# Language Recognition via Ivectors and Dimensionality Reduction

*Najim Dehak*[1], *Pedro A.Torres-Carrasquillo*[2], *Douglas Reynolds*[2], *Reda Dehak*[3]

[1]MIT-CSAIL, Spoken Language System Group, Cambridge, MA, USA
[2]MIT Lincoln Laboratory, Human Language Technology Group, Lexington, MA, USA
[3]Laboratoire de Recherche et de Développement de l'EPITA (LRDE), Paris, France

najim@csail.mit.edu, ptorres@ll.mit.edu, dar@ll.mit.edu, reda@lrde.epita.fr

## Abstract

In this paper, a new language identification system is presented based on the total variability approach previously developed in the field of speaker identification. Various techniques are employed to extract the most salient features in the lower dimensional i-vector space and the system developed results in excellent performance on the 2009 LRE evaluation set without the need for any post-processing or backend techniques. Additional performance gains are observed when the system is combined with other acoustic systems.

**Index Terms**: Ivector representation, Support Vector Machines, Linear Discriminant Analysis, Neighborhood Component Analysis, Within Class Covariance Normalization.

## 1. Introduction

Language identification (LID) refers to the process of automatically identifying the language spoken in a speech sample usually under the assumption that a single language is present. Over the years a number of techniques have been developed to perform this task ranging from high level systems, typically focusing on phones and the frequency of the sequences of phones observed in each target language, to systems based on the spectral characteristics of each language usually referred to as acoustic systems. Gaussian mixture models (GMM) [1, 2] and support vector machines (SVM) [3] have been the classifiers of choice over recent years for acoustic modeling, consistently outperforming their high-level counterparts. For the GMM and SVM classifiers, a number of techniques developed within the speaker recognition area have shown excellent performance when applied to the language identification task. For example, techniques such as nuisance attribute projection (NAP) [3, 4] and factor analysis [5, 11] that have provided notable improvements over the last few years to speaker identification systems have also resulted in performance gains for acoustic language identification systems.

In this paper, we continue the trend of borrowing techniques developed for speaker recognition and applying them to the language identification task. In particular, we described the application of the i-vector or total variability space approach to the language identification task. The i-vector representation is a data-driven approach for feature extraction that provides an elegant and general framework for audio classification and identification. It consists of mapping a sequence of frames for a given utterance into a low-dimensional vector space, referred to as the total variability space, based on a factor analysis technique. We evaluate our system using different techniques of dimensionality reduction in order to compensate for the intersession effects. These approaches include Linear Discriminant Analysis (LDA), Neighborhood Component Analysis (NCA) [6] and their combination with Within Class Covariance Normalization (WCCN) [7]. Every approach defines a new basis that maximizes the discrimination between the different language classes based on the defined criterion.

The remainder of the paper is as follows. Section 2 describes the experimental system including the total variability approach and scoring mechanism with alternatives for processing the vectors within the lower dimensional space. Section 3 presents the experimental setup with section 4 describing the results obtained for the system including fusion with other acoustic systems. Section 5 includes conclusions and avenues for future work.

## 2. System description

### 2.1. Feature extraction

The feature extraction stage used in this work is similar to that employed in [8]. Speech is windowed at 20ms with a 10ms frame rate filtered through a mel-scale filter bank and then RASTA. Each vector is then converted into a 56-dimensional vector following a shifted delta cepstral parameterization using a 7-1-3-7 scheme and concatenation to the static cepstral coefficients. Speech activity detection is then applied and the speech is normalized following a standard normal distribution.

### 2.2. Total variability modeling

The total variability space or i-vector approach concept was first introduced in the context of speaker verification [9, 10]. This approach was motivated by the success of the Joint Factor Analysis [11], which models both speaker and intersession subspaces separately. Unlike JFA, the total variability approach models all the important variability in the same low dimensional subspace. The basic idea of the total variability space consists of adapting the Universal Background Model (UBM) (which is trained on all the available language data for this paper) to a set of given speech frames based on the eigenvoice adaptation technique in order to estimate the utterance dependent GMM. The eigenvoice adaptation technique operates on the assumption that all the pertinent variability is captured by a low rank rectangular matrix $T$ named the Total variability matrix. The GMM supervector (vector created by stacking all mean vectors from the GMM) for a given utterance can be modeled as follows

$$M = m + Tw + \epsilon \quad (1)$$

This work was sponsored by the Department of Defense under Air Force contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

where $m$ is the Universal Background Model supervector, the i-vector $w$ is a random vector having a normal distribution $\mathcal{N}(0, I)$, and the residual noise term $\epsilon \sim \mathcal{N}(0, \Sigma)$ models the variability not captured by the matrix $T$. In our new modeling, we apply an SVM directly to the low dimensional i-vector (which is the coordinate of the speech segment in the total variability space) instead of applying the SVM in the GMM supervector space as done in [12].

