Using Class Purity as Criterion for Speaker Clustering in Multi-Speaker Detection Tasks

Gang Wang, Xiaojun Wu, Thomas Fang Zheng*, Linlin Wang and Chenhao Zhang
Center for Speech and Language Technologies, Division of Technical Innovation and Development,
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, Beijing, 100084
E-mail: {wanggang, wangll, zhangchh}@cslt.riit.tsinghua.edu.cn, {xjwu, fzheng}@tsinghua.edu.cn
* Corresponding Author: fzheng@tsinghua.edu.cn Tel/Fax: +86-10-62796393

***Abstract*—Speaker clustering is an important step in multi-speaker detection tasks and its performance directly affects the speaker detection performance. It is observed that the shorter the average length of single-speaker speech segments after segmentation is, the worse performance of the following speaker recognition will be achieved, therefore a reasonable solution to better multi-speaker detection performance is to enlarge the average length of after-segmentation single-speaker speech segments, which is equivalently to cluster as many true same-speaker segments into one as possible. In other words, the average class purity of each speaker segment should be as bigger as possible. Accordingly, a speaker-clustering algorithm based on the class purity criterion is proposed, where a Reference Speaker Model (RSM) scheme is adopted to calculate the distance between speech segments, and the maximal class purity, or equivalently the minimal within-class dispersion, is taken as the criterion. Experiments on the NIST SRE 2006 database showed that, compared with the conventional Hierarchical Agglomerative Clustering (HAC) algorithm, for speech segments with average lengths of 2 seconds, 5 seconds and 8 seconds, the proposed algorithm increased the valid class speech length by 2.7%, 3.8% and 4.6%, respectively, and finally the target speaker detection recall rate was increased by 7.6%, 6.2% and 5.1%, respectively.**

I. Introduction

Multi-speaker detection [1-3] is a kind of speaker recognition, whose object is to automatically identify which one in a pre-specified set of known target speakers is speaking during a given utterance containing multi-speaker. Multi-speaker detection can be applied in forensic and banking domain and it usually contains three steps, speaker segmentation, speaker clustering and speaker identification. The goal of speaker segmentation [3] is to segment an utterance into acoustically homogeneous segments, each of which contains only one speaker. Usually those segments' lengths are short. As is well known, the shorter the length of the identified utterance is, the worse the performance of the speaker identification system is. Hence, speaker clustering [3] must be implemented to enlarge the average length of after-segmentation single-speaker speech segments. Speaker clustering refers to the task of grouping unknown speaker utterances together based on their associated speakers.

The most commonly used method of speaker clustering is Hierarchical Agglomerative Clustering (HAC) [4]. The merit of HAC algorithm is simple and easy. And the shortcoming is that the performance seriously depends on the predefined threshold. Many improved HAC methods had been proposed, such as [5-10]. However, those methods are not particularly fit to multi-speaker detection tasks. The goal of speaker clustering in multi-speaker detection tasks is to increase the length of segments to reduce the affection of shorter segment length to the speaker identification system performance. However, these current methods much considered the average class purity [11-15], but there is not the direct relationship between the average class purity and the performance of the speaker identification system. Even if the average class purity is high, it is possible that the utterance of the target speaker is mixed with or submerged in another speaker's utterance, which makes the performance of the speaker identification system worse. In multi-speaker detection tasks, the single class's purity is a more notable parameter. The more the number of the classes with higher purity is, the better the speaker identification performance is.

