Community detection with manifold learning on speaker i-vector space for Chinese

Hongcui WANG [1], *Di JIN* [1,*], *Lantian LI* [2], *Jianwu DANG* [1, 3]

[1] Tianjin Key Laboratory of Cognitive Computation & its Applications, Tianjin University, Tianjin, P. R. China

[2] Center for Speaker and Language Technologies (CSLT), Tsinghua University, Beijing, P. R. China

[3] School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa, Japan

hcwang@tju.edu.cn, jindi@tju.edu.cn, lilt@cslt.riit.tsinghua.edu.cn, jdang@jaist.ac.jp

10.21437/Interspeech.2015-100

Abstract

Speaker recognition with clustering speech signals of 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. 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 a community detection model on the 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 speaker content network built from a Chinese speaker recognition database. The results confirmed the effectiveness of this new method.

Index Terms: community detection, speaker clustering, speaker detection, manifold structure

1. Introduction

The problem of speaker clustering is an important speech analysis task. It is mainly considered on large scale data [1][2]. 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 [3], Joint Factor Analysis [4] and i-vectors [5] , and then employed agglomerative hierarchical clustering algorithms for the recognition. 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 on 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 [6]. 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 [7]. To sum up, the proposed method can not only effectively model the intrinsic Riemannian structure of the data space with the idea of local invariance, but also be very efficient because it just works on highly sparse networks.

The most relevant previous work is the method proposed by Shum, Campbell & Reynolds [8], which also used community detection for speaker recognition. 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. As a result, the traditional community detection methods are not enough for speaker recognition, 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. 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 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 Section 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 nonnegative matrix factorization (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.

Recent studies in spectral graph theory [9] and manifold learning theory [10] 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 a supervector 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 zth 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 zth 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, $D_{ii} = \Sigma_j B_{ij}$. $L = D–B$, which is called graph Laplacian [9]. 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:

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

in which the parameter λ 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:

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

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 [11]. 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:

$$\frac{\partial O}{\partial Y} = [\cdot]_+ - [\cdot]_-$$
$$[\cdot]_+ = 4HH^T H + 2\lambda DH \quad (9)$$
$$[\cdot]_- = 4AH + 2\lambda BH$$

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

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

in which η_{ij} denotes a positive learning rate. Thereafter, according to the results in [12], we set $\eta_{ij} = \frac{H_{ij}}{([\cdot]_+)_{ij}}$, and then make the above update rule become a multiplicative update rule:

$$H_{ij} = H_{ij} - \frac{H_{ij}}{([\cdot]_+)_{ij}}([\cdot]_+ - [\cdot]_-)_{ij} = H_{ij}\frac{([\cdot]_-)_{ij}}{([\cdot]_+)_{ij}}$$
$$= H_{ij}\frac{(2AH + \lambda BH)_{ij}}{\left(2HH^T H + \lambda DH\right)_{ij}} \quad (11)$$

According to the analysis in [12], 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 *soft* community membership, which is often used to infer the *deterministic* community membership. Generally speaking, one can simply assign each node i to community r satisfying $r = arg\max_z\{H_{iz} \mid z = 1,2,\ldots,c\}$.

Notice that, the time to calculate AH, BH, $H(H^TH)$ 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 use two databases. The UBM training dataset is CSLT-Chronos [13], in which the speech signals are digitalized at 8kHz kHz sampling rates simultaneously in 16-bit precision. Each speaker read 100 Chinese sentences and 10 isolated Chinese. All data is 124 MB. The test dataset we used is bought from company SpeechOcean, which consists of 500 sentences recorded by 50 native Chinese speakers (25 females and 25 males respectively) from mainly Beijing and Heibei province. Each speaker read ten different texts extracted from newspaper. They are required to speak Mandarin Chinese. The length of each sentence ranges from 8 to 30 Chinese characters with an average of 14.

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) was trained with 30 female and 30 male utterances (about 4 hours in total) from the database CSLT-Chronos. The UBM involves 512 Gaussian components. And the dimension of i-vectors is 200. The acoustic feature used is 13-dimensional MFCCs (Mel-Frequency Cepstrum Coefficients) together with 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) [14], which is based on the information theory. This measure is formally described by the following formula:

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

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

3.4. Results

We conducted the experiments on the test database. 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.

Three other highly related methods are selected and compared with the proposed model. The first is the standard NMF algorithm [15] for data clustering. This is similar with our formulation in Eq. (6), but it works on the full similarity matrix S rather than the sparse graph A. The second and third are the Shum's methods [8], which work on the weighted and unweighted nearest neighbor networks, respectively. But in

order to make the Shum's methods more comparable with our method, we used their main idea of community detection, but replaced their original suggested community detection methods (i.e., spectral clustering, Markov clustering, and Infomap) by the standard NMF method.

Table 1. *NMI (%) results of four methods on five scales of complicated networks ('num' denotes the number of the utterances by each speaker)*

NMI (%)	*Standard NMF*	*Shum's (weighted)*	*Shum's (unwighted)*	*Proposed method*
num=10	50.59	62.68	62.72	**63.73**
num=8	52.23	64.05	64.56	**65.18**
num=6	55.69	66.04	67.66	**68.53**
num=4	59.86	74.15	73.81	**74.51**
num=2	67.25	84.87	84.64	**85.87**

Table 1 presents the recognition results of the four methods. As we can see from this table, in general our method gives the best accuracy performance on five different scales of databases. And for every method, the accuracy result becomes worse along with the network's complexity from *num*=2 (2 utterances by each speaker) to *num*=10 (10 utterances by each speaker). Actually, the num is not a parameter of our proposed method, so its value can be fixed randomly. Specifically, the accuracy of our method is 25.19% on average better than the baseline NMF method, 1.77% on average better than the Shum's unweighted community detection method, and 1.25% on average better than Shum's weighted community detection method. These results further confirm the effectiveness of our new model and method.

