IEICE TRANS. INF. & SYST., VOL.E102–D, NO.10 OCTOBER 2019

PAPER

# Scalable Community Identification with Manifold Learning on Speaker I-Vector Space

**Hongcui WANG**[†,††], **Shanshan LIU**[†], **Di JIN**[†a)], **Lantian LI**[†††], ***Nonmembers***, ***and*** **Jianwu DANG**[†,††††], ***Member***

**SUMMARY** Recognizing the different segments of speech belonging to the same speaker is an important speech analysis task in various applications. Recent works have shown that there was an underlying manifold on which speaker utterances live in the model-parameter space. However, most speaker clustering methods work on the Euclidean space, and hence often fail to discover the intrinsic geometrical structure of the data space and fail to use such kind of features. For this problem, we consider to convert the speaker i-vector representation of utterances in the Euclidean space into a network structure constructed based on the local ($k$) nearest neighbor relationship of these signals. We then propose an efficient community detection model on the speaker content network for clustering signals. The new model is based on the probabilistic community memberships, and is further refined with the idea that: *if two connected nodes have a high similarity, their community membership distributions in the model should be made close*. This refinement enhances the local invariance assumption, and thus better respects the structure of the underlying manifold than the existing community detection methods. Some experiments are conducted on graphs built from two Chinese speech databases and a NIST 2008 Speaker Recognition Evaluations (SREs). The results provided the insight into the structure of the speakers present in the data and also confirmed the effectiveness of the proposed new method. Our new method yields better performance compared to with the other state-of-the-art clustering algorithms. Metrics for constructing speaker content graph is also discussed.

***key words:*** *community detection, i-vector space, manifold learning, speaker clustering, speaker graph content*

## 1. Introduction

With the increasing use of speech in the network, there are amassing of large volumes of audio, including broadcasts, voice mails, meetings and other "spoken documents". Therefore, there is a growing need for speaker diarization systems which are, given an unlabeled audio file, to mark where speaker changes occur (segmentation), and then associate the different segments of speech belonging to the same speaker (clustering) [1]. Many new approaches are explored and proposed in this area [2].

The speaker clustering is the focus of our research efforts in this paper. It is one of the most important tasks in many applications. The outputs of speaker clustering can be used to help speech recognition, to facilitate the searching and indexing of audio archives, and to increase the richness of automatic transcriptions, making them more readable. Next we consider to use community detection algorithms to perform large-scale clustering on network structure graphs. We refer a unique speaker in the process of clustering as a community.

Nowadays, the problem of speaker clustering is mainly considered on large scale data [3], [4]. The state-of-the-art methods often first converted each speech signal into a high dimensional vector-based representation space using the techniques such as GMM supervectors [5], Joint Factor Analysis [6] and i-vectors [7], and and then employed agglomerative hierarchical clustering (AHC) algorithms [8]–[10] for the recognition. This method starts by initializing each speech segments as a singleton cluster and then iteratively merges the nearest pair of speech segments clusters by calculating the distance between different speech segments clusters. The stopping criterion is vital for the performance [9]. However, mapping an entire corpus of speech utterances to a set of vectors will lead to questions about the structure of the underlying manifold. But most data clustering methods work on the Euclidean space, and hence often fail to discover the intrinsic geometrical and discriminating structure of the data space, which limits their application on some complicated speaker recognition situations.

In order to model the underlying manifold structure of the data space, we convert the i-vector representation of speech signals in the Euclidean space into a network structure constructed based on the local ($k$) nearest neighbor relationship of these signals. We then propose a community detection model for the recognition of speakers, which is built upon the assumption that the group of speech signals corresponding to a same speaker will be densely connected with respect to the rest of the network [11]. Furthermore, the similarities of speech signals are also not useless. Here we refine the model with the idea that: if two speech signals have a high similarity in the local Euclidean space, their community membership distributions should be made close in the model. This further enhances its local invariance, i.e., if two data points are close in the intrinsic geometry of the data distribution, then the new representations of the two points with respect to the new basis, are also close to each other, which is essential to respect the manifold structure [12]. To sum up, the proposed method can not only effectively model the intrinsic Riemannian structure of the data space with the

Manuscript received October 22, 2018.
Manuscript revised May 18, 2019.
Manuscript publicized July 10, 2019.

[†]The authors are with College of Intelligence and Computing, Tianjin University, Tianjin, P. R. China.
[††]The author is with Zhejiang University of Water Resouces and Electric Power, Hangzhou, Zhejiang, 310018, P. R. China.
[†††]The author is with Tsinghua University, Beijing, P. R. China.
[††††]The author is with School of Information Science, Japan Advanced Institute of Science and Technology, Nomi-shi, 923–1292 Japan.
a) E-mail: jindi@tju.edu.cn (Corresponding author)
DOI: 10.1587/transinf.2018EDP7356

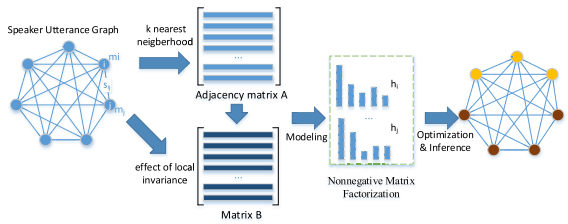


**Fig. 1** Idea of our community detection framework

idea of local invariance, but also be very efficient, or say scalable, because it just works on highly sparse networks.

