Interspeech 2018
2-6 September 2018, Hyderabad

Cosine Metric Learning for Speaker Verification in the i-Vector Space

Zhongxin Bai, Xiao-Lei Zhang, and Jingdong Chen

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

zxbai@mail.nwpu.edu.cn, xiaolei.zhang9@gmail.com, jingdongchen@ieee.org

Abstract

It is known that the equal-error-rate (EER) performance of a speaker verification system is determined by the overlap region of the decision scores of true and imposter trials. Also, the cosine similarity scores of the true or imposter trials produced by the state-of-the-art i-vector front-end approximate to a Gaussian distribution, and the overlap region of the two classes of trials depends mainly on their between-class distance. Motivated by the above facts, this paper presents a cosine similarity learning (CML) framework for speaker verification, which combines classical compensation techniques and the cosine similarity scoring for improving the EER performance. CML minimizes the overlap region by enlarging the between-class distance while introducing a regularization term to control the with-in class variance, which is initialized by a traditional channel compensation technique such as linear discriminant analysis. Experiments are carried out to compare the proposed CML framework with several traditional channel compensation baselines on the NIST speaker recognition evaluation data sets. The results show that CML outperforms all the studied initialization compensation techniques.

Index Terms: speaker verification, cosine metric learning, channel and session compensation.

1. Introduction

A speaker verification system consists of two parts, i.e., a front-end and a back-end [1–3]. The front-end extracts identity features from a speaker utterance. The state-of-the-art identity feature extractor is the factor analysis [1], which extracts identity vectors (i-vectors) from the output supervectors of either a Gaussian mixture model (GMM) based universal background model (GMM-UBM) or a deep neural network based UBM [4–6]. Another popular front-end, called deep vector (d-vector), takes the average of the activations of the last hidden layer of a DNN as the speaker feature [7–9].

The back-end verifies the similarity of two speakers by evaluating the similarity of their identity features. Common back-ends include the cosine similarity scoring, support vector machines, and probabilistic linear discriminant analysis (PLDA). As the output of a front-end is both inter-session and speaker dependent, statistical techniques are usually employed to compensate channel or session variability before scoring. Compensation techniques include linear discriminant analysis (LDA) [10], within class covariance normalization (WCCN) [11] and nuisance attribute projection (NAP) [12]; however, these compensation techniques do not have a direct connection to the final scoring result of speaker verification.

Metric learning was proposed to combine compensation methods with scoring methods [13, 14], which aims to reduce the within-class variation and maximize the between-class distance. In [15], Fang *et al.* also employed neighborhood component analysis to learn a projection matrix that minimizes the average leave-one-out k-nearest neighbor classification error. Some recent works attempt to train the front-end and back-end jointly by end-to-end deep learning. For example, the methods for text-dependent speaker verification [16, 17] learn a deep model that maps a pair of enrollment and test utterances directly to a cosine similarity score. In [18], David *et al.* applied a similar end-to-end framework that jointly trains a deep neural network front-end and a PLDA-like back-end.

Motivated by the work in [19], we propose in this paper a metric learning method, named *cosine metric learning* (CML), for speaker verification. It combines traditional compensation methods with the cosine similarity scoring method for improving the equal error rate (EER). The proposed method takes a linear transform A_0 produced from a compensation method as its initialization, and learns a new linear transform A that minimizes EER directly by minimizing the overlap region of the decision score distributions between true trials and imposter trials. Experimental results on the NIST speaker recognition evaluation (SRE) corpora show that the proposed method can combine traditional compensation methods with the cosine similarity scoring method effectively for optimizing the EER performance.

Note that the focal point of this paper is on describing a general metric learning approach instead of an end-to-end deep learning method. As will be shown later, CML can be extended to end-to-end deep learning.

2. Cosine metric learning

The probability distribution of the decision scores produced from an i-vector speaker verification system is illustrated in Fig. 1. One can see from this figure that the scores produced from the true and imposter trials can be modeled approximately by two Gaussian distributions respectively. The performance of the system is then determined by the overlap region between the two distributions. Since the overlap region is determined by the distance between the means of the two distributions, i.e. between-class distance, and the within-class variance of the two distributions, the proposed CML aims to reduce the overlap region by enlarging the distance between the means of the two distributions, thereby improving the verification performance.

