Out-of-Set i-Vector Selection for Open-set Language Identification

Hamid Behravan, Tomi Kinnunen, Ville Hautamäki

School of Computing
University of Eastern Finland
{behravan,tkinnu,villeh}@cs.uef.fi

Abstract

Current language identification (LID) systems are based on an i-vector classifier followed by a multi-class recognition back-end. Identification accuracy degrades considerably when LID systems face open-set data. In this study, we propose an approach to the problem of out of set (OOS) data detection in the context of open-set language identification. In our approach, each unlabeled i-vector in the development set is given a per-class outlier score computed with the help of non-parametric Kolmogorov-Smirnov (KS) test. Detected OOS data from unlabeled development set is then used to train an additional model to represent OOS languages in the back-end. The proposed approach achieves a relative decrease of 16% in equal error rate (EER) over classical OOS detection methods, in discriminating in-set and OOS languages. Using support vector machine (SVM) as language back-end classifier, integrating the proposed method to the LID back-end yields 15% relative decrease in identification cost in comparison to using all the development set as OOS candidates.

1. Introduction

Language identification (LID) is the task of automatically identifying whether a known target language is being spoken in a given speech utterance [1]. Over the past years, several methods have been developed to perform LID tasks, including phonotactic [2] and acoustic ones [3]. The former uses phone recognizers to tokenize speech utterances into discrete units followed by n-gram statistics accumulation and language modeling back-end [4, 1]. The latter uses spectral characteristics of languages in a form of acoustic features such as shifted delta cepstral (SDC) coefficients [5, 6]. Gaussian mixture models (GMMs) [7] and support vector machines (SVMs) [8] are often used as classifiers. Recently, i-vectors [9] based on bottleneck features [10] have also been extensively explored.

State-of-the-art LID systems [11, 12, 10] achieve high identification accuracy in *closed-set* tasks, where the language of a test segment corresponds to one of the known target (in-set) languages. But in *open-set* LID tasks, where the language of a test segment might not be any of the in-set languages, accuracy often degrades considerably [13, 14]. In open-set LID, the objective is to classify a test segment into one of the pre-defined in-set languages or a single *out-of-set* (OOS) language (or model). Open-set LID is more applicable in real-life scenarios, where speech may come from any language. For example, in multi-lingual audio streaming and broadcasting, it is necessary to filter languages which do not belong to any of the modeled target languages [15].

Different approaches have been explored for OOS modeling both in open-set speaker identification (SID) and LID systems. In the context of open-set SID, the objective is to decide whether to accept or reject a speaker as being one of the enrolled speakers. The authors of [16, 17] used the knowledge of universal background model (UBM) to represent the OOS speakers. Each in-set speaker is modeled using Gaussian mixture models (GMMs) with a UBM and maximum a posteriori (MAP) speaker adaptation [18]. During classification, if any of the in-set speakers is selected, the test speaker is labeled as in-set; otherwise, the UBM has the highest score and the test speaker is classified as OOS. Authors in [19] proposed a system which first finds the best-matched model for a test speaker using vector quantization (VQ) based recognition system [20]. Then, a set-score is formed using support vector machine (SVM) classifier. Finally, a vector distance measurement for each enrolled speaker is used to accept or reject the test speaker as in-set class.

A few prior studies have been carried out on open-set LID tasks as well, as summarized in Table 1. To model OOS languages in open-set LID tasks, some approaches make use of additional OOS speech data derived from languages different from the target (in-set) languages [21, 22]. This additional OOS data is then used for training an OOS model. Obtaining additional data is often done in a supervised or semi-supervised way, which can be time consuming or leads to further sub-problems such as representative data selection to model the OOS languages. The authors of [21] proposed a method for compact OOS candidate language selection based on knowledge of world-language distance. In specific, candidate OOS data came from different language families having different prosody characteristics from the target languages. This method achieved 8.4% relative improvement in classification performance over a baseline system with a random selection of OOS candidates. In [22], a target-independent (TI) Gaussian was trained using development data of all target languages, to represent the OOS languages. Further, adopting maximum mutual information (MMI) [23] approach in [14] allows training an additional OOS Gaussian model using only in-set data. The trained OOS model improved the detection cost considerably, however, the impact of training the OOS model using actual OOS data was not investigated.

A practical key question in representing the OOS data is how to select the most representative OOS candidates to model OOS languages from a large set of unlabeled data. While random selection might be one option, in this study we attempt to specifically identify ”higher quality” OOS utterances. To achieve this, we present a simple approach to find such OOS candidates in i-vector space [9], based on non-parametric Kolmogorov-Smirnov (KS) test [25, 26]. It gives each i-vector a per-class outlier score representing the confidence that an i-vector corresponds to an OOS language. This approach is fast and in contrast to [21, 22], requires no prior language labels of the additional data which may not be available in the real-world applications [27].

Table 1: Summary of the previous studies on open-set language identification task. Different approaches for selecting out-of-set (OOS) data include using in-set data [14], using all development data [22], selecting labeled out-of-set data [21], pooling additional data [13, 24] and finally selecting from unlabeled data (present study).

