LETTER

Deep Nonlinear Metric Learning for Speaker Verification in the I-Vector Space*

Yong FENG[†a)], ***Member***, **Qingyu XIONG**[†], ***and*** **Weiren SHI**[†], ***Nonmembers***

SUMMARY Speaker verification is the task of determining whether two utterances represent the same person. After representing the utterances in the i-vector space, the crucial problem is only how to compute the similarity of two i-vectors. Metric learning has provided a viable solution to this problem. Until now, many metric learning algorithms have been proposed, but they are usually limited to learning a linear transformation. In this paper, we propose a nonlinear metric learning method, which learns an explicit mapping from the original space to an optimal subspace using deep Restricted Boltzmann Machine network. The proposed method is evaluated on the NIST SRE 2008 dataset. Since the proposed method has a deep learning architecture, the evaluation results show superior performance than some state-of-the-art methods.

key words: *speaker verification, restricted Boltzmann machine, nonlinear metric, deep learning*

1. Introduction

Speaker recognition is a form of biometric personal recognition. Usually there are two modes of recognition: verification and identification. Speaker identification aims at identifying who is the speaker while speaker verification focuses on whether the claimed speaker is the true speaker, a yes or no problem. In this paper, only speaker verification is discussed.

Generally, speaker verification consists of three stages: front-end feature extraction, modeling, and back-end scoring of classification. In the front-end feature extraction, MFCC or PLP are usually used. Then some approaches are adopted to model the speaker, such as VQ [1], GMM [2], SVM [3], JFA [4], i-vector [5] et al. In the back-end scoring or classification method, likelihood ratios or SVMs are usually used.

Over the past decades, i-vector model has been the dominant approach for modeling speakers. Since the speaker related information is buried under others, raw i-vectors are not sufficiently discriminative. In order to improve the discriminative capability of i-vectors, various discriminative models have been proposed, including WCCN [6], NAP [7], LDA [8] and PLDA [9]. Among these models, PLDA is regarded as the most effective approach and delivers state-of-art performance. As for the objective function, although PLDA encourages discrimination among speakers, the task of speaker verification is to discriminate true speakers and imposters, which is a binary decision, instead of the multiclass discrimination in PLDA training. As for the Gaussian assumption, it is often over strong and cannot be held in practice, leading to a less representative model.

Metric learning has provided a viable solution for speaker verification by comparing the speech pairs based on the learned metric [10]. The most commonly used metric is Mahalanobis metric. It is equivalent to first applying a linear transformation, then computing Euclidean distance in the new subspace. But in many situations, a linear transformation often fails to give good performance in high-dimensional space, and it is not powerful enough to capture the underlying data manifold. Therefore, we resort to more powerful non-linear transformation. The kernel-based approaches can achieve this goal implicitly.

Recently, deep neural network (DNN) based approaches have been used in many speech processing fields [11]–[14]. Conventionally, bottleneck features are generated by a multi-layer neural network, in which one of the internal layers has a small number of hidden units, relative to the size of other layers [11]. This small layer creates a constriction in the network that forces the information pertinent to classification into a low dimensional representation. Thus, the bottleneck features can be considered as a nonlinear feature transformation and dimensionality reduction technique. Yamada et al. proposed a method using the bottleneck features extracted from DNNs for distant-talking speaker identification [12]. They considered that the bottleneck features can reduce the influence of reverberation and can transform the reverberant speech feature to a new feature space closed to clean speech feature. In [13], a method, which combined the bottleneck feature and a cepstral domain de-noising auto-encoder based de-reverberation, was proposed to improve the speaker identification performance. In [14], an impressive method for i-vector extractor was proposed. It combines the bottleneck feature and DNN posteriors to accumulate multi-model statistics and train the i-vector extractor. All the above methods try to extract frame-level features or middle-level feature (i-vector) from the original acoustics feature by a DNN with a special bottleneck layer and can be regarded as DNN based feature transformation approaches.

