# Language Recognition in iVectors Space

*David Martínez*[1], *Oldřich Plchot*[2], *Lukáš Burget*[2], *Ondřej Glembek*[2] *and Pavel Matějka*[2]

[1]Aragon Institute for Engineering Research (I3A), University of Zaragoza, Spain
[2]Speech@FIT, Brno University of Technology, Czech Republic

david@unizar.es, {iplchot,burget,glembek,matejkap}@fit.vutbr.cz

## Abstract

The concept of so called iVectors, where each utterance is represented by fixed-length low-dimensional feature vector, has recently become very successfully in speaker verification. In this work, we apply the same idea in the context of Language Recognition (LR). To recognize language in the iVector space, we experiment with three different linear classifiers: one based on a generative model, where classes are modeled by Gaussian distributions with shared covariance matrix, and two discriminative classifiers, namely linear Support Vector Machine and Logistic Regression. The tests were performed on the NIST LRE 2009 dataset and the results were compared with state-of-the-art LR based on Joint Factor Analysis (JFA). While the iVector system offers better performance, it also seems to be complementary to JFA, as their fusion shows another improvement.

**Index Terms**: Acoustic Language Recognition, iVectors, Joint Factor Analysis.

## 1. Introduction

Joint Factor Analysis (JFA) [15], which is a statistical model originally proposed for Speaker Recognition, has become very successful also for acoustic Language Recognition (LR) [3, 2]. The idea behind JFA is to consider not only the inter-class variability in the space of model parameters (we have different model parameters for different languages in LR), but also the inter-session variability (parameters for a language can change from utterance to utterance because of the differences in channel, speaker, etc.). We will refer to the latter variability simply as *channel variability*. When the likelihood of a test utterance is evaluated for a certain language, the corresponding model is adapted to the channel of that test utterance. This is done by finding the point MAP (or ML) estimate of a low-dimensional latent variable vector - *channel factors*, which are coordinates in a highly channel-variable subspace of the model parameter space.

Recently, systems based on iVectors [4, 16] have provided superior performance in speaker recognition. iVector is a fixed-length low-dimensional vector, which is extracted for each utterance based on the JFA-like idea of estimating latent variables corresponding to high variability subspace. The principal difference from JFA is that we are not interested in evaluating the adapted model. Instead, the latent variables - iVectors - are used as features for another (possibly very simple) classifier. Also, the underlying model for iVector extraction does not attempt to separate inter-class and channel variability. Instead, it considerers only single *total variability* subspace corresponding to both sources of variability. The advantage is that the model for iVector extraction can be trained in unsupervised manner (without providing speaker or language identities for speaker or language recognition respectively). On the other hand, iVector contains information about both the class and the channel; this has to be taken into account in the following classifier.

Inspired by the success of iVectors in speaker recognition, we apply the same idea in the context of language recognition in this work. As a classifier in the iVector space, we use the linear generative model, where the distribution of iVectors for each language is Gaussian with full covariance matrix shared across languages. This model is analogue to Probabilistic Linear Discriminant Analysis (PLDA) [1], which is currently the most successful model for modeling iVectors in speaker recognition [16, 13]. Unlike in PLDA, we do not need to explicitly model distributions of class means. We deal here only with a closed-set problem, where means for a limited number of classes (languages) can be robustly obtained as the ML estimates. However, note that the PLDA approach, thanks to that inter-class distribution modeling, could be useful when dealing with an open-set LR problem, where also unknown out-of-set languages have to be detected.

Low dimensionality of iVectors makes it also convenient to apply discriminative classifiers. We have experimented with linear Support Vector Machines (SVM) and Logistic Regression in combination with Nuisance Attribute Projection (NAP) [11] as a channel compensation technique.

The performance of the proposed techniques is compared with state-of-the-art JFA based system on the NIST LRE 2009. On 30s condition, the best performing individual system is iVector based generative model, where $C_{avg} = 0.0188$ corresponds to 7% improvement over the JFA baseline. Further improvements (up to 18% over the JFA baseline) can be obtained by fusing the JFA and iVector based systems.