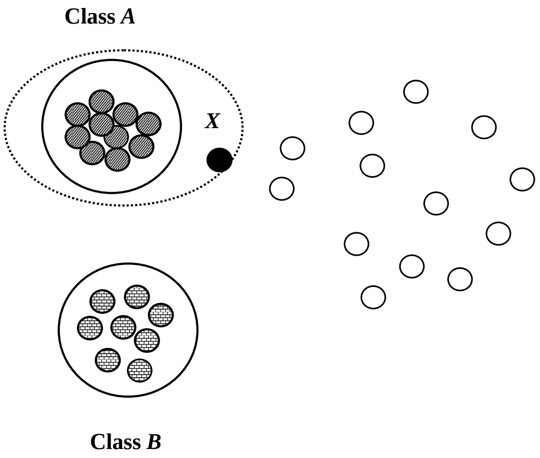

Fig. 1 Conventional HAC Algorithm Block Diagram

Fig. 1 is an illustration of the typical HAC algorithm. The utterances belonging to Class *A* and utterance *X* are the two utterances between them the distance is the smallest. If the distance is smaller than the predefined threshold, the utterance *X* would be merged into class *A*. Obviously, the merging would make the within-class dispersion of class *A* much bigger. In other words, the purity of class *A* would be much worse.

A more reasonable solution is that if the merging makes the within-class dispersion changing smaller, the merging continues. Else, the merging stops and the utterance itself forms a new class (See Fig. 2).

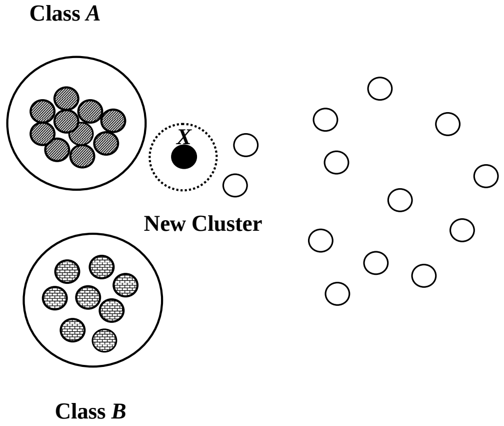

Fig. 2 Modified HAC Algorithm Block Diagram

According to the above idea, a class purity criterion based speaker clustering (CPCSC) algorithm is proposed to alleviate the influence on the performance of the following speaker identification due to the average short length of single-speaker speech after segmentation. Reference Speaker Model (RSM) [16] is used to calculate the distance between two utterances and the minimal within-class dispersion as well as the maximal class purity are taken as the criteria. It may reduce the probability of the speech segments by different speakers being clustered into one same class.

This paper is organized as follows. In Section II, the CPCSC algorithm is introduced. The experimental settings, results and analysis are given in Section III. Conclusions and future work are presented in Section IV.

II. Class purity Criterion based Speaker Clustering

A. The Class Purity Criterion

As is well known, on the condition of the purity of each class is higher, if the number of class is larger than the true speaker number in the utterance, the clustering's affection to multi-speaker detection is less than that the number of class is smaller than the true speaker number. Therefore, an assumption is given in CPCSC algorithm that the utterance contains two speakers at least. First, two segments are chosen as the two initial classes of which the probability belonging to the same speaker is the lowest. Then, a segment is chosen that is the nearest to the two initial classes. If the class purity change is smaller after merging, the segment would be merged into the nearest class to itself. If the class purity change is larger, the segment would be used to create a new class. Meanwhile, if the length of some class is equal or longer than the shortest length requirement of the speaker identification system, the class would not take part in the succedent clustering process, which would guarantee the single class's purity high although it makes the speaker identification's computation load bigger. But the affection of the increased computation is so little that it can be ignored, considering the benefit of higher purity brought by the CPCSC algorithm.

B. The CPCSC Algorithm

(1) Two class sets (C_I and C_S) are defined and set empty. All segments are labeled the status 'not clustered'. Those segments, which are longer than the shortest identification length (SIL), are chosen and put into C_I. SIL is decided by the performance of the speaker identification system and is set 5 seconds in the following experiments.

(2) Two segments are chosen from the segments whose statuses are 'not clustered' and the distance between them are the largest. Meanwhile, the lengths of the chosen segments are larger than 2 seconds. The two segments' statuses are set 'clustered' and put into class set C_S. The correlation based on RSM [16] is used as the distance measure between segments. The RSM set is denoted as R_1 and contains 256 RSM [16].

