CSLT TECHNICAL REPORT-20150026 [Monday 19th October, 2015]

Binary Speaker Embedding

Lantian Li[1,3], Chao Xing[1], Dong Wang[1*], Kaimin Yu[1] and Thomas Fang Zheng[1]

[*]Correspondence: wang-dong99@mails.tsinghua.edu.cn
[1]Center for Speech and Language Technology, Research Institute of Information Technology, Tsinghua University, ROOM 1-303, BLDG FIT, 100084 Beijing, China
Full list of author information is available at the end of the article

Abstract

The popular i-vector model represents speakers as low-dimensional continuous vectors (i-vectors), and hence is a way of continuous speaker embedding. In this paper, we instigate binary speaker embedding, which transforms i-vectors to binary vectors (codes) by a hash function. We start from local sensitive hashing (LSH), a simple binarization approach where binary codes are derived from a set of random hash functions. A potential problem of LSH is that the randomly sampled hash functions might be suboptimal, we therefore propose an improved hamming distance learning approach, where the hash function is learned by a variable-sized block training that projects each dimension of the original i-vectors to variable-sized binary codes independently.

Our experiments show that binary speaker embedding can deliver competitive or even better results on both speaker verification and identification tasks, while the memory usage and the computation cost are significant reduced.

Keywords: speaker recognition; metric learning; i-vector

1 Introduction

The state-of-art i-vector model for speaker recognition assumes that a speech segment can be represented as a continuous vector (i-vector) in a subspace that involves both speaker and channel variances [1, 2]. Normally the cosine distance is used as the distance measure in this i-vector space. Various discrimination or normalization approaches have been proposed to improve the i-vector model, e.g., linear discriminant analysis (LDA) [3], with-in class covariance normalization (WCCN) [4], probabilistic linear discriminant analysis (PLDA) [5]. We prefer LDA because it is simple and effective, achieving similar performance as the complex PLDA while preserving the simple cosine distance, which is highly important for large-scale applications. In this paper, whenever we mention the i-vector model or i-vectors, we mean i-vectors with LDA employed.

The i-vector model can be regarded as a *continuous* speaker embedding, which projects a complex and high-dimensional structural data (speech signal) to a simple speaker space that is low-dimensional and continuous. Despite the broad success of this approach, there are some potential problems associated with the continuous embedding. Firstly, although i-vectors are quite compact representations of speakers (compared to conventional method based on Gaussian mixture models, or GMMs), memory usage and computation cost are still demanding for large-scale tasks. For example, if the dimensionality of an i-vector is 150 and each dimension is a float (4 bytes), representing one billion people (the number in China) requires 600 GB memory. To search for a people given a reference i-vector, the computation cost involves one billion cosine distance calculation. Note that the computation will be

prohibitive if the model is based on GMMs or the scoring is based on PLDA, which is why we focus on the LDA-projected i-vector model in this paper.

Another potential problem of the continuous speaker embedding, as we conjecture, is the over sensitivity to non-speaker variances. We argue that since the vectors are continuous and can be changed by any small variance in the speech signal, i-vectors tend to be 'over representative' for subtle information that are irrelevant to speakers. LDA can solve part of this problem, but it is the nature of continuous representations that are fragile with corruptions. This resembles to the fact that analog signals tend to be impacted by transmission errors.

In this paper, we propose to use binary speaker embedding to solve the above problem. More specifically, we project i-vectors to binary vectors (codes) on the principle that the cosine distance in the original i-vector space is largely preserved in the new binary space measured by the Hamming distance. The binary embedding leads to significant reduction in storage and computing cost; additionally, since binary vectors are less sensitive to subtle change, we expect more robustness in conditions with noise or channel mismatch.

We start from the simple binary embedding method based on locality sensitive hashing (LSH) [6, 7, 8], and then extend to a Hamming distance learning method [9]. Particularly, we propose a variable-sized block training algorithm that can improve the learning speed and allocate more bits for important dimensions.