With the idea of DNN based feature transformation, we propose a nonlinear metric learning method for speaker

Manuscript received May 16, 2016.
Manuscript revised August 31, 2016.
Manuscript publicized October 4, 2016.

[†]The authors are with School of Automation, Chongqing University, Chongqing, China.

[*]This work is supported by the Natural Science Foundation Project of Chongqing under Grant No. CSTC2016shmszx00013.

a) E-mail: fengyong023@126.com

DOI: 10.1587/transinf.2016EDL8106

verification in the i-vector space by using deep Restricted Boltzmann Machine (RBM) [15]. Different from the methods proposed in [11]–[14], the proposed method tries to extract some high level features from the i-vector space and can be seemed as a classifier for the intended speaker verification task directly.

Specifically, we regard the RBM network as a nonlinear function, which transforms the features from the original space (the i-vector space) to another subspace. And in order to identify good discriminative features, we combine the side information constraints of metric learning with RBM, and stack the RBM networks in deep architecture. We formulate the proposed method as an appropriate optimization problem, and employ discriminative pre-training and fine-tuning methods to get the optimal solution. The proposed method is evaluated on the SRE08 core test set. Results demonstrate superior performance over some state-of-art methods.

This paper is organized as follows. Section 2 introduces the traditional metric learning briefly. Section 3 explains the details of our deep nonlinear metric learning method. Section 4 gives the experiments and result analysis. And the last section has the conclusion.

2. Traditional Supervised Metric Learning

Distance metrics are fundamental concepts in machine learning. The label information in distance metric learning is usually specified in the form of pairwise constraints on the data: (1) equivalence constraints, which state that the given pairs are semantically-similar and should be close in the learned metric; and (2) inequivalence constrains, which indicate that the given pairs are semantically-dissimilar and should not be near in the learned metric [3]. The objective of metric learning is to find a distance metric that keep all the data pairs in the equivalence constraints close while separating those in the in-equivalence constraints. The most representative work is [16], which formulates distance metric learning as a constrained convex programming problem.

Let $\boldsymbol{C} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T\}$ be a collection of data points, where T is the number of samples in the collection. Each $\boldsymbol{x}_i \in \boldsymbol{R}^n$ is a data vector where n is the dimension of features. Let the set of equivalence constraints denoted by:

$$S = \{(\boldsymbol{x}_i, \boldsymbol{x}_j) \mid \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ belong to the same class}\} \quad (1)$$

and the set of in-equivalence constraints denoted by:

$$D = \{(\boldsymbol{x}_i, \boldsymbol{x}_j) \mid \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ belong to different class}\} \quad (2)$$

Let the distance metric denoted by matrix $\boldsymbol{A} \in \boldsymbol{R}^{n\times n}$, and the distance between any two data points $\boldsymbol{x}$ and $\boldsymbol{y}$ expressed by:

$$d_A^2(x, y) = \| x - y \|^2 = (x - y)^T A(x - y) \quad (3)$$

Given the constraints in S and D, [12] formulates the problem of metric learning into the following convex programming problem:

$$\begin{aligned} &\min_{A\in R^{n\times n}} \sum_{(x_i,x_j)\in s} d_A^2(x_i, x_j) \\ &s.t. \quad A \geq 0, \sum_{(x_i,x_j)\in D} d_A^2(x_i, x_j) \geq 1 \end{aligned} \quad (4)$$

The objective term is to make the distance between similar pairs as small as possible. The positive semi-definite constraint $\boldsymbol{A} \succcurlyeq 0$ is needed to ensure the nonnegative distance between any two data points and the triangle inequality. The third term is to make the distance between dissimilar pairs at least larger than 1. A is symmetric positive semi-definite, so it can be decomposed as $\boldsymbol{A} = \boldsymbol{W}^T\boldsymbol{W}$, and in the learned metric, the distance between any two points can be written as:

$$d_A^2(x, y) = (x-y)^T A(x-y) = (Wx-Wy)^T(Wx-Wy) \quad (5)$$

So the traditional metric learning is equivalent to learn a linear transformation matrix $\boldsymbol{W}$, and then compute Euclidean distance in the transformed subspace.

