$ denotes the number of overlapping pitch frames for $\mathbf{g}$, and a and b are constants controlling the steepness of the penalty and tolerance to overlapping errors, respectively. The penalty function has a value ranging from 0 to 1. It will be 1/2 when there are b overlapping frames. Since a is negative, $P(\mathbf{g})$ will saturate to 1 as $m_{\mathbf{g}}$ increases and to zero when $m_{\mathbf{g}}$ is significantly smaller than b.

Adding the penalty, the objective function becomes

$$J(\mathbf{g}) = \lambda O(\mathbf{g}) - (1 - \lambda)cP(\mathbf{g}), \quad 0 \leq \lambda \leq 1 \tag{5}$$

where c is a constant which scales $P(\mathbf{g})$ to the range of $O(\mathbf{g})$, and λ controls the tradeoff between these two terms. In this work, we set c to be $\max_{\mathbf{g}} O(\mathbf{g})$. Empirically, we find that λ needs to be greater or equal than 0.5 to achieve good results.

Our objective function pools multiple simultaneous streams for clustering. We have also considered clustering simultaneous streams iteratively using a GMM based likelihood function [7] but obtained worse results. It is probably because an individual simultaneous stream does not contain sufficient speaker information. In addition, a sum-of-squared-error objective function is not chosen considering its sensitivity to outliers [13].

2.4. Search

Given the objective function, the clustering problem can thus be formulated as an optimization problem, i.e. to find a binary label vector $\mathbf{g}$ that maximizes the objective function (5). In principle, the optimal solution can be found by an exhaustive search. However, this method is only feasible when there is a relatively small number of simultaneous streams. When the system needs to process longer mixtures, this brute force method becomes computationally expensive. To overcome this difficulty, we use a genetic algorithm (GA) [14] to search for the optimal grouping.

In GA, each chromosome, often encoded as a binary string, represents a potential solution for a given problem. To start the search, GA randomly generates a set of chromosomes to form a seed population. These chromosomes are evaluated in parallel using a fitness function and according the fitness scores, individual chromosomes are altered through a set of operations including selection, crossover, and mutation to generate a new population. This procedure is repeated until the maximum number of generations is reached. The chromosome with the highest fitness score in the final population is taken as the GA solution.

Our clustering problem fits into the GA framework. In our task, each chromosome corresponds to a binary label vector $\mathbf{g}$. The fitness function for evaluating the partitions is the objective function in (5). For selection, we employ the linear ranking method [15] to prevent premature convergence. This method sorts chromosomes by their fitness values in an increasing order, and uses the output ranks to determine their corresponding numbers of offsprings. Crossover among chromosomes is performed by swapping the subsequences of the two chromosomes between two random points. A crossover probability is used to control the percentage of newly generated offsprings in the whole population. Mutation is carried out by replacing one element of a chromosome by a random number with a mutation probability. In this study, we set the initial population size, number of generations, and the crossover probability to be 500, 50, and 0.8, respectively. We have also tried other parameters and obtained similar results.

3. Evaluation and comparison

Following [3], we evaluate our algorithm by measuring the target speaker segregation performance. Two types of simultaneous streams, either estimated using the tandem algorithm or generated directly from the IBM, are employed for sequential grouping. For estimated simultaneous streams, we take the resynthesized speech from the voiced IBM as the ground truth and measure the SNR of segregated target speech as

$$\mathrm{SNR} = 10\log_{10}(\sum_n S_I^2[n]/\sum_n (S_I[n]-S_E[n])^2), \quad (6)$$

where $S_I[n]$ and $S_E[n]$ are the target signals resynthesized from the voiced IBM and the estimated voiced IBM, respectively. The voiced IBM is generated by taking the portions of the IBM with pitched frames. We also evaluate the system using ideal simultaneous streams derived from ground-truth pitch contours and IBMs. Specifically, ground-truth pitch contours are detected for each speaker from the premixed utterance using Praat [16] and the corresponding portions of the IBM are taken as simultaneous streams. To prevent the SNR to become infinity in this case, we compare the estimated masks with the overall IBM (i.e. not the voiced IBM). In addition, since our algorithm is unsupervised, we treat the segregated signal matching $S_I[n]$ better as the estimated target and the other as interference.