One may argue that the binary embedding is a retraction back to the historical one-hot encoding, and binary codes is less representative than continuous vectors unless a very large dimensionality is used. However, our experiments showed that this is not the truth: very compact binary vectors can represent tens of thousands of speakers pretty well, and binary vectors work even better in some circumstances. These observations indicate that binary embedding is not an odd retraction to the one-hot encoding; it is essentially a simple speaker information distillation via hashing.

The rest of this paper is organized as follows. Section 2 describes the related work; Section 3 presents the LSH-based binary embedding, and Section 4 presents the variable-sized block training. The experiments are presented in Section 5, and Section 6 concludes the paper.

2 Related work

Binary embedding has not been fully recognized in the speaker recognition community. The limited research focuses on employing the advantages of binary codes in robustness and fast computing. For example, [10] proposed a time-spectral binary masking approach to improve robustness of speaker recognition in conditions with high interference. The work proposed in [11] is more relevant to our proposal. By their approach, a universal back ground model (UBM) is employed to divide the acoustic space into subregions, and each subregion is populated with a set of Gaussian components. Each acoustic frame is then converted to a binary vector by evaluating the Gaussian components that the frame belongs to, and the frame-level vectors are finally accumulated to produce the segment-level speaker vector. Better robustness compared with the conventional GMM-UBM approach was reported by the authors.

3 Binary speaker embedding with LSH

Let x denote a length-normalized i-vector, and the similarity between i-vectors is measured by the cosine distance. Our goal is to project a continuous vector x to a binary code $h(x)$ of b bits. The LSH approach [6, 7, 8] seeks for a hash function operating on x, such that more similar i-vectors have more chance to coincide after hashing.

We employ a simple LSH approach proposed in [7]. It selects b hash functions $h_r(\cdot)$, each of which simply rounds the output of the product of x with a random hyperplane defined by a random vector r:

$$h_r(x) = \begin{cases} 1 & if\ r^T x \geq 0 \\ 0 & otherwise \end{cases}$$

where r is sampled from a zero-mean multivariate Gaussian $N(0; I)$. It was shown by [12] that the following LSH requirement is satisfied:

$$P[h(x_i) = h(x_j)] = 1 - \frac{1}{\pi}\theta(x_i, x_j)$$

where $\theta(x_i, x_j)$ is the angle between x_i and x_j and is closely related to their cosine distance. Intuitively, this means that similar i-vectors have more chance to be encoded by the same binary vector than dissimilar ones, which just coincides our goal of preserving similarities of i-vectors with the binary codes.

4 Binary embedding with variable-sized block training

A potential problem of the LSH embedding is that x is not necessarily uniformly distributed on the hyper sphere, and so the uniformly sampled hash functions $\{h_r\}$ might be suboptimal. A better approach would be derive the hash function by learning from data. An interesting method of this category is the Hamming distance learning proposed by [9]. This section presents this approach first, and then proposes a variable-sized block training method that can improve training speed and quality.

4.1 Hamming distance learning

The Hamming distance learning approach [9] learns a projection function $f(x; w)$ where x is the input (an i-vector in our case) and w is the model parameter. Once the projection function is learned, the binary code for x is obtained simply by $b(x; w) = sign(f(x; w))$. Different choices of f lead to different learning methods, though the simple linear model $f(x; w) = w^T x$ is chosen in this study. Note that if w is randomly sampled from $N(0; I)$, this approach is equivalent to LSH without any training.

The Hamming distance learning defines a loss function on triplets (x, x^+, x^-), where x is an i-vector of a particular speaker, x^+ is another i-vector of the same speaker derived from a different speech segment, and x^- is the i-vector of an imposter. The goal of Hamming distance learning is to optimize w such that $b(x; w)$ is closer to $b(x^+; w)$ than $b(x^-; w)$ in terms of Hamming distance. Denoting (h, h^+, h^-) as the binary codes obtained by applying $b(x, w)$ to the triplet (x, x^+, x^-), the loss

function of the learning is:

$$l(h, h^+, h^-) = [||h - h^+||_H - ||h - h^-||_H + 1]_+$$

where $||\cdot||_H$ is the Hamming distance, defined as the number of $1's$ in the vector. Adding the loss function and a regularization term, the training objective function with respect to w is defined as follows:

$$L(w) = \sum_{(x,x^+,x^-)\in D} l(b(x;w), b(x^+;w), b(x^-;w)) + \frac{\lambda}{2}||w||^2$$

where $D = (x_i, x_i^+, x_i^-)_{i=1}^n$ denotes the training samples, and λ is a factor to scale the contribution of the regularization term. Note that this approach has been employed to image retrieval in [9], though in this paper we use it for speaker recognition.

