INTERSPEECH 2009 BRIGHTON

Language Recognition Using Language Factors

Fabio Castaldo[1], *Sandro Cumani*[1], *Pietro Laface*[1], *Daniele Colibro* [2]

[1] Politecnico di Torino, Italy, [2] Loquendo, Torino, Italy

{Fabio.Castaldo, Sandro.Cumani, Pietro.Laface}@polito.it

Daniele.Colibro@loquendo.com

Abstract

Language recognition systems based on acoustic models reach state of the art performance using discriminative training techniques.

In speaker recognition, eigenvoice modeling of the speaker, and the use of speaker factors as input features to SVMs has recently been demonstrated to give good results compared to the standard GMM-SVM approach, which combines GMMs supervectors and SVMs.

In this paper we propose, in analogy to the eigenvoice modeling approach, to estimate an eigen-language space, and to use the language factors as input features to SVM classifiers. Since language factors are low-dimension vectors, training and evaluating SVMs with different kernels and with large training examples becomes an easy task.

This approach is demonstrated on the 14 languages of the NIST 2007 language recognition task, and shows performance improvements with respect to the standard GMM-SVM technique.

Index Terms: language recognition, factor analysis, language space, support vector machine, discriminative training, GMM-SVM.

1. Introduction

State of the art in automatic language recognition is typically achieved by combining classifiers based on two main features. A first class of classifiers is based upon phonetic decoding and language modeling [1-3]. The second approach is based on acoustic features and has the advantage of not requiring phonetic knowledge, labeled speech, and phonetic decoders. The key for the success of the acoustic approach has been modeling the languages by means of GMMs using shifted-delta cepstral features [4], and mostly the use of discriminative training techniques. Examples of discriminative training techniques are Maximum Mutual Information Estimation of the GMMs [5] and Support Vector Machine (SVM) classifiers using the GMM mean supervectors [6-7]. A successful technique that exploits the information given by the support vectors estimated by an SVM classifier has been recently proposed to obtain discriminative GMMs [8].

This paper focuses on the acoustic approach to language recognition based on SVM methods where a GMM is adapted from a Universal Background Model (UBM) by using the frames of a single utterance. The adapted means of a GMM are stacked in a supervector, which is given as input feature to an SVM classifier. We refer to this approach, which combines GMMs supervectors and SVMs, as GMM-SVM.

Using the language GMM supervectors as input features, however, is not the only possible choice. It has been recently shown that good results can be obtained in speaker recognition by modeling the speaker GMMs with eigenvoices [9-10]:

$$\mathbf{s} = \boldsymbol{\mu}_{UBM} + \mathbf{V}\mathbf{y} \qquad (1)$$

and using as input features to SVMs the vectors of the speaker factors $\mathbf{y}$ obtained by Joint Factor Analysis rather than the much larger GMM supervectors [11].

Building on this suggestion, our starting hypothesis for this work has been to estimate an eigen-language space, in analogy to the eigen-voice space, and to use the language factors as input features to SVMs. Since the language factors are low-dimension vectors, it is possible to perform several experiments by training and evaluating SVMs with different kernels, and with a large number of training examples.

In this paper we propose a simple approach to estimate a space where languages are better discriminated, and we show that using the language factors with SVMs we achieve the same or better performance with respect to the standard GMM-SVM technique on the NIST 2007 language recognition task [12] with much smaller models.

The paper is organized as follows: Section 2 introduces the nuisances affecting the language models, and how we deal with them. Section 3 details how we estimate the principal components of the space that discriminate a language from the others. The description of our system and of the train and development corpora is given in Section 4. Section 5 illustrates how we compute the standard GMM-SVM kernels staying in the language factor space. The language models that we train are illustrated in Section 6. Experimental results are presented and commented in Section 7, and the conclusions are drawn in Section 8.

2. Nuisance compensation

The variability of the speaker, channel and environment are among the most important nuisance factors affecting the performance of automatic language recognition systems.

