IVECTOR-BASED PROSODIC SYSTEM FOR LANGUAGE IDENTIFICATION

David Martínez[1], *Lukáš Burget*[2], *Luciana Ferrer*[2], *Nicolas Scheffer*[2]

[1]Aragon Institute for Engineering Research (I3A), University of Zaragoza, Spain
[2]Speech Technology and Research Laboratory, SRI International, Menlo Park, CA, USA

ABSTRACT

Prosody is the part of speech where rhythm, stress, and intonation are reflected. In language identification tasks, these characteristics are assumed to be language dependent, and thus the language can be identified from them. In this paper, an automatic language recognition system that extracts prosody information from speech and makes decisions about the language with a generative classifier based on iVectors is built. The system is tested on the NIST LRE09 dataset. The results are still not comparable to state-of-the-art acoustic and phonotactic systems. However, they are promising and the fusion of the new approach with an iVector-based acoustic system is found to bring further improvements over the latter.

Index Terms— Language Identification, Prosody, iVectors, Joint Factor Analysis.

1. INTRODUCTION

In recent years, we have seen great improvements in acoustic and phonotactic language identification (LID) systems. Among the most popular modeling techniques used in acoustic systems are joint factor analysis (JFA) [2] and iVectors [1], which are usually applied to model spectral features such as mel frequency cepstral coefficients (MFCC). In contrast, phoneme n-gram statistics are modeled in order to recognize languages in phonotactic approaches [3, 4].

Several approaches have been also investigated to extract prosodic information from speech and employ it in LID systems. In [6], the authors extract a set of features based on the three components of prosody: rhythm, stress, and intonation. However, the extraction procedure is computationally expensive since an automatic speech recognition (ASR) system is required. In [7], pitch contours are approximated using Legendre polynomials over long temporal intervals, which seems to be logical and useful for prosody modeling. This approach has also been recently adopted for speaker identification (SID) [8, 9, 10], where pitch contours and also energy contours are approximated using linear combination of Legendre polynomials over syllable or syllable-like units. The regression coefficients together with durations of corresponding segments are the features describing the three characteristics of prosody.

When modeling prosodic features for SID, different techniques have been proposed in the literature [8, 9, 10, 11, 12]. Until recently, one of the most popular approaches was to use a standard JFA model [8, 10]. Recently, the standard iVector approach [14], initially proposed to model MFCC features, was tested on polynomial coefficient prosodic features [11], showing remarkable performance on a speaker verification task, comparable to that obtained using the JFA approach. Note that these approaches are applicable only to features that are always defined and are relatively low-dimensional, like the polynomial coefficient features described above. For more complex sets of features, another subspace modeling technique called the subspace multinomial model (SMM) [12] was introduced, which models the vector of weights from a background Gaussian mixture model (GMM) that takes into account probabilities of undefined values. Recently, SMM-based iVectors were also successfully used as low-dimensional representations of n-gram counts in a phonotactic LID system [5].

In our work, we adopt the standard iVector paradigm [14] to model the prosodic polynomial features for LID, and create a classification system similar to the one from [1], where an iVector system is built based on acoustic features, and a generative Gaussian model for each of the languages with a shared covariance matrix is used as the classifier. Our systems are tested on the NIST LRE 2009 dataset [16], on which no previous results based on prosodic features are available. We hope that this can be useful as a baseline for future research on this topic.

The rest of the paper is organized as follows: in Section 2, the prosodic feature extraction process is described; in Section 3, the generative Gaussian LID system based on iVectors is revised; in Section 4, the experimental setup and results are shown; in Section 5, the conclusions are drawn.

2. PROSODIC FEATURE EXTRACTION

2.1. Pitch and Energy Contour Extraction

Our prosodic features carry information about the evolution of pitch and energy along time. To extract pitch and energy contours we use The Snack Sound Toolkit [15]. The pitch and energy values are converted to log domain, to simulate human

perception. In the next step, energy is normalized by subtracting its maximum value in the log scale. This makes it more robust to language-independent phenomena such as channel variations. The log pitch values are normalized by subtracting mean and dividing by standard deviation estimated over each recording. In SID no normalization of pitch is required, since the absolute value contains information about the speaker. In LID, we are interested only in the information about the language and we believe that pitch normalization reduces the unwanted across-speaker variability. We have also experimented with only mean normalization, which resulted in very similar performance to mean and variance normalization, and for this reason, only results for mean and variance normalization will be shown.

