An Investigation of Universal Background Sparse Coding Based Speaker Verification on TIMIT

Xiao-Lei Zhang

Center for Intelligent Acoustics and Immersive Communications, School of Marine Science and Technology, Northwestern Polytechnical University, China

xiaolei.zhang@nwpu.edu.cn, xiaolei.zhang9@gmail.com

Abstract

In this paper, we propose a universal background model, named universal background sparse coding (UBSC), for speaker verification. The proposed method trains an ensemble of clusterings by data resampling, and produces sparse codes from the clusterings by one-nearest-neighbor optimization plus binarization. The main advantage of UBSC is that it does not suffer from local minima and does not make Gaussian assumptions on data distributions. We evaluated UBSC on a clean speech corpus—TIMIT. We used the cosine similarity and inner product similarity as the scoring methods of a trial. Experimental results show that UBSC is comparable to Gaussian mixture model.

Index Terms: multilayer bootstrap network, speaker verification, universal background sparse coding

1. Introduction

Speaker verification has long been a fundamental task in speech processing. In speaker verification, the recognizer verifies an identity claim made by a test speaker, and decides to accept or reject the claim. Based on the input speech material, speaker verification can be either *text-dependent* or *text-independent*. The former constrains the speaker to pronounce a prescribed text, while the latter does not constrain the speech contents. Here we study text-independent speaker verification.

The first and earliest speaker verification method, i.e. feature averaging, learns the utterance-level feature of an utterance by averaging the frame-level acoustic features [1]. The method requires long speech utterances to reach stable speech statistics.

The second method estimates the density of speech frames by statistical models. Early approaches of this kind build a model, e.g. vector quantization [2] or Gaussian mixture model (GMM) [3, 4], for each training speaker. These approaches are inefficient when the number of training speakers is large. To alleviate this problem, GMM-based universal background model (GMM-UBM), which builds a single GMM from the pool of all training speakers, was developed [5]. GMM-UBM is a fundamental method of the later research. To deal with noise factors, such as utterance variations and channel variations, many approaches were proposed along with GMM-UBM, where i-vectors [6, 7] are among the effective ones. They first extract high-dimensional supervectors from the first- and second-order statistics of GMM-UBM, and then reduce the noise factors by factor analysis.

The third method is based on deep neural networks (DNNs). It can be roughly categorized to two approaches. The first approach uses a DNN to extract bottleneck features that are then used as the input of GMM-UBM, e.g. [8]. The second approach takes a DNN trained for a different task, e.g. speech recognition, to generate class posteriors of speech frames [9], which is an alternative to GMM-UBM. To demonstrate the advantages of the two approaches, their DNNs need to be trained with additional data [10].

After feature extraction by the aforementioned methods, speaker verification needs to score the similarity of two speakers in a trial. The scoring methods include maximum a posteriori estimation [5], support vector machines [11], cosine similarity measurement [7], probabilistic linear discriminative analysis [12], etc.

Besides the aforementioned methods, sparse coding is another emerging topic [13–18]. Many sparse coding methods focus on post processing [13–17], including modeling and classification of speakers with GMM-UBM supervectors and i-vectors. The orthogonal matching pursuit (OMP) method proposed by Boominathan and Murty [18] is an alternative to GMM-UBM, which directly builds a sparse model for each test utterance from original acoustic features, and produces a sparse code for each speech frame.

In this paper, we propose a new UBM, named universal background sparse coding (UBSC), which directly builds a UBM from original acoustic features by data resampling and one-nearest-neighbor optimization. We compared UBSC with GMM-UBM on TIMIT. To analysis the difference between GMM-UBM and UBSC, we used neither denoising frontend nor additional data sets, and took the simple *cosine similarity* measurement as the scoring method of a trial. Experimental results show that UBSC is comparable to GMM-UBM.

2. Universal background sparse coding

2.1. Model training

UBSC trains an ensemble of k-centers clusterings. The centers of a k-centers clustering is trained simply by random sampling. Specifically, suppose we have a number of training speakers, and each speaker contains several utterances. The training process is as follows:

- The first step extracts frame-level acoustic features, e.g. mel-frequency cepstral coefficients (MFCC), from the speech signals, and then pools all acoustic features together into a set, denoted as $\mathcal{X} = \{\mathbf{x}_i\}_i$, where $\mathbf{x}_i$ denotes the acoustic feature of the i-th frame.
- The second step builds V random models $\{\mathbf{W}_v\}_{v=1}^{V}$, where the model $\mathbf{W}_v$ is a random sample of k frames from $\mathcal{X}$ without replacement[1], denoted as $\mathbf{W}_v = [\mathbf{w}_{v,1}, \ldots, \mathbf{w}_{v,k}]$.

