## **DESCRIPTION OF SYSTEM FOR MCE 2018**

Qingyu Guo, Xuming Pan

Team [ATS]

### **ABSTRACT**

The task of MCE 2018 is to determine whether or not a recorded utterance was spoken by one of a large number of "blacklisted" speakers [1]. Each utterance is represented by a single i-vector [2]. If we verify that the utterance was spoken by one "blacklisted" speaker, the specific Blacklist speaker needs to be found out. In our submission, we tackle the two sub-tasks with specified algorithms. For the top-S detector, we treat the problem as a binary classification. Linear Discriminant Analysis (LDA) [3] is applied to reduce the dimension of data space and transforms the training data to make it easier to classify. An SVM classifier with polynomial kernel plays the role of top-S detector. For the task of top-1 detection, we combine two of the PLDA variants, PLDA1 [4] and PLDA<sub>max</sub> to achieve good performance. Evaluation on the development dataset shows that our top-S detector achieves EER of 1.28% and an EER of 7.58% is reported on top-1 detector, with total confusion error of 269.

*Index Terms*— Multi-target detector, LDA, SVM, PLDA, speaker verification and identification

#### 1. INTRODUCTION

The Multitarget Challenge aims to assess how well current speech technology is able to determine whether or not a recorded utterance was spoken by one of a large number of blacklisted speakers. It is a form of multi-target speaker detection based on real-world telephone conversations. Data recordings are generated from call center customer-agent conversations. Each conversation is represented by a single i- vector [5]. Given a pool of training and development data from non-Blacklist and Blacklist speakers, the task is to measure how accurately one can detect 1) whether a test recording is spoken by a Blacklist speaker, and 2) which specific Blacklist speaker was talking. Although the primary task will restrict participants to the provided data, participants are allowed to submit secondary systems that use additional data in order to achieve better performance.

## 2. TOP-S DETECTION: LDA+SVM

The task of top-S detection asks: given a recorded utterance, determine whether or not it was spoken by one of the "black-

listed" speakers. We regarded it as a binary classification and provide a top-S detector which combines LDA and SVM. Let  $\mathcal{X}$  be the set of i-vectors in training set and  $\mathbf{x}^i$  be the i-th i-vector in  $\mathcal{X}$ . Let  $I = \{1, 2, ..., |\mathcal{X}|\}$ . Let  $S^{bl}$  be the set of blacklist speakers and  $S^{bg}$  be the set of background speakers in the training set. For each speaker  $s \in S$  with  $S = S^{bl} \cup S^{bg}$ , let  $I_s \subset I$  be the index set of i-vectors in  $\mathcal{X}$  corresponding to speaker s. Let  $\bar{\mathbf{x}}_s$  be the average i-vector for speaker s in training set and  $\bar{\mathbf{x}}$  be the average i-vector over  $\mathcal{X}$ .

LDA is a widely used technique for dimensionality reduction. It finds a set of orthogonal axes that minimize the within-class variation and maximize the between-class variation. The projection matrix W is computed by solving the following optimization:

$$W = \arg\max_{W} Tr\{\frac{W^t S_B W}{W^t S_W W}\}\tag{1}$$

where  $S_B$  is the between-class covariance matrix and  $S_W$  is the within-class covariance matrix.

$$S_B = \sum_{s \in S} (\bar{\mathbf{x}}_s - \bar{\mathbf{x}})(\bar{\mathbf{x}}_s - \bar{\mathbf{x}})^t$$

$$S_W = \sum_{s \in S} \frac{1}{|S|} \sum_{i \in I_s} (\mathbf{x}^i - \bar{\mathbf{x}}_s)(\mathbf{x}^i - \bar{\mathbf{x}}_s)^t$$
(2)

The projection W is applied to both the training data  $\mathcal{X}$  and the test set  $\mathcal{Z}$ , and the projected datasets are  $\mathcal{X}'$  and  $\mathcal{Z}'$  respectively. Let each instance  $\mathcal{X}'$  be labeled with 1 if it is an (projected) i-vector of a blacklist speaker, and 0 otherwise. The problem becomes a standard binary classification and we adopt the SVM with polynomial kernel as our detector. For degree-d polynomials, the kernel function is defined as:

$$K(\mathbf{u}, \mathbf{v}) = (\mathbf{u}^t \mathbf{v} + c)^d. \tag{3}$$

We implement LDA and SVM using scikit-learn package in Python 3.6 [6]. To output a score for each tested i-vector for computing the EER, we call the predict\_log\_proba function of SVC class to return the log probabilities of possible outcomes for the instance and take the value for outcome of label 1 as the final score.

