CRSS systems for the NIST i-Vector Machine Learning Challenge

*Gang Liu, Chengzhu Yu, Navid Shokouhi, Abhinav Misra, Hua Xing, John H. L. Hansen**

Center for Robust Speech Systems (CRSS)
University of Texas at Dallas, Richardson, TX 75080
{gang.liu,chengzhu.yu,navid.shokouhi,abhinav.misra,hxx093020,john.hansen}@utdallas.edu

Abstract

This paper describes the systems developed by the Center for Robust Speech Systems (CRSS), Univ. of Texas - Dallas, for the National Institute of Standards and Technology (NIST) i-Vector challenge. Since the emphasis of this challenge is on utilizing unlabeled development data, our system development focuses on: 1) unsupervised clustering methods to estimate development data labels; 2) build efficient classifier without clustering method. Our results indicate substantial improvements obtained from incorporating one or more of the aforementioned techniques.

Index Terms: i-Vector challenge, clustering, speaker verification, development data, data labeling.

1. Introduction

In large scale speaker verification tasks, such as the NIST Speaker Recognition Evaluation (SRE) [1] and DARPA RATS (Robust Automatic Transcription of Speech) [2], it is shown that low dimensional feature vectors (namely i-Vectors) and probabilistic linear discriminant analysis (PLDA) modeling are two of the main constituents of state-of-the-art technology [3~12]. In the 2013-2014 Speaker Recognition i-Vector Machine Learning Challenge, some new challenges are introduced. One of which is that no label information is provided for development data. This study will investigate ways to improve performance in this scenario. One of our approaches is using clustering to recover the label information. Another is to build classifiers without using any clustering since non-ideal clustering will inevitably introduce erroneous information.

In 2014, in an effort to encourage all researchers involved in pattern recognition and machine learning research, NIST held a new competition. With the goal of reaching out to a broader community, NIST collaborated with Johns Hopkins University's Human Language Technology Center of Excellence and MIT Lincoln Laboratory to supply participants with i-vectors [13], instead of speech recordings for model/test speakers and development data. This approach short-circuits variations such as feature extraction and other signal related aspects, allowing those that are less familiar with the signal processing aspects of speech to be able to participate. Five i-Vectors are provided for each model speaker. Development and test i-vectors are unlabeled, preventing participants from using commonly used channel compensations methods such as applying PLDA to the i-Vector space. Other information provided by NIST includes file durations (as a form of meta-data) and evaluation guidelines. The distribution of file durations is provided in Fig. 1. The semi-normal distribution of log-duration values indicates that using file durations as meta-data could be useful in the decision-making process.

NIST provides an implementation of cosine scoring as a baseline system [3, 13]. The cosine scoring system uses a global mean and variance to project the i-Vector space onto a unit sphere, making it possible to compare different i-Vectors.

Evaluations are performed in two phases. The first phase is a gradual process where participants can submit their system outputs (in the form of score vectors) and see the min-DCF value for a subset of the trials (*40%*) [13]. Participants can also see where they stand in the overall competition. This phase started from November of 2013 and lasted until Apr-07-2014. The second phase of the evaluation is a one-time event in which NIST releases the min-DCF values for all submissions over a different trial subset (the remaining 60%). Relying on the second evaluation phase has the benefit of ruling out over-tuned submissions.

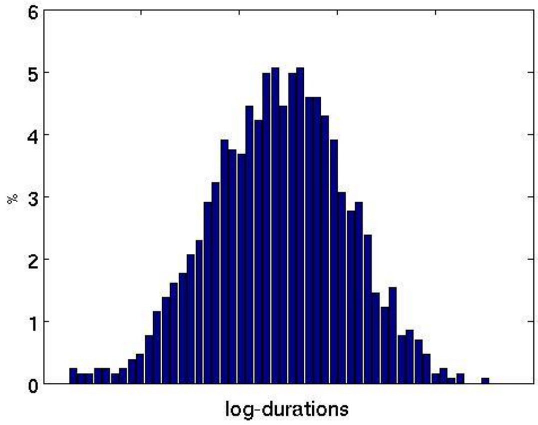

Figure 1. *Histogram of the logarithm of file durations in the i-Vector challenge data.*

2. Baseline System

The baseline system is cosine distance scoring (CDS) [13]. One of the useful properties of CDS is symmetry, that is in the scoring process there is no difference between model and test i-Vectors in each trial. The system description in summary is:

1. Use the unlabeled development data to estimate a global mean and covariance.

2. Center and whiten the evaluation i-Vectors based on the computed mean and variance.

3. Project all the i-Vectors into the unit sphere.

4. For each model, average its five i-Vectors and then project the resulting averaged model i-Vector into the unit sphere.

