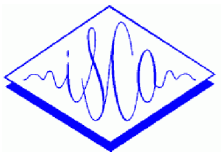
ISCA Archive
http://www.isca-speech.org/archive



INTERSPEECH 2012
ISCA's 13th Annual Conference
Portland, OR, USA
September 9-13, 2012

Exemplar-Based Sparse Representation for Language Recognition on I-Vectors

Bing Jiang, Yan Song, Wu Guo, LiRong Dai

Department of Electronic Engineering and Information Science,
University of Science and Technology of China, Hefei, Anhui, China
bing2010@mail.ustc.edu.cn, {songy, guowu, lrdai}@ustc.edu.cn

10.21437/Interspeech.2012-549

Abstract

In this paper, a new automatic language identification method using sparse representation on i-vectors in low-dimensional total variability space is proposed. It is mainly based on the recently proposed i-vector based language recognition systems. In our proposed method, an over-complete dictionary is first constructed by randomly sampling of the low-dimensional total variability space after Within-Class Covariance Normalization (WCCN) and Linear Discriminate Analysis (LDA). And then for each test sample, the classification score is derived from sparse linear representation with respect to the over-complete dictionary. Furthermore, a random subspace method, which combines different sparse representation classifiers, is introduced to address the possible over-fitting issue. Evaluations on NIST LRE 2007 dataset show that the proposed method outperforms the *state-of-the-art* i-vector based language recognition system. Especially for 30s test condition, our proposed method achieves relative reduction of 29.6% on Equal Error Rate (EER) compared with the baseline system.

Index Terms: language recognition, ivector, sparse representation

1. Introduction

Recently, sparse representation of signals has emerged as a major research area in statistical signal processing. It is a algorithmic problem of computing the sparse linear combinations with respect to an over-complete dictionary composed of the base elements [1]. According to recent development of sparse representation research, the resulting optimization problem can be formed as the l^1-norm penalization over the coefficients in the linear combination.

Sparse representation can be broadly categorized as the exemplar-based techniques, e.g. *K*-nearest neighbors (KNNs), which exploit the information from individual training sample to estimate the classification score. In contrast, the parametric methods generally pool the information about all training examples together to estimate the model parameters. Exemplar-based sparse representation has been shown to outperform the GMMs for several classification tasks in terms of accuracy [2, 3]. In [4], Naseem *et.al*. introduce a classifier based on sparse representation using the GMM mean supervector for speaker recognition. The experimental results on TIMIT database show that the sparse representation classifier performs better than baseline GMM-SVM system. In [5], Li *et.al*. employ the sparse representation to model the i-vectors in low-dimensional total variability space after performing the WCCN and LDA instead of the high-dimensional GMM supervectors.

In this paper, we mainly focus on exemplar-based sparse representation using i-vectors in low-dimensional total variability space for language identification tasks. Specifically, we first construct the over-complete dictionary using random sampling scheme. And then for each test sample, the sparse linear combination via l^1-norm with respect to the over-complete dictionary is calculated, and the classification score is further derived for making the final decision.

In addition, we propose three methods to improve the robustness and performance of our language identification system. First, we experiment with different dictionary construction methods, such as 1) using all i-vectors samples, 2) *K*-means clustering methods and 3) random sampling of i-vector belonging to each language. Second, we empirically evaluate the relationship between the sparsity of the representation coefficients and the final language identification performance, which can be further used to tune the tolerance parameter in sparse representation. Finally, we propose to use the random subspace method to combine multiple sparse representation classifiers for robustness of the estimation, and prevent the over-fitting issue of the sparse representation. To evaluate the effectiveness of the proposed method, we conduct extensive experiments on NIST LRE 2007 dataset. The experimental results show the advantage of the proposed method in terms of EER. Especially, the performance of our on 30s test condition achieves reduction of 29.6% on EER compared with the *state-of-the-art* i-vector based system.