Study	**Data**	**OOS selection**	**OOS modeling**
Zhang and Hansen [21]	NIST LRE 2009	Supervised candidate selection	General Gaussian back-end
BenZeghiba *et al.* [22]	NIST LRE 2007	All development data as OOS	General Gaussian back-end
McCree [14]	NIST LRE 2011	No OOS detection	Gaussian discriminative training using in-set data
Torres-Carrasquillo *et al.* [13]	NIST LRE 2009	Additional OOS data	Several spectral and token classifiers
Torres-Carrasquillo *et al.* [24]	NIST LRE 2007	Additional OOS data	Several spectral and token classifiers
Present study	NIST i-vector challenge 2015	OOS selection from unlabeled data	Gaussian, cosine and SVM classifiers

2. Open-set language identification

In closed-set LID, the objective is to classify a test segment X into one of the pre-defined set of target (in-set) languages $\{L_m | m = 1, ..., M\}$, where M is total number of target languages. To classify X, the decision of the most similar language $\tilde{L}$ is chosen to maximize the *a posteriori* probability [28],

$$\tilde{L} = \underset{1 \le m \le M}{\operatorname{argmax}} \, p(L_m|X) = \underset{1 \le m \le M}{\operatorname{argmax}} \, p(X|L_m)p(L_m), \qquad (1)$$

where the language likelihood $p(X|L_m)$ and language *a priori* probability $p(L_m)$ are assumed known. In *open-set* LID, the objective is to classify X into one of the $M+1$ languages, with M in-set languages and a single additional OOS language (or model).

Figure 1 shows a block diagram of a general open-set LID system used in this paper. OOS data selection block performs OOS detection on unlabeled development data to find best representative OOS data for training an additional OOS model.

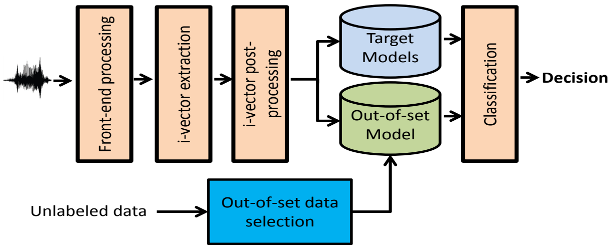


Figure 1: Block diagram of open-set language identification. The best-limited out-of-set (OOS) candidates are selected from the unlabeled development data for OOS modeling. We propose a simple method based on Kolmogorov-Smirnov test to find out-of-set data in the i-vector space.

In this study, we use i-vectors to represent utterances and consider the following three back-end language classifiers: **Gaussian** [22, 29], **cosine** [30] and **SVM** scoring [8, 31], to model both the target and the OOS languages. In the first case, for a given test i-vector, $\mathbf{w}_{\text{test}}$, the log-likelihood for a target language m is computed as

$$ll^m_{\mathbf{w}_{\text{test}}} = (\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_m)^T \mathbf{w}_{\text{test}} - \frac{1}{2}\boldsymbol{\mu}_m^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_m \qquad (2)$$

where $\boldsymbol{\mu}_m$ is the sample mean vector of target language m, and $\boldsymbol{\Sigma}$ is a shared covariance matrix common for all the languages. Having access to the training i-vectors, we form the maximum likelihood estimates of $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}_m$'s, and use Eq. (2) to compute a language similarity score.

Cosine scoring is a dot product between test i-vector, $\mathbf{w}_{\text{test}}$, and language model mean, $\boldsymbol{\mu}_m$

$$\text{score}^m_{\mathbf{w}_{\text{test}}} = \frac{\mathbf{w}^T_{\text{test}} \; \boldsymbol{\mu}_m}{||\mathbf{w}_{\text{test}}|| \; ||\boldsymbol{\mu}_m||}. \qquad (3)$$

In addition, we used one-versus-all version of support vector machine (SVM) classifier with second order polynomial kernel [31] after experimenting with different kernel types. In the training phase, all samples of a target language and OOS languages are considered as positive instances with all the other languages corresponding to negative instances. The number of class separators equals the number of target languages plus one, the last coming from the OOS model. During testing phase, the highest score of a separator determines the class label of a test segment.

A simple LID baseline system is to treat the problem as a closed-set task without the OOS model. NIST has provided such a system in the download package of the recent 2015 language recognition i-vector challenge [32]. It is based on cosine scoring in which development data is used to estimate global mean and covariance to center and whiten the evaluation i-vectors. We will refer to this closed-set LID system as the **NIST baseline** in our results, in contrast to the open-set systems containing an additional OOS model.

3. Out-of-set data selection for OOS modeling

The objective in OOS detection is to assign each i-vector with an *outlier score*, higher value indicating higher confidence that the i-vector is an OOS observation (none of the known target languages). Since the main aim of this study is to select most representative OOS candidates to model OOS languages, we investigate three commonly used OOS detection methods, in general outlier detection context, as our baselines: (i) one-class SVM, (ii) k-nearest neighbour (kNN) and (iii) distance to cluster centroid. Each of these methods provides an outlier score for each of the scored unlabeled utterances. Then, for the purpose of OOS modeling (Figure 1), we apply 3-sigma-rule [33] for OOS selection, provided that the distribution of outlier scores for these three methods can be assumed normal.