The process of training the total variability matrix $T$ is a little bit different compared to learning the eigenvoice adaptation matrix [13]. In eigenvoice training for speaker recognition, all the recordings of a given speaker are considered to belong to the same person; in the case of the total variability matrix however, we pretend that every utterance from a given speaker is produced by different speakers. If we follow the same total variability matrix training process for language identification, we assume that every utterance for a given language class is considered a different class. Additional details on the i-vector extraction procedure are described in [10].

### 2.3. Support Vector Machine and cosine kernel

Support vector machines are supervised binary classifiers. Proposed in [14], they are based on the idea of finding, from a set of learning examples $X = \{(w_1, y_1), (w_2, y_2), ..., (w_N, Y_N)\}$, the best linear separator $H$ for distinguishing between the positive examples ($y_i = +1$) and negative examples ($y_i = -1$). When the kernel function is used the SVM separator is defining as follow.

$$\begin{aligned} H : \mathbb{R}^N &\rightarrow \mathbb{R} \\ w &\mapsto H(w) = \sum_{i=1}^{m} \alpha_i^* y_i k(w, w_i) + b_0 \end{aligned} \tag{2}$$

where $\alpha_i^*$ and $b_0$ are the SVM parameters set during the training step. As in the case of speaker identification, we considered several kernel functions with the best set of results obtained by using the cosine kernel function. This kernel is linear and computed as follows:

$$k(w_1, w_2) = \frac{w_1^t \,.\, w_2}{\|w_1\| \; \|w_2\|} \tag{3}$$

where $w_1$ and $w_2$ correspond to two i-vectors. There are two strategies to extend the SVM approach to a multi-class problem. The first strategy is the one versus one separator, which consists of estimating a separator between the target language class and each of the competing classes with the final decision obtained by a majority vote over all classifiers. The second approach, which is used in our system, is based on the one versus all strategy. For each target class, we consider its samples as positive examples with all the other classes samples corresponding to negative examples. The number of separators in this approach corresponds to the number of classes. The class label of a given test sample is based on the separator that obtains the highest score.

### 2.4. Linear Discriminant Analysis

Linear discriminant analysis (LDA) is a very popular technique for dimension reduction in the machine learning field. It has the advantage of defining new axes that maximize the discrimination between the different classes. In the context of language recognition, each class represents a different language. The LDA procedure consists of finding the basis that maximizes the between classes variability while minimizing the intra-class variability. The LDA axes are then defined by a projection matrix A, which contains the eigenvectors corresponding to the highest eigenvalues in the decomposition. The solution is obtained by solving the general eigenvalue problem.

$$\Sigma_b v = \lambda \Sigma_w v \tag{4}$$

where $\lambda$ is the diagonal matrix of eigenvalues. The matrices $\Sigma_b$ and $\Sigma_w$ correspond to the between classes and within class covariance matrices, respectively.

$$\Sigma_b = \sum_{i=1}^{L} (w_i - \overline{w})(w_i - \overline{w})^t \tag{5}$$

$$\Sigma_w = \sum_{l=1}^{L} \frac{1}{n_l} \sum_{i=1}^{n_l} \left(w_i^l - \overline{w_l}\right)\left(w_i^l - \overline{w_l}\right)^t \tag{6}$$

where $\overline{w_l} = \frac{1}{n_l}\sum_{i=1}^{n_l} w_i^l$ is the mean of the i-vectors for each language class, $L$ is the total number of language classes and $n_l$ is the number of utterances for each language $l$. We assume that the mean vector of the entire i-vectors $\overline{w}$ is equal to the null vector since they have a standard Normal distribution $w \sim \mathcal{N}(0, I)$ with zero mean.