The rest of this paper is organized as follows: In Section 2, we introduce a basic general framework for classification using sparse representation. Section 3 shows how to apply this general classification framework in language recognition focusing on three fundamental issues, sample representation, dictionary construction and scoring, and then propose a random subspace method to further improve performance. We describe our experimental setup in Section 4 and discuss the experimental results in Section 5. The conclusions and the future work are introduced in Section 6.

2. Sparse Representation Classifier

In our sparse classification, we exploit the discriminative nature of sparse representation to perform classification. As in [1], we assume a test sample $\boldsymbol{w}$ can be represented as a linear combination in a given dictionary $\boldsymbol{D}$:

$$\boldsymbol{w} = \sum_{i=1}^{n} \boldsymbol{d}_i \beta_i = \boldsymbol{D}\boldsymbol{\beta} \tag{1}$$

where $\boldsymbol{d}_i$ is a base element of the dictionary $\boldsymbol{D} \in \mathbb{R}^{m \times n}$, and $\boldsymbol{\beta} = [\beta_1, \beta_2, \cdots, \beta_n] \in \mathbb{R}^n$ is a coefficient vector. In sparse representation, the dictionary $\boldsymbol{D}$ needs to be over-complete, that is the system of equation (1) is required to be under-determined,

i.e. $m < n$. It has been shown that $\boldsymbol{\beta}$ can be obtained by searching the sparest solution as follows[6]:

$$\boldsymbol{\beta}^* = \arg\min \|\boldsymbol{\beta}\|_0 \quad \text{s.t.} \quad \boldsymbol{D}\boldsymbol{\beta} = \boldsymbol{w} \tag{2}$$

It is obvious that the above problem is l^0 optimization problem, which is known as non-convex and NP-hard. Fortunately, in optimization field the solution of the l^0 minimization problem is approximately equal to the solution of the following l^1 minimization problem [7]:

$$\boldsymbol{\beta}^* = \arg\min \|\boldsymbol{\beta}\|_1 \quad \text{s.t.} \quad \|\boldsymbol{D}\boldsymbol{\beta} - \boldsymbol{w}\|_2 < \varepsilon \tag{3}$$

There are some techniques to solve this problem. In this paper, Lasso-based method [8] is used to find the solution. For each test sample, the sparse representation $\boldsymbol{\beta}$ can be derived using each element in the training set. And the corresponding labels can be directly used to make the classification decision. This is the major difference between the sparse representation classifier and other exemplar-based methods, such as KNN and SVM.

3. Language Recognition using Sparse Representation

In this section, we will first introduce three modules in our proposed sparse representation framework for language recognition. And then we present the random subspace method to further improve the performance based on this framework.

3.1. Language Recognition via Sparse Representation

Our proposed language recognition framework consists of three modules: 1) sample representation, 2) dictionary construction, and 3) classification scoring, which will be detailed as follows.

3.1.1. Sample Representation

As aforementioned, the sample dimension m should be smaller than the number of dictionary element n. In addition, the high dimension will lead to heavy memory demands and computation cost, which make the sparse representation classifier based on supervector infeasible for language identification tasks. In this work, the i-vector which is in a low-dimensional total variability space is proposed for sparse representation system.

I-vector was first introduced in speaker recognition [9]. Sample representations based on i-vectors have achieved the *state-of-the-arts* performance in speaker and language recognition [10]. Unlike Joint Factor Analysis (JFA) which models separately two distinct spaces, namely Eigen-Voice space and Eigen-Channel space, i-vector is an effective representation which aims at capturing both speaker and channel variabilities in GMM supervectors. Given an utterance, the speaker- and channel-dependent GMM supervector M can be modeled as follows

$$\boldsymbol{M} = \boldsymbol{m} + \boldsymbol{T}\boldsymbol{w} \tag{4}$$

where $\boldsymbol{m}$ is the UBM supervector which is trained on all available data. $\boldsymbol{T}$ is a low rank rectangular matrix, termed as total variability matrix. And $\boldsymbol{w}$ is the i-vector with standard normal distribution $N(0, \boldsymbol{I})$.

