INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Universal Background Sparse Coding and Multilayer Bootstrap Network for Speaker Clustering

Xiao-Lei Zhang[1,2]

[1]Center of Intelligent Acoustics and Immersive Communications,
School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, China
[2]Department of Computer Science and Engineering, The Ohio State University, Columbus, OH, USA

xiaolei.zhang9@gmail.com

Abstract

We apply multilayer bootstrap network (MBN) to speaker clustering. The proposed method first extracts supervectors by a universal background model, then reduces the dimension of the high-dimensional supervectors by MBN, and finally conducts speaker clustering by clustering the low-dimensional data. We also propose an MBN-based universal background model, named universal background sparse coding. The comparison results demonstrate the effectiveness and robustness of the proposed method.

Index Terms: multilayer bootstrap network, speaker clustering, universal background sparse coding, unsupervised learning

1. Introduction

Speaker clustering aims to clustering speech segments that are belonged to the same speaker into a single cluster. It is important in many speech systems, such as speaker diarization, language recognition, and speech recognition.

Existing speaker clustering methods mainly include principle component analysis (PCA), k-means clustering, Gaussian mixture model (GMM), agglomerative hierarchical clustering, and joint factor analysis. For example, Wooters and Huijbregts [1] used agglomerative clustering to merge speaker segments by Bayesian information criterion. Iso [2] used vector quantization to encode speech segments and used spectral clustering, which is a k-means clustering applied to a low-dimensional subspace of data, for speaker clustering. Nwe *et al.* [3] used a group of GMM clusterings to improve the individual base GMM clusterings. Some methods apply clustering techniques, e.g. variational Bayesian expectation-maximization (EM) GMM [4] and spectral clustering [5], to i-vectors [6].

Because little prior knowledge of data is known beforehand, an unsupervised method should satisfy the following conditions: (i) no need for manually-labeled training data; (ii) no hyperparameter tunning for a satisfied performance; and (iii) robustness to different data or modeling conditions. Due to these strict requirements, speaker clustering is a very difficult task.

In this paper, we present a *multilayer bootstrap network* (MBN) [7] based algorithm, which contains two novel points. The first novel point is to generate high-dimensional supervectors of speech segments by *universal background sparse coding* (UBSC), a novel MBN-based universal background model. The second one is to reduce the dimensionality of the supervectors by MBN. Experimental results show that the proposed method satisfies these requirements.

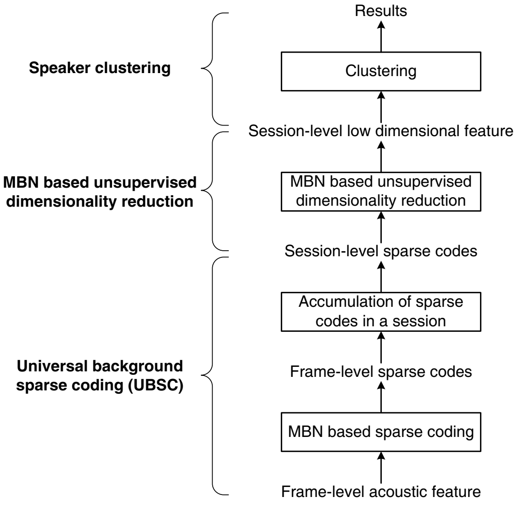
Results
Speaker clustering
Clustering
Session-level low dimensional feature
MBN based unsupervised dimensionality reduction
MBN based unsupervised dimensionality reduction
Session-level sparse codes
Accumulation of sparse codes in a session
Universal background sparse coding (UBSC)
Frame-level sparse codes
MBN based sparse coding
Frame-level acoustic feature

Figure 1: UBSC+MBN speaker clustering system.

This paper is organized as follows. In Section 2, we present the MBN-based system. In Section 3, we present the MBN algorithm. In Section, 4, we present the UBSC model. In Section 5, we present the merits of the method. In Section 6, we report comparison results. In Section 7, we conclude this paper.

2. System

We propose the following speaker clustering algorithm:[1]

- The first step trains a speaker- and session-independent universal background model (UBM), which produces a d-dimensional supervector for each session.