From the above, we can see that UBSC has two hyperparameters k and V.

[1]The word "without replacement" means that the k frames are different observations in $\mathcal{X}$.

Algorithm 1

Input: Input utterance $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ and UBSC model $\{\mathbf{W}_v\}_{v=1}^V$ where $\mathbf{W}_v = [\mathbf{w}_{v,1}, \ldots, \mathbf{w}_{v,k}]$

Output: High-dimensional supervector $\mathbf{z}$

for $i = 1, \ldots, N$ **do**
  for $v = 1, \ldots, V$ **do**
    for $j = 1, \ldots, k$ **do**
      $d_{i,v,j} \leftarrow \|\mathbf{x}_i - \mathbf{w}_{v,j}\|_2$.
    end for
    Learn a sparse code $\mathbf{s}_{i,v} = [s_{i,v,1}, \ldots, s_{i,v,k}]^T$ by

$$s_{i,v,j} \leftarrow \begin{cases} 1, \text{ if } d_{i,v,j} = \min_{\forall l \in \{1,\ldots,k\}} d_{i,v,l} \\ 0, \text{ otherwise} \end{cases} \quad (1)$$

  end for
  $\bar{\mathbf{s}}_i \leftarrow [\mathbf{s}_{i,1}^T, \ldots, \mathbf{s}_{i,V}^T]^T$
end for
return $\mathbf{z} \leftarrow \frac{1}{N} \sum_{i=1}^N \bar{\mathbf{s}}_i$

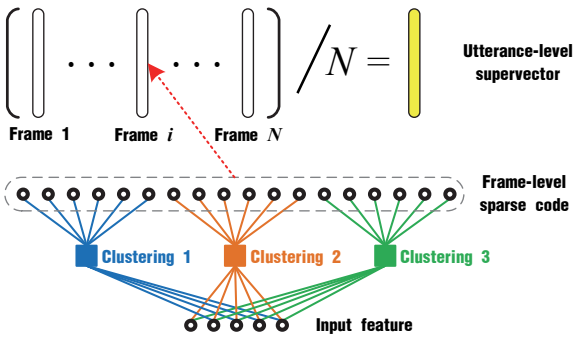

Figure 1: *Principle of UBSC. The three base clusterings are drawn in different colors.*

2.2. Sparse representation learning

In the feature learning stage, given an utterance $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ of a speaker to be processed in either the training, enrollment, or test stage, a high-dimensional supervector $\mathbf{z}$ is extracted from $\mathbf{X}$ by Algorithm 1.

Algorithm 1 first calculates frame-level sparse features $\{\bar{\mathbf{s}}_i\}_{i=1}^N$ and then averages the frame-level features for an utterance-level supervector $\mathbf{z}$, where $\bar{\mathbf{s}}_i$ is the concatenation of a group of one-hot codes $\{\mathbf{s}_{i,v}\}_{v=1}^V$, each of which is produced from a random model $\mathbf{W}_v$. The learning process is illustrated in Fig. 1.

3. Motivation and related work

UBSC was motivated from multilayer bootstrap networks [19]. They share the same theoretical base (see [19] for the theoretical base of UBSC). However, UBSC may not be replaced by multilayer bootstrap networks. Because the scoring process of speaker verification is a supervised classification problem, learning a multilayer network in an unsupervised manner loses much information needed for the supervised problem. Empirically, we observed performance drop by using multilayer bootstrap networks.

UBSC is a comparable model to GMM-UBM. Comparing to GMM-UBM, UBSC does not make model assumptions on data distributions, since each base model of UBSC is a random sample of data. UBSC does not suffer from local minima, since it is optimized by one-nearest-neighbor. UBSC may be implemented easily. Its training process is also fast. A drawback of UBSC is that its network is usually large, so that its test complexity may be high. But it supports parallel computing naturally, which may alleviate this drawback.