3.1.2. Dictionary construction

Given i-vector of each training utterance, the dictionary H for sparse representation can be constructed by concatenating L sub-dictionaries according to the training labels.

$$\boldsymbol{H} = [\boldsymbol{H}_1, \boldsymbol{H}_2, \cdots, \boldsymbol{H}_L] \tag{5}$$

where L is the number of target languages, $\boldsymbol{H}_i$ is the i-th language sub-dictionary which is constructed from the samples in language i.

It is straightforward to construct the dictionary with all the training samples as shown [1]. However, it is still computational costly to solve the equation (3) with large dictionary. To address this issue, we further experiment with another two dictionary construction methods, i.e. K-means and random sampling.

Perhaps K-means algorithm [12] is the most common clustering method for dictionary construction. Given the n_i samples of i-th language, we can train the corresponding sub-dictionary $\boldsymbol{H}_i$ with k centroids as follows:

$$\boldsymbol{\mu}_{i,k} = \frac{\sum_{j=1}^{n_i} r_{k,j} \boldsymbol{w}_{i,j}}{\sum_{j=1}^{n_i} r_{k,j}} \tag{6}$$

where $\boldsymbol{w}_{i,j}$ is the jth training sample of the i-th language. $\boldsymbol{\mu}_{i,k}$ is the kth center of the ith language. $r_{k,n}$ is the binary indicator variable $r_{k,n} \in \{0,1\}$. If $\boldsymbol{w}_{i,j}$ is assigned to cluster k then $r_{k,j} = 1$, otherwise $r_{k,j} = 0$. In this case, the sub-dictionary $\boldsymbol{H}_i$ is defined as $\boldsymbol{H}_i = [\boldsymbol{\mu}_{i,1}, \boldsymbol{\mu}_{i,2}, \cdots, \boldsymbol{\mu}_{i,K}] \in \mathbb{R}^{m \times K}$, and the final dictionary is then formed as $\boldsymbol{H} = [\boldsymbol{H}_1, \boldsymbol{H}_2, \cdots, \boldsymbol{H}_L] \in \mathbb{R}^{m \times LK}$.

Random sampling is another simple and effective dictionary construction method, especially for large dataset. For dictionary construction, we random select M samples from language to construct the sub-dictionary $\boldsymbol{H}_i$. And then the dictionary $\boldsymbol{H}$ is constructed by concatenating all sub-dictionaries as a $d \times LM$ matrix.

3.1.3. Scoring

After solving the l^1 problem, the final classification score is needed to be derived from the solution coefficients. For each language i, let δ_i: $\mathbb{R}^n \longrightarrow \mathbb{R}^n$ be the characteristic function that selects the coefficients corresponding to language i. $\delta_i(\boldsymbol{\beta}) \in \mathbb{R}^n$ is a new vector, in which the entries of the other languages are set to zero and only the nonzero coefficients in associated language are considered. With the new language specific coefficient vector δ_i, i=1,..,L, two scoring schemes are proposed, 1) Summation Scoring Scheme and 2) Residual Scoring Scheme.

Summation Scoring

The summation scoring scheme is derived directly from the language specific coefficient vector for each language. The scores can be calculated as follows

$$S(\boldsymbol{w}) = [sum(\delta_1(\boldsymbol{\beta})), sum(\delta_2(\boldsymbol{\beta})), \cdots sum(\delta_L(\boldsymbol{\beta}))] \tag{7}$$

where S($\boldsymbol{w}$) is used as the score vector for making the final decision.

Residual Scoring

Inspired by [1], we can also derive the classification score according to the reconstruction residuals. With the coefficient vector $\delta_i(\boldsymbol{\beta})$, the test sample $\boldsymbol{w}$ can reconstructed as $\boldsymbol{w}_i' = \boldsymbol{H}\delta_i(\boldsymbol{\beta})$. And the residual according to i-th sub-dictionary can be defined as

$$s_i(\boldsymbol{w}) = 1 - r_i(\boldsymbol{w}) \tag{8}$$

$$r_i(\boldsymbol{w}) = \|\boldsymbol{w} - \boldsymbol{H}\delta_i(\boldsymbol{\beta})\|_2 \tag{9}$$

where $s_i(\mathbf{w})$ is score for each language. and residual derived from the sparse representation with respect to $\delta_i(\boldsymbol{\beta})$, $r_i(\mathbf{w}) \in [0, 1]$. The score vector $S(\mathbf{w})=[s_1(\mathbf{w}),\ldots,s_L(\mathbf{w})]$ is further normalized using

$$\tilde{S}(\boldsymbol{w}) = \frac{S(\boldsymbol{w})}{\|S(\boldsymbol{w})\|_2} \tag{10}$$