(3) RSM is a group of models and covers the whole acoustical space, so one segment is only nearer to the small part of the RSM. The small part RSM is used to calculation the distance, which would make the distance measure more accuracy. Therefore, the *N* RSM is chosen from R_1 which are the top *N* nearest to the segments in class set C_S. The *N* RSM is as a new RSM set and denoted as R_2. The value of *N* is 64.

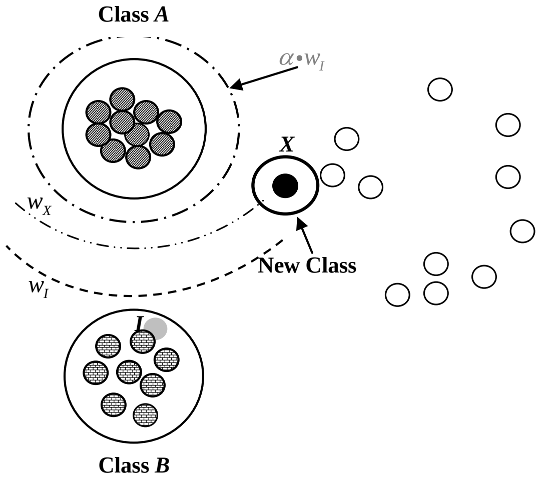

Fig. 3 CPCSC Algorithm Block Diagram

(4) The RSM set R_2 is used to calculate the distances between the segments whose statuses are 'not clustered' and the classes in class set C_S, respectively. Denote the segment as *X* between which and C_{Si} the distance is the smallest. C_{Si} is a

class in class set C_S. Calculate the within-class dispersion w_X with equation (1) after X is merged into class C_{Si}. Choose class B in class set C_S between which and C_{Si} the distance is the smallest. Choose the segment in class B between which and C_{Si} the distance is the smallest. And denote the segment as I. Calculate the within-class dispersion w_I after segment I is merged into class C_{Si}.

$$w = \frac{1}{C_N} \sum_{i=1}^{C_N} d\left(X_i, \overline{X_i}\right) \qquad (1)$$

where w is within-class dispersion and C_N is the number of segments in one class. X_i is the i-th segment in this class. $\overline{X_i}$ denotes the other segments in this class except segment X_i. $d(X_i, \overline{X_i})$ denotes the distance between segment X_i and segments $\overline{X_i}$. The distance calculation is the same as step (2) and RSM is used to calculate the distance.

$$w_X \leq \alpha \cdot w_I \qquad (2)$$

if equation (2) is satisfied, the segment X would be merged into class C_S. If equation (2) is not satisfied, the segment X would be as a new class and put into class set C_S. α is an adjustable parameter and $0 < \alpha < 1$.

Because class purity is not been obtained in the clustering process and the class purity is negative correlation with the within-class dispersion, within-class dispersion is used to evaluate the class purity. The larger the within-class dispersion is, the smaller the class purity is. Moreover, at the beginning of the algorithm running, the number of segments in each class is only 1, the calculation of within-class dispersion is impossible. Each segment is separated into two segments in the following experiments to solve it.

(5) If the segment length sum in class set C_S is larger than SIL, the class would be removed from class set C_S. Choose one segment from the segments whose statuses are ‘not clustered’ and between which and the class set C_S the distance is the largest. The segment is joined into class set C_S and labeled ‘clustered’.

(6) If there is not a segment whose status is ‘not clustered’, the clustering processing is terminated. Otherwise, go to step (3).

III. Experimental Results and Analysis

A. Database and experiment set up

The database for experiments is the National Institute of Standards and Technology (NIST) [1] Speaker Recognition Evaluation (SRE) 2006 database. 100 female speakers and 100 male speakers were chosen for speaker clustering. One three-minute utterance per speaker was segmented slice from 0.5 second to 10 seconds. Three kinds of average length databases were obtained (2 seconds, 5 seconds and 8seconds. Denote Set1, Set2 and Set3). All the utterances are in 8 kHz sampling rate with 8-bit width.