3.1. One-class SVM

SVMs [34] are most commonly used as two-class classifiers. An SVM projects the data into a high-dimensional space and finds a linear separator between classes. In contrast, *one*-class SVM was proposed for out-of-set detection in [35]. In the training phase, the detector constructs a decision boundary to achieve maximum separation between the training points and the origin. A given unlabeled utterance is then projected into the same high-dimensional space. The distance between the unlabeled utterance and the linear separator is used as the outlier score. We use LIBSVM[1] (version 3.21) to train an individual one-class SVM for each in-set class using the polynomial kernel [34] and the default parameters of the software package. The maximum score over in-set languages determines the outlier score for a given unlabeled utterance.

3.2. K-nearest neighbour (kNN)

In this technique, the outlier score for an observation is computed by the sum of its distances from its k nearest neighbours [36]. In this study, for each unlabeled utterance, the outlier scores are computed using k = 3 within each in-set language using Euclidean distance. Then, the maximum of outlier scores over all the in-set languages is used as the outlier score for that utterance.

3.3. Distance to class centroid

This is a simple classical approach to detect OOS data [37]. We assume that OOS data are far away from the class centroids. For instance, if the data follows a normal distribution, observations beyond two or three standard deviations above and below the class mean can be considered as OOS data [37]. This technique consists of two steps. First, the centroid of each in-set language is computed. Then, the distance between a data to the class centroid is computed as the outlier score. The maximum distance over all the in-set languages determines the outlier score for a given unlabeled utterance. We consider the in-set languages as different classes and the mean of each class as class centroids. Euclidean distance is chosen to compute the distance between each test data and the class means.

4. Proposed method

By now we have reviewed three commonly used OOS detection methods. Here we propose a simple and effective technique to find OOS data in the i-vector space. To this end, we adopt the non-parametric *Kolmogorov-Smirnov* (KS) test [25, 26]. It is used to decide whether a sample is drawn from a population with a known distribution (one-sample KS test) or to compare whether two samples have the same underlying distribution (two-sample KS test).

For any i-vector, $\mathbf{w}_i$, the distances of $\mathbf{w}_i$ to other i-vectors in language m has an empirical cumulative distribution function (ECDF) $F_{\mathbf{w}_i}(x)$ evaluated at x. The KS statistic between i-vector $\mathbf{w}_i$ and any other i-vector $\mathbf{w}_j$ in m can be computed by

$$\mathrm{KS}(\mathbf{w}_i, \mathbf{w}_j) = \max_x |F_{\mathbf{w}_i}(x) - F_{\mathbf{w}_j}(x)| \tag{4}$$

Given language m with the total number of instances N, the outlier score for i-vector $\mathbf{w}_i$ is then defined as the average of these KS test statistics:

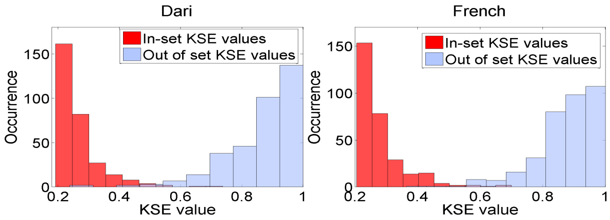
$$\mathrm{KSE}(\mathbf{w}_i) = \frac{1}{N-1} \sum_{\substack{j=1 \\ j \neq i}}^{N} \mathrm{KS}(\mathbf{w}_i, \mathbf{w}_j) \tag{5}$$

The average of KS statistics in Eq. (5), lies between 0 and 1; value close to 1 correspond to points with higher likelihood of being an OOS. Algorithm 1 shows a pseudo-code for computing the outlier score for a particular unlabeled i-vector.

Algorithm 1 Outlier score computation for an unlabeled i-vector using KSE.

```
Let $L = \{l_1, l_2, \dots, l_M\}$ be the set of $M$ in-set languages
Let $W_m = \{\mathbf{w}_{m1}, \mathbf{w}_{m2}, \dots, \mathbf{w}_{mN}\}$ be the set of i-vectors in in-set language $l_m$
Input $\mathbf{w}$ as an unlabeled i-vector
for $l_m \in L$ do
    $temp \leftarrow 0$
    for $\mathbf{w}_{mk} \in W_m$ do
        $KS \leftarrow$ compute KS value between $\mathbf{w}$ and $\mathbf{w}_{mk}$ using Eq. (4)
        $temp \leftarrow temp + KS$
    end for
    $KSE[m] \leftarrow$ divide $temp$ by $N-1$, Eq. (5)
end for
$outlierscore \leftarrow$ Multiply $KSE[m]$ by -1 and select the maximum value
```

Figure 2 shows the distribution of per-language (in-set) and OOS KSE values for Dari and French. For inset values, only i-vectors of these languages were used to plot the distributions (in other words, i-vectors i and j in Eqs. (4) and (5) are both from the same languages). KSEs within each language have values close to zero. For the OOS KSE values in Figure 2, a set of i-vectors which do not belong to these languages were used to plot the same distributions (in other words, i-vector i does not belong to the language class of i-vector j, in Eqs. (4) and (5)). These i-vectors are considered as OOS to these languages. As expected, the KSE values tend to values close to 1.