3.2. Random Subspace Method

In this section, we will present a novel language identification method based on the random subspace method under the sparse representation based classification framework.

In the proposed random subspace method, the basic classifier is sparse representation classifier based on the randomly constructed dictionary. It is mainly motivated from observations that when using the random sampling to construct the dictionary, the system performance depends on the dictionary. To further improve system performance, the combination scheme is as follows.

1. Repeat for $j = 1,2,\ldots,p$:
   a) Random select M samples from every language training set respectively to construct all L sub-dictionaries, then the j-th dictionary is $\boldsymbol{H}^j$;
   b) Construct a sparse classifier using the dictionary $\boldsymbol{H}^j$ and obtain the decision score $\boldsymbol{S}^j$;
2. Combine decision scores $\boldsymbol{S}^j$, $j = 1,2,\ldots,p$, by a liner fusion

$$\boldsymbol{S}^* = \frac{1}{p}\sum_{j=1}^{p}\boldsymbol{S}^j \tag{11}$$

The $\boldsymbol{S}^*$ is the final score that we use to make decision. It is similar as the random subspace methods proposed by Ho in [13]. However, in [13], each classifier is constructed from the randomly selected feature subspace. Our method is based on the classifier with different dictionary.

4. Experimental setup

The training dataset in our proposed system is mainly collected from the conversational telephone speech (CTS). The evaluation dataset we used is achieved from NIST LRE2007 which contains 14 target languages. There are 7530 evaluation segments for 30s, 10s and 3s including the *out-of–set* data. We focus on the closed set problem. The total variability space is based on an UBM comprised by 1024 Gaussian components, whose i-vector dimension's value is 600. The process of training the total variability matrix $\boldsymbol{T}$ is exactly similar as the one in speaker recognition, and i-vector $\boldsymbol{w}$ of each utterance is defined by its posterior distribution conditioned to the Baum-Welch statistics for a given utterance [9].

After obtaining i-vector $\boldsymbol{w}$, two channel compensation techniques, LDA and WCCN, are further applied to improve performance. For convenience, the i-vector is normalized as follows:

$$\boldsymbol{w}^* = \frac{\boldsymbol{B}^T\boldsymbol{A}^T\boldsymbol{w}}{\left\|\boldsymbol{B}^T\boldsymbol{A}^T\boldsymbol{w}\right\|} \tag{12}$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ are the LDA projection matrix and the WCCN projection matrix respectively [10]. It is important to note that the dimension of i-vector is decreased from hundreds to L-1 after using LDA. In this case, the system of equations $\boldsymbol{A}^T\boldsymbol{w}=\boldsymbol{A}^T\boldsymbol{H}\boldsymbol{\beta} \in \mathbb{R}^{L-1}$ is more underdetermined than $\boldsymbol{w}=\boldsymbol{H}\boldsymbol{\beta} \in \mathbb{R}^m$ in the unknown $\boldsymbol{\beta} \in \mathbb{R}^n$ with $L\text{-}1 \ll m \ll n$. Then the solution is sparser and the solving speed is much faster. In our experiments, we will analyze the effects of dimension reduction on solving l^1 minimization problem (5) and the performance.

In this paper, our baseline systems are SVMs with cosine kernel [10] and Cosine Distance Scoring [1]. EER is the main performance measurement. All results we reported don't have backend.

5. Results and Analysis

In this section, we present experiment results and analysis on LRE2007. We will analyze the effect of LDA and WCCN on the sparse representation first. And then we will demonstrate the effectiveness of the proposed dictionary construction methods and the random subspace method and compare its performance using various dictionary sizes. Finally, we explore the impact of sparsity.