  A common choice of UBM is GMM [8]. We further propose another choice, i.e. UBSC, in Section 4.
- The second step reduces the dimension of $\mathbf{x}$ from d to $\bar{d}$ ($\bar{d} \ll d$) by MBN which is introduced in Section 3.
- The third step conducts k-means clustering on the low-dimensional data if the number of the underlying speak-

[1] The source code is downloadable from http://sites.google.com/site/zhangxiaolei321/speaker_recognition

http://dx.doi.org/10.21437/Interspeech.2016-65

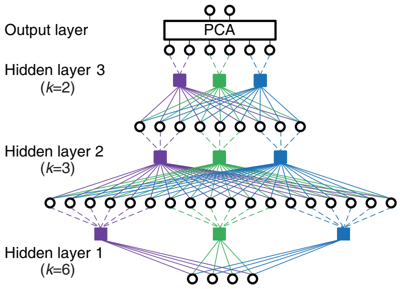

Figure 2: The MBN network. Each square represents a k-centers clustering.

ers is known, or agglomerative clustering if the number of the speakers is unknown.

The system that takes GMM as the UBM is denoted as the GMM+MBN system. The system that takes UBSC as the UBM, which is shown in Fig. 1, is denoted as the UBSC+MBN system.

3. Multilayer bootstrap network

The structure of MBN [7] is shown in Fig. 2. MBN is a multilayer localized PCA algorithm that gradually enlarges the area of a local region implicitly from the bottom hidden layer to the top hidden layer by high-dimensional sparse coding, and gets a low-dimensional feature explicitly by PCA at the output layer.

Each hidden layer of MBN consists of a group of mutually independent k-centers clusterings. Each k-centers clustering has k output units, each of which indicates one cluster. The output units of all clusterings are concatenated as the input of their upper layer [7].

MBN is trained layer-by-layer from bottom up. For training a hidden layer given a d-dimensional input $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, MBN trains each clustering independently [7]:

- **Random feature selection.** The first step randomly selects $\hat{d}$ dimensions of $\mathcal{X}$ ($\hat{d} \leq d$) to form a new set $\hat{\mathcal{X}} = \{\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_n\}$. This step is controlled by a hyperparameter $a = \hat{d}/d$.
- **Random sampling.** The second step randomly selects k data points from $\hat{\mathcal{X}}$ as the k centers of the clustering, denoted as $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$. This step is controlled by a hyperparameter k.
- **Sparse representation learning.** The third step assigns the input $\hat{\mathbf{x}}$ to one of the k clusters and outputs a k-dimensional indicator vector $\mathbf{h} = [h_1, \ldots, h_k]^T$. For example, if $\hat{\mathbf{x}}$ is assigned to the second cluster, then $\mathbf{h} = [0, 1, 0, \ldots, 0]^T$. The assignment is calculated according to the similarities between $\hat{\mathbf{x}}$ and the k centers, in terms of some predefined similarity measurement at the bottom layer, such as the minimum squared loss $\arg\min_{i=1}^{k} \|\mathbf{w}_i - \hat{\mathbf{x}}\|^2$, or in terms of $\arg\max_{i=1}^{k} \mathbf{w}_i^T \hat{\mathbf{x}}$ at all other hidden layers [7].

A suggested parameter setting is given in [7].

4. Universal background sparse coding

The proposed UBSC is shown in Fig. 3. Suppose we have S sessions $\{\mathcal{U}_s\}_{s=1}^{S}$ with the s-th session $\mathcal{U}_s$ defined as $\mathcal{U}_s = \{\mathbf{x}_{s,i}\}_{i=1}^{n_s}$ where $\mathbf{x}_{s,i}$ is the acoustic feature of the i-th frame of $\mathcal{U}_s$. UBSC executes the following steps:

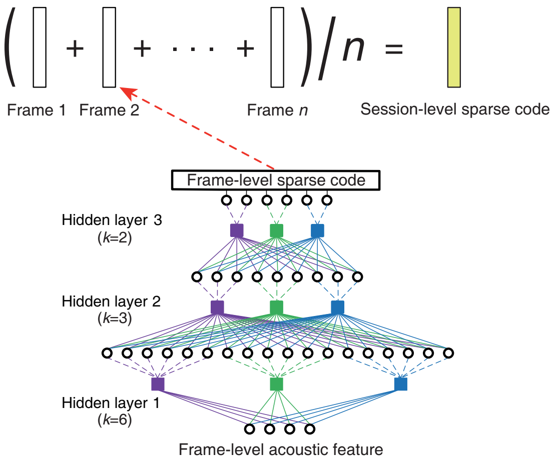

Figure 3: Principle of UBSC. The operator "+" denotes element-wise addition between vectors.