Based on the performance of the combination of the LDA and within class covariance normalization combination for speaker verification [7], we proposed two different combinations. The first combination is exactly the same LDA and WCCN combination as done in [9, 10]

$$k(w_1, w_2) = \frac{\left(A^t w_1\right)^t}{\sqrt{\left(A^t w_1\right)^t W^{-1} \left(A^t w_1\right)}} W^{-1} \frac{\left(A^t w_2\right)}{\sqrt{\left(A^t w_2\right)^t W^{-1} \left(A^t w_2\right)}} \tag{7}$$

where $W$ is the within class covariance matrix estimated as follows:

$$W = \frac{1}{L} \sum_{l=1}^{L} \frac{1}{n_l} \sum_{l=1}^{n_l} \left(w_i^l - \overline{w_l}\right)\left(w_i^l - \overline{w_l}\right)^t \tag{8}$$

where $A$ is the LDA projection matrix, $\overline{w_l} = \frac{1}{n_l}\sum_{i=1}^{n_l} A^t w_i^l$ is the mean of the LDA projected i-vectors for each language $l$, $L$ is the total number of language classes, and $n_l$ is the number of ivectors of each language $l$. The second combination uses the diagonal eigenvalues matrix to normalize the cosine kernel between two i-vectors $w_1$ and $w_2$.

$$k(w_1, w_2) = \frac{\left(A^t w_1\right)^t \lambda \left(A^t w_2\right)}{\sqrt{\left(A^t w_1\right)^t \lambda \left(A^t w_1\right)} \sqrt{\left(A^t w_2\right)^t \lambda \left(A^t w_2\right)}} \tag{9}$$

where $\lambda$ is the diagonal matrix of eigenvalues. Both kernel normalization matrices, WCCN and the diagonal eigenvalues matrix assign more importance to the dimensions with higher between classes variance during the cosine kernel computation.

### 2.5. Neighborhood component analysis

Neighborhood component analysis (NCA) is a dimension reduction technique [6]. It estimates a linear projection matrix based on optimizing the leave-one-out criteria of the nearest neighborhood classifier on a given training data. Given a set of i-vectors $\{w_1, w_2, ..., w_N\}$ of dimension $d$ and the corresponding language label set $\{y_1, y_2, ...., y_N\}$, the NCA approach learns a projection matrix $B$ of dimension $(p \times d)$ that defines a Mahalanobis distance metric, which maximizes the accuracy of the nearest neighbor classifier in the projected space.

$$d\left(w_i, w_j\right) = \left(Bw_i - Bw_j\right)^t \, \left(Bw_i - Bw_j\right) \qquad (10)$$

The differentiable optimization criterion of the NCA is based on a stochastic "soft" neighbor assignment in the projected space instead of using directly the k-nearest neighbored classifier. Every vector $i$ in the training set can select its neighbor $j$ with probability $P_{ij}$, which is a softmax over Euclidean distances in the transformed space. This probability is given by the following equation

$$P_{ij} = \frac{\exp\left(-\left\|Bw_i - Bw_j\right\|^2\right)}{\sum_{k \neq i} \exp\left(-\left\|Bw_i - Bw_k\right\|^2\right)} \qquad (11)$$

The NCA approach consists of maximizing the expected number of samples that are classified using the leave one out strategy on the training dataset. Lets define $p_i = \sum_{j \in C_i} p_{ij}$ which corresponds to the probability that a sample $i$ will be correctly classified and the set $C_i = \{j | \text{class}_i = \text{class}_j\}$ contains all the samples of the same class as vector $i$. The objective function that needs to be optimized can be defined as follows:

$$f\left(B\right) = \sum_i \sum j \in C_i p_{ij} = \sum_i P_i \qquad (12)$$

This objective function can be optimized by differentiating $f(B)$ with the respect to projection matrix $B$, which yields the following gradient optimization rule.

$$\frac{\partial f}{\partial B} = 2B \sum_i \left( p_i \sum_k p_{ik} w_{ik} w_{ik}^t - \sum_{j \in C_i} p_{ij} w_{ij} w_{ij}^t \right) \qquad (13)$$

where $w_{ij} = w_i - w_j$. It is clear that the function $f(B)$ is not a convex function. The choice of the initial matrix $B$ is crucial in the convergence of the algorithm. In our experiments, the first initialization of the matrix $B$ corresponds to the linear discriminant analysis matrix. Similar to LDA, we tested the combination of NCA and WCCN.