In [13], we have proposed an intersession compensation technique in the feature domain for speaker recognition, and we have applied the same approach to the compensation of inter-speaker variations within the same language. In particular, we estimate an inter-speaker subspace matrix $\mathbf{U}$ with a large set of differences between models generated using different speaker utterances of the same language. For the experiments performed in this work, we trained a gender independent matrix $\mathbf{U}$ using the databases detailed in Section 4. The dimension of the space spanned by matrix $\mathbf{U}$ is 100. Since in these databases there are few different sessions for the same speaker, the inter-speaker nuisance is the main factor that is compensated, but the compensation possibly includes session variations. Computing the occupation probability $\gamma_m(t)$ of each Gaussian mixture m of the UBM for a given frame of an utterance i, and the inter-speaker factors $\mathbf{x}^{(i)}$, the speaker-compensated features can be obtained as

$$\hat{\mathbf{o}}^{(i)}(t) = \mathbf{o}^{(i)}(t) - \sum_m \gamma_m(t) \cdot \mathbf{U}_m \cdot \mathbf{x}^{(i)} \qquad (2)$$

10.21437/Interspeech.2009-69

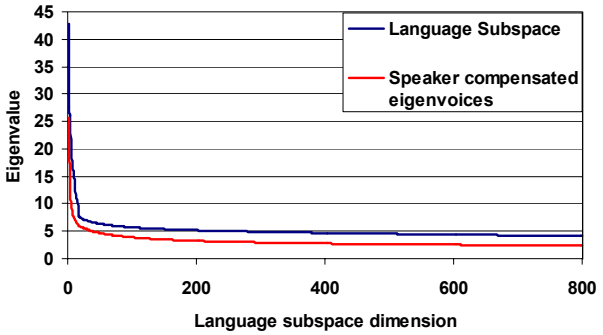

Figure 1. Eigenvalues of two language subspaces

3. Toward an eigen-language space

Residual information about the channel and the speaker remains after the compensation of nuisances of a GMM adapted from the UBM using a single utterance. However, most of the undesired variation is removed as demonstrated by the improvements obtained using this technique [13-14].

In speaker recognition, eigenvoice modelling assumes that a low dimensional eigenvoice matrix **V** exists such that a speaker s can be represented as

$$\mathbf{s} = \mu_{UBM} + \mathbf{V}\mathbf{y} \tag{3}$$

where **y** is a low rank vector including the so called speaker factors.

Estimating the principal directions of the GMM supervectors of all the training segments prior to inter-speaker nuisance compensation would produce a set of language independent, "universal" eigenvoices. After nuisance removal, however, the speaker contribution to the principal components is reduced to the benefit of language discrimination. We will refer to these principal components as "speaker compensated eigenvoices", and to the corresponding matrix as $\mathbf{V}_1$.

To find the directions that further enhance the discrimination among the languages we followed a working hypothesis considering a polyglot speaker that utters a set of phonetically rich sentences in different languages. Computing the differences between the GMM supervectors obtained from utterances of this speaker in different languages would factor out the speaker characteristics and would enhance the acoustic components of a language with respect to the others.

Since we do not have labeled databases including polyglot speakers, we could in principle compute and collect the difference between GMM supervectors produced by utterances of speakers of two different languages irrespective of the speakers' identity, which should have been already compensated in the feature domain using (2). Since the number of these difference supervectors would grow with the square of utterances of the training set, a feasible solution is to perform the Principal Component Analysis on the set of the differences between the set of the supervectors of a language and the average supervector of every other language. We will refer to the subspace derived from these differences as the "language subspace", and to the corresponding matrix as $\mathbf{V}_2$.

Figure 1 shows the eigenvalues obtained estimating up to 800 eigenvectors of the speaker compensated eigenvoices, and of the language subspace respectively. It is interesting noting that the language subspace has higher eigenvalues, and that both curves show a sharp decrease for their first 13 eigenvalues, corresponding to the main language discrimination directions, whereas the remaining eigenvalues decrease slowly indicating that the corresponding directions still contribute to language discrimination, even if they probably still account for residual channel and speaker characteristics.