The most relevant previous works are the method proposed by Shum, Campbell and Reynolds [13], NMF-based method proposed by Nishida et al [14], and spectral clustering (RatioCut method [15], Ng-Jordan-Weiss (NJW) [16]). Although the Shum's method and our method presented here seemed to be similar, they have some key differences. To be specific, the Shum's method employed the ($k$) nearest neighbor network of speech signals to model the manifold structure of the data space, and then directly used the existing community detection methods to detect speakers. But they also noted that, the difference of community detection performance on the weighted and unweighted nearest neighbor networks is negligible. This is in fact reasonable because community detection mainly focuses on unweighted networks. Even though several methods can deal with the weighted networks, they often do not work well in this situation. On the other hand, the weights on the nearest neighbor network respect the local invariance of speech signals, which is essential to model the manifold structure of the data space. Though, the constraint of the membership distributions being forced to be similar if the nodes are close in the local space is introduced in spectral clustering method (e.g. RatioCut), it did not discovers the intrinsic geometrical structure of the data space and ignores the soft assignment to communities, thus fails to use such kind of features. As a result, the traditional community detection methods like spectral clustering and NMF are not enough for speaker clustering, and hence a new type of methods which is specialized suitable for this complicated problem is needed. For the problem, we give a novel 'two-step' idea, shown as Fig. 1. We first propose a probabilistic model on the unweighted nearest neighbor network for the detection of communities. We then refine the model with an intuitive idea that: if two connected nodes have a high similarity, their community membership distributions in the model should be made close; and vice versa. This not only utilizes the advantage of community detection, but also further enhances the local invariance assumption of this problem, and thus better respects the structure of the underlying manifold. This is also partly validated in the experiments. The rest of the paper is organized as follows: Section 2 presents the details of our method; Section 3 gives the experiments and results; the paper is concluded in Sect. 4.

## 2. Methods

We first introduce the method to evaluate the similarities of speech signals and, based on them, we construct the local ($k$)nearest neighbor network to model the manifold structure of the data space. We then propose a community detection model for detecting each group of speech signals corresponding to the same speaker, and we further refine it by incorporating the local invariance of these speech signals. At last, we give a NMF method to learn the parameters of the model.

### 2.1 Speaker Content Networks

The construction of a speaker content graph here assumes that each node $i$ in the graph corresponds to an utterance or speech sentence represented by an identity vector $m_i$ (i-vector). The i-vector approach is an extension to the universal background model-Gaussian mixture model (UBM-GMM) approach. The i-vector space is referred to as the total-variance space (speaker and session variances), and a speech segment can be represented by an identity vector in this space. Then, we define the similarity matrix $S = (S_{ij})_{n\times n}$ of speech signals as:

$$S_{ij} = e^{d(m_i,m_j)} \tag{1}$$

in which the $d(\cdot,\cdot)$ corresponds to the cosine distance between two utterance, where $m_i$ and $m_j$ denote the i-vectors extracted from the $i$-th and $j$-th utterances.

Recent studies in spectral graph theory [17] and manifold learning theory [18] have demonstrated that the local geometric structure can be effectively modeled through the ($k$) nearest neighbor network on a scatter of data points. Consider a network $N$ with $n$ vertices where each vertex corresponds to an i-vector of speech signal. For each speech signal $i$, we find its $k$ nearest neighbors and put edges between $i$ and its neighbors. Then we have the adjacency matrix $A = (A_{ij})_{n\times n}$ of the network $N$ as:

$$A_{ij} = \begin{cases} 1, & \text{if nodes } i \text{ and } j \text{are connected} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Note that this construction implies the minimum degree of each node is $k$, but because the edge construction is done separately at each node, the degree of any particular node could be substantially larger than $k$.

### 2.2 Community Detection Models

We employ $c$ soft communities to describe the network $N$ with adjacency matrix $A$. The model is parametrized by a set of variables $H_{iz}$'s, in which $H_{iz}$ denotes the propensity of node $i$ belonging to the $z$-th community. We then employ $H$ to generate the expected adjacency matrix $\hat{A}$ of the network. Specifically, $H_{iz}H_{jz}$ is employed to present the expected number of links between nodes $i$ and $j$ in the $z$-th

community. Summing over the communities, the expected number of links between nodes $i$ and $j$ in the whole network will be:

$$\hat{A}_{ij} = \sum_z H_{iz} H_{jz} \tag{3}$$

Using squared loss to measure the relaxation error, the model defined in (3) can be fitted and learned by minimizing the following optimization function:

$$O_1(H) = \left\| A - HH^T \right\|_F^2 \tag{4}$$

where $\|.\|_F$ is the Frobenius norm which denotes the likelihood of Gaussian distribution, and $H$ is a nonnegative matrix. Furthermore, in order to incorporate the similarities of speech signals, an intuitive idea is that: if two speech signals $i$ and $j$ have a high similarity $S_{ij}$ in the local Euclidean space, their community membership distributions $H_i$ and $H_j$ should be made close; otherwise, they will be made not close. We use the following term to denote the effect of the local invariance of the similarity matrix $S$:

$$\begin{aligned} R(H) &= \frac{1}{2} \sum_{ij} \left\| H_i - H_j \right\|^2 B_{ij} \\ &= \sum_i H_i^T H_i D_{ii} - \sum_{ij} H_i^T H_j B_{ij} \\ &= Tr(H^T DH) - Tr(H^T BH) = Tr(H^T LH) \end{aligned} \tag{5}$$

in which $B_{ij} = A_{ij} S_{ij}$, $Tr(\cdot)$ denotes the trace of a matrix, and $D$ is a diagonal matrix whose entries are column sums of $B$, $D_{ii} = \sum_j B_{ij}$. $L = D - B$, which is called graph Laplacian [11]. By minimizing $R$, we expect that if two connected nodes $i$ and $j$ are similar (i.e. $A_{ij} = 1$ and $S_{ij}$ is large), their community membership distributions $H_i$ and $H_j$ will be close to each other; and vice versa.

To sum up, by incorporating network topology modeled by (4) and the local similarities of speech signals modeled by (5), the mixed model can be formulated and learned by minimizing the following optimization function:

$$\begin{aligned} O(H) &= O_1(H) + \lambda R(H) \\ &= \left\| A - HH^T \right\|_F^2 + \lambda Tr(H^T LH) \end{aligned} \tag{6}$$

in which the parameter $\lambda$ balances the effect network topology and nodes' local similarities.

