Language Recognition in iVectors Space

David Martínez¹, Oldřich Plchot², Lukáš Burger², Ondřej Glembek² and Pavel Matějka²

¹Aragon Institute for Engineering Research (I3A), University of Zaragoza, Spain
²Speech@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 considers 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 \( C_{\text{avg}} = 0.0188 \) 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
\[ M = m + Tw, \quad (1) \]
where \( m \) is the language- and channel-independent component
of the mean supervector, \( T \) is a matrix of bases spanning the
subspace covering the important variability (both speaker- and
session-specific) in the supervector space, and \( w \) is a standard-
normally distributed latent variable. For each observation se-
quence representing an utterance, our iVector is the Maximum
A Posteriori (MAP) point estimate of the latent variable \( w \). Our
iVector extractor training procedure is based on the efficient im-
plementation 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
\( w \) corresponding to a test utterance, we evaluate log-likelihood
for each language as:
\[
\ln p(w|l) = -\frac{1}{2} w^T \Sigma^{-1} w - \frac{1}{2} \mu_l^T \Sigma^{-1} \mu_l - \frac{1}{2} \text{const},
\]
where \( \mu_l \) is the mean vector for language \( l \), \( \Sigma \) is the com-
mon covariance matrix and const is a language- and iVector-
dependent constant. If the log-likelihoods \( \ln p(w|l) \) were di-
rectly used to decide about the language (or estimate the pos-
terior probability of a language), the quadratic term \( w^T \Sigma^{-1} w \)
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 \( w \). 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 classi-
fiers: linear Support Vector Machines (SVM) and Logistic Re-
gression 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, respec-
tively. Although, we have used binary logistic regression in our
experiments, our problem could be addressed more directly us-
ing a single multi-class logistic regression classifier. For exam-
ple, the experiments in [3], where multi-class logistic regression
was applied to recognize languages from GMM mean supervec-
tors, 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 degra-
dation 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 lan-
guages 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) (in-
cluding \( C_0 \)) 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 ob-
tained, and concatenated to MFCCs, to obtain a final feature
vector of 56 coefficients. For each utterance, the correspond-
ing 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 \( C_{avg} \times 100 \)
de-

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 appro-
imate 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) cal-
ibration back-end is compared to the duration-dependent (DD)
back-end, where a separate back-end is trained for each condi-
tion. 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

indicates that scores obtained from the generative model are in-
dependent of the duration of the test utterances and we can ben-
et 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 re-
duced 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 improve-
ment 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 di-

densions 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 dimension-
ality reduction would not have any effect if the classification
decision was based directly on the generative model (for simi-
lar 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 con-
junction with the following back-end.

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

<table>
<thead>
<tr>
<th>Condition</th>
<th>200D</th>
<th>300D</th>
<th>400D</th>
<th>500D</th>
<th>600D</th>
<th>700D</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 s DD</td>
<td>16.29</td>
<td>15.87</td>
<td>15.63</td>
<td>15.50</td>
<td>15.29</td>
<td>15.25</td>
</tr>
<tr>
<td>10 s DI</td>
<td>4.63</td>
<td>4.33</td>
<td>4.26</td>
<td>4.14</td>
<td>4.04</td>
<td>4.05</td>
</tr>
<tr>
<td>10 s DD</td>
<td>5.55</td>
<td>5.25</td>
<td>5.11</td>
<td>4.90</td>
<td>4.76</td>
<td>4.79</td>
</tr>
<tr>
<td>30 s DI</td>
<td>2.29</td>
<td>2.07</td>
<td>1.94</td>
<td>1.94</td>
<td>1.91</td>
<td>2.01</td>
</tr>
<tr>
<td>30 s DD</td>
<td>2.36</td>
<td>2.08</td>
<td>1.88</td>
<td>1.90</td>
<td>1.88</td>
<td>1.93</td>
</tr>
</tbody>
</table>

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 regular-
ization 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 ob-
tained 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 logis-
tic regression are shown. For both classifiers, we also experi-
mented with three modifications. The first one is the applica-
tion of Nuisance Attribute Projection [11], which projects $N$
directions with the smallest within-class variance 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 gen-
erative classifier. The third modification is iVector length nor-
malization followed by LDA. As we can see, better results are
generally obtained with logistic regression, where particularly
good performance is obtained with NAP and with LDA (with-
out 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 pre-
serving the original dimensionality of iVectors. Although this is
unnecessary with low dimensional iVectors, where appropriate
linear transformation can be applied to remove the correspond-
ding 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 dimen-
sionality 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 ap-
proaches. Both generative and discriminative classifiers based
on iVectors outperform the state-of-the art JFA system and fu-
son of JFA and iVector based systems leads to additional im-
provements. It is interesting to see that most of the improve-

dents.
Table 3: \( C_{\text{avg}} \times 100 \) obtained with SVM classifier. Experiments with 400 dimensional iVectors

<table>
<thead>
<tr>
<th>Condition</th>
<th>SVM</th>
<th>+NAP</th>
<th>+LDA</th>
<th>+NORM+LDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 s</td>
<td>15.84</td>
<td>15.71</td>
<td>14.99</td>
<td>14.66</td>
</tr>
<tr>
<td>10 s</td>
<td>5.16</td>
<td>5.00</td>
<td>4.56</td>
<td>4.39</td>
</tr>
<tr>
<td>30 s</td>
<td>2.24</td>
<td>2.03</td>
<td>2.10</td>
<td>2.28</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Condition</th>
<th>LgR</th>
<th>+NAP</th>
<th>+LDA</th>
<th>+NORM+LDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 s</td>
<td>15.14</td>
<td>13.86</td>
<td>14.05</td>
<td>14.25</td>
</tr>
<tr>
<td>10 s</td>
<td>4.88</td>
<td>4.06</td>
<td>4.03</td>
<td>4.17</td>
</tr>
<tr>
<td>30 s</td>
<td>2.05</td>
<td>1.92</td>
<td>1.93</td>
<td>2.17</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>System</th>
<th>JFA</th>
<th>Generative</th>
<th>SVM+LDA</th>
<th>LgR+LDA</th>
<th>Fus1</th>
<th>Fus2</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 s</td>
<td>14.57</td>
<td>14.10</td>
<td>14.66</td>
<td>14.05</td>
<td>13.88</td>
<td>13.81</td>
</tr>
<tr>
<td>10 s</td>
<td>4.89</td>
<td>4.04</td>
<td>4.39</td>
<td>4.03</td>
<td>3.86</td>
<td>3.82</td>
</tr>
<tr>
<td>30 s</td>
<td>2.02</td>
<td>1.88</td>
<td>2.10</td>
<td>1.90</td>
<td>1.70</td>
<td>1.66</td>
</tr>
</tbody>
</table>

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 Martinez at Brno University of Technology funded by the Spanish Ministry of Science and Innovation under project TIN2008-06856-C05-04.

8. References

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