2.2. Segment Definition

After extracting pitch and energy contours for whole speech recordings, every recording is divided into segments and coefficients describing pitch and energy contours are extracted for each such segment. In [10], different segment definitions were tested and segmentation based on syllables detected using an ASR system was found to perform the best. Since the language is unknown in the case of LID, we wanted to avoid the use of ASR. Therefore, we experimented with the other two segment definitions proposed in [10]: segment boundaries defined by energy valleys and fixed-length segments. For the energy valley based segments, segment boundaries are determined by local minima in the energy contour. This approach tries to find syllable boundaries in a very simple way. In the case of fixed-length segments, the signal is split into segments of 200 ms with an overlap of 150 ms. Compared to the segment length of 300 ms proposed in [10], our segments are closer to the average syllable duration of 120 ms. Also, shorter segments and larger overlap allow us to obtain more training examples for languages with small amounts of training data.

2.3. Contour Modeling

For each segment, we drop all unvoiced frames for which no pitch was detected. Then pitch and energy contours are approximated by linear combination of Legendre polynomials as

$$f(t) = \sum_{i=0}^{M} a_i P_i(t) \tag{1}$$

where $f(t)$ is the contour being modeled and $P_i(t)$ is the i Legendre polynomial. Each coefficient a_i represents a characteristic of the contour shape: a_0 corresponds to the mean, a_1 to the slope, a_2 to the curvature, and higher order represents more precise detail of the contour. In our implementation, Legendre polynomials of order 5 give six coefficients for pitch and six for energy.

Finally, 13-dimensional feature vectors are obtained by augmenting the coefficients with the number of voiced frames in the segment. Thus, we can consider that our features contain information of the three components of prosody: intonation in the pitch, rhythm in the duration, and stress in both the energy and in the duration. These are the features used to build our GMM universal background model (UBM). Supervectors of Baum-Welch statistics can then be estimated for each utterance, as in [14]. They are of dimension 13 times the number of Gaussians in the UBM.

3. IVECTORS AND CLASSIFICATION

3.1. iVector Extraction

The idea behind the iVector approach is that the language- and channel-dependent supervectors of concatenated GMM means can be modeled as

$$\mathbf{M} = \mathbf{m} + \mathbf{T}\mathbf{w}, \tag{2}$$

where $\mathbf{m}$ is a language- and channel-independent supervector of concatenated UBM means, $\mathbf{T}$ is a matrix of bases spanning the subspace covering the important variability (both language- 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 $\mathbf{w}$. For more detail on iVector extraction see [14].

3.2. Classifier

Once the iVectors for our training data are obtained, a linear generative classifier is trained as proposed in [1]. The distributions of iVectors for individual languages are modeled by Gaussian distributions with a single within-class (WC) full covariance matrix shared by all the languages.

For an iVector $\mathbf{w}$ corresponding to a test utterance, the loglikelihood for each language is

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

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 irrelevant for making decisions. The quadratic term $\mathbf{w}^T\boldsymbol{\Sigma}^{-1}\mathbf{w}$, which is constant over classes, would be also irrelevant, if the log-likelihoods were directly used to obtain posterior probabilities of classes. However, since the likelihoods are used only as input features to the calibration backend, it makes a difference in our system, as explained in [1].

3.3. Fusion and Calibration Backend

For calibration, a Gaussian backend followed by discriminative multiclass logistic regression is used to postprocess

scores obtained from the described classifiers. Note that the Gaussian backend 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. When fusing multiple systems, a separate Gaussian backend is trained for each subsystem and outputs of the Gaussian backends are fused by multiclass logistic regression. A detailed description of the backend, which also uses information about the recording duration for calibration, can be found in [13].

4. EXPERIMENTS AND RESULTS

4.1. Test Data

Our results are reported for a closed-set task of 3, 10 and 30 seconds of the NIST LRE 2009 evaluation [16]. The data comprises 31178 recordings of 23 target languages. Results are reported in C_{avg}, which is an error metric defined in [16].