4. System overview

The language models that we use are adapted from a Gaussian mixture Universal Background Model [15]

$$g(\mathbf{x}) = \sum_{i=1}^{N} \lambda_i \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{4}$$

where λ_i are the mixture weights, N is the number of mixture Gaussians, and μ_i and Σ_i are the mean and covariance of the Gaussians distributions, respectively.

In particular, we estimate a gender independent UBM with 2048 Gaussians using the training and development sets of the Callfriend corpus [16]. The observation vector includes Static plus Shifted Delta (SSDC) coefficients [4]. A specific GMM is created by MAP adaptation with a small relevance factor from the common UBM for each segment of a language, both in training and in testing.

The language models, and the matrices $\mathbf{V}_1$ and $\mathbf{V}_2$ that are used in the eigen-language approach, are trained using the following corpora:

- all data of the 12 languages in the Callfriend corpus
- half of the NIST LRE07 development corpus
- half of the OSHU corpus provided by NIST for LRE05
- The Russian through switched telephone network [16]

including a total of approximately 14000 segments.

For development, in particular to train the backend parameters, described in Section 7, the following data sets were used:

- the second half of the NIST LRE07 development corpus
- the second half of the OSHU corpus provided by NIST for LRE05
- development and test set provided by NIST for LRE03

including approximately 6000 segments for each of the duration conditions defined by NIST: 3, 10, and 30 seconds.

5. Kernel functions

In Section 7, devoted to the experimental results, we compare the performance of our reference GMM-SVM system with the SVM models using GMM supervectors obtained by estimating language factors and with SVM models using as input directly the language factors obtained by factor analysis.

An SVM is a two-class classifier based on a kernel function $K(.,.)$

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + d_i \tag{5}$$

The support vectors $\mathbf{x}_i$ and the bias d_i are obtained from the training set by an optimization process.

In this work we compare the performance of two kernels functions: one proposed for the GMM supervectors, and the other based on low dimension language factors.

5.1. Identity kernel function

The language factors, which correspond to the coordinates of an utterance in the space defined by the eigen-language matrix, can be provided as input features to an SVM classifier without any normalization. Good performances have been reported in [11] for speaker recognition using this simple approach.

Since directly using the language factors corresponds to the use of the GMM mean supervectors without any normalization, a kernel based on a distance between the GMMs of two

utterances should be more informative. That is why we compared it to the Kullback-Leibler kernel.

5.2. Kullback-Leibler kernel function

This kernel, proposed in [6], is based on an approximation of the Kullback-Leibler (KL) divergence, which leads to the kernel function

$$K(g_a, g_b) = \sum_{i=1}^{N} \lambda_i \mu_{a,i}^t \Sigma_i^{-1} \mu_{b,i} = \sum_{i=1}^{N} \left(\sqrt{\lambda_i} \Sigma_i^{-\frac{1}{2}} \mu_{a,i} \right)^t \left(\sqrt{\lambda_i} \Sigma_i^{-\frac{1}{2}} \mu_{b,i} \right) \quad (6)$$

The trivial implementation of this kernel for the eigen-language models consists in expanding the language factors to the original GMM space through the eigen-language matrix **V,** to get the supervector

$$\mathbf{g} = \boldsymbol{\mu}_{UBM} + \mathbf{V}\mathbf{y} \quad (7)$$

We can avoid going back to the high-dimension space of the supervectors and directly evaluate the kernel using the language factors.

Since SVM-based classification is invariant to any translation of the feature space, we can ignore the constant term $\boldsymbol{\mu}_{UBM}$, and consider only the supervector $\mathbf{g} = \mathbf{V}\mathbf{y}$.