Note that in [9], another iVector based approach is applied to phonotactic language recognition, where recently proposed Subspace Multinomial Model [5] is used to extract iVector from phone n-gram counts.

The rest of the paper is organized as follows: in Section 2, iVectors fundamentals are revisited; in Section 3, the classifiers used for the experimentation are reviewed; in Section 4, the experimental setup is described; in Section 5, the results are presented; and in Section 6, the conclusions are derived.

## 2. iVectors

The iVector approach has become state-of-the-art in the speaker verification field [4] and, in this work, we show that it can be successfully applied also to language recognition. The approach provides an elegant way of reducing high-dimensional sequential input data to a low-dimensional fixed-length feature vector while retaining most of the relevant information. The main idea is that the language- and channel-dependent supervectors of concatenated Gaussian Mixture Model (GMM) means can be

modeled as

$$\mathbf{M} = \mathbf{m} + \mathbf{T}\mathbf{w}, \quad (1)$$

where $\mathbf{m}$ is the language- and channel-independent component of the mean supervector, $\mathbf{T}$ is a matrix of bases spanning the subspace covering the important variability (both speaker- and session-specific) in the supervector space, and $\mathbf{w}$ is a standard-normally distributed latent variable. For each observation sequence representing an utterance, our iVector is the Maximum A Posteriori (MAP) point estimate of the latent variable $\mathbf{w}$. Our iVector extractor training procedure is based on the efficient implementation suggested in [7].

# 3. Classifiers

## 3.1. Generative model

In the case of the generative model, distribution of iVectors for each language is modeled by a Gaussian distribution, where full covariance matrix is shared across all languages. For an iVector $\mathbf{w}$ corresponding to a test utterance, we evaluate log-likelihood for each language as:

$$\ln p(\mathbf{w}|l) = -\frac{1}{2}\mathbf{w}^T\boldsymbol{\Sigma}^{-1}\mathbf{w} + \mathbf{w}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_l - \frac{1}{2}\boldsymbol{\mu}_l^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_l + const,$$

where $\boldsymbol{\mu}_l$ is the mean vector for language $l$, $\boldsymbol{\Sigma}$ is the common covariance matrix and $const$ is a language- and iVector-independent constant. If the log-likelihoods $\ln p(\mathbf{w}|l)$ were directly used to decide about the language (or estimate the posterior probability of a language), the quadratic term $\mathbf{w}^T\boldsymbol{\Sigma}^{-1}\mathbf{w}$ could be ignored as it is independent of the class thanks to the shared covariance matrix. This would lead to linear classifier as the remaining terms are only linear in $\mathbf{w}$. In our case, however, the log likelihoods are used as inputs to another classifier, the calibration back-end described in section 4.3. For this reason, we include the quadratic term, and thus, we avoid the iVector (utterance) dependent shift in our scores.

## 3.2. Discriminative Classifiers

We have also experimented with discriminative linear classifiers: linear Support Vector Machines (SVM) and Logistic Regression with L2 regularization. In both cases, binary classifiers are trained and one-versus-all strategy is used to obtain scores for all languages. We use implementations from LIBSVM [10] and LIBLINEAR [12] for SVM and logistic regression, respectively. Although, we have used binary logistic regression in our experiments, our problem could be addressed more directly using a single multi-class logistic regression classifier. For example, the experiments in [3], where multi-class logistic regression was applied to recognize languages from GMM mean supervectors, can be now carried out in iVector space with significantly reduced computational cost and space complexity.

# 4. Experimental Setup

## 4.1. Training and Development Data

Our training data were taken from the same databases as in [2]: Callfriend, Fisher English Part 1 and 2, Fisher Levantine Arabic, HKUST Mandarin, Mixer (data from NIST SRE 2004, 2005, 2006, 2008). We have defined two sets with data from the 23 NIST LRE 2009 target languages only: the first contains all the utterances in the databases for these languages and it is further denoted *full*. The second contains a maximum of 500 utterances per language (we do not have 500 utterances for all languages), and it is further denoted *balanced*. For training the iVector extractor, the full dataset has been taken, but no degradation in performance was seen when using the balanced one. For training the classifiers, the balanced dataset has been taken, because it was found that having equal amount of data per class leads to lower error rates.