2.1. Optimization objective

In the development stage, assume that we have a development set $\{x_i, y_i, l_i\}_{i=1}^N$, where x_i and y_i are a pair of speakers, l_i is the ground-truth similarity of the two speakers, and N denotes the total number of speaker pairs. If x_i and y_i come from the same speaker, then $l_i = 1$; otherwise $l_i = -1$. Furthermore, the index sets of the true trials and imposter trials are denoted by $\mathrm{pos} = \{i|l_i = 1\}_{i=1}^N$ and $\mathrm{neg} = \{i|l_i = -1\}_{i=1}^N$, respectively.

In the test stage, suppose that x_{target} and x_{test} represent the

10.21437/Interspeech.2018-1593

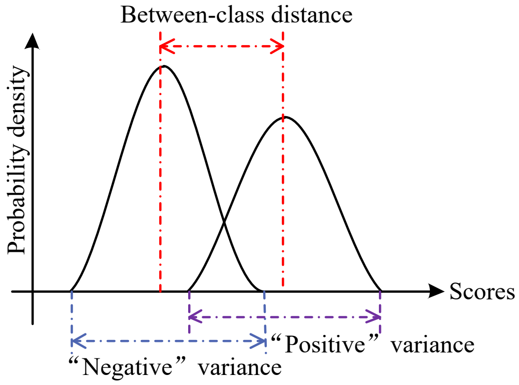

Figure 1: *Probability distribution of the decision scores produced by an i-vector/cosine speaker verification system. "Positive" denotes the true trials, "Negative" denotes the imposter trials.*

i-vectors of a target and a test speaker, respectively. Given a linear transform A that is used to compensate channel and session variability, the cosine similarity scoring back-end is written as:

$$S(x_{\text{targe}}, x_{\text{test}}, A) = \frac{\langle Ax_{\text{target}}, Ax_{\text{test}} \rangle}{\|Ax_{\text{target}}\| \|Ax_{\text{test}}\|}. \tag{1}$$

The proposed CML method aims to maximize the distance between the means of the decision scores of the true trials and imposter trials by optimizing A in the development stage. A possible optimization cost function is defined as follows:

$$f(A) = \max_A \frac{1}{|\text{pos}|} \sum_{i \in \text{pos}} S(x_i, y_i, A) - \frac{1}{|\text{neg}|} \sum_{i \in \text{neg}} S(x_i, y_i, A), \tag{2}$$

where $|\text{pos}|$ and $|\text{neg}|$ denote, respectively, the size of pos and neg. However, directly maximizing (2) may lead to a large variance of the score distribution. To control the variance of the score distribution, we add a regularization term $\|A - A_0\|^2$ to (2), i.e.,

$$\begin{aligned} f(A) = & \\ & \max_A \frac{1}{|\text{pos}|} \sum_{i \in \text{pos}} S(x_i, y_i, A) - \frac{1}{|\text{neg}|} \sum_{i \in \text{neg}} S(x_i, y_i, A) \\ & -\gamma \|A - A_0\|^2, \end{aligned} \tag{3}$$

where γ is a hyperparameter and A_0 is a predefined matrix, which can be any linear transform produced from a traditional compensation technique, such as LDA, WCCN, and NAP. With a simple mathematical manipulation, (3) can be rewritten as:

$$\begin{aligned} f(A) = \max_A & \sum_{i \in \text{pos}} S(x_i, y_i, A) - \alpha \sum_{i \in \text{neg}} S(x_i, y_i, A) \\ & - \beta \|A - A_0\|^2, \end{aligned} \tag{4}$$

where $\alpha = \frac{|\text{pos}|}{|\text{neg}|}$ and $\beta > 0$ are two hyperparameters. β is a free parameter that makes a trade-off between the optimization of A and its negative effect, i.e. enlarging the Gaussian variance of the decision scores. When β tends to infinity, A approaches to A_0. In other words, the performance of the proposed method is lower-bounded by its initialization method, which can be LDA, WCCN, NAP, etc.