### 2.3 Parameters Optimization

According to (6), the optimization of the parameters of our model will be the following minimization problem:

$$H = argmin_{H \geq 0} O(H) \tag{7}$$

This can be also taken as a nonnegative matrix factorization (NMF) problem. In order to infer the multiplicative update rule, we employ a gradient descent approach [19]. First, the gradient of (7) with respect to the parameter matrix $H$ can be calculated as:

$$\frac{\partial O}{\partial H} = 4HH^T H + 2\lambda DH - 4AH - 2\lambda BH \tag{8}$$

This gradient can be decomposed into some positive components as well as some negative components which are presented as:

$$\begin{aligned} &\frac{\partial O}{\partial H} = [\cdot]_+ - [\cdot]_- \\ &[\cdot]_+ = 4HH^T H + 2\lambda DH \\ &[\cdot]_- = 4AH + 2\lambda BH \end{aligned} \tag{9}$$

Then, by using $[\cdot]_+$ and $[\cdot]_-$we can define an update rule based on iterative learning:

$$H_{ij} = H_{ij} - \eta_{ij} \frac{\partial O}{\partial H} = H_{ij} - \eta_{ij} ([\cdot]_+ - [\cdot]_-)_{ij} \tag{10}$$

in which $\eta_{ij}$ denotes a positive learning rate. Thereafter, according to the results in [14], we set $\eta_{ij} = \frac{H_{ij}}{([\cdot]_+)_{ij}}$, and then make the above update rule become a multiplicative update rule:

$$\begin{aligned} H_{ij} &= H_{ij} - \frac{H_{ij}}{([\cdot]_+)_{ij}} ([\cdot]_+ - [\cdot]_-)_{ij} = H_{ij} \frac{([\cdot]_-)_{ij}}{([\cdot]_+)_{ij}} \\ &= H_{ij} \frac{(2AH + \lambda BH)_{ij}}{(2HH^T H + \lambda DH)_{ij}} \end{aligned} \tag{11}$$

According to the analysis in [20], once the parameter matrix $H$ is initialized to be nonnegative, the derived multiplicative update rule will keep its nonnegativity. When $([\cdot]_+)_{ij} = ([\cdot]_-)_{ij}$, the update rule will converge, which means that $\frac{\partial O}{\partial H} = 0$ is the stationary point of the function in (6).

By iteratively updating the multiplicative update rule defined in (11), we can obtain the optimal (or local optimal) community memberships $H$. But in fact, $H_{iz}$ presents a *s*oft community membership, which is often used to infer the *d*eterministic community membership. Generally speaking, one can simply assign each node $i$ to community $r$ satisfying $r = argmax_z \{H_{iz} | z = 1, 2, \cdots, c\}$. Also note that we set the number of communities as the ground-truth of the number of speakers in experiments.

### 2.4 Complexity Analysis

We analyze the time complexity of our community detection models. The main complexity comes from the calculation of $AH$, $BH$, $H(H^T H)$ and $DH$ in (11) are $2mc$, $2mc$, $2nc^2$ *and* $nc$, respectively, where $n$ is the number of nodes, $m$ the number of links, and $c$ the number of communities ($c << m$ *or* $n$). Thus, the time of evaluating (11) once is $O(mc + nc^2)$. Therefore, the calculational complexity of our method is $O(T(mc + nc^2))$, where $T$ is the number of iterations for convergence which is often considered as a constant.

## 3. Experiments

### 3.1 Databases

In this paper, we have two parts of databases. One part is for

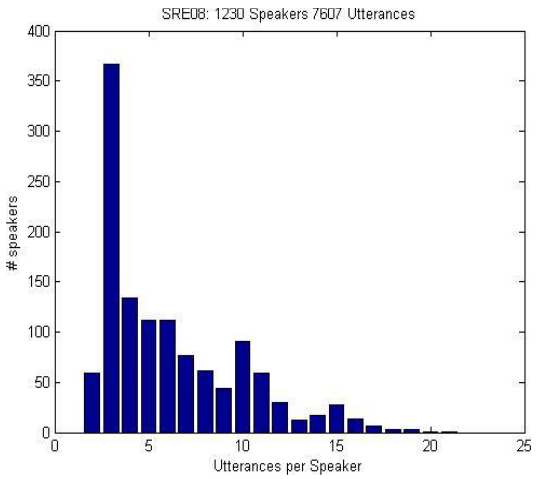


**Fig. 2** Distribution of the number (>= 2) of utterances per speaker on the test dataset of NIST SRE08.

the UBM training stage which has two databases. One UBM training dataset is CSLT-Chronos [21], in which the speech signals are digitalized at 8kHz sampling rates simultaneously in 16-bit precision. It consists of 60 speakers. Each speaker in this databases read 100 Chinese sentences and 10 isolated Chinese. All data is 124 MB. The other UBM training dataset is SRE 03+04 and Fisher. SRE 03+04 consists of over1 20 hours of English conversa-tional telephone speech. Fisher consists of over 11,699 recorded telephone conversations (speakers). To show the efficiency of the method, the test dataset we used has two different scales of databases. The first one, called as CH50, is bought from company SpeechOcean, which consists of 500 utterances recorded by 50 native Chinese speakers (25 females and 25 males respectively) from mainly Beijing and Heibei province. Each speaker read 20 different texts extracted from newspaper, and the duration of each utterance is about 5 to 10 seconds. They are required to speak Mandarin Chinese. The length of each sentence ranges from 8 to 30 Chinese characters with an average of 14. The second one, called SRE08, is the large-scale database, NIST 2008 Speaker Recognition Evaluations, which consists of 7,607 utterances recorded by 1,230 speakers. And the duration of each utterance is about 3 to 5 minutes. This database includes both genders; more detailed information regarding the distribution of utterances per speaker is shown in Fig. 2.