5. Compute the inner product between all the average-model i-Vectors and test i-Vectors.

*This project was funded by AFRL under contract FA8750-12-1-0188 and partially by the University of Texas at Dallas from the Distinguished University Chair in Telecommunications Engineering held by J.H.L. Hansen.

3. Proposed System

This i-Vector challenge is based on NIST SRE data from 2004 to 2012. This suggests some of the popular algorithm may still fit this i-Vector challenge. But it also has its new flavor. One of the major challenges is the development comes with no speaker label information. Here we will mainly focus on two variations of PLDA and an SVM-based approach.

3.1. PLDA

PLDA is a process that follows factor analysis in order to separate the between-speaker and within-speaker variability in the i-vector space. However, to model PLDA properly, a large amount of labeled data is required.

In order to find the development data labels, we employ an iterative bottom-up classification algorithm. To improve both clustering speed and reliability, the i-Vectors extracted from audio files of less than 20 seconds are excluded from the process. We apply a bottom-up hierarchical clustering using k-means algorithm by treating each i-vector as a separate cluster to start with. The similarity between two clusters is then determined by averaging the distance between i-Vectors in the first cluster and those in the second cluster. Here, the distance is defined as the cosine distance between two i-Vectors. The termination criterion of each iteration is set according to the inconsistency coefficient which is a measure of similarity decreasing gradient during clustering. After each iteration, i-Vectors from each cluster are averaged followed by length normalizion. New iteration starts by treating each averaged i-vector as separate cluster. The best performance was achieved when using 4 iterations.

The two algorithms described below are designed to combine the information obtained from different model i-vectors supplied for each speaker. Each of these method results in a different set of scores which contain complementary information useful for the final submission.

3.1.1 Before-scoring Average PLDA (PLDA1):

In this approach, the five i-Vectors of the j^{th} enrollment speaker are grouped and averaged before applying PLDA to perform verification. Using the average model i-Vector helps to omit potential noise and/or channel mismatch.

3.1.2 Post-scoring Average PLDA (PLDA2)

Each i-Vector of the target file is treated as if originated from a different speaker. After applying PLDA, scores of the j^{th} test file against instances of i^{th} enrollment speaker are averaged, and used as the likelihood score of the *(i-j)*th trial. This is equivalent to a majority vote on the decision with the hope that each individual sample/utterance captures some combination of the acoustic-based speaker characteristics and environment distortion. This basically can be considered multi-condition training and is an echo to the multi-condition preparation for the enrollment files, which is neglected in PLDA-1 (described in the previous section).

3.2. SVM

1300 i-Vectors are randomly selected from the development set and used as pseudo model files (also known as imposter model files). An additional 8700 files are selected and used as pseudo test files (also known as imposter test files). The remaining of the development data is used as imposter data to help build SVM models for each enrolment speaker.

4. Score normalization

Score normalization methods are employed to reduce the effects of variability in decision making. They normalize the decision score with mean, *m*, and standard deviation, *v*, derived from some extra data:

$$score(enr_i, tst_j)_{norm} = \frac{score(enr_i, tst_j) - m}{v}\ , \qquad (1)$$

Depending on the background data from which these parameters are derived, score normalization has three variations in current speaker verification systems.