The calibration back-end described in section 4.3 was trained on development dataset, which comprises data from NIST LRE 2007, OGI-multilingual, OGI 22 languages, Foreign Accented English, SpeechDat-East, Switch Board and Voice of America radio broadcast. Only data of the 23 target languages are used. This set was based on segments of previous NIST LRE evaluations plus additional segments extracted from CTS, VOA3 and human-audited VOA2 data, not contained in the training dataset, and is the same as in [2].

## 4.2. Feature Extraction

Standard 7 Mel Frequency Cepstral Coefficients (MFCC) (including $C_0$) are used. Vocal Tract Length Normalization (VTLN) [8] and Cepstral Mean and Variance Normalization is applied in MFCC computation. Then, Shifted Delta Cepstral (SDC) coefficients [6] with usual 7-1-3-7 configuration are obtained, and concatenated to MFCCs, to obtain a final feature vector of 56 coefficients. For each utterance, the corresponding feature sequence is finally converted to an iVector using an iVector extractor based on a GMM with 2048-components trained on pooled features from all 54 languages included in our training data.

## 4.3. Calibration Back-end

For calibration and fusion, a Gaussian Back-end followed by a Discriminative Multi-Class Logistic Regression is used to post-process scores obtained from the described classifiers. Note that the Gaussian Back-end is essentially the same model as our generative classifier. However, its inputs are the scores from the classifiers described above rather than the iVectors. Also, it is trained on the separate development dataset to obtain well-calibrated scores.

# 5. Results

All results are for the closed-set condition. We use the NIST LRE 2009 dataset, which contains 23 target languages, and files of 3, 10 and 30 s. Results are shown in terms of $C_{avg} \times 100$ defined in the NIST LRE 2009 Evaluation Plan[1]. Since at the output of the backend well-calibrated log-likelihoods are obtained, the threshold is set analytically.

## 5.1. Results for Generative Linear Classifier

In Table 1, we show the effect of iVectors dimensionality for three conditions corresponding to the three nominal durations of test utterances (3, 10 and 30 s). We can see that the appropriate iVector dimensionality is 600. A lower dimensionality does not give the same level of accuracy and higher dimensionality does not offer further improvements, while the computational complexity is increased. Also, duration-independent (DI) calibration back-end is compared to the duration-dependent (DD) back-end, where a separate back-end is trained for each condition. As we can see, no significant difference between DI and DD back-end for the 30 s condition is found. However, for the 3 and 10 s conditions, the DI back-end performs better. This

[1] http://www.itl.nist.gov/iad/mig/tests/lre/2009/LRE09_EvalPlan_v6.pdf

| Condition | 200D | 300D | 400D | 500D | 600D | 700D |
|---|---|---|---|---|---|---|
| 3 s DI | 14.78 | 14.54 | 14.35 | 14.30 | **14.10** | 14.12 |
| 3 s DD | 16.29 | 15.87 | 15.63 | 15.50 | 15.29 | 15.25 |
| 10 s DI | 4.63 | 4.33 | 4.26 | 4.14 | **4.04** | 4.05 |
| 10 s DD | 5.55 | 5.25 | 5.11 | 4.90 | 4.76 | 4.79 |
| 30 s DI | 2.29 | 2.07 | 1.94 | 1.94 | 1.91 | 2.01 |
| 30 s DD | 2.36 | 2.08 | 1.88 | 1.90 | **1.88** | 1.93 |

Table 1: $C_{avg} \times 100$ *for the generative model with 200 to 700 dimensions, for the 3, 10 and 30 s conditions, and for the DI and DD back-ends*

indicates that scores obtained from the generative model are independent of the duration of the test utterances and we can benefit from training the back-end on larger amount of data pooled from the three conditions. For this reason, only the DI back-end is used in the remaining experiments.