Feature extraction was performed on a 20 milliseconds frame every 10 milliseconds. The pre-emphasis coefficient was 0.97 and hamming windowing was utilized to each frame. 16-dimensional MFCC features were extracted with 30 triangular Mel filters in the MFCC calculation. For each frame, 32-dimentional feature vectors were extracted and formed of the 16-dimensional MFCC coefficients and their first derivative. Finally, the cepstral mean subtraction (CMS) [17] was applied.

The baseline system was a conventional HAC algorithm. RSM based correlation was used as the distance measure. A threshold was predefined to judge whether clustering process stops or merges two utterances. The proposed method (CPCSC) used the dynamical threshold such as step (4) and α was 0.4. The speaker identification system used the GMM-UBM algorithm [18]. NIST SRE 2004 1C4W dataset was used to train the UBM through EM algorithm [19]. The UBM was represented by an $M = 1{,}024$ Gaussian mixture density function, where the value of M was chosen empirically.

For evaluating the clustering performance, some parameters were defined. If one class contains only one speaker’s utterance, the class is called a valid class. Pure is the ratio between the length sum of valid classes and the length sum of all classes. Pure describes the clustering algorithm’s ability to obtain the valid class. The more the number of the valid class is, the better the clustering performance is and the less the affection to speaker identification is.

B. Experimental Results and Analysis

TABLE I
PURE COMPARISON WITH HAC

Database	GLR+HAC	RSM+HAC	RSM+CPCSC
Set1	78.5	79.6	81.2
Set2	82.5	84.5	86.3
Set3	84.9	87.4	89.5

In Table I, the digital is parameter *pure* (%). Compared with the conventional HAC algorithm, for speech segments with average lengths of 2 seconds (Set1), 5 seconds (Set2) and 8 seconds (Set3), the CPCSC algorithm increased the valid classes’ speech length by 2.7%, 3.8% and 4.6%, respectively.

TABLE II
THE PERFORMANCE OF SPEAKER DETECTION COMPARISON WITH HAC

Data	GLR+HAC	RSM+HAC	RSM+CPCSC
Set1	68.1/21.3	71.0/23.8	75.7/28.3
Set2	76.2/22.2	78.6/26.9	82.4/27.8
Set3	82.5/27.4	83.9/28.3	87.6/27.6

In table II, the digital is the parameter recall rate and precision rate (*R*/ *P*, %). In the speaker identification experiments top three candidate target speakers were given with the highest likelihood score. Compared with the conventional HAC algorithm, for speech segments with average lengths of 2 seconds (Set1), 5 seconds (Set2) and 8 seconds (Set3), the target speaker detection recall rate was increased by 7.6%, 6.2% and 5.1%, respectively. Because the top-3 candidates were used in speaker identification, the recall rate was better and meanwhile the precision rate was worse.

In table III's experiment, the clustered utterances in table II's experiment are concatenated to a long utterance. After the long utterance was segmented by RSM based speaker segmentation tools [16], the segmentation results were used to cluster. The clustering results were used to identify whether there existed the target speaker's speech.

TABLE III
THE PERFORMANCE OF MULTI-SPEAKER DETECTION COMPARISON WITH HAC

Data	Algorithm	*Pure* (%)	*R*/*P* (%)
Set1	HAC	60.9	61.0/22.1
	CPCSC	61.4	65.7/28.9
Set2	HAC	74.2	67.2/25/3
	CPCSC	76.0	73.5/30.6
Set3	HAC	80.5	76.5/27.5
	CPCSC	81.6	85.6/29.2

Due to the miss detection of speaker segmentation, there existed the multi-speaker utterance in the segmentation results, which made the clustering performance worse than the table II's experiments. Furthermore, the speaker detection performance was worse as well. However, the CPCSC algorithm was much better than HAC due to the higher class purity. Because there existed so many short utterances in database Set1 that the miss detection was more than Set2 and Set3, the clustering and speaker detection performance was much worse.