3. Deep Nonlinear Metric Learning with Restricted Boltzmann Machine

Because of the powerful approximate ability of the deep learning architecture to learn functions or distributions, and the virtues brought by the deep architecture, deep learning methods theoretically exhibit powerful learning ability to discover the nature of the dataset. In this work, we use RBM as the basic network, then describe the discriminative training algorithm – Deep Nonlinear Metric Learning with RBM (DNML_RBM), and finally stack the pre-trained RBM for deep metric learning.

3.1 Restricted Boltzmann Machine

RBM is an undirected model, one being visible layer and the other hidden layer (Fig. 1). In graph theory, it can be regarded as a bipartite graph, each edge being attached with a weight, noted as a matrix $\boldsymbol{W}$ [15].

Suppose RBM system has n vertexes in visible layer and m vertexes in hidden layer. Vector $\boldsymbol{h}$ and $\boldsymbol{v}$ stand for the state in hidden layer and visible layer respectively, among which h_i and v_j stand for the state of the ith vertex in hidden layer and the state of the jth vertex in visible layer. Then the energy of the RBM is defined as follows:

$$E(v, h|\theta) = -\sum_{i=1}^{n} a_i v_i - \sum_{j=1}^{m} b_j h_j - \sum_{i=1}^{n}\sum_{j=1}^{m} v_i W_{ij} h_j \quad (6)$$

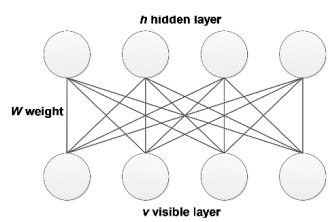

Fig. 1 Restricted Boltzmann Machine (RBM).

The probability that a RBM assigns to a visible vector v is:

$$p(v|\theta) = \frac{1}{\sum_{v,h} e^{-E(v,h|\theta)}} \sum_{h} e^{-E(v,h|\theta)} \tag{7}$$

Since there are no hidden-hidden connections, the activation state of the hidden vertex only depends on the vertex in the visible layer

$$p(h_j = 1|v, \theta) = sigmoid(b_j + \sum_{i=1}^{n} v_i W_{ij}) \tag{8}$$

And vice versa for symmetry:

$$p(v_i = 1|v, \theta) = sigmoid(a_i + \sum_{j=1}^{n} h_j W_{ji}) \tag{9}$$

It can be seen that each layer is related only with the previous layer and the whole procedure can be regarded as layer-wise train. Since the training does not require any label information, the derived feature is considered as unsupervised features.

3.2 Nonlinear Metric Learning with RBM

We regard the RBM network as an explicit nonlinear transformation function: $f(x, W) : R^n \rightarrow R^m$, and we use the side information constraints to get the optimal parameters of the RBM network. In the transformed subspace, we compute distance $d(\overline{x_i}, \overline{x_j})$ between data points, and we assume a logistic regression model when estimating the probability for any two data points (i-vectors) $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ to share the same class (belong to the same person) or be semantically dissimilar, i.e.

$$P(l_{ij}|x_i, x_j) = 1/(1 + \exp(-l_{ij}(d(\overline{x_i}, \overline{x_j}) - \mu))) \tag{10}$$

Where $l_{ij} = 1$ if $(x_i, x_j) \in S$, $l_{ij} = -1$ if $x_i, x_j) \in D$, $\overline{x_i} = f(x_i)$, $\overline{x_j} = f(x_j)$. The parameter $\boldsymbol{\mu}$ is the threshold. Two i-vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ will belong to the same person only when their distance is less than the threshold $\boldsymbol{\mu}$. Then the overall log likelihood for all the equivalence constraints S and the in equivalence constraints D can be written as:

$$\begin{aligned} L(W, \mu) = &- \sum_{(x_i,x_j)\in S} \log(1 + \exp(-d(\overline{x_i}, \overline{x_j}) + \mu)) \\ &- \sum_{(x_i,x_j)\in D} \log(1 + \exp(d(\overline{x_i}, \overline{x_j}) - \mu)) \end{aligned} \tag{11}$$

