UTILIZATION OF UNLABELED DEVELOPMENT DATA FOR SPEAKER VERIFICATION

*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, hua.xing, john.hansen}@utdallas.edu

ABSTRACT

State-of-the-art speaker verification systems model speaker identity by mapping i-Vectors onto a probabilistic linear discriminant analysis (PLDA) space. Compared to other modeling approaches (such as cosine distance scoring), PLDA provides a more efficient mechanism to separate speaker information from other sources of undesired variabilities and offers superior speaker verification performance. Unfortunately, this efficiency is obtained at the cost of a required large corpus of labeled development data, which is too expensive/unrealistic in many cases. This study investigates a potential solution to resolve this challenge by effectively utilizing unlabeled development data with universal imposter clustering. The proposed method offers +21.9% and +34.6% relative gains versus the baseline system on two public available corpora, respectively. This significant improvement proves the effectiveness of the proposed method.

Index Terms— Clustering, Speaker verification, PLDA, i-Vector, Universal imposter clustering

1. INTRODUCTION

Recent large scale speaker verification evaluations, such as the NIST Speaker Recognition Evaluation (SRE) [1] and DARPA RATS (Robust Automatic Transcription of Speech) [2], have shown that low dimensional feature vectors (that is i-Vector) and PLDA modeling is state-of-the-art speaker identification technology [3~12]. Compared to other modeling approaches (such as cosine distance scoring), PLDA provides a more efficient mechanism to separate speaker information from other sources of undesired variabilities. It can learn acoustic variations, such as channel, noise [19, 20], duration, vocal effort, age, and emotion [21, 22], from multiple sessions recorded for each speaker (provided in the form of model development data, which is different from enrollment data) [23]. Usually, the required development data can have more than 1000 speakers with potentially 10~100 sessions available for each speaker. Collecting such large amounts of labeled data is not only time-consuming but also expensive, and therefore unfeasible in many applications. In reality, researchers often collect recording sessions without knowing speaker or session specific information. The question here is: *Can we still make use of such unlabeled / open data?*

There is already ongoing research in this direction. The *domain adaptation challenge* is among the most recent efforts, in which the focus is to use unlabeled data to adapt and apply i-Vector speaker recognition systems based on some out-of-domain labeled i-Vectors[1]. Some progress has been reported in this regard [13-15]. The NIST i-Vector Machine Learning Challenge has a similar goal, but without any out-of-domain labeled i-Vectors [18]. Despite all these efforts, few have investigated methods to utilize the potential information that is stored in enrollment data without having the luxury of labeled development data. We believe this is a crucial aspect of the speaker verification problem, since in most speaker verification tasks (pattern classification applications in general), one must deal with multiple speakers/classes with multiple samples/sessions for which labels are only provided for enrollment data (i.e., the controlled training data).

To avoid confusion, we use the following definitions for the three categories of data available in our speaker verification framework: 1) enrollment data; includes thousands of speakers and several sessions for each speaker (sometimes called training data), 2) test data; data that is compared against enrollment models to form trials, and 3) unlabeled development data; used to train background models.

This paper is organized as follows. The next section begins with a description of baseline system, followed by proposed methods in Sec. 3. PLDA is briefly reviewed in Sec. 4. In Sec. 5, experiment setup is detailed. Experiment results are summarized in Sec. 6. The study is concluded in Sec. 7.

*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.

[1] http://www.clsp.jhu.edu/workshops/archive/ws13-summer-workshop/groups/spk-13/

2. BASELINE

In this study, we always assume the features, i-Vectors, instead of audio files, are already available. Thus, we will only focus on how we can make use of the given i-Vectors to improve system performance. Since PLDA is state-of-the-art technology in the i-Vector framework, it will be used as the baseline system. But to build an efficient PLDA classifier, we need speaker label information. We can either use enrollment data as training, or we can adopt clustering methods on development data to recover label information. As the baseline system, we will simply use enrollment data as PLDA model development data since we can assume the label information is very accurate. The exploration of clustering approach is delegated to Sec. 3.

3. PROPOSED METHODS

We begin by exploring two classical clustering methods in order to classify unlabeled development data, and subsequently apply this to the PLDA framework. Furthermore, we describe our proposed clustering technique.

3.1. Two classical clustering methods

After a series of pilot clustering experiments on potential clustering methods, we narrow down to two promising approaches [16].