### 3.2 Preprocess

The first step before our experiments is to construct the speaker content graph using i-vectors. The i-vector system (including parameters of the UBM and T matrix) for Chinese data was trained with 30 female and 30 male utterances (about 4 hours in total) from the database CSLT-Chronos. Another system was trained with SRE03+04 and fisher databases. The UBM involves 512 Gaussian components. And the dimension of i-vectors is 200. The basic acoustic feature involved 13-dimensional Mel-Frequency Cepstrum Coefficients (MFCCs), in which log energy is included rather than C0. These basic features were augmented by their first and second order derivatives, resulting in 39-dimensional feature vectors.

### 3.3 Measurements

To assess the quality of the results, we adopt a widely-used accuracy metric for data clustering and community detection, named normalized mutual information (NMI) [22], which is based on the information theory. This measure is formally described by the following formula:

$$NMI(A,B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} N_{ij}\log(\frac{N_{ij}N}{N_{i.}N_{.j}})}{\sum_{i=1}^{C_A} N_{i.}\log(\frac{N_{i.}}{N}) + \sum_{j=1}^{C_B} N_{.j}\log(\frac{N_{.j}}{N})} \quad (12)$$

where $C_A$ is the number of real communities and $C_B$ is the number of found or detected communities. In the above equation, $N$ is the confusion matrix where the rows correspond to the real community (ground truth) and columns correspond to the found communities. The element $N_{ij}$ is the number of vertices in the real community $i$ that appear in the detected community $j$. The sum over row $i$ of the matrix $N_{ij}$ is denoted $N_{i.}$ and the sum over column $j$ of the matrix $N_{ij}$ is denoted $N_{.j}$. The value of NMI ranges from 0 to 1 and the higher the value, the better the community structure.

Another measurement is *Pout* which is used to indicate the complexity of the network. It is defined as the ratio $Pout = Zout/(Zin + Zout)$, where *Zout* is the number of nodes in the constructed speaker content graph belonging to different speakers and *Zin* is the number of nodes within the same speaker. The larger the *Pout* value is, the harder the community structure of the network is to be found.

### 3.4 Baseline Methods

We compare our method with two type of methods. The first is to use the network construction method proposed in this work (i.e. constructing the speaker content graph based on i-vetors and the local ($k$) nearest neighbor relationship), and then compare our new NMF approach with five existing community detection methods in this case.

- CNM method [23]. CNM is a classical community detection algorithm, which is to optimize the well-known modularity function in a fast and greedy way. Modularity function $Q$ estimates the goodness of a partition of the network based on a comparison between the graph at hand and a random null model, defined as:

$$Q = \frac{1}{2m}\sum_{ij}[A_{ij} - \frac{k_i k_j}{2m}]\delta(c_i, c_j) \quad (13)$$

  where the $\delta(i, j)$ equals to 1 if $i = j$ and 0 otherwise, $k_i$ is degree of the node $i$, $m$ is the number of edges in the graph, and $c_i$ denotes the community to which node $i$ belongs.

- Infomap method [24]. Infomap uses the probability flow of random walks on a network as a proxy for information flows in the real system and decompose the network into modules by compressing a description of the probability flow. It is to arrive at a two-level description that exploits both the network's structure and the fact that a random walker is statistically likely to spend long periods of time within certain clusters of nodes. Infomap is to find a group division $M$ with $m$ groups, by minimizing the average number of bits per step:

$$L(M) = q_{\curvearrowright} H(Q) + \sum_{i=1}^{m} p^i_{\circlearrowright} H(P^i) \tag{14}$$

  where the first term describes movement between modules, while the second describes movement within modules.
- Markov Clustering (MCL) method [25]. The basic idea of MCL is to find out where the flows is gathered by random walks, and thus discover the clustering. To be specific, this algorithm converts a graph affinity matrix to a stochastic matrix by dividing the elements of each row by their sum and then iterates between two steps, which are the expansion and inflation steps. The expansion step is mainly based on the expansion parameter $e$ to perform exponentiation operator on the transition probability matrix $M$. The formalized formula is:

$$M_{exp} = Expand(M, e) = M^e \tag{15}$$

  The inflation step is mainly based on the inflation parameter $r$ to performs exponentiation operator on each column of the transfer matrix, and then performs a normalization operation on each column. The formalized formula is:

$$M_{inf} = Inflate(M, r) = \frac{M(i,j)^r}{\sum_{k=1}^{n} M(k,j)^r} \tag{16}$$

  MCL iterates between these two parts until convergence.
- RatioCut method [15]. RatioCut is a spectral clustering algorithm. It aims to find a partition of the graph that the edges between different groups have a very low weight, while effectively preventing isolated points from appearing. Specifically, given the number of subsets $k$ of subsets, RatioCut consists in choosing a partition $A_1, \cdots, A_k$ by minimizing:

$$RatioCut(A_1, A_2, \ldots, A_k) = \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \overline{A}_j)}{|A_i|}) \tag{17}$$

  based on spectral optimization.
- Ng-Jordan-Weiss (NJW) method [26]. NJW is also a classic spectral clustering algorithm. It uses the eigenvector corresponding to the top-K largest eigenvalues of the Laplacian matrix $L$ as the representation of the data, and then uses K-means for clustering. The Laplacian matrix $L$ of NJW is defined as:

$$L = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} \tag{18}$$

- Standard NMF method [27]. It is similar to the formulation we introduced in Eq. (6), while it works on the dense full similarity matrix $S$ rather than the sparse graph $A$. The objective function is defined as:

$$min_{M,H\geq 0} = \left\| S - MH^T \right\| \tag{19}$$

  in which $M \in R^{n\times c}$ corresponds to nonnegative basis matrix and $H$ the coefficient matrx.