(a)

(b)

Figure 2: Distribution of in-set and OOS KSE values for two different languages, a) Dari and b) French. KSEs within each language have values close to zero. KSE values for OOS i-vectors tend towards one.

The Table 2 further demonstrates how we label data to evaluate our OOS detectors. Let us consider five i-vectors and their computed KSE values, given three in-set languages. The first three rows correspond to in-set utterances and the last two rows to OOS utterances. If the true language is one of the inset languages, label is set to 1 (e.g. the first row of Table 2), and to 0 otherwise (e.g. the last row of Table 2). The KSE values of each unlabeled utterance is multiplied by -1 and the maximum value is selected as the outlier score.

Following this method, we use box plot [33] to select OOS i-vectors. Box plot uses the median and the lower and upper quartiles defined as 25th and 75th percentiles. The lower quartile, median and upper quartile are often denoted by Q1, Q2 and Q3, respectively. In this study, unlabeled i-vectors with outlier scores above a threshold set at, $\mathrm{Q3} + 2.5 \times \mathrm{IQ}$, are selected for OOS modeling. Interquartile range or IQ denotes the difference $(\mathrm{Q3} - \mathrm{Q1})$.

[1] https://www.csie.ntu.edu.tw/~cjlin/libsvm/

Table 2: Example of test utterance labeling for the evaluation of OOS data detection task given multiple inset languages. KSE values for each data is computed according to Eq. (5).

Data_Id	True language	KSE values			In-set/OOS
		Greek	Dari	Urdu	
1	Greek	**0.29**	0.82	0.84	Inset
2	Dari	0.91	**0.11**	0.79	Inset
3	Urdu	0.85	0.92	**0.21**	Inset
4	Spanish	0.74	0.79	**0.64**	OOS
5	Farsi	0.81	**0.56**	0.77	OOS

5. Experimental set-up

5.1. Training, development and evaluation data

In this study, we used i-vectors provided by the National Institute of Standards and Technology (NIST) in their 2015 language i-vector machine learning challenge [32]. It is based on the i-vector system developed by the Johns Hopkins University Human Language Technology Center of Excellence in conjunction with MIT Lincoln Laboratory [6]. Table 3 shows the distribution of development, training and test sets provided in the challenge. The development set consists of 6500 unlabeled i-vectors intended for general system tuning. The training set consists of a set of 300 i-vectors for each of the 50 target languages, corresponding to 15000 training utterances in total. The test segments include 6500 unlabeled i-vectors corresponding to all of the target languages and an unspecified number of OOS languages. The i-vectors are of dimensionality 400.

Table 3: Distribution of training, development and test sets from the NIST 2015 language i-vector machine learning challenge. The i-vectors are derived from conversational telephone and narrowband broadcast speech data.

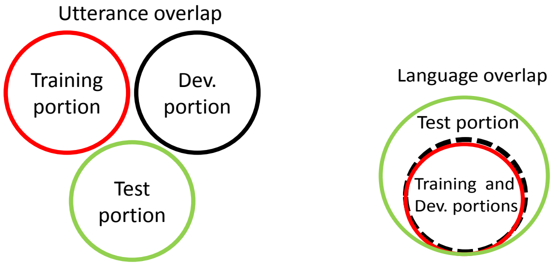
Dataset	#i-vectors	#languages	label
Training set	15000	50	labeled
Development set	6500	*n/a*	unlabeled
Test set	6500	50+OOS	labeled

To evaluate the OOS detection methods, we need both inset and OOS data. Since only the training set has labels, we further split it into three portions of non-overlapped utterances. We name them a development, training and test portion to make a distinction between them and the original training, development and test sets provided by NIST. Table 4 shows the distribution of these portions in our study. Training and development portions include non-overlapped utterances of same languages. These languages are called in-set languages and correspond to 30 different languages. Test portion consists of utterances corresponding to all of those 30 in-set languages plus utterances from 20 additional languages. We call these 20 languages as OOS languages and their corresponding utterances as OOS data.

Figure 3 further shows the Venn diagram illustrating the data overlap, i.e. individual utterances and languages, between training, development and test portions. The development portion is used for general OOS data detection tuning, such as parameter setting for one-class SVM and threshold setting to identify OOS data in the LID task. Training and test portions are used for building and evaluating OOS data detectors, respectively.

Table 4: Distribution of development, training and test portions for the out-of-set (OOS) data detection task. All portions are subsets of the original NIST 2015 LRE i-vector challenge training set.