Algorithm 1 Steepest descent algorithm for the CML back-end.

```
Input: Training pairs ${x_i, y_i, l_i}_{i=1}^N$; predefined matrix $A_0$;
    initial value of the optimization target $A^{(0)}$; hyperparameters $\alpha$ and $\beta$; constant $\epsilon > 0$;
Output: The best variability compensation matrix $A^*$;
Initial iteration index $k = 0$;
repeat
    Compute gradient directions $\nabla f(A^{(k)})$;
    Compute a step size $\lambda_k$ via exact line search [20];
    Compute the next point $A^{(k+1)} = A^{(k)} + \lambda_k \nabla f(A^{(k)})$;
    $k = k + 1$;
until $(\|\nabla f(A^{(k)})\| < \epsilon)$
```

2.2. Optimization algorithm

We use a gradient descent algorithm to solve problem (4). The gradient of $f(A)$ is:

$$\begin{aligned} \nabla f(A) = & \sum_{i \in \text{pos}} \frac{\partial S(x_i, y_i, A)}{\partial A} - \alpha \sum_{i \in \text{neg}} \frac{\partial S(x_i, y_i, A)}{\partial A} \\ & - 2\beta(A - A_0), \end{aligned} \tag{5}$$

where

$$\frac{\partial S(x_i, y_i, A)}{\partial A} = \frac{\partial \left\{ \frac{x_i^T A^T A y_i}{\sqrt{x_i^T A^T A x_i} \sqrt{y_i^T A^T A y_i}} \right\}}{\partial A} \tag{6}$$

For clarity, we denote the numerator and denominator of (6) by $u(A) = x_i^T A^T A y_i$ and $v(A) = \sqrt{x_i^T A^T A x_i} \sqrt{y_i^T A^T A y_i}$, respectively. It follows then that:

$$\nabla \left(\frac{u(A)}{v(A)} \right) = \frac{1}{v(A)} \frac{\partial u(A)}{\partial A} - \frac{u(A)}{v(A)^2} \frac{\partial v(A)}{\partial A} \tag{7}$$

$$\frac{\partial u(A)}{\partial A} = A(x_i y_i^T + y_i x_i^T) \tag{8}$$

$$\frac{\partial v(A)}{\partial A} = \frac{\sqrt{y_i^T A^T A y_i}}{\sqrt{x_i^T A^T A x_i}} A x_i x_i^T + \frac{\sqrt{x_i^T A^T A x_i}}{\sqrt{y_i^T A^T A y_i}} A y_i y_i^T \tag{9}$$

Substituting (6) to (9) into (5) gives the final gradient of $f(A)$.

We apply the steepest descent algorithm to solve this optimization problem, which is summarized in Algorithm 1.

3. Experiments

3.1. Dataset

All experiments were carried out on NIST 2006 SRE (8*conv* condition) and NIST 2008 SRE (8*conv* condition). Both of those are eight two-channel conversation excerpts. Each conversation involves the target speaker on their designated sides. A speaker utterance in a conversation is 1 to 2 minutes long after removing the silence segments by voice activity detection (VAD), where we took the automatic-speech-recognition (ASR) transcript as its VAD label. We divide all the speech signals into 15 second segments.

3.1.1. Development data

We used NIST 2006 SRE data (8*conv* condition) for development, which include 402 female speakers and 297 male speakers. There are a total of 24043 segments for the female speakers and 17765 segments for the male speakers. These data are used

Table 1: *Test conditions. "EN" denotes the number of enrollment speakers, "TN" denotes the number of test speech segments, "Tr-N" denotes the number of trials, and Ci denotes a test condition with i being the number of the enrollment speech segments (varies from 1 to 5).*

Condition	**Female**			**Male**		
Name	EN	TN	Tr-N	EN	TN	Tr-N
C1	394	2748	$1.08M$	238	1650	$393K$
C2	390	2748	$1.07M$	236	1650	$389K$
C3	381	2748	$1.05M$	231	1650	$381K$
C4	362	2748	$995K$	221	1650	$365K$
C5	320	2748	$879K$	201	1650	$332K$

to train gender-dependent speaker verification systems, including GMM-UBM, total variability matrix, LDA, WCCN, NAP, PLDA, and our CML. The number of speaker pairs for the CML model training contains $23K$ true trials and $9.3M$ imposter trials for the females, and $17.8K$ true trials and $5.3M$ imposter trials for the males.