The second type of methods compared is the Shum's method [13]. It works on both the weighted and unweighted nearest neighbor networks, forming two methods which are the weighted and unweighted versions. In addition, to make Shum's methods more comparable with our approach, we used their main idea of community detection, while replaced their original suggested community detection methods (e.g. spectral clustering, Markov clustering, as summarized in [28]) by the standard NMF method. In short, Shum's (weighted/unweighted) methods applies the standard NMF method on sparse matrix ($W/A$), where $A$ is the adjacency matrix defined in Eq. (2), and $W$ is defined as follows:

$$W_{ij} = \begin{cases} e^{-d^2(m_i, m_j)/\sigma}, & \text{if nodes } i \text{ and } j \text{ are connected} \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

where $\sigma$ controls the decay of the exponential function, and $d(\cdot,\cdot)$ corresponds to Euclidean distance between two speaker GMM supervectors.

For all these methods compared, we use implementations provided or agreed by the authors. We set the parameter $k$ the same with the proposed method so that the speaker content graph built from the database is the same. We used default values provided by the authors for other parameters in their own algorithms.

### 3.5 Results and Discussion

We first conducted the experiments on the Chinese database CH50. To investigate the effectiveness of our method in different scales of the complicated networks, we first use 100 utterances (50 speakers and 2 utterances spoken by each speaker) to conduct the experiment, then gradually added the utterances numbers by each speaker to 4, 6, 8, 10, 12, 14, 16, 18, 20.

Table 1 presents the clustering results of the nine methods. As we can see from this table, in general our method gives the best accuracy performance on ten different scales of databases. Shum's weighted and unweighted method as is described in the original paper, gives almost the same performance. And their results are only a little worse than the

**Table 1** NMI (%) average results of 10 runs of nine methods on five scales of complicated networks ('NUM' denotes the number of the utterances by each speaker)

| NMI (%) | CNM | INFOMA | MCL | RATIOCUT | Spectral Clustering | STANDARD NMF | SHUM'S (WEIGHTED) | SHUM'S (UNWIGHTED) | PROPOSED METHOD |
|---|---|---|---|---|---|---|---|---|---|
| num = 20 | 72.33 | 97.92 | 72.21 | 94.63 | 97.34 | 78.49 | 96.60 | 97.45 | **98.62** |
| num = 18 | 73.04 | 98.05 | 71.02 | 93.63 | 97.38 | 79.72 | 95.76 | 96.84 | **98.54** |
| num = 16 | 97.57 | 96.95 | 71.77 | 94.18 | 97.37 | 79.04 | 95.54 | 97.09 | **98.42** |
| num = 14 | 74.78 | 95.42 | 72.02 | 94.58 | 97.57 | 77.97 | 97.49 | 97.27 | **98.65** |
| num = 12 | 89.56 | 96.79 | 71.44 | 93.84 | 96.80 | 76.78 | 95.65 | 97.60 | **98.82** |
| num = 10 | 77.23 | 96.28 | 71.66 | 94.59 | 96.31 | 78.76 | 96.55 | 97.53 | **98.49** |
| num = 8 | 78.96 | 96.13 | 69.47 | 94.47 | 93.67 | 77.73 | 95.60 | 96.69 | **97.61** |
| num = 6 | 75.49 | 92.79 | 67.85 | 94.20 | 90.03 | 74.27 | 94.79 | 94.71 | **96.83** |
| num = 4 | 83.06 | 95.43 | 64.64 | 94.93 | 83.60 | 77.38 | 94.14 | 96.19 | **96.24** |
| num = 2 | 90.78 | 91.48 | 60.19 | 94.03 | 83.87 | 81.68 | **96.08** | 94.66 | 95.78 |

**Table 2** NMI (%) average results of 10 runs of nine methods on ten random scales of complicated networks

| NMI (%) | CNM | INFOMA | MCL | RATIOCUT | Spectral Clustering | STANDARD NMF | SHUM'S (WEIGHTED) | SHUM'S (UNWIGHTED) | PROPOSED METHOD |
|---|---|---|---|---|---|---|---|---|---|
| RANDOM#0 | 75.18 | 89.65 | 69.7 | 89.23 | 92.61 | 75.55 | 94.31 | 92.46 | **95.65** |
| RANDOM#1 | 81.62 | 91.00 | 69.27 | 89.54 | 92.69 | 75.99 | 93.00 | 93.13 | **95.36** |
| RANDOM#2 | 75.18 | 87.62 | 69.96 | 89.72 | 92.08 | 72.86 | 92.85 | 91.99 | **95.28** |
| RANDOM#3 | 93.00 | 94.13 | 71.19 | 89.53 | 92.39 | 83.30 | 93.3 | 94.12 | **96.48** |
| RANDOM#4 | 84.53 | 95.07 | 71.03 | 91.92 | 93.77 | 78.23 | 95.48 | 94.62 | **96.05** |
| RANDOM#5 | 78.62 | 88.12 | 69.4 | 86.66 | 90.29 | 81.07 | 92.63 | 91.87 | **94.62** |
| RANDOM#6 | 85.67 | 89.34 | 71.82 | 89.39 | 91.80 | 83.34 | 90.80 | 90.80 | **94.48** |
| RANDOM#7 | 92.84 | 93.72 | 72.24 | 87.96 | 91.94 | 79.99 | 93.09 | 93.05 | **94.38** |
| RANDOM#8 | 95.53 | 93.83 | 71.21 | 88.71 | 92.21 | 77.23 | 91.85 | 93.65 | **95.77** |
| RANDOM#9 | 76.11 | 91.88 | 72.42 | 89.35 | 91.58 | 76.74 | 92.79 | 91.20 | **95.92** |

**Table 3** NMI (%) average results of 10 runs of nine methods on large scale complicated network of sre08

| NMI (%) | CNM | INFOMA | MCL | RATIOCUT | Spectral Clustering | STANDARD NMF | SHUM'S (WEIGHTED) | SHUM'S (UNWIGHTED) | PROPOSED METHOD |
|---|---|---|---|---|---|---|---|---|---|
| | 77.05 | 80.07 | 69.3 | 74.14 | 83.5 | 70.08 | 82.96 | 83.79 | **84.16** |

proposed method. We also can see that for the proposed method and MCL method, the accuracy result becomes better along with the network's complexity from *num* = 2 (2 utterances by each speaker) to *num* = 10 (10 utterances by each speaker) and then no big change happens if the average nodes are bigger than ten. Specifically, the accuracy of our method is 16.52%, 2.07%, 28.57%, 3.49%, 4.41%, 19.61%, 1.98%, 1.19% on average better than the baseline method CNM, Infomap, MCL, RatioCut, Spectral Clustering, NMF, Shum's weighted, Shum's unweighted community detection. These results further confirm the effectiveness of our new model and method.