4.2. Training and Development Data

Our training data is from the following databases: CALLFRIEND, NIST LRE03, NIST LRE05, NIST LRE07, and VOA3. The data comprises 51 languages, which are all used to train our UBM. For training iVector extractor matrices T, we use data of only the 23 target languages. For training the generative classifier, we use only 500 files per language, in the same way as in [1].

A separate dataset was used for training the fusion/calibration backend, which includes data from the following databases: CALLFRIEND, CALLHOME, Fisher, NIST LRE05, NIST LRE07, Mixer, OGI22, and VOA.

4.3. Results with Prosodic Features

Several parameters can be tuned in the system. We have studied the influence of the number of Gaussians, the iVector dimensionality, and the type of segment definition as described in Section 2.2.

Table 1 compares performance of prosodic features with 1) energy valley based segments and 2) fixed-length segments. UBM with 512 Gaussian components is used in extraction of 300-dimension iVectors. As can be seen, fixed-length segments provide better performance, which is in agreement with the previous experiments on the SID task [10]. Prosodic features with fixed-length segments are used in all the following experiments.

Next, we experimented with the number of Gaussian components in iVector extraction and with iVector dimensionality. Recent experiments in SID [11] show that a reasonable configuration for prosodic systems is 512 Gaussian components and 300-dimension iVectors. Table 2 compares performance of systems with different numbers of Gaussian components. Improvement can be seen when increasing the number of components from 512 to 2048. As for acoustic features [1], increasing the dimensionality of iVectors improves the system accuracy. 400-dimension iVectors were found to be optimal and no additional gains were observed for higher dimensions.

Condition	Energy valley	Fixed length
3 s	35.08	**34.57**
10 s	25.83	**24.45**
30 s	19.27	**17.28**

Table 1. $C_{avg} \times 100$ *on NIST LRE 2009 for the prosodic features with energy valley based segments and fixed-length segments, 512 Gaussian components, 300-dimension iVectors*

Condition	512 Gaussians	1024 Gaussians	2048 Gaussians
3 s	32.56	31.97	**31.76**
10 s	22.52	21.89	**21.12**
30 s	15.58	14.60	**13.78**

Table 2. $C_{avg} \times 100$ *on NIST LRE 2009 for the prosodic features with fixed-length segments, 512, 1024 and 2048 Gaussian components, 400-dimension iVectors*

4.4. Fusion with Acoustic iVectors-based System

4.4.1. Acoustic system

The state-of-the-art-acoustic system is built in the same fashion as in [1]. It uses the same configuration (SDC 7-1-3-7, 2048 Gaussians, 600-dimension iVectors) except for not using vocal tract length normalization (VTLN) and having a different training dataset. The UBM, iVector extractor, Gaussian classifier, and backend are trained in the same way and on the same data as described for the prosodic system in Section 4.1. Therefore, the improvements obtained from fusing the acoustic and prosodic system can be attributed to the complementarity of prosodic and cepstral features and not to combining information from different data sources.

4.4.2. Fusion results

Table 3 shows the results for the state-of-the-art acoustic system, our best prosodic system (2048 Gaussians, 400-dimension iVectors) and the fusion of both systems. As can be seen, the fusion with the prosodic system improves performance in all conditions. The relative improvements obtained over the acoustic system are: 10.93% for 3 seconds; 15.24% for 10 seconds; and 9.39% for 30 seconds.

5. CONCLUSIONS

A LID system based on prosodic features has been introduced. Extraction of the pitch, energy, and duration allows us to represent the three components of prosody: stress,

Condition	Acoustic	Prosodic	Fusion
3 s	19.13	31.76	**17.04**
10 s	6.30	21.12	**5.34**
30 s	3.09	13.78	**2.80**

Table 3. $C_{avg} \times 100$ *for the generative iVectors-based acoustic system, generative iVectors-based prosodic system and fusion of both systems*

intonation, and rhythm. Unvoiced frames where the pitch is undefined are discarded, permitting us to treat the features as continuous. Thus, the same classifier successfully applied for acoustic LID, based on iVectors and a generative model, can be adapted for our prosodic features. Fixed-length segments, 2048 Gaussians, and 400 dimensions, have been found to be a good configuration for the system. Although the performance of the prosodic system alone does not give outstanding results, it is in the fusion with another LID system where this approach is really powerful. The combination with a prosodic system resulted in significant performance improvements over the state-of-the-art iVectors-based acoustic system on all conditions of the NIST LRE 2009 task. We consider this technique to be very promising as there are still many possibilities for experimenting with additional prosodic features such as AM modulation or formants that could provide further improvements. For this reason, we believe that prosodic features can play an important role in future LID systems. At the same time, a baseline for prosodic systems on the NIST LRE 2009 dataset has been established in this work.