UBSC is different from the OMP sparse coding method in [18]. First, OMP learns a sparse model for each test utterance, while UBSC builds a single UBM from all training utterances. Second, OMP is formulated as an NP-hard combinatorial optimization problem, while UBSC is a simple algorithm that contains only data resampling and one-nearest-neighbor optimization. Third, OMP produces the test score of a trial directly, while UBSC produces a high-dimensional supervector for each utterance, leaving the scoring method an open problem. Besides, UBSC has a clear geometric explanation. It also supports parallel computing naturally.

4. Empirical results

In this section, we compared UBSC with GMM-UBM on a clean speech corpus—TIMIT, and adopted a similar experimental setting with that in [3]. All experiments were conducted with MATLAB 2015b on a Linux server running with 2 Inter(R) Xeon(R) E5-2650 CPUs, 4 Nvidia Tesla K80 GPUs (including 8 GPU cores), and 256 GB memory. Here we report the main results, leaving the details in the Supplementary Material in http://www.xiaolei-zhang.net/publications.htm.

4.1. Experimental settings

TIMIT contains 630 speakers, including 438 males and 192 females. Each speaker has 10 clean utterances. Each utterance is roughly 3 seconds long. The sampling rate of TIMIT is 16 kHz. To guarantee the reproducibility of the experiments, we did not adopt voice activity detection. We set frame length to 25 ms and frame shift to 10 ms. Then, we applied Hamming window filter to each frame, and extracted 19 dimensional MFCC with 1 dimensional log power energy by the VOICEBOX toolbox.[2] We further filtered the 20 dimensional features by a Hamming window in the mel-domain.

We adopted the MSR Identity Toolbox as the implementation of the GMM-UBM baseline.[3] In the training stage of GMM-UBM, we initialized the mean (and variance) of each Gaussian component by the mean (and variance) of the MFCC features of a randomly selected utterance, and set all Gaussian components to an equal prior probability. In the test stage, given an utterance, we extracted first- and second-order statistics from each frame [5] which are further concatenated to a frame-level feature. Then, we averaged the frame-level features for an utterance-level supervector.

The parameter settings of the comparison methods are as follows. For GMM-UGM, we searched the number of Gaussian mixtures from $2^{[1:1:11]}$ and the number of EM iterations from $[1, 5, 10, 20, 30]$ in grid, where the symbol $[a : b : c]$ represents a serial integers from a to c with an increment of b. For UBSC, we searched parameter k from $2^{[1:1:17]}$ and parameter V from $[1, 3, 10, 30]$ in grid. For each parameter pair, we recorded the average *equal error rate* (EER) and standard deviation of 10 independent runs.

[2] http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html

[3] https://www.microsoft.com/en-us/research/publication/msr-identity-toolbox-v1-0-a-matlab-toolbox-for-speaker-recognition-research-2/

Table 1: *Number of claimant and imposter trials.*

	#speakers	#true trials	#imposter trials
Male	438	876	765,624
Female	192	384	146,688
Male+Female	630	1,260	1,585,080

Table 2: *EERs (in percent) produced by GMM-UBM and UBSC, when the cosine similarity scoring method is used. The values in brackets are standard deviations.*

	Male	Female	Male+Female
GMM-UBM	**3.92** (0.26)	5.46 (0.55)	**3.35** (0.24)
UBSC	**3.79** (0.28)	**3.99** (0.23)	**3.16** (0.16)

We used the two-tailed t-test to evaluate the statistical significance of the difference between a pair of results, where the null-hypothesis is defined as that the difference is insignificant. The significance level α is set to 0.05. If the p-value is smaller than α, then the null-hypothesis is rejected, in other words, the difference is regarded as statistically significant.

4.2. Results with cosine similarity scoring method

We used the *cosine similarity* measurement as the scoring method of two supervectors [7], which is defined by $\frac{\mathbf{x}^T\mathbf{y}}{\|\mathbf{x}\|_2\|\mathbf{y}\|_2}$ where $\mathbf{x}$ and $\mathbf{y}$ are two vectors.

4.2.1. Main results

For each speaker in TIMIT, we selected the first 8 utterances as training speech with each of the remaining 2 utterances as an individual test. We took each speaker as a claimant with the remaining speakers acting as imposters, and rotated through the tests of all speakers. We investigated the comparison methods on males, females, and both genders of speakers respectively. The number of claimant and imposter trials are summarized in Table 1.