Actually, the *num* is not a parameter of our proposed method, so its value can be fixed randomly. To check if the balanced structure of the data network affects our method's results, we randomly chose two to twenty utterances for each speaker in CH50. We chose ten random times. The result is as Table 2 shows.

Still our method gives best results. Specifically, the accuracy of our method is 11.57%, 3.96%, 24.58%, 6.20%, 3.26%, 16.97%, 2.38%, 2.71% on average better than the baseline method CNM, Infomap, MCL, RatioCut, Spectral Clustering, NMF, Shum's weighted, Shum's unweighted community detection. Compared with the first row in Table 1, the results of Table 2 indicate that our method performs better in the case of non-balanced dataset than that in the balanced dataset.

In order to verify the effectiveness of our method in large scale speaker detection task, SRE08 was employed. Different from the CH50, it is standard speaker recognition evaluation database, in which the utterances contain all kinds of noises. The results are as shown in Table 3.

From Table 3, we see that our method performs a little better than Shum's weighted and unweighted method and much better than other four methods. Specifically, the accuracy of our method is 7.11%, 4.09%, 14.86%, 10.02%, 0.66%, 14.08%, 1.2%, 0.37% better than the baseline method CNM, Infomap, MCL, RatioCut, Spectral Clustering, NMF, Shum's weighted, Shum's unweighted community detection. The results indicate that our method performs efficiently and effectively in the large-scale speaker network.

In addition, we also observe the running time of our algorithm for 10 times. The average computational time of our method on database of the CH50 and SRE08 is approximately 14 seconds and 40 minutes, respectively.

### 3.6 Parameter Selection

We run our algorithm described above on the full testing

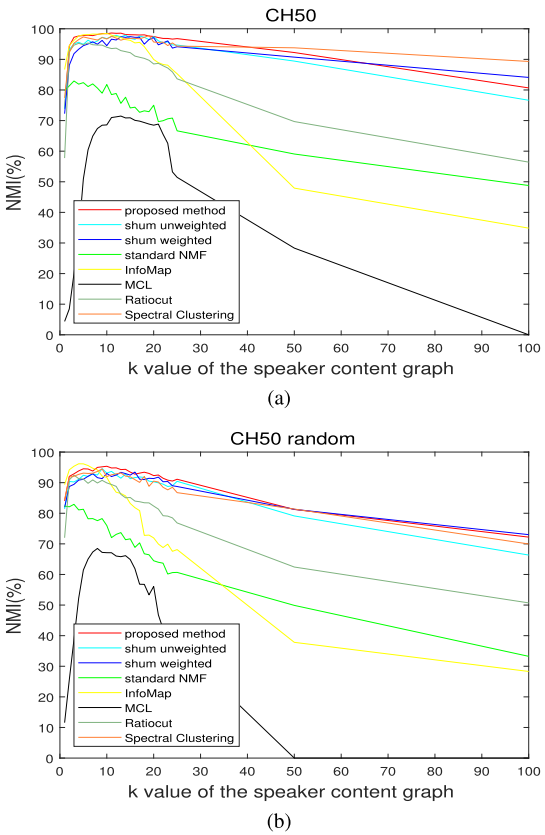


**Fig. 3** The NMI result of our method with the different condition of $k$ (a) on the full database (20 utterances of each speaker) and (b) on a random test database (random times of utterances of each speaker). $k$ is the number of neighbors of each utterance when constructing the speaker content graph. The average utterance number of one speaker in (b) is 9.5.

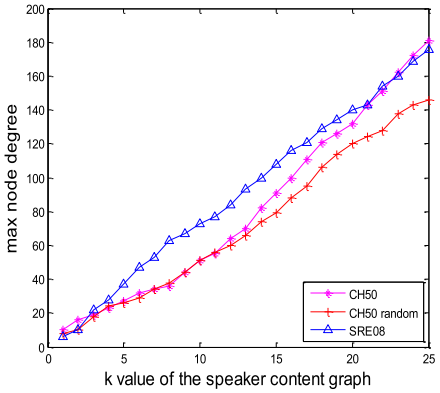


**Fig. 4** The max node degree changes with the different $k$ value of the speaker contend graph. In the three datasets, they all show the similar linear relationship.

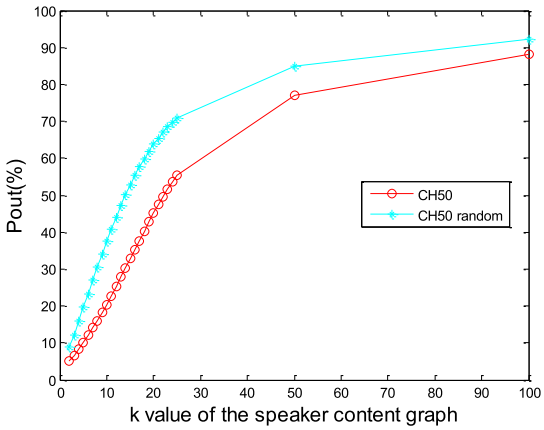


**Fig. 5** The Pout performance of our method on the 500 utterances with the different condition of $k$. The $k$ is the number of neighbors of each utterance when constructing the speaker content graph. Pout denotes the complexity of the network. The larger the Pout value is, the harder the community structure of the network is to be detected.

database (50 speakers and each speaker speaks 20 utterances) and on the first random testing database (50 speakers and each speaker speaks random number utterances from 2 to 20) following the different setting for edge degree of $k$ parameter. The results are summarized in Fig. 3.