3.1.1. K-means clustering

The k-means algorithm is an iterative clustering method which partitions a given dataset into a user specified number of clusters, *k*. The algorithm operates on a set of d-dimensional vectors $D = \{X_i \mid i = 1,...,N\}$ where $X_i \in R^d$ denotes the i^{th} data point and N is the size of the dataset. The first iteration is initialized by picking *k* points in R^d as the initial *k* cluster representatives or "centroids". There are several alternative techniques for selecting these initial seeds. Random sampling from the dataset is used in this study. The algorithm iterates between the following two steps until convergence [16]:

Step 1: Data Assignment. Each data point is assigned to its closest centroid, with ties broken arbitrarily. This partitions the data samples.

Step 2: Re-allocation of "means". Each cluster representative is re-allocated to the center of all data points assigned to it.

One open question is to decide the number of clusters: *k*. We resolve this issue through trial and error. Note that each iteration requires $N \times k$ comparisons, which signifies the time complexity of each iteration. Cosine similarity scores are used as the metric to measure the distance in this study due to its superior performance on i-Vector features.

$$\cos(a,b) = \frac{a^T b}{\| a \| \bullet \| b \|} \quad (1)$$

3.1.2. Agglomerative hierarchical clustering (AHC)

Hierarchical clustering algorithms can be either top-down or bottom-up. We adopt the *bottom-up* Agglomerative hierarchical clustering (AHC) approach, in which each observation starts in its own cluster, and pairs of clusters are merged as one move up the hierarchy. Cosine similarity is used as the distance metric. The inconsistency *cutoff* value is found through grid search. *Cutoff* is a threshold for cutting the hierarchical tree generated by AHC into clusters. Clusters are formed when inconsistent values are greater than *cutoff*[2]. The optimal *cutoff* value is determined via hill-climbing method.

3.2. Proposed method: Universal Imposter Clustering (UIC)

As is known, for traditional universal background modeling (UBM), different speaker data are used collectively to model the generic speaker space. Along the same idea, we propose to use all the development data as if they belong to a single speaker/class and add this new class to the enrollment data (assume it has *S* speakers/classes) to form new development data, which has *S*+1 classes. Next, we build our PLDA model based on this new development data. This method is called Universal Imposter Clustering (UIC). The rationale behind this is as follows:

a) Clustering by itself cannot draw a clear line between different speakers. For example, some data may be grouped together due to characteristics of channel instead of speaker. Potentially, this type of clustering result will mislead the modeling process.

b) By taking the development data set as a whole without distinguishing speaker information, we consider this as the $(S+1)^{th}$ speaker class. This new super "speaker" will be used to represent the residual speaker information in the data. This can be interpreted as a "speaker" that has a relatively larger variation across different sessions (in the data investigated in this study, there are nearly 40,000 sessions). The efficiency of this method is validated in Sec. 6.

4. PLDA

The extracted i-Vector of each speech utterance contains both inter-speaker and intra-speaker variabilities. Therefore, the PLDA classifier is normally employed to separate out the

[2] The *clusterdata* Matlab built-in function is adopted for the Agglomerative hierarchical clustering. To facilitate potential comparison, we report the usage here: speaker_label = *clusterdata*(unlabeled_data,'criterion','inconsistent','cutoff',0.8,'distance','cosine')

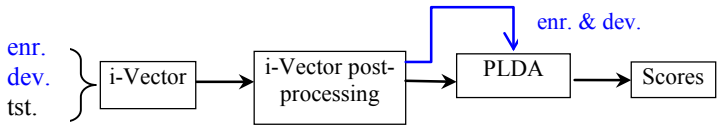

Figure 1: *Block diagram of i-Vector based speaker verification system. Enr=enrolmment data; dev.=development data; tst.=test data. Labeled development data is used to assist model developing. Enrollment data is used to register speakers. During testing phase, only test data is imputed.*

intra speaker variability caused by channel mismatch in the i-Vector system [4]. In the PLDA framework, a D-dimensional i-Vector ω can be described as:

$$\omega = m + \Phi\beta + \varepsilon, \quad (2)$$

where Φ is a $D \times R$ rectangular matrix that represents the subspace containing speaker specific information, β is an R-dimensional latent vector assumed to have a standard normal distribution, and ε is the D-dimensional vector representing the full covariance of the residual noise. Here, the reduced dimension R denotes the number of columns in the matrix Φ. To train the PLDA model, we need labeled data. Note that length normalization is applied to all i-Vectors before PLDA classification [17]. The implementation of scoring follows [4].