4.1. Zero Normalization (ZNorm)

Zero Normalization (ZNorm) can compensate inter-speaker variability. It estimates the mean and standard variation of scores of a list of pseudo test files (called ZList) against a target model. That is, it assumes the scores take the following form:

$$score(enr_i, ZList) \sim N(m_i, v_i) \qquad (2)$$

Note that this only depends on the i^{th} target model. This normalization is a function of the speaker models. From (1),

$$score(enr_i, tst_j)_{ZNorm} = \frac{score(enr_i, tst_j) - m_i}{v_i} \qquad (3)$$

4.2. Test Normalization (TNorm)

Test Normalization (T-norm) can compensate inter-session variability. It estimates the mean and standard variation of scores from a list of pseudo model files (called TList) against test utterances. That is, it assumes their scores take the following form:

$$score(TList, tst_j) \sim N(m_j, v_j) \qquad (4)$$

In T-norm, normalization parameters are a function of test utterances.

$$score(enr_i, tst_j)_{TNorm} = \frac{score(enr_i, tst_j) - m_j}{v_j} \qquad (5)$$

4.3. Symmetric Normalization (SNorm)

The abovementioned score normalization techniques are either model dependent or test dependent. For some symmetric back-ends, such as cosine distance scoring, it is desirable to preserve the symmetry between the test and model set for better performance. This is done by combining normalization results from T-norm and Z-norm.

$$\begin{aligned} & score(enr_i, tst_j)_{SNorm} \\ & = \frac{score(enr_i, tst_j) - m_i}{v_i} + \frac{score(enr_i, tst_j) - m_j}{v_j} \qquad (6) \\ & = score(enr_i, tst_j)_{ZNorm} + score(enr_i, tst_j)_{TNorm} \end{aligned}$$

4.4. Straightforward Implementation

Figure 2 illustrates the score-normalization process. All cohort score matrices needed by the three score normalization methods mentioned above can be derived in one run.

5. Experiment Results

The results from all the methods are summarized in Table 1. From experimental observation, SNorm fails to boost the performance of PLDA. So it is only used to other systems

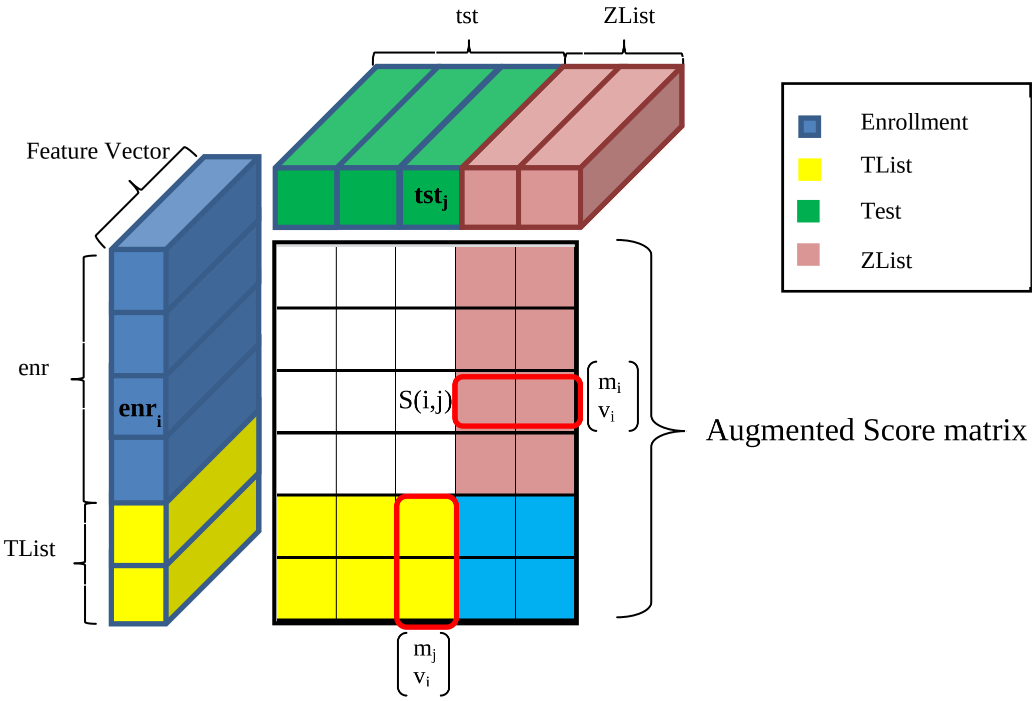

Figure 2. Straight implementation of score normalization. Enrollment list can be augmented by appendix pseudo model list, that is, TList. Test list can be augmented by appendix pseudo model list, that is, ZList. Then we can use any proper back-end to do the scoring to derive an augmented score matrix, which include original score matrix (Left-top sub matrix), TNorm score matrix (Left-bottom), and ZNorm score matrix (Right-top sub matrix). Here, the feature vector can be any dimension-fixed feature, such as i-Vector.