The normalized inner product of two supervectors expanded from the language factor vectors $\boldsymbol{y_a}$ and $\boldsymbol{y_b}$, can be evaluated as

$$\mathbf{g_a^t g_b} = \left(\boldsymbol{\lambda}^{\frac{1}{2}} \boldsymbol{\Sigma}^{-\frac{1}{2}} \mathbf{V} \mathbf{y_a} \right)^t \left(\boldsymbol{\lambda}^{\frac{1}{2}} \boldsymbol{\Sigma}^{-\frac{1}{2}} \mathbf{V} \mathbf{y_b} \right) = \mathbf{y_a^t} \hat{\mathbf{V}}^t \hat{\mathbf{V}} \mathbf{y_b} \quad (8)$$

In (8) $\boldsymbol{\lambda}$ is a diagonal matrix whose values are p repetitions of the weights λ_i, where p is the dimension of the observation vectors, $\boldsymbol{\Sigma}$ is a diagonal matrix whose values are the Σ_i and $\hat{\mathbf{V}} = \boldsymbol{\lambda}^{\frac{1}{2}} \boldsymbol{\Sigma}^{-\frac{1}{2}} \mathbf{V}$.

Performing the Cholesky decomposition $\hat{\mathbf{V}}^t \hat{\mathbf{V}} = \boldsymbol{\Lambda}^t \boldsymbol{\Lambda}$, we can evaluate the kernel function using the language factors $\hat{\mathbf{y}} = \boldsymbol{\Lambda}\mathbf{y}$, normalized by means of the low-dimension matrix $\boldsymbol{\Lambda}$, without actually returning to the supervector space. The Gram matrix allowing to effectively train the SVMs, thus, be computed by means of a dot product of the normalized language factors.

$$\mathbf{g_a^t g_b} = \hat{\mathbf{y}}_\mathbf{a}^\mathbf{t} \hat{\mathbf{y}}_\mathbf{b} \quad (9)$$

6. Language models

The straightforward usage of the language factors consists in creating the language models by training SVMs using one of the above mentioned kernels and a one versus all strategy. During testing, scoring an unknown utterance is performed using (5). In [7] we proposed an alternative way to perform scoring, which consists in exploiting the information given by the SVM classifier to create discriminative GMMs. Scoring test utterances is then done on these models, avoiding the need to estimate the language factors, which can lead to unreliable estimations on short test utterances. Another method, based on the use of the support vectors identified by the SVM, has been proposed in [8] for creating discriminative models. Since a support vector corresponds to a GMM supervector, the location of the positive boundaries of the SVM can be modeled by a weighted combination of the support vectors associated to positive Lagrange multipliers. In analogy, an anti-model can be obtained by the weighted sum of the support vectors associated with negative Lagrange multipliers.

$$\mathbf{g}_{positive} = \sum_{i|\alpha_i>0} \frac{\alpha_i}{\sum_{j|\alpha_j>0} \alpha_j} \cdot \mathbf{g}_i \qquad \mathbf{g}_{negative} = \sum_{i|\alpha_i<0} \frac{\alpha_i}{\sum_{j|\alpha_j<0} \alpha_j} \cdot \mathbf{g}_i \quad (10)$$

Thus, for each language, a GMM is created together with its anti-model, and the log likelihood ratio score for a test utterance including a sequence of T observation vectors $\mathbf{x_t}$ will be given by

$$llr = \sum_{t=1}^{T} \log(g_{positive}(\mathbf{x_t})) - \sum_{t=1}^{T} \log(g_{negative}(\mathbf{x_t})) \quad (11)$$

The same approach can be followed to obtain discriminative GMMs from the language factors

$$\mathbf{g}_{positive} = \boldsymbol{\mu}_{UBM} + \mathbf{V} \cdot \sum_{i|\alpha_i>0} \frac{\alpha_i}{\sum_{j|\alpha_j>0} \alpha_j} \cdot \mathbf{y}_i \qquad \mathbf{g}_{negative} = \boldsymbol{\mu}_{UBM} + \mathbf{V} \cdot \sum_{i|\alpha_i<0} \frac{\alpha_i}{\sum_{j|\alpha_j<0} \alpha_j} \cdot \mathbf{y}_i \quad (12)$$

7. Experiments

Experiments were performed on the 14 languages of the NIST 2007 language recognition task (LRE07) [12].