We list the best EERs of the comparison methods in Table 2 with examples of the detection error tradeoff (DET) curves shown in Fig. 2 and Figs. S1 and S2 in the Supplementary Material. From the comparison, we observe that UBSC shows statistically significant improvement over GMM-UBM in the Female experiment with a p-value of 0, and slightly better performance than GMM-UBM in the Male and the Male+Female experiments with p-values of 0.2849 and 0.0532 respectively.

4.2.2. Effect of parameters k and V

We report the EER with respect to parameters k and V in Fig. 3 and Figs. S3 and S4 in Supplementary Material. From the figures, we observe the following two phenomena. (i) If k is fixed, then enlarging V reduces EER, and moreover, setting V to 30 balances the performance and computational complexity. (ii) Given V fixed, we can find an optimal k.

4.2.3. Effect of number of training speakers

To study how the number of training speakers affect the performance, we randomly select 10, 30, 100, and 300 speakers from males, females, and both genders respectively. We ran the experiment 10 times and report the average performance in Fig. 4. From the figure, we find that the empirical conclusions in Section 4.2.1 are not affected by the number of training speakers.

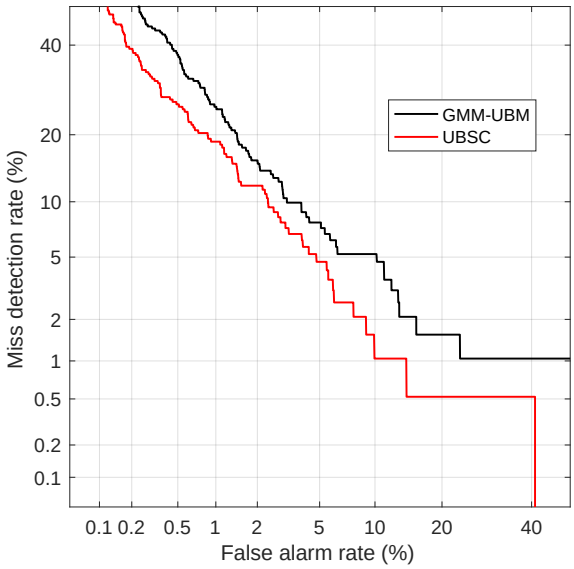

Figure 2: *DET curves produced by GMM and UBSC in the Female experiment, with the cosine similarity scoring method.*

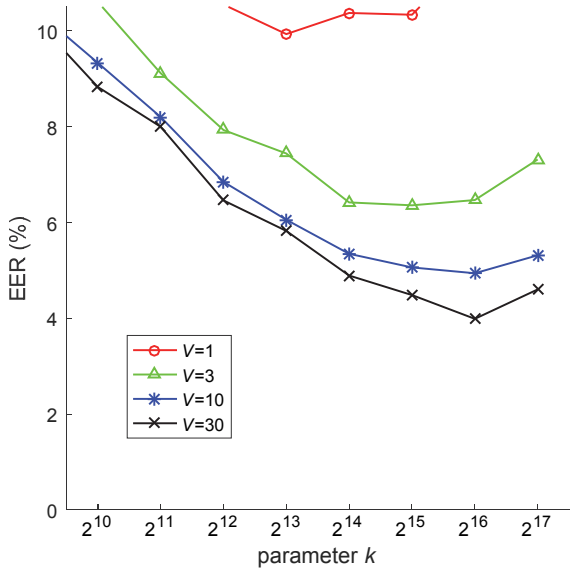

Figure 3: *EER curves produced by the UBSC with the cosine similarity scoring method with respect to parameters V and k in the Female experiment.*

Moreover, the advantage of UBSC with a small number of training speakers are more apparent than that with a relatively large number of training speakers. Possible explanations include that, when only a small number of training speakers are used, (i) the local minima of the expectation-maximization algorithm of GMM affects its performance heavily, and (ii) the data distribution may not be Gaussian.