- The first step mixes all sessions into a large corpus $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N}$, where $N = \sum_{s=1}^{S} n_s$.
- The second step trains an MBN with $\mathcal{X}$, and generates a D-dimensional sparse vector $\mathbf{y}_i$ for each frame $\mathbf{x}_i$. Note that, different from [7], MBN does not further reduce the feature to a low-dimensional feature by PCA.
- The third step generates session-level supervectors $\{\bar{\mathbf{y}}_s\}_{s=1}^{S}$ by conducting an element-wise average over the frames that belong to the same session: $\bar{y}_{s,d} = \frac{1}{n_s} \sum_{i=1}^{n_s} y_{s,i,d}$, $\forall d = 1, \ldots, D$, where $\mathbf{y}_{s,i} = [y_{s,i,1}, \ldots, y_{s,i,D}]^T$ and $\bar{\mathbf{y}}_s = [\bar{y}_{s,1}, \ldots, \bar{y}_{s,D}]^T$.

Based on the principle of MBN, one layer is enough, particularly for supervised learning. However, in practice, we may also train multiple layers for reducing the random noise of data.

5. Merits of the proposed method

One of the main problems of a learning system is the similarity problem between data points, which can be decomposed to two factors: (i) similarity metric, and (ii) nonlinearity.

Regarding the similarity metric, speech frames are not distributed uniformly in the original feature space. That is to say, Euclidean distance is not a suitable similarity metric. Therefore, we cannot average the time-frequency energy of speech frames directly for a session-level feature. Traditional methods fit data to a predefined model template, e.g. GMM, where the original feature space is projected to a rescaled space defined by the model. After the projection, we can average frame-level features for session-level supervectors. MBN-based methods provide an adaptive similarity metric, which is the proportion of the nearest neighbors that fall into the intersection of two local regions, by a concatenation of the uniform resampling, nearest neighbor optimization, and binarization. They do not rely on model templates, which may work better than traditional methods.

Regarding the nonlinearity, because the supervectors are high-dimensional, it is very likely that they contain some nonlinearity. That is to say, two speech frames that are faraway (dissimilar) in the original high-dimensional space may not

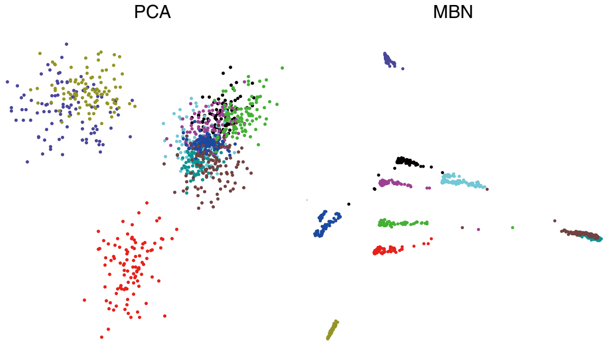

Figure 4: Visualizations of 10 speakers by GMM+PCA and GMM+MBN respectively, where a 16-mixture GMM-UBM with 20 EM iterations is used to produce their input supervectors. The speakers are labeled in different colors.

be so far apart after projecting the original space to a linear space by some nonlinear dimensionality reduction method, and vice versa. However, most traditional dimensionality reduction methods are linear methods, e.g. PCA. Although some kernel based nonlinear methods have been tried, they have to tune the free parameters of the kernels, which limits their practical use, particularly in an unsupervised setting where no information is available for the parameter tuning. MBN-based methods are nonlinear methods without parameter tuning, thanks to the binarization (the third step of MBN), which may work better than linear methods and is more practical than existing nonlinear methods. See [7] for more information.