5. EXPERIMENT SETUP

In this section, the configurations of the i-Vector based speaker recognition system used in our experiments are briefly described. The flow chart is illustrated in Fig. 1.

5.1. Database

Experiments are performed on two data sets: NIST SRE2012 and NIST i-Vector Machine Learning Challenge [18]. The data investigated here are all 600-dimensional i-Vector feature vectors. Although the two data sets may share some similar raw audio data source, they differ in many implementation details, such as VAD, low-level feature extraction, noise introduction into the training file. We will focus on the relevant details and summarize in the following two sub-sections.

5.1.1. NIST SRE2012 in house data set

In preparing this data set, we maintained a close collaboration with the I4U consortium in the NIST SRE2012 [11] challenge[3]. Only male speakers are used for simplicity. Recordings are collected from a total of 763 SRE 2012 target speakers belonging to SRE'06-10 corpora. For these speakers, train/test-trials are prepared for the evaluation. The training list includes multiple sessions per speaker and the test list includes both known and unknown non-target speakers, following SRE 2012 protocol (Table 1). On average, each enrollment speaker has 39 sessions. To assist in a potential comparison, i-Vectors for this corpus can be downloaded online[4].

5.1.2. i-Vector Machine Learning Challenge data set

This data set is provided by NIST as part of the i-Vector Machine Learning Challenge (noted as IVC) [18].

The statistics of this data set are detailed in Table 2. There are a total of 12,582,004 (1306 by 9634) trails. Each enrollment speaker has exactly 5 sessions. These trials are divided into two subsets: 1) a progress subset; used to keep track of each participant's progress, and 2) an evaluation subset; used for the final evaluation. The progress subset comprises 40% of the total trials and the evaluation subset contains the remaining 60%. The results reported in this paper are based on scores obtained from the progress subset[5]. We also note that this data set may contain both male and female speakers, since gender information was not provided for the i-Vectors.

5.2. i-Vector processing and PLDA

The original dimension of all i-Vectors in this study is 600. LDA is applied to reduce the dimension to 400, which also removes some channel and noise distortion, followed by length normalization and PLDA modeling.

6. EXPERIMENT RESULTS AND ANALYSIS

To validate our proposed framework, experiments are performed on the two corpora introduced in Sec. 5.1. First, we explore the verification performance of classical clustering techniques, followed by an investigation of the proposed method. The results are reported in four different metrics; namely EER, and minDCF values extracted by applying protocols from 2008, 2010, and 2012 NIST SREs [1].

The baseline in this study uses enrolment data as the development set both for LDA and PLDA modeling.

[3] [online] http://cls.ru.nl/~saeidi/file_library/I4U.tgz

[4] [online] https://sites.google.com/site/gangliuresearch/codes

[5] It is noted that the evaluation rule in i-Vector Machine Learning Challenge forbids a collective use of different enrollment speakers. So, use caution when comparing the result of this study with that of the Challenge. But this is allowed in NIST SRE2012.

6.1. Verification based on classical clustering

We start by evaluating the performance on the NIST SRE2012 data set. As mentioned in Sec. 2.1.1, the k-means algorithm requires specifying k, the number of clusters. A grid search is implemented to find the optimal k and detailed in Fig. 2. The grid search result of AHC is displayed in Fig. 3. The optimal value for the *cutoff* is 0.8. Both clustering techniques fail to surpass the baseline system (see Fig. 2). This brute-force paradigm is unavoidable, since one does not have access to the number of speakers in unlabeled development data. So this is an apparent disadvantage of k-means clustering method.

For the i-Vector Machine Learning Challenge, k-means and AHC results are documented in Table 3 and 4, respectively. Compared with the baseline system, optimal k-means only offers minor improvements. Optimal AHC, however, can offer +16.1% relative gain when the *cutoff* value is 0.8.