From these two figures, we can find out a systematic dependency between clustering performance and graph edge density. The best performance for our method is achieved where $k$ is around half or full of the graph edge density, which means the nodes between the same speaker are fully connected; even the network is non-balanced, in which people speak different number of utterances. Also this dependency rule is true for other methods, except the standard NMF method. The figure also indicate that out method have a relative better robust to the graph edge density than the Infomap method and MCL method, although the bigger edge density really impact the algorithm running time.

As Fig. 4 shows, the bigger the graph edge density is, the larger value of the nodes degree, which means the low sparsity of the whole network. From Fig. 5, it is obvious that the network is becoming complicated with the increase of $k$-value as *Pout* is getting larger. However, from these figures, we observe that the NMI performance of our method is not become low all the time with the increase of Pout. That may because our performance is not only influenced by the complexity of the network, but also by the information of speakers. Generally, it makes sense that our best result is obtained on the 10-NN graphs, since the 10 nearest neighbors of an utterance spoken by a speaker should ideally be the rest of the utterances spoken by that same speaker on average. Finally we chose $k = 10$ as the general default value for our method even the network is large scale and more complicated such as SRE08. Of course, if one have the assumption of the network, it is better to select half of the average node number of the speakers. To validate this assumption, we run our method on database of Ch50 when $num = 5, 10, 20$ (50 speakers and each speaker speaks 5, 10, 20 utterances). As shown in Fig. 6, when $k$ is half of the number of utterances spoken by the same speaker, our result is the best in general.

After $k$ is fixed as ten, then we try to get a proper $\lambda$ parameter. We illustrate the variation of the algorithm with $\lambda$ in Fig. 7. At first when $\lambda$ increases, more content information is combined together with link structure and it performs better. Then, such advantages drop off, as too much content

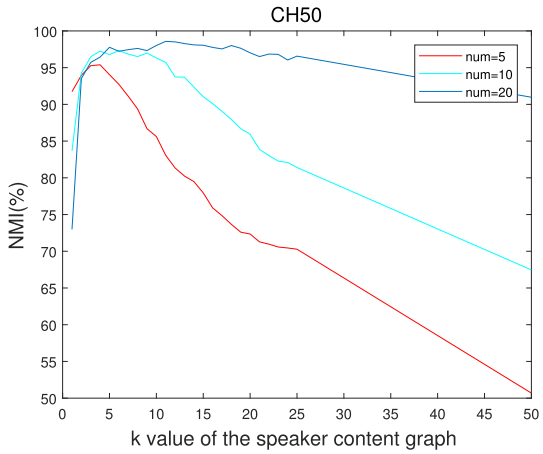


**Fig. 6** The proposed algorithm performance with variation of parameter $k$ when num is 5, 10 and 20 on CH50 database.

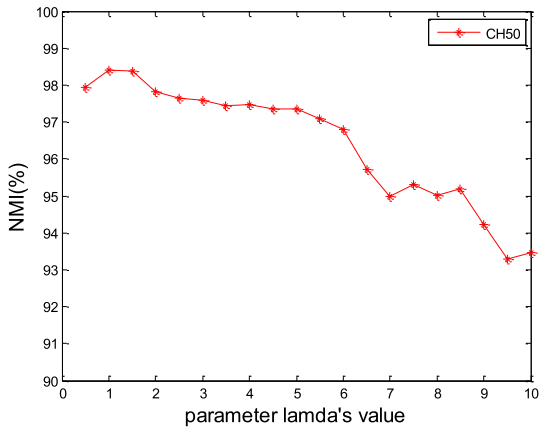


**Fig. 7** The proposed algorithm performance with variation of parameter $\lambda$ when $k$ is 10 on the whole CH50 databases.

information may reduce the impact of the link structure. We notice that the peak appear on $\lambda = 1$. And we used this value in the experiments.

## 4. Conclusions

In this paper, we propose a community detection model to cluster speakers on the speaker content graph, which is constructed based on the i-vector represented utterances. The results of the experiments on two Chinese and one English database show that, the accuracy of our method is much larger than that of the baseline methods of standard NMF, CNM, Infomap, MCL and RatioCut; the accuracy of our method is also larger than the spectral clustering (NJW) and Shum's methods. The results also indicate that our method performs better in the large-scale speaker networks. We also see that all the methods' performances become worse with the increase of the network's complexity. And when we try the algorithm on the condition of the random number utterances of each speaker, the results show that our method still performs better in the case of non-balanced dataset than that in the balanced dataset. However, one problem of our method on this stage is the lack of flexibilities for the data changing or increasing every day. Next we hope to consider to improve our method to fit the real condition on that speech data is increasing or dynamically changing.

## Acknowledgements

The research is partly supported by the National Key R&D Program of China (2018YFC0809800), the Natural Science Foundation of China (61772361), the Natural Science Foundation of Zhejiang Province under Grant Y19E050078. We would like to thank Prof. Patrick Kenny from CRIM and Dr. Stephen H. Shum for providing the useful advices on the speaker content graph construction for this work.