3.1.2. Evaluation data

We used NIST 2008 SRE data ($8conv$ condition) for evaluation, which include 395 female speakers and 238 male speakers. In the enrollment stage, we selected 1 to 5 speech segments from the first conversation of a speaker as his/her enrollment data. In the test stage, we selected 1 speech segment from each of the remaining 7 conversations of the speaker for test, which corresponds to 7 test speech segments. We took each speaker as a claimant with the remaining speakers acting as imposters, and rotated through the tests of all speakers. We conducted the experiments on the female and male speaker respectively. The number of trials are summarized in Table1.

3.2. Experimental setup

We used 19 Mel frequency cepstral coefficients (MFCCs), 13 relative spectral filtered perceptual linear predictive cepstral coefficients (RASTA-PLP) and the log energy of each frame. Their delta and double delta coefficients are included. So, the total dimension per frame [21] is 99 (33 × 3). The frame length was set to 25 milliseconds, and the frame shift was set to 10 milliseconds. Feature warping with a window size of 3 seconds was applied after the acoustic feature extraction. We employed the MSR Identity Toolbox [22] to extract i-vectors and train LDA/PLDA. The number of mixture components of GMM-UBM was set to 2048. The dimension of the total variability matrix was set to 400.

We used the LDA, WCCN or NAP model for the initialization of CML. The CML with the three initialization models are denoted as LDA+CML, WCCN+CML, and NAP+CML respectively. For comparison, the initialization methods were also evaluated by the cosine similarity scoring. The output dimension of LDA was set to 200. The corank numbers [1] of NAP was set to 350 for the females and 100 for the males.

We also present the performance of the cosine similarity scoring back-end without any compensation techniques, denoted as "NULL", as well as the LDA+PLDA back-end for comparison. The evaluation metrics are EER and detection error tradeoff (DET) curve.

Table 2: *EER comparison between CML and its initialization channel compensation techniques on the female speakers.*

Method	C1	C2	C3	C4	C5
NULL	9.73%	6.63%	5.27%	4.62%	4.12%
LDA	6.88%	5.09%	4.37%	4.05%	3.74%
LDA+CML	4.35%	3.55%	3.23%	3.13%	2.97%
WCCN	5.34%	4.27%	3.74%	3.65%	3.37%
WCCN+CML	4.74%	4.11%	3.64%	3.55%	3.28%
NAP	5.57%	4.74%	4.35%	4.20%	3.94%
NAP+CML	4.92%	4.28%	3.94%	3.87%	3.63%
LDA+PLDA	4.11%	3.76%	3.50%	3.50%	3.44%

Table 3: *EER comparison between CML and its initialization channel compensation techniques on the male speakers.*

Method	C1	C2	C3	C4	C5
NULL	7.72%	5.90%	5.07%	4.61%	4.63%
LDA	6.62%	5.51%	4.89%	4.49%	4.55%
LDA+CML	5.50%	4.89%	4.55%	4.25%	4.35%
WCCN	5.71%	4.92%	4.56%	4.24%	4.28%
WCCN+CML	5.50%	4.85%	4.53%	4.22%	4.27%
NAP	7.16%	5.74%	4.95%	4.54%	4.59%
NAP+CML	5.66%	4.88%	4.51%	4.21%	4.28%
LDA+PLDA	5.29%	4.90%	4.72%	4.40%	4.40%

3.3. Main results

Table 2 lists the EER results on the female speakers. From the table, one can see that the proposed CML methods outperform their initialization methods. Specifically, LDA+CML achieves 2.5% absolute improvement over LDA in the C1 condition, and approximately 20% relative improvement over LDA in the C2 to C5 conditions. WCCN+CML outperforms WCCN slightly. NAP+CML outperforms NAP slightly.

Table 3 lists the EER resulta on the male speakers. From the table, we see that LDA+CML achieves 1.12% absolute improvement over LDA, and NAP+CML achieves 1.5% absolute improvement over NAP in the C1 condition. The performance of WCCN+CML is comparable to that of WCCN.

The performance of LDA+PLDA back-end is also presented in Tables 2 and 3 although it is not fair to compare CML with the LDA+PLDA back-end since the former only combines with an initialization method. It is seen that LDA+PLDA achieves EERs of 4.11% to 3.44% for the female task, and 5.29% to 4.40% for the male task. Our CML obtained better performance than LDA+PDLA in the C2 to C5 conditions. For example, LDA+CML achieves approximately 13% relative improvement in the C5 condition for the female task.