4.2 Variable-sized block training

A particular problem of the Hamming distance learning is the high computation demand if the dimensions of the continuous and/or binary vector are large. Additionally, the learning algorithm treats each dimension of the input continuous vector equally, which is not optimal for the LDA-projected i-vectors for which the low dimensions involve more discriminative information. We propose a variable-sized blocking training approach to solve this problem.

Considering that the expected number of bits of the binary codes is b, we hope these bits are distributed to the dimensions of the original i-vectors unequally, subjected to the constraints $\sum_{i=1}^{D} T_i = b$ where T_i is the number of bits allocated to dimension i, which is linearly descended according to the dimension:

$$T_i = \frac{D + 1 - i}{D} T_1$$

This leads to $T_i = \frac{2b(D+1-i)}{D(D+1)}$, and the ceil value $T_i = \lceil \frac{2b(D+1-i)}{D(D+1)} \rceil$ is selected as the number of encoding bits for the i-th dimension.

Specifically, the variable-sized block training first defines the number of bits T_i, and then the Hamming distance learning is employed to learn the projection matrix w_i, which in turn is used to embed the i-th dimension of the i-vectors to binary codes. Since the learning and embedding for every dimension i is independent, this in fact leads to a block diagonal parameter matrix w (so the block training is named):

$$w = \begin{pmatrix} w_1 & 0 & 0 & \cdots & 0 \\ 0 & w_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & \cdots & w_D \end{pmatrix}.$$

Note that this block training learns each dimension independently so is faster than the conventional Hamming distance learning where the projection matrix w is learned as a whole. Additionally, because more bits are allocated for the low dimensions, the resultant binary codes are more representative and discriminative.

Condition	Description
C1	interview speech in training and test
C2	interview speech from the same microphone type in training and test
C3	interview speech from different microphones types in training and test
C4	interview training speech and telephone test speech
C5	telephone training speech and noninterview microphone test speech
C6	telephone speech in training and test
C7	English language telephone speech in training and test
C8	English language telephone speech spoken by a native U.S. English speaker in training and test

Table 1 NIST SRE2008 test conditions [13].

5 Experiments

In our experiments, both speaker verification and identification tasks are chosen to evaluate the proposed binary speaker embedding. We first present the data and settings used in the experiments, and then report the results on the verification and identification tasks respectively.

5.1 Data and experimental setup

The Fisher database was used to train the i-vector system. We selected 7196 speakers to train the i-vector model and the LDA model. The NIST SRE 2008 database [13] was used for testing. We selected 1997 female utterances from the core evaluation data set and constructed 59343 trials, including 12159 target trials and 47184 imposter trials. The NIST SRE 2008 test conditions are reproduced in Table 1.

The acoustic feature used is 12-dimensional Mel frequency cepstral coefficients (MFCCs) together with the log energy. The first and second order derivatives are augmented to the static features, resulting in 39-dimensional feature vectors. The UBM involves 2048 Gaussian components and was trained with about 4000 female utterances selected from the Fisher database randomly. The i-vector system was trained with all the female utterances in the Fisher database, and the dimension of the i-vectors is 400. The LDA model was trained with utterances of 7196 female speakers, again randomly selected from the Fisher database. The dimension of the LDA projection space is set to 150. For the variable-sized block training, we selected 5000 trials (5000 i-vector pairs) from the SRE08 core evaluation data set (short2-short3) randomly.