Using the maximum likelihood estimation, we will cast the problem of distance metric learning into the following optimization problem:

$$\begin{aligned} \min_{W,\mu} E = &- L(W, \mu) + \lambda \sum_{t=1}^{T} \sum_{i=1}^{m} f_i(x^t, W) \\ s.t. \quad &WW^T = I \end{aligned} \tag{12}$$

The first term $L(\boldsymbol{W}, \boldsymbol{\mu})$ is the log likelihood of the side information constraints, which encourage the margin between positive and negative samples to be large. The second term is the mapping function of RBM, which encourages the sparsity of the transformed features. The hard orthonormality constraints is used to prevent degenerate solution of $\boldsymbol{W}$.

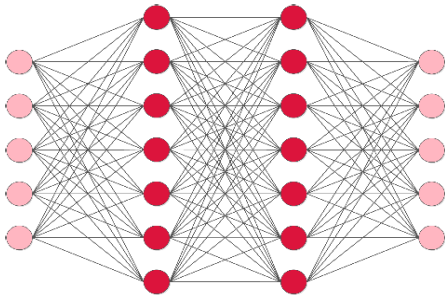

Fig. 2 The deep architecture of stacked RBM.

We adopt gradient descend method for objective optimization, in which the gradient can be computed as:

$$\begin{aligned} \partial E/\partial W_{jq} = &- \sum\nolimits_{(x,y)\in S} (P_S(x, y) - 1) \frac{\partial d(\overline{x}, \overline{y})}{\partial W_{jq}} \\ &- \sum\nolimits_{(x,y)\in D} (P_D(x, y) - 1) \frac{\partial d(\overline{x}, \overline{y})}{\partial W_{jq}} \\ &+ \lambda \sum_{t=1}^{T} \sum_{i=1}^{m} \frac{\partial f_i(x^t, W)}{\partial W_{jq}} \end{aligned} \tag{13}$$

$$\begin{aligned} &\partial E/\partial \mu \\ &= -(\sum\nolimits_{(x,y)\in S} (1 - P_S(x, y)) + \sum\nolimits_{(x,y)\in D} (0 - P_D(x, y))) \end{aligned} \tag{14}$$

After obtaining the gradient, the parameter $\boldsymbol{W}$ and $\boldsymbol{\mu}$ can be updated by following until convergence:

$$W = W - \alpha \frac{\partial E}{\partial W} \tag{15}$$

$$\mu = \mu - \alpha \frac{\partial E}{\partial \mu} \tag{16}$$

3.3 Stacked RBM Network for Deep Metric Learning

In the last section, we describe the RBM network and combine the side information to discriminatively train the RBM network. But the single layer RBM network has limited ability to map the gap between i-vector and class label. So we stack multiple RBM networks to form a deep architecture network. Specifically, the i-vectors are input to the first RBM network, and then the responses of the first RBM network are treated as inputs to the next RBM network, and the output of the last RBM network is regarded as the transformed feature of the original i-vector.

The whole model can be seemed as a stacked RBM network. Similar to other algorithms proposed in the deep learning literature, our stacked RBM model is trained greedily layerwise in the pretraining phase, but we use the discriminative pretraining algorithm. In the fine-tuning phase,

the objective function is similar to (12), but the mapping function is a stacked RBM network. We adopt the conjugate gradients method for objective optimization, and the gradient-computing steps are similar to those steps in last section.