Figure 2 plots the DET curves produced by LDA, LDA+CML and NULL in the C1 condition on the female speakers. From the figure, one can see that our LDA+CML yields a significantly better DET curve than LDA.

Figure 3 plots the score distributions of LDA and LDA+CML. One can see from this figure that LDA+CML yields a larger between-class distance than LDA while keeping a similar within-class variance, which results in a smaller over-

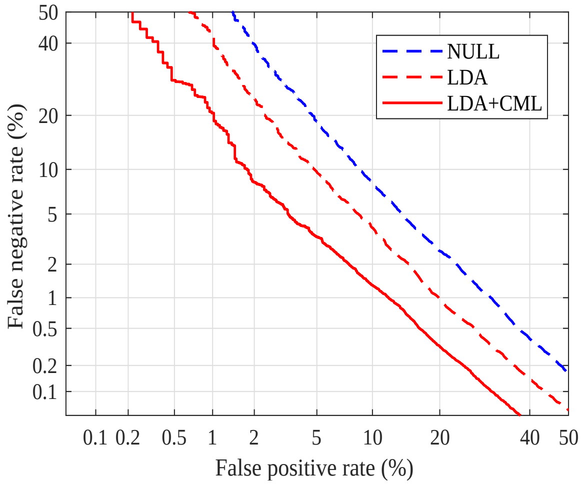

Figure 2: *DET curves produced by LDA, LDA+CML and NULL in the C1 condition on the females.*

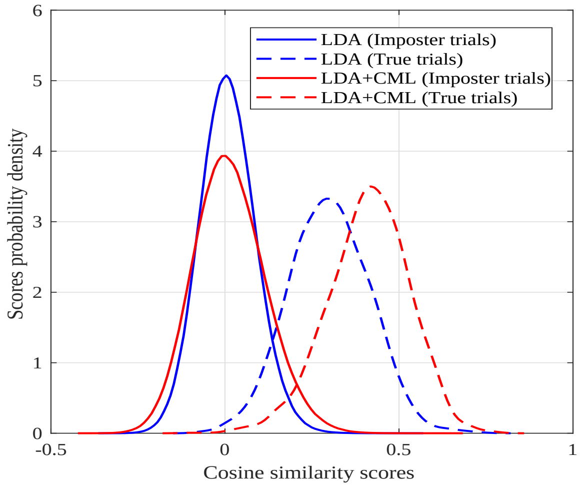

Figure 3: *Score distributions of LDA and LDA+CML in the C1 condition on the females.*

lap region of the decision scores between the true and imposter trials than LDA. As a result, LDA+CML has a smaller EER.

3.4. Effect of the value of β on performance

This experiment investigates the impact of the value of the hyperparameter β on performance and the result for the the female speakers in the C1 condition is plotted in Fig. 4. It is seen from the figure that the EER of LDA+CML first decreases and then increases with the value of β. The underlying reason can be explained as follows. When β is small, the regularization term in (4) does not pay an important role. In this situation, CML does not only increase the between-class distance but also increases the within-class variance as a side effect. The negative effect of using the cost function (2) offsets its positive effect. As a result, the verification performance is not increased. On the other hand, if the value of β is large, CML approaches to its initial point. Therefore, it is important to set β to a proper value to increase the between-class distance while maintaining the within-class variance relatively unchanged. While Fig. 4 was plotted

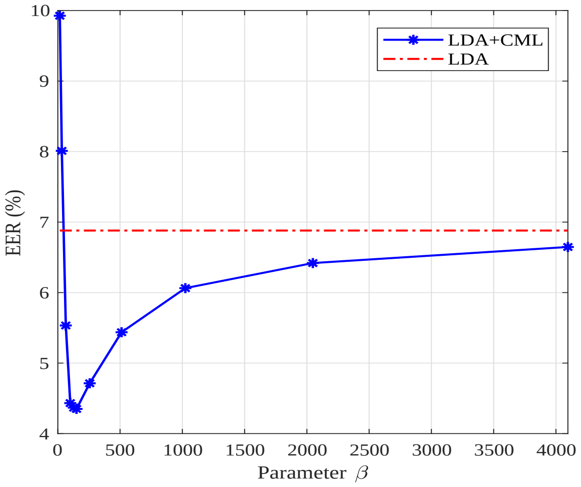

Figure 4: *Performance of LDA+CML with respect to hyperparameter β in the C1 condition on the female speakers.*

for LDA+CML, we observed a similar result for WCCN+CML and NAP+CML, which is not shown here for conciseness.