Results are given for the official LRE07 test set, including approximately 6500 utterances uniformly distributed for durations of 3, 10, and 30 seconds.

The reported evaluation measures are the NIST defined minimum Decision Cost Function score (minDCF) and the percent Equal Error Rate (% EER) uniformly weighted over the languages [12].

For all systems, the raw scores are transformed by means of a backend. In particular, a small linear SVM has been trained using as input features the raw scores of the development set described in Section 4. Since the distribution of utterances of different languages was not uniform in our development set, care has been taken to balance the a priori probability of the classes by appropriately setting the cost-factor parameter in the SVM train procedure.

The results of our reference system were obtained using the KL kernel in the GMM-SVM approach, and are shown in the first row on Table 1.

The first set of experiments were done to evaluate the performance of the identity kernel SVM classifiers as a function of the modeling eigen-language matrices, $\mathbf{V}_1$ or $\mathbf{V}_2$, and as function of the language subspace dimensions. The results for the 30s condition, shown in Figure 2, clearly demonstrate the advantage of using the language subspace modeled by $\mathbf{V}_2$ with respect to the speaker compensated eigenvoices subspace represented by matrix $\mathbf{V}_1$. The minDCF obtained with the language factors is always better and more stable compared with the minDCF achieved using the factors estimated with the speaker compensated eigenvoices matrix. Using only 100 language factors we get only a 10% relative decrease of the performance with respect to the best one obtained with 600 factors. All the other experiments were, thus, performed using 600 language factors of subspace $\mathbf{V}_2$.

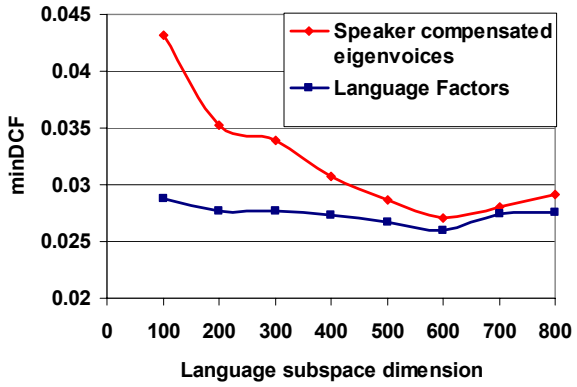

Figure 2. *MinDCF as a function of the number of language factors estimated using two language subspaces*

Table 1. *Min DCFs and (%EERs) for the general closed set tests in LRE07.*

Models	30s	10s	3s
GMM-SVM (KL kernel)	0.029 (3.43)	0.085 (9.12)	0.201 (21.3)
GMM-SVM (Identity kernel)	0.031 (3.72)	0.087 (9.51)	0.200 (21.0)
LF-SVM (KL kernel)	0.026 (3.13)	0.083 (9.02)	0.186 (20.4)
LF-SVM (Identity kernel)	0.026 (3.11)	0.083 (9.13)	0.187 (20.4)
Discriminative GMMs	0.021 (2.56)	0.069 (7.49)	0.174 (18.45)
LF-Discriminative GMMs (KL kernel)	0.025 (2.97)	0.084 (9.04)	0.186 (19.9)
LF-Discriminative GMMs (Identity kernel)	0.025 (3.05)	0.084 (9.05)	0.186 (20.0)

In the second set of experiments we compare the results of the language factor based SVM systems with the reference GMM-SVM system. As shown in Table 1, the low dimension language factor approach outperforms the standard GMM-SVM system, not only, as expected, in the short duration conditions, but also in the 30s condition. Surprisingly, the more theoretically sound KL kernel, does not give significant performance improvement. In light of these experimental findings, we trained a GMM-SVM using as input features the GMM supervector means without any KL normalization. The results shown in the second row of Table 1 confirm that the KL kernel is better than the identity kernel, but the relative performance gap is not marked.