	In-set	Out-of-set
Total number of languages	30	20
Count of dev. portion files	1500	—
Count of training portion files	6000	—
Count of test portion files	1500	6000



(a) (b)

Figure 3: Venn diagram illustrating the data overlap between training, development and test portions, all being subsets of the original NIST 2015 LRE i-vector challenge training set. a) Utterance overlap. b) Language overlap.

5.2. i-Vector post-processing

The sample mean and the sample covariance of the unlabeled development set are computed to center and whiten all the i-vectors [38]. Then, length-normalization [38] is applied to project all the i-vectors onto a unit ball. These i-vectors are then further transformed using principal component analysis (PCA) [39], keeping the 99% of the cumulative variance. The resulting i-vector dimensionality is 391, just slightly smaller than original 400. Then, linear discriminant analysis (LDA) [6] is applied to reduce the dimensionality of i-vectors to the maximum number of classes minus one, in our case 49 dimensions. Following PCA and LDA, within-class covariance normalization (WCCN) [40] is used as a supervised transformation technique to further suppress unwanted within-language variation. For open-set LID, the projection matrices of PCA, LDA and WCCN are computed using the training set. The order of post-processing techniques follows the same order as [6].

5.3. Tasks and performance measure

We use detection error tradeoff (DET) curve [41] to evaluate the OOS data detection performance. It plots the false acceptance rate (FAR) versus false rejection rate (FRR), using a normal deviate scale. Here the task is to identify those test portion data which do not conform to any of the training portion classes.

The performance measure of open-set LID task as defined in the NIST 2015 language recognition i-vector challenge task is defined as follows [32]:

$$\text{Cost} = \frac{(1-P_{\text{oos}})}{N}\sum_{k=1}^{N} P_{\text{error}}(k) + P_{\text{oos}} \times P_{\text{error}}(\text{oos}) \quad (6)$$

where $P_{\text{error}}(k) = (\frac{\#errors\_class\_k}{\#trials\_class\_k})$, $N = 50$, and $P_{\text{oos}} = 0.23$.

Open-set LID is performed using the training and test sets (not portions) for model training and evaluation, respectively. Detected OOS data from the development set is then used for OOS modeling.

6. Results

In order to evaluate our proposed OOS selection method, we separate the experiments into two parts. First, we evaluate the performance of the proposed and the baseline OOS detectors. Then, we assess the open-set language identification task with the OOS model trained by the additional data selected by our proposed OOS detector.

6.1. Stand-alone out-of-set data selection

Figure 4 shows the impact of parameter k on kNN-based OOS detection task, in terms of EER. As shown, no considerable change is observed by changing the value of k. For the remaining experiments, we arbitrarily fix $k = 3$.

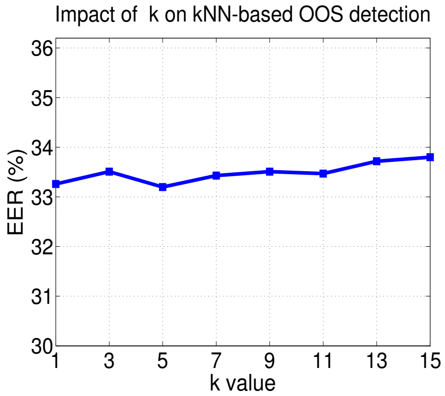


Figure 4: Impact of k value on kNN-based OOS detection method in terms of EER (%). No improvement is observed by increasing k.

Next, as KSE is based on the distribution of distances between points, here we study the impact of different distance metrics [42] on our proposed OOS detector. To this end, we vary the distance metric used in computing ECDFs in Eq. (4). Figure 5 shows the results. Euclidean and city-block distances achieve the highest performance with EERs of 28.80% and 28.46%, respectively. For the cosine distance with EER of 32.27% and Pearson correlation distance with EER of 32.35%, the performance degradation is more pronounced. For the remaining experiments, we fix the Euclidean distance metric.

Figure 6 shows the DET curve comparison between the proposed KSE and the three baseline methods on the test portion data. The results indicate that the proposed KSE method outperforms the baselines in terms of EER. KSE outperforms kNN and one-class SVM by 14% and 16% relative EER reductions, respectively. From the baselines, the distance to class mean method obtains the lowest performance with EER of 36.34%.

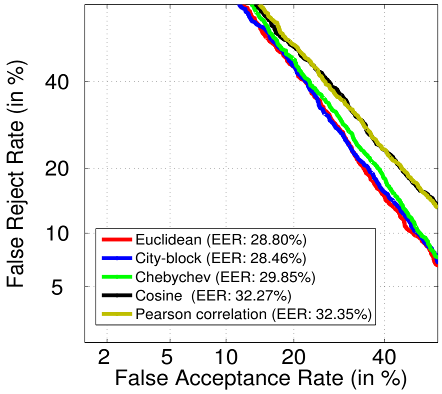


Figure 5: Performance of proposed OOS detection method under different distance metrics. Euclidean and city-block distances achieve the highest performance.