4. Experiments

4.1 Experimental Setup

To evaluate the effectiveness of the proposed method, we perform experiments on the core condition of the telephone speech NIST 2008 speaker recognition evaluation (SRE) list [17]. We use the speaker data from Fisher, SwitchBoard, NIST 2004, 2005 and 2006 datasets to train the UBM, i-vector model and LDA model. The same data are also used to conduct the proposed deep metric learning. We select 1997 female utterances from the core evaluation dataset of NIST 2008 as the test set. And based on the constructed 59343 trials, it includes 12159 target trials and 47184 imposter trials.

We use a UBM containing 2048 Gaussians, operated on a 60-dimensional feature which is formed by 20-dimensional MFCC appended with the first and second order derivatives. The classical Total Variability Model (TVM) based i-vector extractor is referred to as the baseline system. The dimension of i-vector it produces is 800 in our experiments. We apply the channel compensation on i-vector by LDA projection with speaker factor of dimension 256. For metric learning, utterance in the Fisher database are sampled randomly to build the equivalence and inequivalence pairs. And in this study, 800-dimensional ivector feature is used as the input of the proposed Deep Nonlinear Metric Learning with RBM (DNML_RBM). There are 500 hidden units in each hidden layer, and 256 units in the output layer.

4.2 Effectiveness of the Proposed DNML_RBM

In this experiment, we evaluate the effectiveness of our proposed DNML_RBM. The test is based on the NIST SRE 2008 core task, which is divided into 8 test conditions according to the channel, language and accent [17]. Table 1 shows the EER results of our method in the pre-training and fine-tuning phase.

From these results, we can see that the performance improves significantly as the number of layers increase, and the fine-tuning can further improve the performance over the second layer. In addition, the improvement of fine-tuning is near 2% over single layer (first layer). Thus, it proves the effectiveness of the deep learning architecture.

4.3 Performance Comparison between Different Methods

Table 2 gives the fair comparison between the proposed DNML_RBM with Raw_i-vectors (Original i-vector), Raw_RBM (stacked RBMs without metric learning), LDA, PLDA and Cosine Similarity Metric Learning (CSML) [18], which is a typical linear metric learning method. It can be observed that the proposed method significantly improves the discriminative capability than Raw_i-vectors and Raw_RBM. And it also outperforms LDA, PLDA in condition 1-4 (which takes the major proportion of the test data). However, in condition 5-8, PLDA wins the competition. We attribute this discrepancy to the data imbalance in the development set: condition 5-8 involves complex pattern (e.g. multilingual speakers, different accents) that were not involved in the Fisher database that was used to train the models. This leads to performance degradation on these conditions with our proposed method that we found heavily relies on large training data. For LDA and PLDA, the Gaussian assumption improves generalizability on unseen conditions, thus resulting in superior performance than DNML_RBM, a purely discriminative approach. Nevertheless, since condition 1-4 takes a large proportion of the data, the proposed DNML_RBM gets the best overall performance. We also compared the performance of CSML, a typical linear metric learning method, with DNML_RBM, the proposed nonlinear metric learning method. The evaluation results are in the last two columns of Table 2. And we can see that the proposed method DNML_RBM produces about 3.7% relative improvement in EER over CSML.

Table 1 Performance of the proposed method in different phase.

Condition	Original	DNML_RBM		
		First Layer	Second Layer	Fine-tuning
C1	29.34	15.61	14.50	14.75
C2	4.78	3.10	2.05	1.19
C3	29.66	16.21	15.25	15.23
C4	18.92	14.32	12.10	10.91
C5	20.31	15.13	12.27	15.38
C6	12.47	12.10	11.50	16.02
C7	7.73	7.31	6.50	10.77
C8	7.37	6.53	5.56	10.43
All	25.58	17.11	16.28	14.94

Table 2 Performance comparisons between the proposed method and other State-of-the-art methods.