We create cochannel speech mixtures using the speech separation challenge (SSC) corpus [17]. The SSC corpus contains 34 speakers with both males and females. We use the test part of this corpus to generate two-talker mixtures. All utterances are first downsampled from 25 kHz to 20 kHz. For each utterance deemed a target, another utterance is randomly selected from other speakers and mixed with the target. The interfering utterance is either cut or concatenated with itself to match the length of the corresponding target signal. In total, we have created 100 mixtures at 0 dB for evaluation. Among them, 49 are mixtures of different gender (DG) talkers, and 51 are same gender (SG) mixtures. For feature reconstruction, we build a speech prior by training a 64-component GMM model of 128-dimensional GFs for each speaker, and then pooling the GMMs of all speakers other than those of the target and interfering speakers. For the penalty term in (4), a and b are ideally set to -10 and 0.5, respectively, to penalize concurrent pitches in a single speaker. However, since the tandem algorithm may overdetect pitches for a single speaker, we set a and b to -0.3 and 15, respectively, to tolerate such errors.

We compare our method to the background model (BM) based method of [3] since both algorithms operate on simultaneous streams for sequential grouping. For each input mixture, the BM method forms a target speaker set by randomly selecting a group of 10 speakers including the target one, and constructs the interferer model using all speakers other than the two constituent speakers. We emphasize that our method is completely speaker independent (SI) while the BM method needs to know the target speaker. The results with both ideal and estimated simultaneous streams are shown in Table 1, where the "SI" column under "Proposed" describes our performance, and the "BM" column shows that using the background model. Compared with the BM method, the proposed algorithm improves the segregation performance by 3.4 dB on average in the ideal case and 0.4 dB in the estimated case. The improvement decreases in the latter case, suggesting that our algorithm is more sensitive to errors in simultaneous stream estimation and pitch detection. On the other hand, our method would benefit more from improved simultaneous streams. Note that our method performs better for both SG and DG mixtures.

To test the speaker dependency of our method, we have also used SD models for feature reconstruction. The results are shown in the "SD" column under "Proposed" in Table 1. We have also incorporated different levels of prior information in the BM method: the "Target" column denotes the scenario where the target identity is provided directly, and the "SD" column represents that identities of both target and interfering speakers are given. As expected, the performance improves as more prior information is incorporated. In the SD case, our method performs a little better with ideal simultaneous streams but a little worse with estimated simultaneous streams. However, our method with a speaker-independent speech prior performs comparably or better than the BM method with prior target information.

To evaluate the effectiveness of the GA search, we show the grouping performance using exhaustive search in Table 1. The exhaustive search method performs only marginally better: an improvement of 0.5 dB in the ideal case and 0.2 dB in the estimated case. This indicates that our GA search does a good job in approximating the optimal solution. To establish a performance upper-bound, we also perform ideal sequential grouping (ISG). In ISG, a simultaneous stream is grouped as target if more than half of its energy is retained by the IBM.

Table 1: Comparisons of output SNRs (in dB) between the proposed method and a speaker-model based method

Simultaneous streams	Gender	MODEL-BASED			PROPOSED		EXHAUSTIVE		ISG
		BM	Target	SD	SI	SD	SI	SD	
Ideal	SG	7.9	8.9	13.0	12.2	14.1	13.2	14.1	14.4
	DG	11.5	11.9	15.3	14.1	15.2	14.1	15.2	15.7
	Both	9.7	10.4	14.1	13.1	14.6	13.6	14.6	15.0
Estimated	SG	3.5	3.9	5.5	3.7	5.1	3.8	5.2	6.5
	DG	6.8	7.1	8.6	7.4	8.2	7.7	8.5	9.0
	Both	5.1	5.5	7.0	5.5	6.6	5.7	6.8	7.7

Our method with SD reconstruction in the ideal case performs only 0.4 dB worse than the ISG, but the gap increases to 1.1 dB in the estimated case.