In speaker recognition, significantly improved performance was observed when the dimensionality of iVectors was reduced by LDA and/or length of each iVector was normalized to unity [14] prior to applying the PLDA model. In Table 2, we can see that none of these techniques leads to an improvement in LR. The maximum number of useful dimensions that LDA can identify is the number of classes minus one. Since we have only 23 target languages, iVectors are reduced to 22 dimensions when applying LDA. Note that, since LDA and the generative model are both based on the same assumption of the common within-class covariance matrix, LDA dimensionality reduction would not have any effect if the classification decision was based directly on the generative model (for similar reasons as described in section 3.1). However, LDA causes utterance-dependent shifts to the likelihood scores (common to all classes) corresponding to the discarded dimensions, which makes the difference when using the generative model in conjunction with the following back-end.

| Condition | Generative | +NORM | +LDA |
|---|---|---|---|
| 3 s | **14.10** | 14.57 | 14.41 |
| 10 s | **4.04** | 4.32 | 4.13 |
| 30 s | **1.91** | 2.03 | 1.96 |

Table 2: $C_{avg} \times 100$ *for the iVectors and generative models*

### 5.2. Results for Discriminative Classifiers

First, we carried out experiments to find appropriate regularization constant for both SVM and logistic regression. Figure 1 and Figure 2 show performance obtained with SVM and logistic regression for different values of regularization parameter $C$ as defined in LIBSVM and LIBLINEAR (smaller $C$ leads to more aggressive regularization). The optimal performance was obtained with 400 dimensional iVectors and C=0.001 in the case of SVM, and with 600 dimensional iVectors and C=0.01 in the case of logistic regression. The following results are reported for these configurations.

In Tables 3 and 4, results obtained with SVM and logistic regression are shown. For both classifiers, we also experimented with three modifications. The first one is the application of Nuisance Attribute Projection [11], which projects *N* directions with the largest channel variability out of the iVectors. The second modification is the LDA dimensionality reduction of iVectors applied in the same way as in the case of the generative classifier. The third modification is iVector length normalization followed by LDA. As we can see, better results are

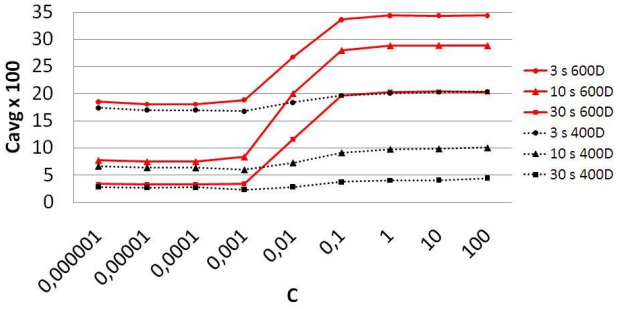


Figure 1: *Tuning of C value for SVM with iVectors of dimension 400 and 600 with the DI back-end*

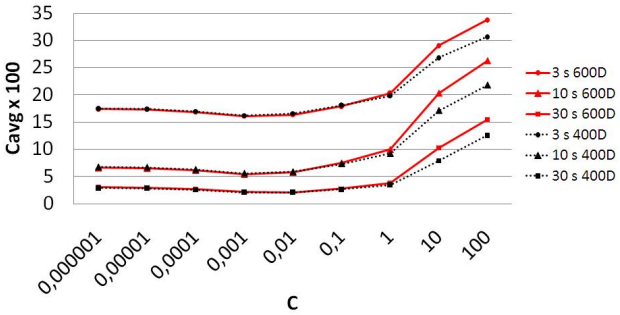


Figure 2: *Tuning of C value for logistic regression with iVectors of dimension 400 and 600 with the DI back-end*

generally obtained with logistic regression, where particularly good performance is obtained with NAP and with LDA (without iVector normalization).