6. Experiments

We first evaluate the GMM+MBN system, comparing with GMM+PCA. Then, we evaluate UBSC+MBN, comparing with GMM+PCA and the proposed GMM+MBN.

In both evaluations, we used the training corpus of speech separation challenge (SSC) [9]. The training corpus of SSC contains 34 speakers, each of which has 500 clean utterances.

For each speaker clustering job, we assumed that the number of speakers was known. We took the original feature or the low dimensional feature as the input of k-means clustering. Because the k-means clustering suffers from local minima, we ran it 50 times and picked the clustering result that corresponded to the optimal objective value (i.e., the minimum mean squared error) among all 50 candidate objective values as the final clustering result. We ran each experiment 10 times and reported the average performance.

6.1. Evaluation of GMM+MBN

We selected the first 100 utterances (a.k.a., sessions) of each speaker for evaluation, which amounts to 3400 utterances. We set the frame length to 25 milliseconds and frame shift to 10 milliseconds, and extracted a 25-dimensional MFCC feature.

For the proposed GMM+MBN, we set $V = 400$, $a = 0.5$, and k to 1700-850-425-212-106-53 (i.e. $k_{l+1} = \lfloor 0.5k_l \rfloor$ where l denotes the l-th layer). The output of PCA was set to $\{2, 3, 5, 10, 30, 50\}$ dimensions respectively.

We compared with PCA and k-means clustering. For the PCA-based method, we first used the same GMM-UBM as that in GMM+MBN to extract high-dimensional supervectors, then reduced the dimension of the supervectors to $\{2, 3, 5, 10, 30, 50\}$ respectively, and finally evaluated the low-dimensional output of PCA by k-means clustering. For the k-means-clustering-based method, we apply k-means clustering to the high-dimensional supervectors directly.

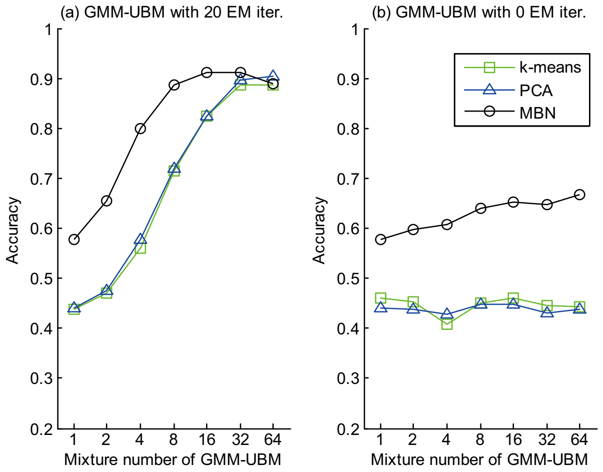

Figure 5: Accuracy comparison (in terms of NMI) between k-means clustering-, PCA-, and MBN-based methods with respect to the mixture number of GMM-UBM. (a) Comparison when the EM iteration number of GMM-UBM is set to 20. (b) Comparison when the EM iteration number of GMM-UBM is set to 0. Note that given a mixture number of GMM-UBM, the accuracy of a method is the best result among the results produced from 6 candidate output dimensions of the method, except k-means clustering.

The performance was measured by normalized mutual information (NMI) [10]. MNI was proposed to overcome the label indexing problem between the ground-truth labels and the predicted labels. It is one of the standard evaluation metrics of unsupervised learning. The higher the NMI is, the better the performance is. Note that NMI has a strong one-to-one correspondence with classification accuracy.

Results: Because all comparison methods use GMM-UBM to extract speaker- and session-independent supervectors, we need to study how they behave in different GMM-UBM settings, in terms of mixture number and expectation-maximization (EM) iterations. (i) The mixture number reflects the capacity of GMM-UBM for modelling an underlying data distribution: if the mixture number is smaller than the number of speakers, GMM-UBM is likely *underfitting*, i.e. it cannot grasp the data distribution well. To study this effect, we set the mixture number to $\{1, 2, 4, 8, 16, 32, 64\}$ respectively. (ii) The number of EM iterations reflects the quality of the acoustic feature produced by GMM-UBM: if the EM optimization is not sufficient, the acoustic feature is noisy. To study this effect, we set the number of EM iterations to $\{0, 20\}$ respectively, where setting the number of iterations to 0 means that GMM-UBM is initialized with randomly sampled means without EM optimization, which is the worst case.