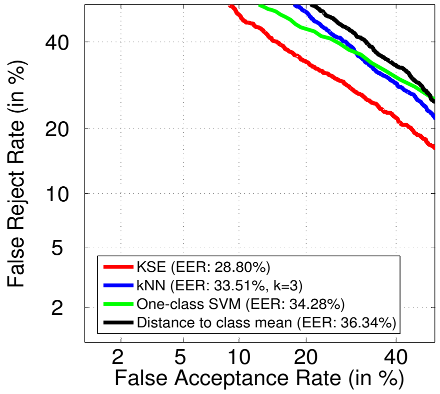


Figure 6: Comparison between the performance of the proposed OOS detection and three different baseline methods. KSE method shows the best performance compared to baseline methods.

Now we turn our attention to the effects of system fusion on the OOS detection performance at the score level. To this end, we adopt linear score fusion function optimized with the cross-entropy objective [43] using the BOSARIS Toolkit [44]. We use our development portion to find optimal classifier weights. Figure 7 shows the results of fusion of KSE to baseline OOS detection methods (2-way score fusion) on the test portion data. Interestingly, fusion of KSE to baseline systems improves the accuracy substantially. A relative decrease of 27% over KSE is achieved by fusing KSE and one-class SVM, yielding EER of 20.93%. EER of 28.25% is obtained by fusing KSE and kNN, yielding relative decrease of 2% and 16% over KSE and kNN, respectively. Fusion of all four methods (4-way score fusion) achieves EER of 22.09%, i.e. relative decrease of 23% and 27% over KSE and fusion of all baseline OOS detection methods (3-way score fusion), respectively.

6.2. Language identification

Up to this point, we have discussed OOS data detection accuracy. Now we turn our attention to the full language identification system in Figure 1 with the OOS model being trained by the additional data selected using one of the OOS detection methods. Table 5 shows the identification results for both closed-set

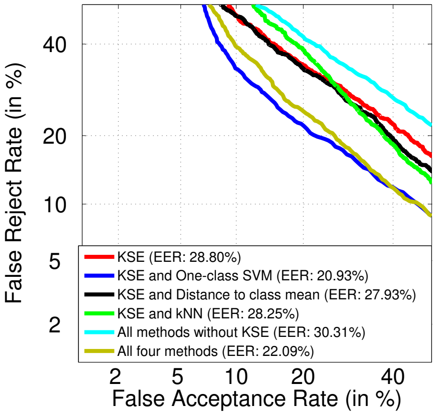


Figure 7: Fusion of KSE to baseline OOS detection methods. Fusion of KSE to one-class SVM yields the best performance. "All methods without KSE" refers to the score fusion of one-class SVM, kNN, and distance to class mean methods (3-way score fusion). "All four methods" indicates fusion of all four methods including KSE (4-way score fusion).

and open-set LID systems for different classifiers. The rows of Table 5 differ based on the data selected for target-independent OOS modeling. Rows 2 and 3 correspond to systems in which the data of *all* the training and development sets are used for OOS modeling, respectively, inspired by the work described in [22]. Row 4 corresponds to pooling the data in both development and training sets. The proposed selection method refers to OOS data selection using KSE. Finally, for reference purposes, the last row shows the closed-set results, where the language of a test segment can be only one of the target languages (no OOS modeling is performed).

We observe, firstly, that integrating the proposed selection method to open-set LID system with an SVM language classifier outperforms the other systems. The lowest identification cost achieved, 26.61, outperforms the NIST baseline system by 33% relative improvement. Using all the training or development data for OOS modeling does not necessarily lead to considerable improvement over closed-set results. For example, taking Gaussian scoring and training data (second row), identification cost decreases from 37.07 in closed-set to 34.15 in open-set, yielding a relative improvement of 8%. Furthermore, assuming an open-set LID system based on random OOS data selection (first row), the proposed method achieves a relative improvement of 17% using SVM language classifier[2]. It is worth mentioning that since test set contains a considerable amount of OOS utterances, the closed-set LID system incorrectly classifies them as one of the target languages. This explains why the closed-set LID system generally underperforms the open-set one.

Now, we treat the open-set LID as a binary classification task, discriminating in-set and OOS test data. In-set test data refers to those test files having the same language labels as one of the target languages. Table 6 shows the confusion matrix of in-set/OOS classification using KSE and three different language classifiers. Results correspond to the fifth row of Table 5. The results indicate that from 1500 OOS test data, 1012 are classified correctly as OOS using KSE with SVM language classifier. This number is 317 and 49 for Gaussian and cosine scoring, respectively.

Table 5: Language identification results for both open-set and closed-set set-ups. Rows differ based on the data used for OOS modeling. Results are reported based on identification cost, lower cost indicating higher performance. Numbers in parentheses indicate amounts of selected data for OOS modeling. The results are reported from the evaluation online system provided by the NIST in the i-vector challenge.