4. Summary

Motivated by the fact that improving the performance of speaker verification can be transformed to a problem of decreasing the overlap region of the decision scores of true and imposter trials, we presented in this paper a cosine metric learning (CML) framework for speaker verification. CML attempts to minimize the overlap region by increasing the between-class distance of the two Gaussian distributions of the decision scores with a regularization term to control the within-class variance. It can be used to improve any traditional channel or session compensation technique, which is used as an initialization of CML. Experiments on NIST SRE demonstrate that LDA+CML, WCCN+CML, and NAP+CML outperforms, respectively, LDA, WCCN, and NAP. Furthermore, LDA+CML has achieved a competitive performance as the state-of-the-art PLDA back-end.

Work is in progress to find better channel or session compensation matrix A_0, given the fact that CML is lower bounded by the pre-defined matrix A_0. We are also working to extend CML to an end-to-end deep learning method since it is natural to propagate the gradient in (5) to a deep neural network by back-propagation.

5. Acknowledgements

This paper was supported in part by the National Natural Science Foundation of China under grant No. 61671381, and in part by the Shenzhen Science and Technology Plan under grant No. JCYJ20170815161820095.

6. References

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

[2] N. Li and M.-W. Mak, "Snr-invariant plda modeling in nonparametric subspace for robust speaker verification," *IEEE/ACM*

Transactions on Audio, Speech and Language Processing (TASLP), vol. 23, no. 10, pp. 1648–1659, 2015.

[3] T. Hasan and J. H. Hansen, "Maximum likelihood acoustic factor analysis models for robust speaker verification in noise," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 2, pp. 381–391, 2014.

[4] P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet, and J. Alam, "Deep neural networks for extracting baum-welch statistics for speaker recognition," in *Proc. Odyssey*, 2014, pp. 293–298.

[5] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1695–1699.

[6] D. Garcia-Romero, X. Zhang, A. McCree, and D. Povey, "Improving speaker recognition performance in the domain adaptation challenge using deep neural networks," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 378–383.

[7] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4052–4056.

[8] F. Richardson, D. Reynolds, and N. Dehak, "A unified deep neural network for speaker and language recognition," *arXiv preprint arXiv:1504.00923*, 2015.

[9] D. Wang, L. Li, Z. Tang, and T. F. Zheng, "Deep speaker verification: Do we need end to end?" *arXiv preprint arXiv:1706.07859*, 2017.

[10] Y. Anzai, *Pattern recognition and machine learning*. Elsevier, 2012.

[11] A. O. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for svm-based speaker recognition," in *Ninth international conference on spoken language processing*, 2006.

[12] W. M. Campbell, D. E. Sturim, D. A. Reynolds, and A. Solomonoff, "Svm based speaker verification using a gmm supervector kernel and nap variability compensation," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1. IEEE, 2006, pp. I–I.

[13] W. Ahmad, H. Karnick, and R. M. Hegde, "Cosine distance metric learning for speaker verification using large margin nearest neighbor method," in *Pacific Rim Conference on Multimedia*. Springer, 2014, pp. 294–303.

[14] L. Li, D. Wang, C. Xing, and T. F. Zheng, "Max-margin metric learning for speaker recognition," in *Chinese Spoken Language Processing (ISCSLP), 2016 10th International Symposium on*. IEEE, 2016, pp. 1–4.

[15] X. Fang *et al.*, "Bayesian distance metric learning on i-vector for speaker verification," Ph.D. dissertation, Massachusetts Institute of Technology, 2013.

[16] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5115–5119.

[17] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," *arXiv preprint arXiv:1710.10467*, 2017.

[18] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016, pp. 165–170.

[19] H. V. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," in *Asian conference on computer vision*. Springer, 2010, pp. 709–720.

[20] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[21] J. Chang and D. Wang, "Robust speaker recognition based on dnn/i-vectors and speech separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5415–5419.

[22] S. O. Sadjadi, M. Slaney, and L. Heck, "Msr identity toolbox v1. 0: A matlab toolbox for speaker-recognition research," *Speech and Language Processing Technical Committee Newsletter*, vol. 1, no. 4, pp. 1–32, 2013.