6. ACKNOWLEDGMENTS

This work was done during an internship of David Martínez at SRI International funded by the Spanish Ministry of Science and Innovation under project TIN2008-06856-C05-04.

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. D10PC20024. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of the Defense Advanced Research Projects Agency (DARPA); or its Contracting Agent, the U.S. Department of the Interior, National Business Center, Acquisition & Property Management Division, Southwest Branch.

7. REFERENCES

[1] D. Martínez, O. Plchot, L. Burget, O. Glembek, P. Matějka, "Language identification in iVectors space", *Proc. Interspeech 2011*, Florence.

[2] N. Brümmer, A. Strasheim, V. Hubeika, P. Matějka, P. Schwarz, J. Černocký, "Discriminative acoustic language recognition via channel-compensated GMM statistics", *Proc. Interspeech 2009*, Brighton.

[3] M.A. Zissman, "Comparison of four approaches to automatic language identification of telepone speech", *IEEE Trans. Speech and Audio Proc.*, vol. 4, no. 1, pp. 31-44, 1996.

[4] T. Mikolov, O. Plchot, O. Glembek, P. Matějka, L. Burget, J. Černocký, "PCA-based feature extraction for phonotactic language recognition", *Proc. Odyssey 2010 - The Speaker and Language Recognition Workshop*, Brno.

[5] M. Soufifar, M. Kockmann, L. Burget, O. Plchot, O. Glembek, "iVector based approach to phonotactic language recognition", *Proc. Interspeech 2011*, Florence.

[6] L. Mary, B. Yegnanarayana, "Extraction and representation of prosodic features for language and speaker recognition", *Speech Communication* 50 (2008) p. 782-796.

[7] Chi-Yueh Lin, Hsiao-Chuan Wang, "Language identification using pitch contour information", *Proc. ICASSP 2005*, Philadelphia.

[8] N. Dehak, P. Demouchel, P. Kenny, "Modeling prosodic features with joint factor analysis for speaker verification", *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 7, pp. 2095-2103, Sept. 2007.

[9] L.Ferrer, N. Scheffer, E. Shriberg, "A comparison of approaches for modeling prosodic features in speaker recognition", *Proc. ICASSP 2010*, Dallas.

[10] M. Kockmann, L. Burget, J. Černocký, "Investigations into prosodic syllable contour features for speaker recognition", *Proc. ICASSP 2010*, Dallas.

[11] M. Kockmann, L. Ferrer, L. Burget, and J. H. Cernock, "iVector fusion of prosodic and cepstral features for speaker verification", *Proc. Interspeech 2011*, Florence.

[12] M. Kockmann, L. Burget, O. Glembek, L. Ferrer, J. Černocký, "Prosodic speaker verification using subspace multinomial models with intersession compensation", *Proc. Interspeech 2010*, Makuhari.

[13] Z. Jančík, O. Plchot, N. Brümmer, L. Burget, O. Glembek, V. Hubeika, M. Karafiát, P. Matějka, T. Mikolov, A. Strasheim, J. Černocký, "Data selection and calibration issues in automatic language recognition - investigation with BUT-AGNITIO NIST LRE 2009 system", *Proc. Odyssey 2010 - The Speaker and Language Recognition Workshop*, Brno.

[14] 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.

[15] K. Sjölander, "The Snack Sound Toolkit", http://www.speech.kth.se/snack/.

[16] "The 2009 NIST Language Recognition Evaluation Plan (LRE09)", http://www.itl.nist.gov/iad/mig/tests/lre/2009/LRE09_EvalPlan_v6.pdf.