Condition	Raw_ ivector	Raw_ RBM	LDA	PLDA	CSML	DNML _RBM
C1	29.34	27.82	22.11	18.57	15.21	14.75
C2	4.78	4.23	1.19	1.79	1.19	1.19
C3	29.66	26.87	22.65	18.70	15.64	15.23
C4	18.92	17.45	12.91	14.41	12.56	10.91
C5	20.31	19.36	14.42	10.58	15.47	15.38
C6	12.47	12.38	10.75	9.42	16.85	16.02
C7	7.73	7.05	5.58	4.06	11.02	10.77
C8	7.37	6.95	5.52	4.21	10.64	10.43
All	25.58	24.50	20.96	19.13	15.52	14.94

5. Conclusions

In this paper, a Deep Nonlinear Metric Learning approach for speaker recognition is proposed. Unlike tradition linear or kernel based metric learning methods, the proposed approach learns an explicit nonlinear transformation. More specifically, we use the RBM as the basic network and stack multiple RBMs in a deep architecture. With the stacked

RBM networks, every instance can be transformed nonlinearly to a compact vector for effective verification. The proposed method is evaluated on the NIST SRE 2008 core test dataset and achieve better results than some state-of-the-art methods.

References

[1] A. Kabir and S.M.M. Ahsan, "Vector quantization in text dependent automatic speaker recognition using mel-frequency cepstrum coefficient [C]," Proc. 6th WSEAS International Conference on Circuits, Systems, Electronics, Control and Signal Processing, Cairo, Egypt, 2007.

[2] D.A. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models [J]," Speech communication, vol.17, no.1-2, pp.91–108, 1995.

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

[4] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Speaker and session variability in GMM-based speaker verification [J]," IEEE Transactions on Audio, Speech and Language Processing, vol.15, no.4, pp.1448–1460, 2007.

[5] D.N. Dehak, R. Dehak, P. Kenny, et al., "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification [C]," Conference of the International Speech Communication Association, 2009.

[6] A.O. Hatch, S.S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," INTERSPEECH, 2006.

[7] A. Solomonoff, C. Quillen, and W.M. Campbell, "Channel compensation for SVM speaker recognition," Proc. Odyssey, Speaker Language Recognition Workshop 2004, pp.57–62, 2004.

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

[9] S. Ioffe, "Probabilistic linear discriminant analysis," Computer Vision ECCV, Springer Berlin Heidelberg, pp.531–542, 2006.

[10] L. Yang and R. Jin, "Distance metric learning: A comprehensive survey," Michigan State University, May 2006.

[11] D. Yu and M. Seltzer, "Improved Bottleneck Features Using Pretrained Deep Neural Networks," Proc. Interspeech, 2011.

[12] T. Yamada, L. Wang, and A. Kai, "Improvement of distant-talking speaker identification using bottleneck features of DNN," Proc. Interspeech, pp.3361–3364, 2013.

[13] Z. Zhang, L. Wang, A. Kai, T. Yamada, W. Li, and M. Iwahashi, "Deep neural network-based bottleneck feature and denoising autoencoder-based dereverberation for distant-talking speaker identification," Eurasip Journal on Audio, Music and Speech Processing, 2015:12, 2015.

[14] F. Richardson, D. Reynolds, and N. Dehak, "Deep Neural Network Approaches to Speaker and Language Recognition," Signal Processing Letters, IEEE, vol.22, no.10, pp.1671–1675, 2015.

[15] A. Fischer and C. Igel, "An introduction to restricted Boltzmann machines," L. Alvarez et al. (Eds.): CIARP 2012, LNCS 7441, pp.14–36, 2012.

[16] E.P. Xing, A.Y. Ng, M.I. Jordan, et al., "Distance metric learning with application to clustering with side-information [J]," Advances in Neural Information Processing Systems, pp.505–512, 2003.

[17] NIST, The NIST year 2008 speaker recognition evaluation plan, Online: http://www.itl.nist.gov/iad/mig/tests/sre/2008/sre08evalplanrelease4.pdf, 2008.

[18] H.V. Nguyen and L. Bai, "Cosine Similarity Metric Learning for Face Verification [C]," Computer Vision - ACCV 2010, Asian Conference on Computer Vision, Queenstown, New Zealand, pp.709–720, Nov. 2010.