In summary, the performance of k-means is relatively stable over a broad spectrum of k values. A relatively good candidate can be found for k by a brute-force grid search, but the actual improvement is minor. This also suggests the automatic determination of clustering number warrant further investigation [24, 25]. AHC may or may not help depending upon the actual corpus. Classical clustering may fail to provide sufficient meaningful information to help boost performance depending on both parameters and data specifications. Plus, some parameter may need heavy tuning before offering reasonable performance.

6.2. Verification based on proposed universal imposter clustering

The performance of the proposed universal imposter clustering (UIC) on the NIST SRE2012 data set is provided in Table 5. We show that the proposed method, UIC, offers significant improvements across all 4 metrics. This is especially true for the system which obtains +21.9% relative gain of minDCF2012, which was the metric adopted in the NIST SRE2012. The results for the i-Vector Machine Learning Challenge are detailed in Table 6. There, UIC offers +34.6% relative improvement against the baseline system, and +22.1% against the agglomerative hierarchical clustering approach. Overall, our proposed Universal imposter clustering solution can consistently offer superior results without requiring parameter-tuning.

The results are very interesting. Given the goal of clustering is to discover potential environmental variation of individual speaker/cluster. However, the experimental results indicated that, the extra information, the speaker label of the development data recovered by clustering, does not help substantially. This may be caused by the fact that the clustering results are very noisy. The gain from speaker label recovering is actually severely undermined by the clustering noise. On the other hand, by grouping all the i-Vectors of development data (close to 40 thousands), UIC can more effectively account for many unseen variation and therefore help boost the classification performance.

We acknowledge that this is just some initial analysis, further exploration is warranted here.

7. CONCLUSIONS

In this study, we investigated a series of algorithms for speaker verification involving unlabeled development data. We explored two approaches to utilize the unlabeled development data. It was shown that traditional clustering methods may or may not provide reasonable improvements depending upon specifics of the data set and parameter configuration of the clustering algorithm in question. Compared with the baseline system which only uses enrollment data for model training, the proposed Universal imposter clustering (UIC) offers +21.9% and +34.6% relative improvements on the two data sets: our SRE 2012 i-Vector development data, and the publicly available NIST i-Vector Machine Learning Challenge data set. Even compared with the popular clustering methods, the proposed UIC also provides significant improvements. This study also confirms that the contribution from unlabeled / open data to boost the performance of speaker identification.

This is only an initial exploration in this direction. Although the proposed methods can provide significant improvement, it still leaves much to be desired. Investigation on more efficient clustering method and decision weighting will be next step of this study.

8. 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, May 25-31, 2013. pp. 6783-6787.
[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, Vancouver, Canada, May 2013, pp. 6768–6772
[13] S. Shum, D. Reynolds, D. Garcia-Romero, and A. McCree, "Unsupervised clustering approaches for domain adaptation in speaker recognition systems", in Proc. Odyssey 2014, The speaker and language recognition workshop, Joensuu, Finland, June 2014.
[14] D. Garcia-Romero, A. McCree, S. Shum, N. Brummer, and C. Vaquero, "Unsupervised domain adaptation for i-vector speaker recognition", in Proc. Odyssey 2014, The speaker and language recognition workshop, Joensuu, Finland, June 2014.
[15] J. Villalba and E. Lleida, "Unsupervised adaptation of PLDA by using variational Bayes methods", in Proc. ICASSP, Florence, Italy, May 2014, pp.744-748
[16] X. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q, Yang, H. Motoda, G.J. McLachlan et al. "Top 10 algorithms in data mining." Knowledge and Information Systems 14, no. 1 (2008): 1-37.
[17] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-Vector length normalization in speaker recognition systems," in Proc. Interspeech, Florence, Italy, Aug. 2011, pp. 249–252.
[18] "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
[19] C. Yu, G. Liu, J.H.L. Hansen,"Acoustic Feature Transformation using UBM-based LDA for Speaker Recognition," in Proc. Interspeech 2014, Singapore, Sep. 2014.
[20] C. Yu, G. Liu, S. Hahm, J.H.L. Hansen, "Uncertainty Propagation in Front End Factor Analysis For Noise Robust Speaker Recognition," in Proc. ICASSP 2014, Florence, Italy, pp. 4045-4049
[21] G. Liu, J.H.L. Hansen, "Supra-Segmental Feature Based Speaker Trait Detection," in Proc. Odyssey 2014, The speaker and language recognition workshop, Joensuu, Finland, June 2014.
[22] G. Liu, Y. Lei, J.H.L. Hansen, "A Novel Feature Extraction Strategy for Multi-stream Robust Emotion Identification", in Proc. Interspeech, Makuhari Messe, Japan, 2010. pp.482-485
[23] G. Liu, J.H.L. Hansen, "An Investigation into Back-end Advancements for Speaker Recognition in Multi-Session and Noisy Enrollment Scenarios," Accepted to IEEE Trans. on Audio Speech and Lang. Process., 2014
[24] R. Tibshirani, G. Walther, T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63(2), 2001. pp.411-423.
[25] D. Pelleg, A. W. Moore, "X-means: Extending K-means with Efficient Estimation of the Number of Clusters," in Proc. ICML. 2000. pp.727-734.