4.3. Effect of scoring methods

To investigate how scoring methods affect the performance of UBSC, we compared the cosine similarity scoring method with

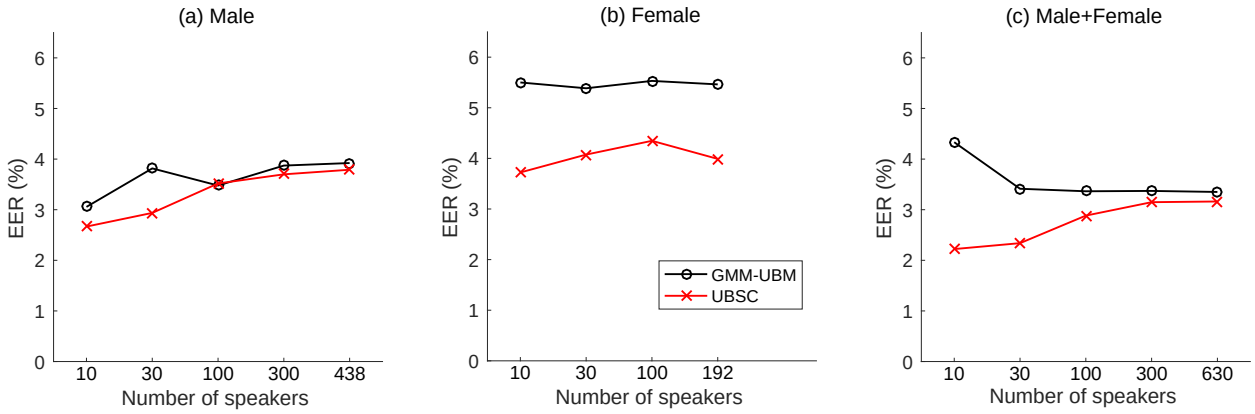

Figure 4: *EER curve comparison of UBSC and GMM-UBM with respect to different number of training speakers, when the cosine similarity scoring method is used.*

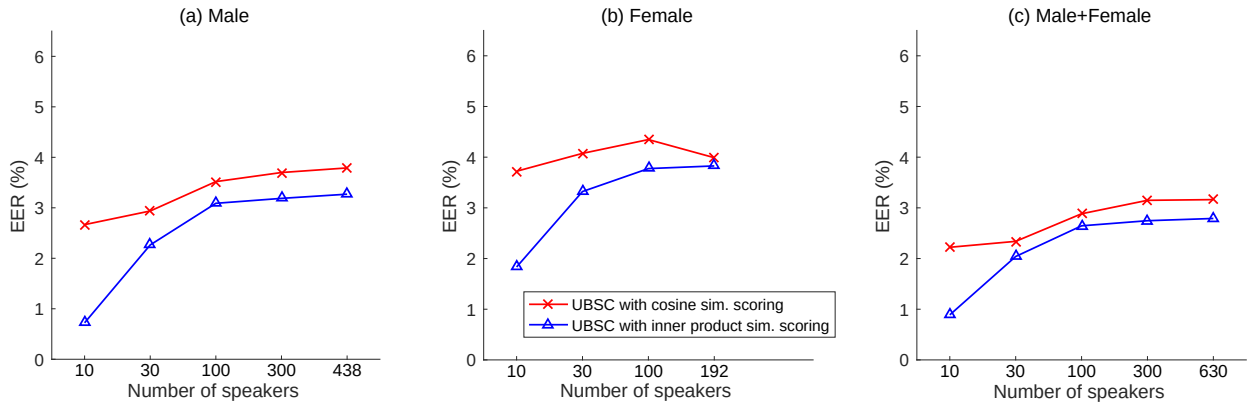

Figure 5: *EER curve comparison between the UBSC with different scoring methods with respect to different number of training speakers.*

Table 3: *EERs (in percent) produced by UBSC. The values in brackets are standard deviations. The word "sim." is short for similarity.*

	Male	Female	Male+Female
UBSC with cosine sim.	3.79 (0.28)	**3.99** (0.23)	3.16 (0.16)
UBSC with inner product sim.	**3.27** (0.21)	**3.83** (0.26)	**2.79** (0.18)

the *inner product similarity* scoring method, where the inner product similarity is defined by $\mathbf{x}^T\mathbf{y}$.