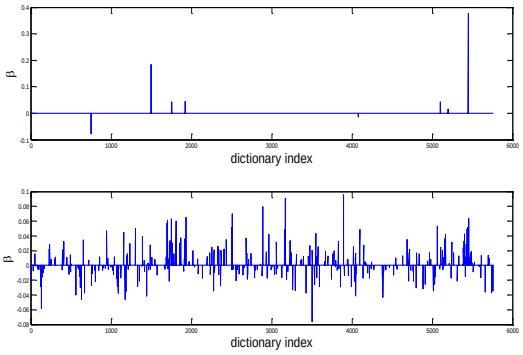


Figure 1: *The sparse solution comparison of a test trial with LDA (above) and WCCN respectively.*

Table 1: *EER (%) comparison of performance for 30s evaluation.*

Methods	SVM	Cosine	Sparse representation	
			Summation score	Residual score
Original	6.81	8.53	**5.97**	23.04
WCCN	6.49	7.55	**5.83**	15.76
LDA	4.63	4.96	**4.22**	4.45
LDA +WCCN	4.63	4.96	**4.22**	4.45

Table 1 shows the comparison of performance of our proposed SR system with the SVM and Cosine Distance Scoring (CDS) i-vector systems using different compensation techniques for 30s test. In this experiment, the dictionary $\boldsymbol{H}$ consists of all training data samples. There are some interesting results in Table 1. Firstly, our proposed system consistently outperforms the baseline system in all cases, which validates the effectiveness of the proposed method. Secondly, the performance from the summation score is better than that from the residual score. It is clear that the configuration for the best performance of our proposed system is also the combination of these two techniques. In addition, we find that adding WCCN does not help a lot in all listed systems after LDA. Fig. 1 shows the sparse solutions of one test sample after using the WCCN and LDA techniques respectively.It is obvious that more undetermined the equations a re, the sparser the solutions under the same condition are. In our subsequent experiments, we will use the combination of LDA and WCCN and adopt the summation score.

Fig.2 shows the comparison of the three dictionary construction methods and the random subspace method with different size of dictionary. The EER of random sampling approach is the average value of 20 random rounds. In this figure, we can easily find that when the dictionary size is big enough, with the increase of the dictionary number, the random sampling

performance curve is converged to the performance curve using all training data. This explains that we don't need to use all training samples to construct the dictionary. We also find that the *K*-means method and the random subspace method can further improve the performance. As well as the random sampling, the performance of *K*-means also does not change obviously with different *K* if *K* is enough large. The random subspace method can lead to the best performance, which EER decreases from 4.40% to 3.57% in the size of 120. For the comparison, the *p* of the random subspace method is also 20. Because the total number of training samples is not infinite, some sub-dictionaries of random selection and *K*-means may be the same as that using all training data. So the curves of these three methods, namely K-means, random selection and random subspace, trend to converge to that of using all training data.

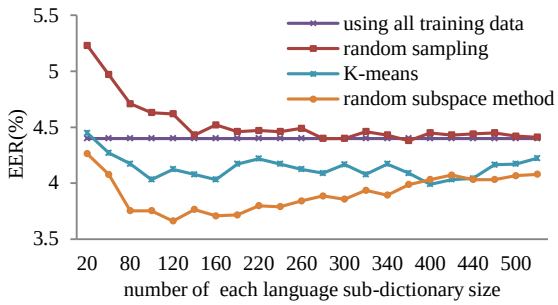


Figure 2: *The EER comparison of performance across different dictionary sizes for 30s evaluation.*

Furthermore, we evaluate the relationship between the solution sparsity and the system performance. Here the solution sparsity ratio is defined as follows

$$\rho = 1 - \frac{\|\beta\|_0}{n} \tag{13}$$

where n is the number of elements in the dictionary. Fig. 3 displays the relationship between EER and sparsity ρ in 30s test when the size of each language dictionary is 120. We use tolerance ε to control the ratio ρ, the ratio grows as ε increases. To a certain extent, the EER performance will be improved with the increase of solution sparsity ratio ρ.