### References

[1] S.E. Tranter and D.A. Reynolds, "An overview of automatic speaker diarization systems," IEEE Trans. Audio, Speech, Language Process., vol.14, no.5, pp.1557–1565, Sept. 2006.
[2] D. Reynolds, P. Kenny, and F. Castaldo, "A Study of New Approaches to Speaker Diarization," Interspeech, 2009.
[3] M. Huijbregts and D.A. van Leeuwen, "Large-Scale Speaker Diarization for Long Recordings and Small Collections," IEEE Trans. Audio, Speech, Language Process., vol.20, no.2, pp.404–413, 2012.
[4] Y. Hu, D. Wu, and A. Nucci, "Fuzzy-Clustering-Based Decision Tree Approach for Large Population Speaker Identification," IEEE Trans. Audio, Speech, Language Process., vol.21, no.4, pp.762–774, 2013.
[5] M. Mehrabani and J.H.L. Hansen, "Singing speaker clustering based on subspace learning in the GMM mean supervector space," Speech Communication, vol.55, no.5, pp.653–666, 2013.
[6] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," IEEE Trans. Audio, Speech, Language Process., vol.15, no.4, pp.1435–1447, 2007.
[7] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," IEEE Trans. Audio, Speech, Language Process., vol.19, no.4, pp.788–798, 2011.
[8] G. Friedland, A. Janin, D. Imseng, X.A. Miro, L. Gottlieb, M. Huijbregts, M.T. Knox, and O. Vinyals, "The ICSI RT-09 speaker diarization system," IEEE Trans. Audio, Speech, Language Process., vol.20, no.2, pp.371–381, 2012.
[9] K.J. Han and S.S. Narayanan, "A robust stopping criterion for agglom-erative hierarchical clustering in a speaker diarization system," AQ 8th Annual Conference of the International Speech Communica-tion Association, 2007.
[10] G. Sell, A. McCree, and D. Garcia-Romero, "Priors for Speaker Counting and Diarization with AHC," INTERSPEECH, pp.2194–2198, 2016.
[11] M. Girvan and M.E.J. Newman, "Community structure in social and biological networks," Proceedings of National Academy of Science, vol.99, no.12, pp.7821–7826, 2002.
[12] D. Cai, X. He, J. Han, and T.S. Huang, "Graph Regularized Nonnegative Matrix Factorization for Data Representation," IEEE Trans. Pattern Anal. Mach. Intell., vol.33, no.8, pp.1548–1560, 2011.
[13] S.H. Shum, W.M. Campbell, and D.A. Reynolds, "Large-scale community detection on speaker content graphs," in Acoustics, Speech and Signal Processing (ICASSP), Vancouver, BC, pp.7716–7720, 2013.
[14] M. Nishida and S. Yamamoto, "Speaker Clustering Based on Non-Negative Matrix Factorization Using Gaussian Mix-ture Model in Complementary Subspace," CBMI 2017, pp.7:1–7:5, 2017.
[15] U.V. Luxburg, "A tutorial on spectral clustering." Statistics and

Computing, vol.17, no.4, pp.395–416, 2007.
[16] N. Tawara, T. Ogawa, and T. Kobayashi, "A comparative study of spectral clustering for i-vector-based speaker clustering under noisy condi-tions," 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, pp.2041–2045, 2015.
[17] F.R.K. Chung, "Spectral Graph Theory," Regional Conference Series in Mathematics, vol.92, 1997.
[18] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," Advances in Neural Information Processing Systems (NIPS) 14, pp.585–591, 2002.
[19] J. Kivinen and M.K. Warmuth, "Additive versus exponentiated gradient updates for linear prediction," Association for Computing Machinery, pp.209–218, 1995.
[20] E. Oja, "Principal components, minor components, and linear neural networks," Neural Networks, vol.5, no.6, pp.927–935, 1992.
[21] L. Wang and F. Zheng, "Creation of time-varying voiceprint database," Technical Session-6 (Oral), Oriental-COCOSDA, Kathmandu, Nepal, Nov. 24–25, 2010.
[22] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," Journal of Statistical Mechanics: Theory and Experiment, vol.2005, p.09008, 2005.
[23] A. Clauset, M.E.J. Newman, and C. Moore, "Finding community structure in very large networks," Phys. Rev. E, vol.70, no.6, 2004.
[24] M. Rosvall and C.T. Bergstrom, "Maps of random walks on complex networks reveal community structure," Proceedings of the National Academy of Sciences, vol.105, no.4, pp.1118–1123, 2008.
[25] S. Van Dongen, "Graph Clustering by Flow Simulation," Ph.D. thesis, University of Utrecht, vol.30, no.1, pp.121–141, 2008.
[26] A.Y. Ng, M.I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," Advances in Neural Infor-mation Processing Systems, 2002.
[27] D.D. Lee and H.S. Seung, "Learning the parts of objects by nonnegative matrix factorization," Nature, vol.401, no.6755, pp.788–791, 1999.
[28] A. Lancichinetti and S. Fortunato, "Community detection algorithms: a comparative analysis," Physical Review E, vol.80, no.5, 2009.
[29] N. Kawasaki, "Parametric study of thermal and chemical nonequilibrium nozzle flow," M.S. thesis, Dept. Electron. Eng., Osaka Univ., Osaka, Japan, 1993.

**Hongcui Wang** received the B.S. and M.S. degree in Computer Science, from Shandong University in 2003 and from Institute of the Computing Technology, Chinese Academic and Science, in 2006. She received Ph.D. degree in School of Informatics, Kyoto University, Kyoto, Japan, in 2009. She worked for Tianjin University from 2010. Since 2018, she has moved to Zhejiang University of Water Resouces and Electric Power. Her current research interests are speaker recognition and signal processing.

**Shanshan Liu** is pursuing her MS degree in College of Intelligence and Computing, Tianjin University, China. Her current research interests include social network analysis and machine learning.

**Di Jin** received his B.S., M.S., and Ph.D. degrees in computer science from Jilin University, Changchun, China, in 2005, 2008, and 2012, respectively. He is currently an Associate Professor with the College of Intelligence and Computing, Tianjin University, Tianjin, China. He has published more than 50 papers in international journals and conferences in the areas of community detection, social network analysis, and machine learning.

**Lantian Li** received the B.S. degree in China University of Mining and Technology, Beijing, China, in 2013. He received the Ph.D. degree in Tsinghua University, Beijing, China, in 2018. His research interests include speaker recognition, signal processing, and deep learning.

**Jianwu Dang** graduated from Tsinghua University, China, in 1982, and got his M.S. at the same university in 1984. He worked for Tianjin University as a lecture from 1984 to 1988. He was awarded the Ph.D. from Shizuoka University, Japan in 1992. Since 2001, he has moved to Japan Advanced Institute of Science and Technology (JAIST). His research interests are in all the fields of speech production, speech synthesis, and speech cognition.