Table 1. System *Performance on NIST i-Vector Challenge Progress set. The final fusion includes PLDA1 scores without SNorm, and system 1 and 4 with SNorm.*

system	minDCF	With SNorm
System 1: baseline	0.386	0.384
System 2: PLDA1	0.349	/
System 3: PLDA2	0.576	/
System 4: SVM	0.334	0.312
Fusion: 1+2+4		**0.287**

The system is then fused with another pre-fused system of Agnito and BUT (its minDCF on progress set is: 0.271) and get 0.256 on progress set (equal weight is applied)[i].

6. Acknowledgement

The authors want to thank the constructive suggestion from Taufiq Hasan.

7. References

[1] The NIST year 1997 - 2012 speaker recognition evaluation plans, [Online]. Available: http://www.nist.gov..

[2] K. Walker and S. Strassel, “The RATS radio traffic collection system,” in ISCA Speaker Odyssey, 2012.

[3] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Frontend factor analysis for speaker verification,” IEEE Trans. Audio, Speech, and Lang. Process., vol. 19, no. 99, pp. 788 – 798, May 2010.

[4] P. Kenny, "Bayesian Speaker Verification with Heavy-Tailed Priors," in Proc. Odyssey, Brno, Czech, Jun. 2010.

[5] L. Ferrer, M. McLaren, N. Scheffer, Y. Lei, M. Graciarena, and V. Mitra, "A Noise-Robust System for NIST 2012 Speaker Recognition Evaluation," in Proc. Interspeech, Lyon, France, Aug. 2013, pp. 1981-1985.

[6] D. Colibro, C. Vair, K. Farrell, N. Krause, G. Karvitsky, S. Cumani, P. Laface, “Nuance - Politecnico di Torino’s 2012 NIST Speaker Recognition Evaluation System,” in Proc. Interspeech, Lyon, France, Aug. 2013, pp. 1996-2000

[7] N. Brummer, et. al., “ABC System description for NIST SRE 2012,” in Proc. NIST Speaker Recognition Evaluation, Orlando, FL, USA, Dec. 2012.

[8] G. Liu, T. Hasan, H. Boril, J.H.L. Hansen, "An investigation on back-end for speaker recognition in multi-session enrollment", in Proc. ICASSP, Vancouver, Canada, May 25-31, 2013. pp. 7755-7759.

[9] T. Hasan, S.O. Sadjadi, G. Liu, N. Shokouhi, H. Boril, J.H.L. Hansen, "CRSS systems for 2012 nist speaker recognition evaluation", in Proc. ICASSP, Vancouver, Canada, pp. 6783-6787, 2013.

[10] V. Hautamaki et al., "Automatic regularization of cross-entropy cost for speaker recognition fusion", in Proc. INTERSPEECH, Lyon, France, 25-29 Aug.,2013.

[11] R. Saeidi et al., "I4U submission to NIST SRE 2012: A large-scale collaborative effort for noise-robust speaker verification", in Proc. INTERSPEECH, Lyon, France, 25-29 Aug.,2013.

[12] O. Plchot et al., “Developing a speaker identification system for the DARPA RATS Project,” in Proc. ICASSP ’13, 2013, pp. 6768–6772.

[13] “The 2013-2014 Speaker Recognition i-Vector Machine Learning Challenge”. [Online] Available: http://www.nist.gov/itl/iad/mig/upload/sre-i-Vectorchallenge_2013-11-18_r0.pdf.

[i] See AGNITIO and BUT’s system description for details.