We used the same experimental setting as that in Section 4.2.1. We list the best EERs of the two scoring methods in Table 3. From the comparison, we observe that the inner-product similarity scoring method shows statistically significant improvement over the cosine similarity scoring method in the Male and the Male+Female experiments with p-values of 0.0001 and 0.0001 respectively, and is slightly better performance than the latter in the Female experiment with a p-value of 0.1539.

To study how the number of training speakers affect the performance, we followed the same experimental setting as that in Section 4.2.3. We report the average performance in Fig. 5. From the figure, we find that the experimental conclusions on the two scoring methods of UBSC are not affected by the number of training speakers.

5. Conclusions and future work

In this paper, we have introduced a universal background model, called UBSC, for speaker verification. UBSC is trained simply by data resampling where each random sample of data forms the centers of a base clustering. In the test stage, given an utterance, UBSC first learns a frame-level sparse code by concatenating the one-hot output codes produced from the base clusterings, and then averages the frame-level sparse codes of all frames for an utterance-level supervector. UBSC does not make model assumptions and does not suffer from local minima. It is easily implemented and used. It supports parallel computing naturally.

We compared UBSC with GMM-UBM on the clean corpus—TIMIT. We used the cosine similarity and inner product similarity as the scoring methods. Experimental results show that, when the scoring method is the cosine similarity measurement, UBSC performs better than GMM on females, and is comparable to GMM on males and both genders of speakers. Moreover, the UBSC with the inner product similarity performs better than that with the cosine similarity. The conclusion is consistent with different number of speakers.

In the future, we will develop a denoising frontend and a backend classifier for UBSC. We will compress the scale of the UBSC model for speeding up the extraction of the supervectors. We will also investigate the density estimation ability of UBSC on more complicated data distributions.

6. References

[1] J. Markel, B. Oshika, and A. Gray, "Long-term feature averaging for speaker recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 25, no. 4, pp. 330–337, 1977.

[2] F. K. Soong, A. E. Rosenberg, B.-H. Juang, and L. R. Rabiner, "Report: A vector quantization approach to speaker recognition," *AT&T Tech. J.*, vol. 66, no. 2, pp. 14–26, 1987.

[3] D. A. Reynolds, "Speaker identification and verification using gaussian mixture speaker models," *Speech Commun.*, vol. 17, no. 1, pp. 91–108, 1995.

[4] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *IEEE Trans. Speech, Audio Process.*, vol. 3, no. 1, pp. 72–83, 1995.

[5] 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.

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

[7] N. Dehak, P. J. 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.

[8] A. K. Sarkar, C.-T. Do, V.-B. Le, and C. Barras, "Combination of cepstral and phonetically discriminative features for speaker verification," *IEEE Signal Process. Lett.*, vol. 21, no. 9, pp. 1040–1044, 2014.

[9] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 1695–1699.

[10] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Process. Lett.*, vol. 22, no. 10, pp. 1671–1675, 2015.

[11] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Process. Lett.*, vol. 13, no. 5, pp. 308–311, 2006.

[12] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Proc. Odyssey Workshop*, 2010, pp. 14–25.

[13] I. Naseem, R. Togneri, and M. Bennamoun, "Sparse representation for speaker identification," in *Proc. Int. Conf. Pattern Recogn.* Elsevier, 2010, pp. 4460–4463.

[14] J. M. K. Kua, E. Ambikairajah, J. Epps, and R. Togneri, "Speaker verification using sparse representation classification," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, 2011, pp. 4548–4551.

[15] M. Li, X. Zhang, Y. Yan, and S. Narayanan, "Speaker verification using sparse representations on total variability i-vectors," in *Proc. Interspeech*, 2011, pp. 2143–2157.

[16] B. C. Haris and R. Sinha, "Robust speaker verification with joint sparse coding over learned dictionaries," *IEEE Trans. Inform. Forensics, Security*, vol. 10, no. 10, pp. 2143–2157, 2015.

[17] J. M. K. Kua, J. Epps, and E. Ambikairajah, "i-vector with sparse representation classification for speaker verification," *Speech Commun.*, vol. 55, no. 5, pp. 707–720, 2013.

[18] V. Boominathan and K. S. R. Murty, "Speaker verification using sparse representation classification," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, 2012, pp. 4381–4384.

[19] X.-L. Zhang, "Multilayer bootstrap networks," *arXiv preprint arXiv:1408.0848*, pp. 1–41, 2014.