5.2 Speaker verification task

The first experiment investigates the performance of binary speaker embedding on the speaker verification task. All the i-vector have been transformed by LDA, and the dimensionality is 150. The performance is evaluated in terms of equal error rate (EER), and the results are shown in Table 2 for the LSH approach, and Table 3 for the variable-sized block training. In each table, the performance with binary codes (denoted by 'b-vector') of various sizes are reported. Note that for variable-sized block training, the bits are not precisely equally to the pre-defined values due to the ceiling operation when determining T_i. We didn't report the time cost in this experiment since the computation is not a serious problem in speaker verification, although binary vectors are certainly faster.

From the results in Table 2 and Table 3, it can be observed that binary vectors can achieve performance comparable to the conventional i-vectors, in spite of the

	i-vector	b-vector			
Bits	4800	150	300	600	900
C1	**22.11**	27.28	26.63	24.67	22.38
C2	**1.19**	7.46	4.18	3.28	2.98
C3	**22.65**	28.30	26.89	25.71	22.84
C4	**12.91**	27.18	21.62	19.82	16.37
C5	**14.42**	24.52	21.39	19.47	16.23
C6	**10.75**	16.41	14.75	12.69	12.47
C7	**5.58**	11.66	9.89	7.48	7.48
C8	**5.26**	11.32	10.26	7.11	6.58
Overall	**20.96**	23.51	22.82	22.19	21.40

Table 2 EER% with LSH-based binary embedding.

	i-vector	b-vector			
Bits	4800	150	375	675	975
C1	22.11	28.06	20.92	20.17	**20.05**
C2	**1.19**	5.37	2.39	2.09	1.79
C3	22.65	28.41	21.54	20.64	**20.70**
C4	**12.91**	22.07	15.32	14.26	13.51
C5	**14.42**	24.28	17.55	15.75	15.75
C6	**10.75**	17.02	13.14	13.08	13.03
C7	**5.58**	11.03	7.86	7.61	7.35
C8	**5.26**	11.32	7.37	6.84	6.84
Overall	20.96	24.15	19.66	19.41	**19.19**

Table 3 EER% with variable-sized block training.

much smaller vector size. For example, with the largest binary codes, the number of bits is only one fifth of that of the original i-vectors. When compared the two binary embedding methods, it is clear that the variable-sized block training performs better consistently. In condition 1 and 3, the binary codes derived by the variable-sized block training work even better than the i-vectors. Note that the conditions where the binary codes perform better than i-vectors are all with the microphone channel, which is different from the condition of the training data (Fisher database) that are all recorded by telephone. This seems support our conjecture that binary codes are more robust to speaker-irrelevant variations.

5.3 Speaker identification task

The advantage of the binary embedding is more evident on the speaker identification task, where heavy computation is required when computing the k-nearest candidates of a given speaker vector. We use the 1997 female speakers as the speaker set, and the 12159 target trials as the speaker correspondence set V. For each trial $(x, y) \in V$, x and y are speaker vectors of two utterances spoken by the same person. In speaker identification, given a test utterance whose speaker vector is x, the task is to search for the k-nearest speaker vectors around x. If a vector y in the k-nearest candidates and (x, y) is in the speaker correspondence set V, then a top-k hit is obtained. We evaluate the performance of speaker identification by the top-k accuracy, which is defined as the proportion of the top-k hits in all the trials. Note that we use only a naive k-nearest search which calculates the distance of the test vector to all the speaker vectors and select the k-nearest candidates. In fact, various methods can be employed to improve efficiency of the search in particular for binary codes, e.g., the PLEB algorithm [14, 7]. We focus on computation cost of the basic algorithm in this paper.