Note that LDA dimensionality reduction and NAP are very similar techniques when applied in iVector space. First, NAP projects out the high channel variability directions while preserving the original dimensionality of iVectors. Although this is unnecessary with low dimensional iVectors, where appropriate linear transformation can be applied to remove the corresponding dimensions, just like in the case of LDA. Furthermore, the iVector extractor is trained in such a way that iVectors (at least those corresponding to training utterances) are standard normal distributed (i.e. variance of iVectors is one in all directions). Therefore, the directions with the largest ratio between across-class and within-class variance (preserved by LDA) are also the directions with the smallest within-class variance (preserved by NAP). However, unlike in the case of LDA, NAP allows us to preserve more than 22 dimensions, which might be found useful by the discriminative classifier. The search for optimal dimensionality of channel subspace in NAP is shown in Figure 3, for both SVM and logistic regression (only the 10 s condition is plotted for a clearer representation, the 3 s and 30 s condition follow the same trend). In both cases the optimal dimension is $N = 60$, and this is the dimension used to run experiments.

### 5.3. Comparison with JFA and fusion

Table 5 shows results for JFA (as described in [3]), for the best performing iVector based systems, and for fusion of both approaches. Both generative and discriminative classifiers based on iVectors outperform the state-of-the art JFA system and fusion of JFA and iVector based systems leads to additional improvements. It is interesting to see that most of the improve-

| Condition | SVM | +NAP | +LDA | +NORM+LDA |
|---|---|---|---|---|
| 3 s | 15.84 | 15.71 | 14.99 | **14.66** |
| 10 s | 5.16 | 5.00 | 4.56 | **4.39** |
| 30 s | 2.24 | **2.03** | 2.10 | 2.28 |

Table 3: $C_{avg} \times 100$ *obtained with SVM classifier. Experiments with 400 dimensional iVectors*

| Condition | LgR | +NAP | +LDA | +NORM+LDA |
|---|---|---|---|---|
| 3 s | 15.14 | **13.86** | 14.05 | 14.25 |
| 10 s | 4.88 | 4.06 | **4.03** | 4.17 |
| 30 s | 2.05 | **1.92** | 1.93 | 2.17 |

Table 4: $C_{avg} \times 100$ *obtained with logistic regression classifier. Experiments with 600 dimensional iVectors*

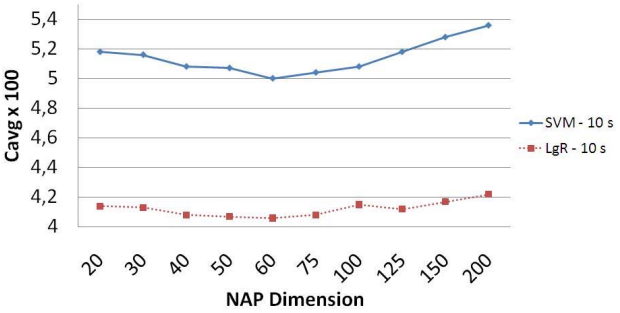


Figure 3: *Tuning of NAP dimensionality for SVM with 400D iVectors and LgR with 600D iVectors, for the 10 s condition*

| System | JFA | Generative | SVM+LDA | LgR+LDA | Fus1 | Fus2 |
|---|---|---|---|---|---|---|
| 3 s | 14.57 | 14.10 | 14.66 | 14.05 | 13.88 | **13.81** |
| 10 s | 4.89 | 4.04 | 4.39 | 4.03 | 3.86 | **3.82** |
| 30 s | 2.02 | 1.88 | 2.10 | 1.90 | 1.70 | **1.66** |

Table 5: $C_{avg} \times 100$ *for the JFA system from [3], the best performing iVector based systems, and for fusion of both approaches:*
*Fus1: fusion of JFA and Generative*
*Fus2: fusion of JFA, Generative, SVM+LDA and LgR+LDA*

ment is obtained when fusing JFA with only one single iVector system based on generative model and that fusion of all the individual systems in Table 5 leads only to insignificant additional $C_{avg}$ reductions.