IV. CONCLUSIONS AND FUTURE WORK

In this paper, we propose to use Class Purity as the criterion for speaker clustering in multi-speaker detection tasks. In the criterion, the minimal within-class dispersion and the shortest identification length are taken as objective functions in order to guarantee high class purity. The experiments on the NIST SRE 2006 database show that the proposed algorithm can increase the valid class speech length and the target speaker recall rate for speech segments with different average length to a certain degree, compared with the conventional HAC algorithm.

However, the proposed algorithm is specifically proposed for multi-speaker detection tasks, it is not guaranteed to be useful to other tasks. What's more, it also brings some extra computation load to the speaker identification though not extreme. In the future, speaker-clustering algorithms with high purity without using SIL should be deeply studied.

REFERENCES

[1] NIST Speaker Recognition Evaluation Plan 2002, Online Available http://www.nist.gov/speech/tests/sre/.

[2] J. P. Campbell. Speaker recognition: A tutorial. Proceedings of the IEEE, vol. 85, pp. 1437--1462, September 1997.

[3] M. Kotti, V.Moschou, C. Kotropoulos. Speaker segmentation and clustering. Signal Processing, 2008, Vol. 88, pp: 1091-1124.

[4] H. Gish, M. H. Siu and R. Rohlicek. Segregation of speakers for speech recognition and speaker identification. IEEE International Conference on Acoustics Speech and Signal Processing, 1991: 873-876.

[5] H. Jin, F. Kubala and R. Schwartz. Automatic speaker clustering. DARPA Speech Recognition Workshop, 1997.

[6] A. Solomonoff, A. Mielke, M. Schmidt and H. Gish. Clustering speakers by their voices. In Proceedings of International Conference on Acoustics Speech and Signal Processing ICASSP, 1998, pp: 757-760.

[7] S. E. Johnson, P. C. Woodland. Speaker clustering using direct maximization of the MLLR-adapted likelihood. In Proceedings of the International Conference on Spoken Language Processing, ICSLP, 1998, Vol. 5, pp: 1775-1778.

[8] R. Faltlhauser and G. Ruske. Robust speaker clustering in eigenspace. In Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding, 2001.

[9] Y. Moh, P. Nguyen, J. C. Junqua. Towards domain independent speaker clustering. In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003, pp: 85-88.

[10] D. Liu, F. Kubala. Online speaker clustering. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP, 2004, Vol. 1, pp: 333-336.

[11] W. M. Rand, Objective criteria for the evaluation of clustering methods. Journal American Stat. Assoc. 1971, Vol. 66, pp: 846-850.

[12] W.-H. Tsai and H.-M. Wang. Speaker clustering of unknown utterances based on maximum purity estimation. In Proceedings of the European Conference on Speech Communication and Technology, Interspeech, 2005, pp: 3069-3072.

[13] W.-H. Tsai and H.-M. Wang. Evolutionary Minimization of the Rand Index for Speaker Clustering. Computer, Speech and Language, 2009, 23: 165-175.

[14] K. J. Han and S. S. Narayanan. A robust stopping criterion for agglomerative hierarchical clustering in a speaker diarization system. In Proceedings of the 8th Annual Conference of the International Speech Communication Association, Interspeech, 2007, pp: 1853-1856.

[15] X. Anguera, C. Wooters and J. Hernando. Purity algorithms for speaker diarization of meetings data. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 2006, pp: 1025-1028.

[16] Gang Wang and Thomas Fang Zheng. Speaker segmentation based on between-window correlation over speakers' characteristics. In Proceedings of APSIPA ASC, Japan, 2009.

[17] S. Furrui, Cepstral analysis technique for automatic speaker verification, IEEE Trans on Acoust. Speech Signal Processing 1981. Vol. 29, pp: 254-272.

[18] D. A. Reynolds, T. Quatieri, R. Dunn. Speaker verification using adapted Gaussian Mixture Models. Digital Signal Processing, 2000, Vol. 10, pp: 19-41.

[19] A. Dempster, N. Laird and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. J. Roy. Stat. Soc. 1977, Vol. 39, pp: 1–38.