We have also measured the speed of the proposed algorithm as well as the BM method on an Intel Xeon 2.5 GHz server with 8 GB of RAM. The operating system is Linux Red Hat Enterprise 5.4. Table 2 summarizes the average runtime per mixture (about $1s$ long) for different methods as well as the speedup with respect to the BM method.

Table 2: Comparisons of average per-mixture runtime between the proposed methods and the BM method

	BM	Exhaustive	GA
Runtime (s)	61.4	55.2	26.5
Speedup	-	10%	57%

As shown in Table 2, the time complexity of the exhaustive search is not too bad in our case since the mixture length is around $1s$. It is even 10% faster than the BM method. By using the GA algorithm, the system speeds up the sequential grouping process by more than 50% compared to the BM method.

4. Conclusion

We have proposed a novel unsupervised clustering method for sequential organization in cochannel speech. With enhanced GFCC features, our method searches for the best clustering by maximizing the acoustic difference of two simultaneous stream groups. Our method does not use pretrained speaker models for separation. Systematic evaluations and comparisons show that our method outperforms a previous speaker-model based method with both ideal and estimated simultaneous streams. In addition, the proposed method is computationally more efficient.

5. Acknowledgements

This research was supported by an NSF grant (IIS-0534707), an AFOSR grant (FA9550-08-1-0155), and the VA Biomedical Laboratory Research and Development Program.

6. References

[1] A. Bregman, *Auditory Scene Analysis*. Cambridge, MA: MIT press, 1990.

[2] Y. Shao and D. L. Wang, "Model-based sequential organization in cochannel speech," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 1, pp. 289–298, 2006.

[3] ——, "Sequential organization of speech in computational auditory scene analysis," *Speech Comm.*, vol. 51, pp. 657–667, 2009.

[4] J. Barker, A. Coy, N. Ma, and M. Cooke, "Recent advances in speech fragment decoding techniques," in *Proc. Interspeech*, 2006, pp. 85–88.

[5] A. Reddy and B. Raj, "Soft mask methods for single-channel speaker separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 6, pp. 1766–1776, 2007.

[6] R. Weiss and D. Ellis, "Speech separation using speaker-adapted eigenvoice speech models," *Comput. Speech Lang.*, vol. 24, no. 1, pp. 16–29, 2010.

[7] B. Narayanaswamy, R. Gangadharaiah, and R. Stern, "Voting for two speaker segmentation," in *Proc. Interspeech*, 2006, pp. 2086–2089.

[8] U. Ofoegbu, A. Iyer, R. Yantorno, and S. Wenndt, "Unsupervised indexing of conversations with short speaker utterances," in *Proc. IEEE Aerospace Conference*, 2006, pp. 1–11.

[9] D. L. Wang and G. Brown, Eds., *Computational auditory scene analysis: Principles, algorithms, and applications*. Hoboken NJ: Wiley & IEEE Press, 2006.

[10] Y. Shao, "Sequential organization in computational auditory scene analysis," Ph.D. dissertation, Dept. of Comput. Sci. & Eng., The Ohio State Univ., 2007.

[11] G. Hu and D. L. Wang, "A tandem algorithm for pitch estimation and voiced speech segregation," *IEEE Trans. Audio, Speech, Lang. Process.*, in press, 2010.

[12] D. L. Wang, "On ideal binary mask as the computational goal of auditory scene analysis," in *Speech Separation by Humans and Machines*, P. Divenyi, Ed. Norwell, MA: Kluwer Academic press, 2005, pp. 181–197.

[13] R. Xu and D. Wunsch, *Clustering*. Hoboken NJ: Wiley & IEEE Press, 2009.

[14] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston MA: Addison-Wesley press, 1989.

[15] J. Baker, "Adaptive selection methods for genetic algorithms," in *Proc. ICGA 1*, 1985, pp. 101–111.

[16] P. Boersma and D. Weenink, "Praat: doing phonetics by computer (version 5.0.02)," Online: http://www.fon.hum.uva.nl/praat, 2007.

[17] M. Cooke and T. Lee, "Speech separation and recognition competition," Online: http://www.dcs.shef.ac.uk/ martin/SpeechSeparationChallenge.htm, 2006.