Fig. 4 and Supplementary-Fig. 1 give a comparison example between PCA and MBN in visualizing the first 10 speakers, where a 16-mixtures GMM-UBM with 20 and 0 EM iteration are used to generate their inputs respectively. From the figures, we can see that MBN produces good visualizations.

Fig. 5 reports results with respect to the mixture number of GMM-UBM. Fig. 6 reports results with respect to the number

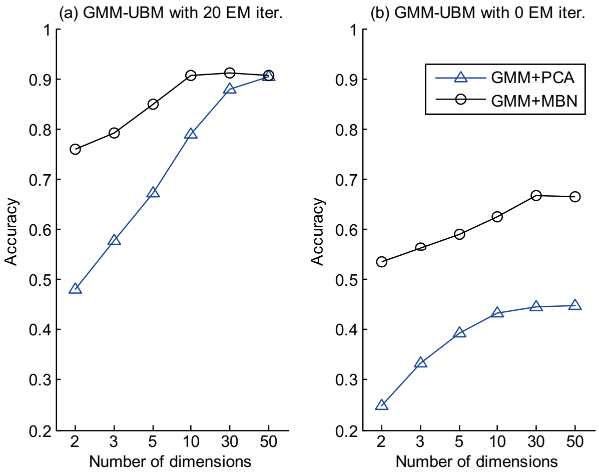

Figure 6: Accuracy comparison (in terms of NMI) between PCA- and MBN-based methods with respect to the number of output dimensions. (a) Comparison when the EM iteration number of GMM-UBM is set to 20. (b) Comparison when the EM iteration number of GMM-UBM is set to 0. Note that given a number of output dimensions, the accuracy of a method is the best result among the results produced from 7 candidate GMM-UBMs.

of output dimensions. Supplementary-Tables 1 and 2 report the detailed results of the two figures. From the figures and tables, we observe the following phenomena: (i) GMM+MBN outperforms GMM+PCA and the k-means-clustering-based method, with a relative improvement of 8% when GMM-UBM is optimized by 20 iterations, and with an relative improvement of 40% when GMM-UBM is optimized by 0 iteration; (ii) GMM+MBN is less sensitive to different parameter settings of both GMM-UBM and MBN itself; (iii) GMM+PCA is sensitive to both the mixture number of GMM-UBM and the number of output dimensions, and strongly relies on the effectiveness of GMM-UBM.

6.2. Evaluation of UBSC+MBN

We selected the first 10 utterances of the first 10 speakers, which amounts to 100 utterances containing 17,385 frames.

For UBSC+MBN, UBSC adopted the following typical parameter setting: $V = 400$, $a = 0.5$, and k were set to 2000-1000-500-250-125 (i.e. $k_{l+1} = \lfloor 0.5k_l \rfloor$). MBN took $V = 400$, $a = 0.5$, and k were set to 50-35-24-16 (i.e. $k_{l+1} = \lfloor 0.7k_l \rfloor$). The output of PCA was set to $\{2, 3, 5, 10, 30, 50\}$ dimensions respectively.

We compared the two universal background models, i.e. UBSC and GMM-UBM, given either PCA or MBN as the dimensionality reduction toolbox. We searched the mixture number of GMM-UBM through $\{2, 4, 8, 16, 32, 64\}$ and found that setting the mixture number of GMM-UBM to 32 performs the best. Therefore, we reported the result of GMM-UBM with 32 mixtures. The MBN in both GMM+MBN and UBSC+MBN adopted the same hyperparameters.

Results: Fig. 7 gives a comparison between GMM+PCA, UBSC+PCA, GMM+MBN, and UBSC+MBN on visualization. From the figure, we observe that, (i) when PCA is used as the dimensionality reduction tool, UBSC+PCA outperforms GMM+PCA apparently, such as differentiating the speakers with yellow and deep-blue colors. Because GMM-UBM has enough mixtures for modeling the 10 speakers, the only reason for their differences is that the data distributions of the speakers are not exactly Gaussian. (ii) When MBN is used as the dimensionality reduction tool, UBSC+MBN performs at least as equally as GMM+MBN with a smaller within-class variance than GMM+MBN.