### 2.6. Backend

The backend system employed for the experiments described in this paper is similar to the backend discussed in [8] and consists of a per-system calibration stage followed by a linear fusion. As in [8], the calibration stage employs a single discriminatively trained Gaussian with shared covariance and is used for all durations through a duration based parametric function.

## 3. Experimental setup

The data used for development of the systems are also similar to that employed in [8] and includes two main sources of data, conversational telephone speech (CTS) and broadcast news (BN). The CTS partition includes data from multiple corpora including CallFriend, CallHome, Mixer, OHSU and the OGI-22 collections. The BN partition includes data from VoA as supplied by NIST and is processed as described in [8]. The data are pooled and then divided into two partitions, a development partition and a test partition with similar number of cuts on each set.

Table 1: Results obtained for various i-vector dimensions and speech durations. Performance is shown in EER.

| | 30s | 10s | 3s |
|---|---|---|---|
| Dim = 200 | 4.3% | 9.3% | 18.3% |
| Dim = 300 | 4.4% | 8.6% | 18.3% |
| Dim = 400 | **3.9%** | **8.2%** | **17.9%** |
| Dim = 500 | 4.0% | 8.5% | 17.6% |

Table 2: Comparison of various ivector systems using different intersession compensation techniques with GMM-MMI and SVM-GSV. The results are given in EER on the NIST 2009 LRE.

| | 30s | | 10s | | 3s | |
|---|---|---|---|---|---|---|
| | BB | AB | BB | AB | BB | AB |
| LDA | 4.6 | 2.4 | 9.2 | 4.8 | 19.2 | 14.2 |
| LDA+eigen | 4.1 | 2.4 | 8.1 | 4.8 | 18.1 | 14.2 |
| LDA+WCCN | 4.2 | 2.4 | 8.5 | 4.8 | 18.6 | 14.2 |
| NCA | 4.3 | 2.3 | 9.3 | 5.2 | 19.1 | 14.9 |
| NCA+WCCN | 3.9 | 2.3 | 8.6 | 5.2 | 18.5 | 14.9 |
| MMI | 7.9 | 2.3 | 10.8 | 4.4 | 17.9 | 12.9 |
| SVM-GSV | 7.5 | 2.3 | 11.2 | 5.0 | 20.4 | 15.4 |

The evaluation data used is the data defined by NIST for the 2009 LRE and includes evaluation segments for 30s, 10s and 3s and covers 23 language classes. For the results in this paper, we focus on the closed set problem and do not include the out-of-set data. The total variability space is based on an UBM comprised by 2048 Gaussian components.

## 4. Results

In this section results are reported for an SVM classifier across different dimensions of the total variability space. These experiments are carried out for all different durations described in the previous section. The purpose of these experiments is to define the optimal i-vector dimensionality for language identification. The results given in Table 1 are obtained without a backend post-processing stage.

The results show that the most consistent performance across all speech duration conditions is obtained with the 400 dimensional i-vectors. We will use this dimensionality for the remainder of the paper.

### 4.1. Dimensionality reduction techniques

We evaluated several dimensionality reduction approaches and results are presented in this section. All systems are based on SVM approaches. We found that the best performances using LDA and NCA were obtained when we reduce the dimension from 400 to 23. We also compared the i-vector approach with two other well known language identification systems. The first system is the GMM approach based on the Maximum Mutual Information (MMI) criteria [8]. The second system is a support vector machines system [7] based on the GMM supervector (GSV). The results are reported before (BB) and after (AB) applying the backend.

There are a number of interesting results observed in Table 2. First, the NCA and WCCN system outperforms all

Table 3: Score fusion results between i-vector systems with the GMM-MMI and SVM-GSV systems. The results are given in EER on the NIST 2009 LRE.

| **ivector + MMI + SVM-GSV** | 30s | 10s | 3s |
|---|---|---|---|
| LDA | 2.2% | **3.9%** | **11.8%** |
| LDA+eigen | 2.3% | 3.9% | 11.9% |
| LDA+WCCN | 2.3% | 3.9% | 11.9% |
| NCA | 2.2% | 4.0% | 12.0% |
| NCA+WCCN | 2.3% | 4.0% | 13.0% |
| GMM-MMI+SVM-GSV | **2.2%** | 3.9% | 12.1% |

other systems before the backend and results in similar performance to other systems on the 30s task. For the 10s and 3s tasks the best pre-backend performance is provided by the LDA+eigenvalue normalization system with post-backend performance being similar for the LDA systems. It is also clear that the i-vector systems outperform the MMI and GSV systems before backend post-processing likely providing better calibration without the need for a backend. However, the GMM-MMI system still provides the best performance after the backend particularly on 10s and 3s tasks.