## 3. TOP-1 DETECTION: PLDA1+PLDA $_{MAX}$

PLDA is a commonly adopted back-end in speaker verification [7, 8]. PLDA is a generative modeling approach, which assumes that the i-vector  $\mathbf{x}^i$  for speaker s can be decomposed as:

$$\mathbf{x}^i = V\mathbf{v}^s + U\mathbf{w}^i + \mathbf{z}^i \tag{4}$$

where V and U are rectangular matrices and represent eigenvoice and eigenchannel subspaces, respectively. Additionally,  $\mathbf{y}^s$  and  $\mathbf{w}^i$  are the speaker and channel factors, respectively.  $\mathbf{z}^i$  is the residual term, typically assumed to be a Gaussian distribution with a zero mean and diagonal covariance.

For verification involving two i-vectors  $\mathbf{x}^i$  and  $\mathbf{x}^j$ , PLDA explores the following hypothesis:  $H_1$  that  $\mathbf{x}^i$  and  $\mathbf{x}^j$  come from the same speaker and  $H_0$  that  $\mathbf{x}^i$  and  $\mathbf{x}^j$  come from the different speakers, which is unified by calculating the following log likelihood ratio:

$$r = \log \frac{p(\mathbf{x}^i, \mathbf{x}^j | H_1)}{p(\mathbf{x}^i, \mathbf{x}^j | H_0)}$$
 (5)

The core of our top-1 detector is PLDA1 algorithm [4], which, for each testing i-vector t, computes

$$r_s = \log \frac{p(\bar{\mathbf{x}}_s, \mathbf{t}|H_1)}{p(\bar{\mathbf{x}}_s, \mathbf{t}|H_0)} \quad \forall s \in S^{bl},$$
 (6)

and predicts  $s^* = \arg\max_{s \in S^{bl}} r_s$  as the identified speaker for  $\mathbf{t}$ .

However, we notice several validation utterances from development set that are consistently misclassified by PLDA1. We argue that it is because  $\bar{\mathbf{x}}_s$  cannot capture the spreading ivectors sometimes. For instance, Fig. 1 shows a 2D visualization of the training i-vectors from blacklist speakers. Yellow marker denotes a testing utterance  $\mathbf{t}$  from speaker s whose i-vectors in training set are marked in red color. PLDA1 recognizes  $\mathbf{t}$  to be spoken by s' whose i-vectors in training set are marked in green color. We notice that  $\mathbf{t}$  is "close" to one i-vector from s, while "far" from the other two. We handle such case with PLDA $_{max}$  algorithm, which calculates the following value:

$$r_s = \max_{i \in I_s} \log \frac{p(\mathbf{x}^i, \mathbf{t}|H_1)}{p(\mathbf{x}^i, \mathbf{t}|H_0)} \quad \forall s \in S^{bl}$$
 (7)

and predicts  $s^* = \arg\max_{s \in S^{bl}} r_s$  as the identified speaker for  $\mathbf{t}$ .

We fuse PLDA1 with PLDA $_{max}$  as follows. We use cross-validation to identify the blacklist speakers that PLDA1 frequently makes mistakes, and denote them by  $\tilde{S}^{bl}$ . For an incoming test utterance  $\mathbf{t}$ , we fuse the "score\_normalized" score  $\mathbf{r}^1$  by PLDA1 and  $\mathbf{r}^{max}$  by PLDA $_{max}$  with the following result

$$r_s = \begin{cases} r_s^{max} & \forall s \in \tilde{S}^{bl} \\ r_s^1 & \forall s \in S \setminus \tilde{S}^{bl}. \end{cases}$$
 (8)