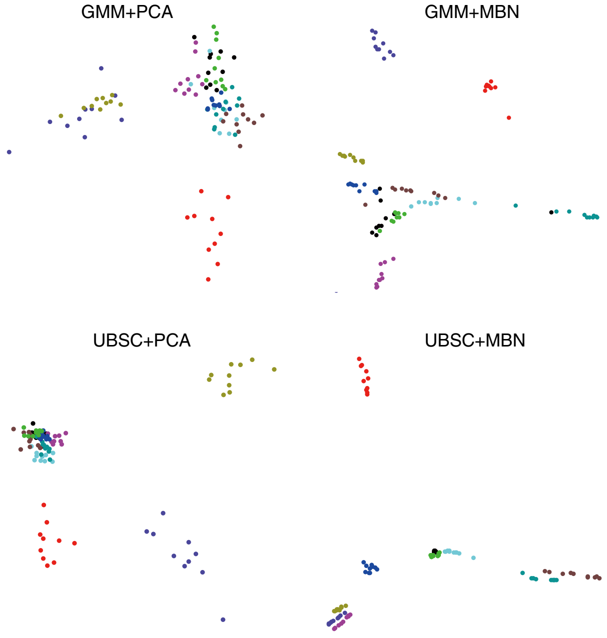

Figure 7: Visualizations of 10 speakers by PCA and MBN at layer 3 respectively, where a 16-mixtures UBM with 20 EM iterations is used to produce their input supervectors. The speakers are labeled in different colors.

Table 1: Accuracy comparison (in terms of NMI) of speaker clustering algorithms.

	2-dim	3-dim	5-dim	10-dim	30-dim	50-dim
GMM+PCA	64.44	71.09	77.74	84.48	86.93	83.90
UBSC+PCA	73.78	73.79	85.25	97.96	96.81	94.82
GMM+MBN	82.74	87.59	90.76	91.72	91.27	90.91
UBSC+MBN	81.40	91.86	95.38	99.11	97.15	97.39

Table 1 lists the comparison result on speaker clustering. From the table, we observe that, (i) UBSC significantly outperforms GMM-UBM, and (ii) MBN significantly outperforms PCA.

7. Conclusions

In this paper, we have proposed a multilayer bootstrap network based speaker clustering algorithm. It uses GMM-UBM or the novel UBSC as the universal background model to extract a high-dimensional feature from the original MFCC acoustic feature, then uses MBN to reduce the high-dimensional feature to a low-dimensional space, and finally clusterings the low-dimensional data. We have compared it with GMM-UBM-, PCA-, and k-means-clustering-based methods. Experimental results have shown that the proposed method outperforms the referenced methods. Moreover, it is insensitive to parameter settings, which facilitates its practical use.

8. References

[1] C. Wooters and M. Huijbregts, "The ICSI RT07s speaker diarization system," in *Multimodal Technologies for Perception of Humans*. Springer, 2008, pp. 509–519.

[2] K.-i. Iso, "Speaker clustering using vector quantization and spectral clustering," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2010, pp. 4986–4989.

[3] T. L. Nwe, H. Sun, B. Ma, and H. Li, "Speaker clustering and cluster purification methods for rt07 and rt09 evaluation meeting data," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 2, pp. 461–473, 2012.

[4] S. H. Shum, N. Dehak, R. Dehak, and J. R. Glass, "Unsupervised methods for speaker diarization: An integrated and iterative approach," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 10, pp. 2015–2028, 2013.

[5] N. Tawara, T. Ogawa, and T. Kobayashi, "A comparative study of spectral clustering for i-vector-based speaker clustering under noisy conditions," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 2041–2045.

[6] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, 2011.

[7] X.-L. Zhang, "Multilayer bootstrap networks," *arXiv preprint arXiv:1408.0848*, 2014.

[8] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Process.*, vol. 10, no. 1, pp. 19–41, 2000.

[9] M. Cooke and T.-W. Lee, "Speech separation challenge," http://staffwww.dcs.shef.ac.uk/people/M.Cooke/SpeechSeparationChallenge.htm, 2006.

[10] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, 2003.