Table 1. *Details of data used for the NIST SRE2012 in-house evaluation.*

data	train (spk.#)	test (spk.#)	dev.	trial	target trial: non-target trial
#	29961 (763)	21837 (804)	39375	16661631	15483:16646148

Table 2. *Details of data used for the NIST i-Vector Machine Learning Challenge.*

data	train (spk.#)	test	dev.	trial	progress subset: eval. subset
#	6530 (1306)	9634	36572	12582004	40% : 60%

Table 3. *Grid search of Kmeans on the NIST i-Vector Machine Learning Challenge.*

		cluster #				
	baseline	8k	10k	12k	14k	16k
minDCF	0.454	0.459	0.454	0.454	**0.443**	0.589

Table 4. *Cutoff value grid search of AHC on the NIST i-Vector Machine Learning Challenge.*

		cutoff			
	baseline	0.5	0.7	0.8	0.9
minDCF	0.454	0.39	0.389	**0.379**	0.381

Table 5. *Performance comparison of proposed UIC on the NIST SRE2012.*

	baseline	proposed UIC	Gain
EER	2.55	**1.49**	+41.6%
minDCF2008	0.088	**0.057**	+35.2%
minDCF2010	0.267	**0.221**	+17.2%
minDCF2012	0.210	**0.164**	+21.9%

Table 6. *Performance comparison of proposed UIC on the NIST i-Vector Challenge.*

system	minDCF
System 1: baseline	0.454
System 2:AHC (cutoff=0.8)	0.379
System 3: UIC(proposed)	**0.297**
Gain (system 3 vs. 1)	+34.6%

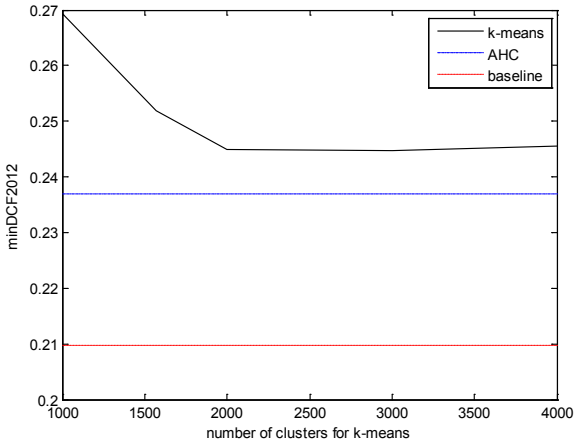

Figure 2: *Illustration of k-means on the SRE2012 (black line). The blue dash line is the optimal AHC performance. The red line is the baseline system.*

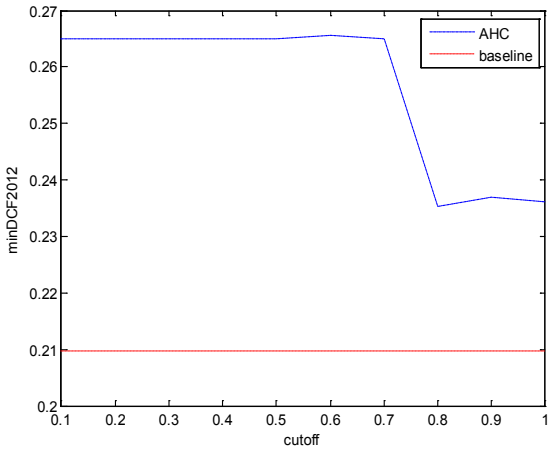

Figure 3: *Grid search of AHC on the NSIT SRE2012 data set.*