The last set of experiments has been devoted to the analysis of the performance of the discriminative GMMs (the so called pushing approach in [8]). Comparing rows one and five of Table 1, it can be noticed that the discriminative GMMs give a relevant improvement with respect to the GMM-SVM approach. Unfortunately, as shown in the last two rows of the table, the pushing approach applied to the language factors as illustrated in (12) did not produce any improvement. The possible reasons are that the weighted combination of the eigen-language utterance models is not as effective as the combination of the MAP adapted models. We also found that, keeping fixed the SVM training parameters, the average number of support vectors trained for each language factor SVM is a small subset (approximately 1/10 in size) of the support vectors estimated for GMM-SVM.

8. Conclusions

We have proposed a language recognition approach in which we estimate an eigen-language space, in analogy to the eigen-voice space, and we use the language factors as input features to SVMs. The kernels based on the language factors performed well on the last NIST LRE07, outperforming our baseline KL GMM-SVM approach. State of the art results for acoustic only systems have been obtained using the pushing approach combining the support vectors produced by training the KL GMM-SVM. Further analysis and experiments are required to find a supervector combination that can achieve in the language factor SVM framework, the best performance provided by the pushing approach.

9. References

[1] M. Zissman, "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Trans. Speech and Audio Processing*, vol. 4, no. 1, pp. 31–44, 1996.

[2] J.L. Gauvain, A. Messaoudi, H. Schwenk. "Language recognition using phone lattices", Proc. ICSLP 2004, pp. 1283-1286, 2004.

[3] W. Campbell, F. Richardson, D. Reynolds, "Language recognition with word lattices and Support Vector Machines", Proc. ICASSP 2007, pp. 989-992, 2007.

[4] P. Torres-Carrasquillo, E. Singer, M. Kohler, R. Greene, D. Reynolds, J. Deller Jr., **"**Approaches to language identification using Gaussian Mixture Models and Shifted Delta Cepstral features", Proc. ICSLP-2002, pp. 33–36, 82–92, September 2002.

[5] L. Burget, P. Matejka, J. Cernocky, "Discriminative training techniques for acoustic language identification," Proc. ICASSP, 2006, pp. 209–212, 2006.

[6] W. Campbell, D. Sturim, D. Reynolds, A. Solomonoff, "SVM based speaker verification using a GMM supervector kernel and NAP variability compensation," Proc. ICASSP 2006, pp. 97–100, 2006.

[7] F. Castaldo, D. Colibro, E. Dalmasso, P. Laface, C. Vair, "Acoustic language identification using fast discriminative training," Proc. Interspeech, 2007, pp. 346-349, 2007.

[8] W. Campbell, "A covariance kernel for SVM language recognition", Proc. ICASSP 2008, pp. 4141-4144, 2008.

[9] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of inter-speaker variability in speaker verification," *IEEE Transactions on Audio, Speech and Language Processing,* Vol. 16, n. 5, pp. 980-988, 2008.

[10] P. Kenny, G. Boulianne, P. Ouellet, P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Trans. Audio Speech and Language Processing*, Vol. 15, n. 4, pp.1435-144, 2007.

[11] N. Dehak, P. Kenny, R. Dehak, O. Glembek, P. Dumouchel, L. Burget, V. Hubeika, F. Castaldo, "Support Vector Machines and Joint Factor Analysis for speaker verification", Proc. ICASSP 2009, to appear.

[12] Avaliable at www.nist.gov/speech/tests/lang/2007/

[13] F. Castaldo, D. Colibro, E. Dalmasso, P. Laface, C. Vair, "Compensation of Nuisance Factors for Speaker and Language Recognition", *IEEE Trans. on Audio, Speech, and Language Processing.* Vol. 15, n.7, pp. 1969-1978, 2007.

[14] W. Campbell, D. Sturim, P. Torres-Carrasquillo, D. Reynolds**, "**A Comparison of Subspace Feature-Domain Methods for Language Recognition", Proc Interspeech 2008, pp. 309-312, 2008.

[15] D. Reynolds, T. Quatieri and R. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models," *Digital Signal Processing*, Vol. 10, pp. 19-41, 2000.

[16] Available at http://www.ldc.upenn.edu/Catalog