### 4.2. Fusion

In this section, fusion results are presented between the i-vector systems, the GMM-MMI system and the SVM-GSV system. The fusion is based on logistic regression as presented in [8]. We also present pairwise fusion results for the best combination of a total variability system and either the GMM-MMI or the GSV system for additional comparisons. The results are given in Table 3.

The results show that adding the i-vector system in the fusion did not help a lot in both speech duration conditions 30s and 10s. However, we obtained more improvement in the 3s condition when the EER decrease from 12.1% without i-vector system to 11.8% with our system. We also notice that the fusion based on the i-vector with LDA only achieved the best performance in almost all conditions. Additional combination of the GSV system with the NCA system shows a minor gain on the 30s task.

## 5. Conclusions

In this paper, we have presented results for a language recognition system using a total variability subspace approach and various techniques for enhancing the discriminative power within the subspace. Results obtained are very competitive with state of the art acoustic LID systems and generally do not require backend post-processing to improve performance. The obtained results showed additional improvements when combined with these state of the art systems resulting in EER of 2.2% on the 2009 evaluation set.

In the future, we intend to continue exploring new ideas for constructing the variability subspace and other techniques for improving the discriminative power in the low dimensional subspace. Additionally, we would like to extend the evaluation to other classification techniques besides SVMs.

## 6. References

[1] L. Burget, P. Matejka, and J. Cernocky, “Discriminative Training Techniques for Acoustic Language Identification,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2006.

[2] V. Hubeika, L. Burget, P. Matejka, and P. Schwarz, “Discriminative Training and Channel Compensation for Acoustic Language Recognition,” Brisbane, AU, 2008.

[3] W. Campbell, J. Campbell, D. Reynolds, E. Singer, and P. Torres-Carrasquillo, “Support vector machines for speaker and language recognition,” *Computer Speech and Language*, vol. 20, no. 2-3, 2006.

[4] A. Solomonoff, C. Quillen, and W. M. Campbell, “Channel Compensation for SVM Speaker Recognition,” in *IEEE Odyssey: The Speaker and Language Recognition Workshop*, 2004.

[5] N. Brummer, A. Strasheim, V. Hubeika, P. Matejka, L. Burget, and O. Glembek, “Discriminative Acoustic Language Recognition via Channel-Compensated GMM Statistics,” Brighton, GB,, September 2009.

[6] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, “Neighbourhood Component Analysis,” in *Neural Information Processing Systems (NIPS)*, 2004.

[7] A. Hatch, S. Kajarekar, and A. Stolcke, “Within-Class Covariance Normalization for SVM-Based Speaker Recognition,” in *International Conference on Spoken Language Processing*, Pittsburgh, PA, USA, September 2006.

[8] P. Torres-Carrasquillo, E. Singer, T. Gleason, A. McCree, D. Reynolds, F. Richardson, and D. Sturim, “The MITLL NIST LRE 2009 Language Recognition System,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Dallas, TX, 2010, pp. 4994–4997.

[9] N. Dehak, R. Dehak, P. Kenny, N. Brummer, P. Ouellet, and P. Dumouchel, “Support Vector Machines versus Fast Scoring in the Low Dimensional Total Variability Space for Speaker Verification,” in *Interspeech*, Brighton, 2009.

[10] N. Dehak, P. Kenny, R. Dehak, P. Ouellet, and P. Dumouchel, “Front End Factor Analysis for Speaker Verification,” *to appear in IEEE Transactions on Audio, Speech and Language Processing*, 2011.

[11] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, “A Study of Interspeaker Variability in Speaker Verification,” *IEEE Transaction on Audio, Speech and Language*, vol. 16, no. 5, pp. 980–988, july 2008.

[12] W. Campbell, “A Covariance Kernel For SVM Language Recognition,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2008.

[13] P. Kenny, G. Boulianne, and P. Dumouchel, “Eigenvoice modeling with sparse training data,” *IEEE Transaction on Speech Audio Processing*, vol. 13, no. 3, may 2005.

[14] V. Vapnick, *The Nature of Statistical Learning*. Springer, 1995.