The Top-k accuracy with the two binary embedding approaches are reported in Table 4 and Table 5 respectively. For comparison, the bits of the vectors and the

	i-vector	b-vector		
Bits	4800	150	450	900
Top-20	**50.53**	35.64	43.75	47.75
Top-100	**66.67**	55.58	61.32	65.14
Top-200	**74.32**	66.44	69.94	73.42
Speed up	×1	×50	×20	×11

Table 4 Top-k accuracy (Acc%) with binary embedding based on LSH.

	i-vector	b-vector		
Bits	4800	150	525	975
Top-20	50.53	39.55	51.28	**51.72**
Top-100	66.67	58.22	69.31	**69.67**
Top-200	74.32	67.84	77.02	**77.33**
Speed up	×1	×50	×18	×10

Table 5 Top-k accuracy (Acc%) with binary embedding based on variable-sized block training.

computation cost (relative to the i-vector system) are also reported. From these results, we observe that binary vectors can approach to the performance of the conventional i-vectors with much fewer bits and much faster computation. In particular with the variable-sized block training, binary vectors even outperform i-vectors on all the Top-k metrics. Note that the Top-k accuracies are relative lower than a typical SID system, because we didn't optimize the classification model with respect to the speaker set and therefore the results are actually for an open-set system where new speakers can be added freely. Nevertheless, the results we obtained in Table 5 clearly demonstrate that the binary embedding performs faster and better than the conventional continuous embedding, and thus is highly suitable for large-scale SID tasks, e.g., in national-wide criminal search.

6 Conclusions

This paper investigated the binary embedding approach for speaker recognition. We studied two binarization approaches, one is based on LSH and the other is based on Hamming distance learning. Our experiments on both speaker verification and identification tasks show that binary speaker vectors can deliver competitive results with smaller vectors and less computation compared to the conventional i-vectors. This is particularly true with the proposed variable-sized block training algorithm, an extension of the conventional Hamming distance learning method.

Although it has not completely beat the continuous i-vectors, the binary speaker embedding proposed in this paper is still very promising. Further work will study more powerful methods to learn the hash function, and investigate the methods to learn binary vectors from speech signals directly.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant No. 61371136 and No. 61271389, it was also supported by the National Basic Research Program (973 Program) of China under Grant No. 2013CB329302.

Author details
[1]Center for Speech and Language Technology, Research Institute of Information Technology, Tsinghua University, ROOM 1-303, BLDG FIT, 100084 Beijing, China. [2]Center for Speech and Language Technologies, Division of Technical Innovation and Development, Tsinghua National Laboratory for Information Science and Technology, ROOM 1-303, BLDG FIT, 100084 Beijing, China. [3]Department of Computer Science and Technology, Tsinghua University, ROOM 1-303, BLDG FIT, 100084 Beijing, China.

References

1. P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, pp. 1435–1447, 2007.
2. P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Speaker and session variability in gmm-based speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, pp. 1448–1460, 2007.
3. N. Dehak, P. 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.
4. Andrew O. Hatch, Sachin Kajarekar, and Andreas Stolcke, "Within-class covariance normalization for svm-based speaker recognition," *Spoken Lang. Process*, 2006.
5. S. Ioffe, "Probabilistic linear discriminant analysis," *Computer Vision ECCV 2006, Springer Berlin Heidelberg*, pp. 531–542, 2006.
6. Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al., "Similarity search in high dimensions via hashing," in *VLDB*, 1999, vol. 99, pp. 518–529.
7. Moses S Charikar, "Similarity estimation techniques from rounding algorithms," in *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*. ACM, 2002, pp. 380–388.
8. Alexandr Andoni and Piotr Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*. IEEE, 2006, pp. 459–468.
9. Mohammad Norouziy, David J. Fleety, and Ruslan Salakhutdinovy, "Hamming distance metric learning," *NIPS*, 2012.
10. Yang Shao and DeLiang Wang, "Robust speaker recognition using binary time-frequency masks," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*. IEEE, 2006, vol. 1, pp. I–I.
11. Jean-Francois Bonastre, Pierre-Michel Bousquet, Driss Matrouf, and Xavier Anguera, "Discriminant binary data representation for speaker recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5284–5287.
12. Michel X Goemans and David P Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *Journal of the ACM (JACM)*, vol. 42, no. 6, pp. 1115–1145, 1995.
13. NIST, "The nist year 2008 speaker recognition evaluation plan," *Online: http://www.itl.nist.gov/iad/mig/tests/sre/2008/sre08_evalplan_release4.pdf*, 2008.
14. Piotr Indyk and Rajeev Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. ACM, 1998, pp. 604–613.