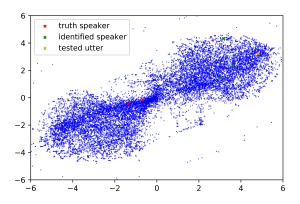


Fig. 1. One confusion error by PLDA1

# 4. CONFIGURATION, DATA USAGE AND EVALUATION

The detailed configuration of our top-S detector is as follows. We call LDA to project both training data and test data to dimension n=600. Thus, we didn't reduce the dimension and use LDA as a linear projection. The projected data are normalized with length normalization to have unit norm. We use SVC class in scikit-learn package with probability set to True and kernel set to linear to train a classifier and all other parameters follow the default setting. The log probabilities on class 1 (blacklist) returned by predict\_log\_proba function are taken as the final score.

The detailed configuration of our top-1 detector is as follows. We first use LDA to reduce the dimension of i-vector space from 600 to 400. We follow the instruction in [4] to implement PLDA1. The platform is Octave and we use the Gaussian-PLDA package as our solver<sup>2</sup>. We use the ivectors of blacklist speakers in the training set as dev data in Gaussian-PLDA and the averaged i-vectors of blacklist speakers  $\bar{\mathbf{x}}_s$  with  $s \in S^{bl}$  as mod data. The testing i-vectors are set as tst data. nPhi is set to 200 and all other parameters follow their default settings.  $PLDA_{max}$  is implemented in similar way except that all the i-vectors of blacklist speakers in the training set is set as both the dev data and mod data. The final prediction follows the equations (7)–(8). The set  $\tilde{S}^{bl}$  is generated as follows. As we use cross-validation to evaluate the performance of the model. We run five random trials with PLDA1 and PLDA<sub>max</sub>, and  $\tilde{S}^{bl}$  is set to be blacklist speakers that PLDA1 failed to identify in at least 4 trials while  $PLDA_{max}$  successfully identified in at least 4 trials.

We tune these parameters with cross-validation, where we combine training data and development data as a full dataset.

<sup>&</sup>lt;sup>1</sup>The score normalization operation is performed on a vector  $\mathbf{r}$  as:  $(\mathbf{r} - mean(\mathbf{r}))/std(\mathbf{r})$ .

 $<sup>^2</sup>$ https://sites.google.com/site/dgromeroweb/software

We randomly separate it into a training subset and validation subset, with a ratio of 3:1 in blacklist i-vectors, same as that between training data and development data. The final system is trained over the full dataset.

We evaluate our system on development set. The top-S EER is 1.28% and an EER of 7.58% is reported on top-1 detector, with total confusion error of 269.

### 5. REFERENCES

- [1] Suwon Shon, Najim Dehak, Douglas Reynolds, and James Glass, "Mce 2018: The 1st multi-target speaker detection and identification challenge evaluation (mce) plan, dataset and baseline system," *ArXiv e-prints arXiv:1807.06663*, 2018.
- [2] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.
- [3] Geoffrey McLachlan, Discriminant analysis and statistical pattern recognition, vol. 544, 2004.
- [4] Gang Liu, Taufiq Hasan, Hynek Boril, and John HL Hansen, "An investigation on back-end for speaker recognition in multi-session enrollment," in *Acoustics, Speech and Signal Processing (ICASSP)*, 2013 IEEE International Conference on, 2013, pp. 7755–7759.
- [5] Tao Xiong and Vladimir Cherkassky, "A combined svm and Ida approach for classification," in *Neural Networks*, 2005. *IJCNN'05*. *Proceedings*. 2005 *IEEE International Joint Conference on*, 2005, vol. 3, pp. 1455–1459.
- [6] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al., "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.
- [7] Sergey Ioffe, "Probabilistic linear discriminant analysis," in Computer Vision ECCV 2006, 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part IV, 2006, pp. 531–542.
- [8] Tomi Kinnunen, Lauri Juvela, Paavo Alku, and Junichi Yamagishi, "Non-parallel voice conversion using i-vector PLDA: towards unifying speaker verification and transformation," in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017, 2017, pp. 5535–5539.