3.5. Parameter analysis

We run our algorithm described above on the full testing database (50 speakers and each speaker speaks 10 utterances) following the different setting for edge degree of *k* parameter. The results are summarized in Figure 1 and Figure 2. From Figure 1, we can find out a systematic dependency between clustering performance and graph edge density. From Figure 2, it is obvious that the network is becoming complicated with the increase of *k*-value as *P*out is getting larger. However, from these two figures, we observe that the NMI performance of our method is not become low all the time with the increase of *P*out. 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.

5. Acknowledgements

The research is partly supported by the National Basic Research Program of China (No. 2013CB329301), the National Natural Science Foundation of China (No. 61303110 and No. 61303109), and the PhD Programs Foundation of Ministry of Education of China (No. 20130032120043 and No. 20120032120043). 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.

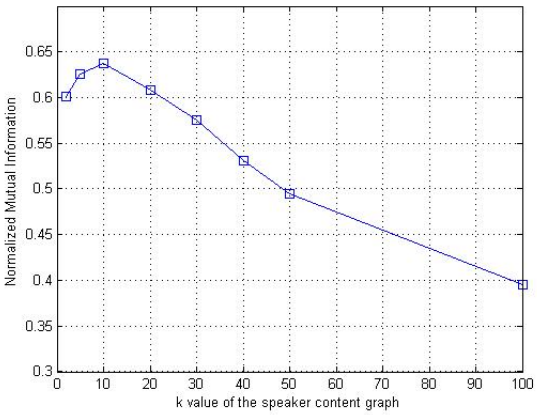

Figure 1: *The Normalized Mutual Information result of our method on the 500 utterances with the different condition of k. The k is the number of neighbors of each utterance when constructing the speaker content graph.*

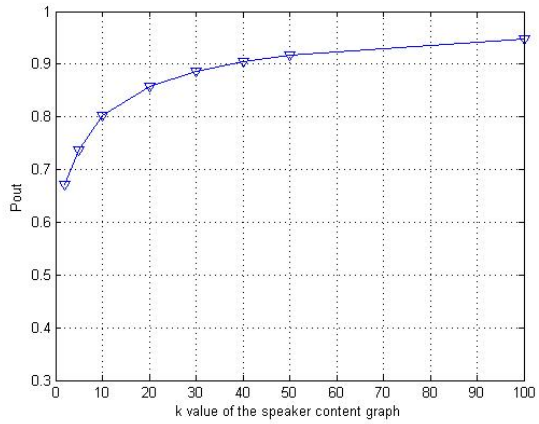

Figure 2: *The Pout performance of our method on the 500 utterances with the different condition of k. The k is the number of neighbors of each utterance when constructing the speaker content graph. Pout denotes the complexity of the network. The larger the Pout value is, the harder the community structure of the network is to be detected.*

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 a Chinese database show that our method accuracy is larger than other three methods. And all the methods' performances become worse with the network's complexity. However, we would not ignore the fact that the whole performance is not high enough. One reason we think may come from the data. All native speakers are from the northern provinces of China and do not have much accent when speaking mandarin. So it is hard to distinguish between different speakers. Next we hope to record some new data with speakers from different parts of China. And also plan to use SRE databases to verify the proposed method.

5. References

[1] M. Huijbregts and D. A. van Leeuwen, "Large-Scale Speaker Diarization for Long Recordings and Small Collections," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 20, pp. 404-413, 2012.

[2] Y. Hu, D. Wu and A. Nucci, "Fuzzy-Clustering-Based Decision Tree Approach for Large Population Speaker Identification," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 21, pp. 762-774, 2013.

[3] M. Mehrabani and J. H. L. Hansen, "Singing speaker clustering based on subspace learning in the GMM mean supervector space," *Speech Communication,* vol. 55, pp. 653 - 666, 2013.

[4] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1435–1447, 2007

[5] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[6] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of National Academy of Science*, vol. 9, no. 12, pp. 7821-7826, 2002.

[7] D. Cai, X. He, J. Han, and T. S. Huang, "Graph Regularized Nonnegative Matrix Factorization for Data Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 33, pp. 1548-1560, 2011.

[8] 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, 2013, pp. 7716-7720.

[9] F. R. K. Chung, "Spectral Graph Theory," *Regional Conference Series in Mathematics*, 1997.

[10] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in Neural Information Processing Systems (NIPS)* 14, 2002, pp. 585–591.

[11] J. Kivinen and M. K. Warmuth, "Additive versus exponentiated gradient updates for linear prediction," *Association for Computing Machinery*, pp. 209-218, 1995.

[12] E. Oja, "Principal components, minor components, and linear neural networks," *Neural Networks,* vol. 5, pp. 927-935, 1992.

[13] L. Wang and F. Zheng, "Creation of Time-Varying Voiceprint Database," *Technical Session-6(Oral), Oriental-COCOSDA,* Nov. 24-25, 2010, Kathmandu, Nepal.

[14] L. Danon, J. Duch, A. Diaz-Guilera and A. Arenas, "Comparing community structure identification," *Journal of Statistical Mechanics: Theory and Experiment, Vol.* 24, P09008, 2005.

[15] D. D. Lee and H. S. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, pp.788-791, 1999.