Data selected for OOS modeling (#)	Cosine	Gaussian	SVM
Random (1067)	36.25	34.20	32.11
Training (15000)	36.35	34.15	32.61
Development [22] (6431)	36.10	32.87	31.23
Training+Dev. (21431)	36.46	33.38	31.74
proposed selection method (1067)	34.28	32.23	**26.61**
Closed-set (no OOS model)	39.59*	37.07	37.23

*From the NIST baseline result

Table 6: Confusion tables for in-set and OOS classification using KSE with three different language classifiers corresponding to the fifth row of Table 5. In total, test set corresponds to 5000 and 1500 in-set and OOS data, respectively.

(a) SVM scoring

True Pred.	Inset	OOS
Inset	4134	866
OOS	488	1012

(b) Gaussian scoring

True Pred.	Inset	OOS
Inset	4867	133
OOS	1183	317

(c) Cosine scoring

True Pred.	Inset	OOS
Inset	4999	1
OOS	1451	49

Fixing SVM as the best language classifier, Table 7 compares the results of using different OOS detectors for OOS modeling in our open-set LID task. For comparison, we also include the closed-set LID results based on SVM in the last row of Table 7. The open-set LID system based on KSE outperforms the other methods, in terms of identification cost. Using KSE as an OOS detector brings relative improvements of 9% and 13% over kNN and one-class SVM, respectively.

Table 7: Open-set language identification results for different OOS data selection methods using SVM language classifier. Results are reported based on identification cost, lower cost indicating higher performance. The results are reported from the evaluation online system provided by the NIST in the i-vector challenge.

Method used for OOS modeling	Cost
KSE	**26.61**
kNN	29.33
One-class SVM	30.66
Distance to class mean	31.48
Closed-set	37.23

[2] Training the OOS model with known identities of OOS languages was not possible since NIST had not provided the class labels of the development set utterances.

7. Conclusion

We focused on the problem of OOS data selection in the i-vector space in the context of open-set LID problem. We proposed an approach based on non-parametric Kolmogorov-Smirnov test to effectively select OOS candidates from an unlabeled development set. Our proposed OOS detection method outperforms the one-class SVM baseline by 16% relative improvement, in terms of EER. We then used OOS candidates to train an additional model to represent the OOS languages in the open-set LID task. The baseline system was realized by using all the development and/or training data as OOS candidates. Using SVM as language classifier, with the proposed OOS data selection method, identification cost was relatively decreased by 15% over using all the development set as OOS candidates in the open-set LID task.

In our future work, we plan to explore possible extensions of the Kolmogorov-Smirnov test, such as weighted Kolmogorov-Smirnov test, to improve OOS detection accuracy. In addition, we will investigate clustering and modeling of KSE scores.

8. References

[1] K.J. Han and J. Pelecanos, "Frame-based phonotactic language identification," in *Proc . of SLT*, 2012, pp. 303–306.

[2] H. Li and B. Ma, "A phonotactic language model for spoken language identification," in *Proc. of ACL*, 2005, pp. 515–522.

[3] N. Brümmer, A. Strasheim, V. Hubeika, P. Matějka, L. Burget, and O. Glembek, "Discriminative acoustic language recognition via channel-compensated GMM statistics," in *Proc. of INTERSPEECH*, 2009, pp. 2187–2190.

[4] J. L Gauvain, A. Messaoudi, and H. Schwenk, "Language recognition using phone latices," in *Proc. of INTERSPEECH*, 2004, pp. 1283–1286.

[5] H. Li, B. Ma, and C.-H. Lee, "A vector space modeling approach to spoken language identification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 1, pp. 271–284, 2007.

[6] N. Dehak, P. A. Torres-Carrasquillo, D. A. Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction," in *Proc. of INTERSPEECH*, 2011, pp. 857–860.

[7] P. A. Torres-Carrasquillo, E. Singer, M. A. Kohler, R. J. Greene, D. A. Reynolds, and J. R. Deller Jr., "Approaches to language identification using Gaussian mixture models and shifted delta cepstral features," in *Proc. of SLP*, 2002, pp. 89–92.

[8] W. M. Campbell, E. Singer, P. A. Torres-Carrasquillo, and D. A., "Language recognition with support vector machines," in *Proc. of Speaker Odyssey*, 2004, pp. 41–44.

[9] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[10] Y. Song, Xinhai Hong, B. Jiang, R. Cui, I. Vince McLoughlin, and L.-R. Dai, "Deep bottleneck network based i-vector representation for language identification," in *Proc. of INTERSPEECH*, 2015, pp. 398–402.

[11] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, D. Martinez, J. Gonzalez-Rodriguez, and P. Moreno, "Automatic language identification using deep neural networks," in *Proc. of ICASSP*, 2014, pp. 5337–5341.

[12] M. Van Segbroeck, R. Travadi, and S.S. Narayanan, "Rapid language identification," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 23, no. 7, pp. 1118–1129, 2015.

[13] P. A. Torres-Carrasquillo, E. Singer, T. P. Gleason, A. McCree, D. A. Reynolds, F. Richardson, and D. E. Sturim, "The MITLL NIST LRE 2009 language recognition system," in *Proc. of ICASSP*, 2010, pp. 4994–4997.

[14] A. McCree, "Multiclass discriminative training of i-vector language recognition," in *Proc. of Speaker Odyssey*, 2014, pp. 166–171.