Table 2: *EER (%) comparison of the final sparse representation system with two baseline systems.*

	30s	10s	3s
SVM	4.63	11.17	22.57
Cosine	4.96	11.49	23.12
SR	**3.57**	**10.51**	**22.43**

Finally, Table 2 shows the final comparison of performance in all 30s, 10s and 3s evaluations of LRE 2007. Our proposed system yielded 3.57%, 10.51% and 22.43%, while the baseline system yielded 4.63%, 11.17%, and 22.57%, respectively in 30s, 10s and 3s. Especially, our performance on 30s test condition achieves reduction of 29.6% on EER compared with the baseline.

6. Conclusion

In this paper, we proposed exemplar-based sparse representation for language recognition on i-vectors. Results obtained are consistent better than the baseline systems without backend. The proposed language identification system based on random subspace method can achieve better performance than the other ones. In addition, the random subspace method can further improve the performance. We also discuss the sparsity effects on the performance.

In the future, we intend to continue study the process of score and design an effective backend to further improve the performance.

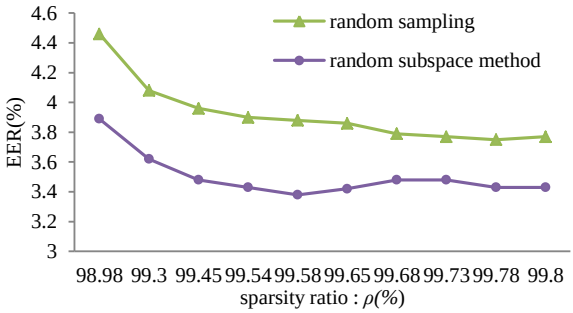


Figure 3: *The EER to sparsity ratio mapping.*

7. References

[1] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust Face Recognition via Sparse Representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.31, no.2, 2009, pp. 210-227.

[2] J. F. Gemmeke and T. Virtanen, "Noise robust exemplar-based connected digit recognition", in *Proc. of ICASSP*, Dallas, Texas, U.S.A, 2010, pp. 4546-4549.

[3] T. N. Sainath, A. Carmi, D. Kanevsky, and B. Ramabhadran, "Bayesian compressive sensing for phonetic classification," in *Proc. of ICASSP*, Dallas, Texas, U.S.A, 2010, pp. 4370-4373.

[4] I. Naseem, R. Togneri, and M. Bennamoun, "Sparse Representation for Speaker Identification", in *Proc. of ICPR*, 2010, pp. 4460-4463.

[5] M. Li, X. Zhang, Y. H. Yan, and S. Narayanan, "Speaker Verification using Sparse Representations on Total Variability I-vector," in *Proc. of INTERSPEECH*, 2011, pp. 2729-2732.

[6] E. J. Candès and M. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, 2008, pp. 21-30.

[7] D. Donoho, "For most large underdetermined systems of linear equations the minimal l_1-norm solution is also the sparsest solution," *Communications on Pure and Applied Mathematics*, vol. 59, no. 6, 2006, pp. 797-829.

[8] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of Royal Statistical Society. Series B (Methodological)*, vol.58, no. 1, 1996, pp. 267-288.

[9] N. Dehak, P. Kenny, R. Dehak, P. Ouellet, and P. Dumouchel, "Front End Factor Analysis for Speaker Verification," IEEE Trans. on Audio, Speech, and Language Processing, 19(4):788-798,2011.

[10] N. Dehak, P. A. Torres-Carrasquillo, D.Reynolds, and R. Dehak, "Language Recognition via I-vector and Dimensionality Reduction," in Proc. of INTERSPEECH, 857-860, 2011.

[11] N. Dehak, R. Dehak, P. Kenny, N. Brummer, P. Ouellet, and P. Dumouchel, "Support Vector Machines versus Fast Scoring in the Low-Dimensional Total Variability Space for Speaker Verification," in *Proc. of INTERSPEECH*, 1559-1562, 2009.

[12] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[13] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transcations on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, 1998, pp. 832-844.