## 6. Conclusions

We have introduced a novel approach for language recognition. Three classifiers (linear generative model, SVM and logistic regression) have been tested in the iVector space, and all outperform the state-of-the-art JFA system. Very simple and fast classifier based on linear generative model provides excellent performance over all conditions. The advantage of this classifier is also its scalability: addition of a new language only requires estimating the mean over the corresponding iVectors. Most of the computational load is in the iVector generation. Hence, as a next step, we will try to obtain iVectors from the utterances and the corresponding sufficient statistics in a more direct way.

## 7. Acknowledgements

This work was partly supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. D10PC20015, and by Czech Ministry of Education project No. MSM0021630528. It was done during an internship of David Martínez at Brno University of Technology funded by the Spanish Ministry of Science and Innovation under project TIN2008-06856-C05-04.

## 8. References

[1] Simon J. D. Prince and James H. Elder, "Probabilistic Linear Discriminant Analysis for Inference About Identity", in Proc. 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, Oct. 2007.

[2] Z. Jančik et al., "Data Selection and Calibration Issues in Automatic Language Recognition - Invesigation with BUT-AGNITIO NIST LRE 2009 System", in Proc. Odyssey 2010 - The Speaker and Language Recognition Workshop, Brno, CZ.

[3] N. Brümmer, A. Strasheim, V. Hubeika, P. Matějka, P. Schwarz, J. Černocký, "Discriminative Acoustic Language Recognition via Channel-Compensated GMM Statistics", in Proc. Interspeech 2009, Brighton, GB.

[4] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel and P. Ouellet, "Front-End Factor Analysis for Speaker Verification", IEEE Trans. on Audio, Speech and Language Processing, vol. 19, pp. 788-798, May 2011.

[5] M. Kockmann et al., "Prosodic speaker verification using subspace multinomial models with intersession compensation", in Proc. Interspeech, Tokyo, 2010

[6] 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 Cepstal Features", in Proc. International Conferences on Spoken Language Processing, Sept. 2002.

[7] O. Glembek, L. Burget, P. Matějka, M. Karafiat, P. Kenny, "Simplification and Optimization of i-vector Extraction", accepted to ICASSP 2011, Prague.

[8] L. Weling, S. Kanthak and H. Ney, "Improved Methods for Vocal Tract Normalization", in Proc. ICASSP 1999, Phoenix.

[9] M. Soufifar et al., "iVector Based Approach to Phonotactic Language Recognition", submitted to Interspeech 2011.

[10] Chin-Chung Chang and Chih-Jen Lin, "LIBSVM: a Library for Support Vector Machines", 2001, http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[11] W.M. Campbell, D.E. Sturim, D.A. Reynolds, A. Solomonoff, "SVM Based Speaker Verification using a GMM Supervector Kernel and NAP Variability Compensation", IEEE International Conference on Acoustics, Speech, and Signal Processing, Toulouse, 2006, vol. 1, pp. 97-100.

[12] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A Library for Large Linear Classification", Journal of Machine Learning Research 9 (2008), 1871-1874. http://www.csie.ntu.edu.tw/ cjlin/liblinear.

[13] L. Burget, O. Plchot, S. Cumani, O. Glembek, P. Matějka, N. Brümmer, "Discriminatively Trained Probabilistic Linear Discriminant Analysis for Speaker Verification", accepted to ICASSP 2011, Prague.

[14] D. García-Romero and C. Espy-Wilson, "Analysis of i-vector Length Normalization in Speaker Recognition Systems", submitted to Interspeech 2011.

[15] P. Kenny et al., "Joint Factor Analysis versus Eigenchannes in Speaker Recognition", IEEE Transactions on Audio, Speech, and Language Processing, vol. 15, pp. 1435-1447, May 2007.

[16] P. Kenny, "Bayesian Speaker Verification with Heavy-Tailed Priors", in proc. of Odyssey 2010 - The Speaker and Language Recognition Workshop, Brno, CZ.