[15] M. Adda-Decker, *Automatic Language Identification*, In Spoken Language Processing, J.J. Mariani ed., Wiley-ISTE, Chapter 8, 2009.

[16] J.H. L. Hansen, J.-W. Suh, and M. R. Leonard, "In-set/out-of-set speaker recognition in sustained acoustic scenarios using sparse data," *Speech Communication*, vol. 55, no. 6, pp. 769–781, 2013.

[17] P. Angkititrakul and J.H.L. Hansen, "Discriminative in-set/out-of-set speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 2, pp. 498–508, 2007.

[18] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[19] J. Deng and Q. Hu, "Open set text-independent speaker recognition based on set-score pattern classification," in *Proc. of ICASSP*, 2003, vol. 2, pp. II–73–6 vol.2.

[20] J. Pelecanos, S. Myers, S. Sridharan, and V. Chandran, "Vector quantization based Gaussian modeling for speaker verification," in *Proc. of ICPR*, 2000, vol. 3, pp. 294–297 vol.3.

[21] Q. Zhang and J. H. L. Hansen, "Training candidate selection for effective rejection in open-set language identification," in *Proc. of SLT*, 2014, pp. 384–389.

[22] M.F. BenZeghiba, J. Gauvain, and L. Lamel, "Gaussian backend design for open-set language detection," in *Proc. of ICASSP*, 2009, pp. 4349–4352.

[23] L. Burget, P. Matejka, and J. Cernocky, "Discriminative training techniques for acoustic language identification," in *Proc. of ICASSP*, 2006, vol. 1, pp. I–I.

[24] P. A. Torres-Carrasquillo, E. Singer, T. P. Gleason, A. McCree, D. A. Reynolds, F. Richardson, W. Shen, and D. E. Sturim, "The MITLL NIST LRE 2007 language recognition system," in *Proc. of INTERSPEECH*, 2008, pp. 719–722.

[25] N. V. Smirnov, "Estimate of deviation between empirical distribution functions in two independent samples," *Bulletin Moscow University*, pp. 2: 3–16, 1933.

[26] M.S. Kim, "Robust, scalable anomaly detection for large collections of images," in *Proc. of SocialCom*, 2013, pp. 1054–1058.

[27] H. Lee, *Unsupervised Feature Learning Via Sparse Hierarchical Representations*, Stanford University, 2010.

[28] M.A. Zissman, "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 4, no. 1, pp. 31–34, 1996.

[29] M. F. BenZeghiba, J.-L. Gauvain, and L. Lamel, "Language score calibration using adapted Gaussian backend.," in *Proc. of INTERSPEECH*, 2009, pp. 2191–2194.

[30] N. Dehak, R. Dehak, P. Kenny, N. Brümmer, 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*, 2009, pp. 1559–1562.

[31] S. Yaman, J. W. Pelecanos, and M. K. Omar, "On the use of non-linear polynomial kernel svms in language recognition," in *Proc. of INTERSPEECH*, 2012, pp. 2053–2056.

[32] "The 2015 language recognition i-vector machine learning challenge," `https://ivectorchallenge.nist.gov/`.

[33] M. Natrella, *NIST/SEMATECH e-Handbook of Statistical Methods*, NIST/SEMATECH, chapter 7, 2010.

[34] I. Steinwart and A. Christmann, *Support Vector Machines*, Springer, 2008.

[35] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1472, 2001.

[36] J. Zhang and H. H. Wang, "Detecting outlying subspaces for high-dimensional data: the new task, algorithms, and performance," *Knowledge and Information Systems*, vol. 10, no. 3, pp. 333–355, 2006.

[37] J.R. Kornycky and D.C. Ferrier, *Method for determining whether a measured signal matches a model signal*, Google Patents, 2014.

[38] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proc. of INTERSPEECH*, 2011, pp. 249–252.

[39] W. Rao and M.-W. Mak, "Alleviating the small sample-size problem in i-vector based speaker verification," in *Proc. of ISCSLP*, 2012, pp. 335–339.

[40] A. O. Hatch, S. S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," in *Proc. of INTERSPEECH*, 2006.

[41] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The det curve in assessment of detection task performance," in *Proc. of EUROSPEECH*, 1997, pp. 1895–1898.

[42] E. Deza and M.-M. Deza, *Dictionary of Distances*, Elsevier, 2006.

[43] N. Brümmer, L. Burget, J. H. Cernocky, O. Glembek, F. Grezl, M. Karafiat, D. A. van Leeuwen, P. Matejka, P. Schwarz, and A. Strasheim, "Fusion of heterogeneous speaker recognition systems in the STBU submission for the NIST speaker recognition evaluation 2006," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 15, no. 7, pp. 2072–2084, 2007.

[44] N. Brümmer and E. de Villiers, "The BOSARIS toolkit user guide: Theory, algorithms and code for binary classifier score processing,," in *Technichal Report*, 2011, p. [Online]. Available: https://sites.google.com/site/nikobrummer.