DISCRIMINATIVE CLASSIFIERS FOR PHONOTACTIC LANGUAGE RECOGNITION WITH IVECTORS

Mehdi Soufifar[1,3], *Sandro Cumani*[2], *Lukáš Burget*[4,1], *Jan "Honza" Černocký*[1]

[1] Brno University of Technology, Speech@FIT, Czech Republic
[2] Politecnico di Torino, Italy
[3] Department of Electronics and Telecommunications, NTNU, Trondheim, Norway
[4] Speech Technology and Research Laboratory, SRI International, Menlo Park, CA, USA

qsoufifar@stud.fit.vutbr.cz, sandro.cumani@polito.it, {cernocky,burget}@fit.vutbr.cz

ABSTRACT

Phonotactic models based on bags of n-grams representations and discriminative classifiers are a popular approach to the language recognition problem. However, the large size of n-gram count vectors brings about some difficulties in discriminative classifiers. The subspace Multinomial model was recently proposed to effectively represent information contained in the n-grams using low-dimensional iVectors. The availability of a low-dimensional feature vector allows investigating different post-processing techniques and different classifiers to improve recognition performance. In this work, we analyze a set of discriminative classifiers based on Support Vector Machines and Logistic Regression and we propose an iVector post-processing technique which allows to improve recognition performance. The proposed systems are evaluated on the NIST LRE 2009 task.

Index Terms— Phonotactic iVector, Discriminative classifier, Support vector machine, Logistic regression

1. INTRODUCTION

Language recognition (LRE) techniques can be divided into phonotactic and acoustic approaches. In phonotactic LRE, one or more tokenizers (e.g. phone recognizers) are used to convert speech utterances into sequences of discrete tokens (e.g. phonemes). N-gram counts can be extracted from the decoder output. The n-gram statistics stacked in a fixed length vector can then be used as features for generative [1] or discriminative classifiers such as Support Vector Machines (SVM) [2]. However, in the latter case, we should deal with the large size of n-gram count vectors. A possible solution based on fast algorithms for linear SVM training was proposed in [3]. Another way to approach the problem is to reduce the dimensionality of the n-gram count vectors: in [4], discriminative selection of n-gram counts was proposed and in [5], dimensionality reduction through principle component analysis was used. Recently, we proposed phonotactic iVectors as an alternative method for reducing n-gram vectors dimensions [6]. The iVectors based on Subspace Multinomial model [7] allow us to effectively represent the information contained in the n-grams using low-dimensional vectors, which greatly simplifies training of the discriminative classifiers and allows us to study the effectiveness of different classification approaches.

In this paper, we analyze the behavior of classifiers based on SVM and Logistic Regression (LR), both in their binary and multi-class formulations. We also propose a technique to obtain multi-class scores starting from binary pairwise classifiers (i.e. classifiers trained to discriminate between a pair of languages) Further, we introduce an effective iVector post-processing step, which allows us to improve recognition accuracy. The experiments are conducted on the NIST LRE 2009 [8] task and all results are given in terms of the average decision cost function (C_{avg}) as defined by NIST for LRE 2009 [8].

The paper is organized as follows. Section 2 explains subspace modeling and iVector extraction. In section 3, different approaches for discriminative training of the classifiers are presented. Section 4 presents the experimental setup. Results and conclusions are given in sections 5 and 6, respectively.

2. SUBSPACE MODELS

2.1. Total variability subspace model

The original total variability subspace model [9] was proposed for continuous features using Gaussian mixture models (GMM). The basic assumption is that feature sequence of each utterance was generated from an utterance specific GMM model. More precisely, we assume that the utterance specific mean super-vector

This work was partly supported by Technology Agency of the Czech Republic grant No. TA01011328, and Czech Ministry of Education project No. MSM0021630528.

$$\boldsymbol{\phi} = \mathbf{m} + \mathbf{T}\boldsymbol{\omega} \tag{1}$$

is constrained to live in a low-dimensional subspace space with origin in $\mathbf{m}$ and spanned by the columns of the matrix $\mathbf{T}$. An iVector, which serves as the low-dimensional representation of a given utterance, is then computed as the maximum–a–posteriori (MAP) point estimate of the latent vector $\boldsymbol{\omega}$ adapting GMM to the feature sequence of the given utterance.

2.2. Multinomial subspace model

Discrete features can also be modeled in a subspace paradigm [7]. Utterances represented by sequences of discrete events can be modeled using multinomial distributions and, similar to the continuous feature case, we can assume that there is a low-dimensional subspace in which the parameters of the multinomial distributions for individual utterances live. In the context of phonotactic LRE, every speech utterance can be represented by a fixed-length vector containing discrete n-gram statistics. The log-likelihood of the n^{th} utterance represented by an E-dimensional vector of n-gram counts ($\boldsymbol{\nu}_n$) can be computed as

$$\log(P(\boldsymbol{\nu}_n \mid \boldsymbol{\phi}_n)) = \sum_{e=1}^{E} \nu_{ne} \log \phi_{ne}, \tag{2}$$

where ν_{ne} is occupation count for n-gram e in utterance n. The ϕ_{ne} is the utterance-dependent model parameter representing probability for the corresponding n-gram. Log-likelihood of a set of utterances is given by

$$\sum_{n=1}^{N} \log P(\boldsymbol{\nu}_n \mid \boldsymbol{\phi}_n), \tag{3}$$

where N is the number of utterances. We constrain the utterance dependent model parameters ϕ_{ne} to live in a low-dimensional manifold using log-linear model:

$$\phi_{ne} = \frac{\exp(m_e + \mathbf{t}_e \boldsymbol{\omega}_n)}{\sum_{i=1}^{E} \exp(m_i + \mathbf{t}_i \boldsymbol{\omega}_n)}, \tag{4}$$

where $\boldsymbol{\omega}_n$ is an utterance-dependent latent variable and $\boldsymbol{t}_e$ is the e^{th} row of subspace matrix $\mathbf{T}$.

Given the parameters $\mathbf{m}$ and $\mathbf{T}$, we can estimate $\boldsymbol{\omega}$ to maximize the log-likelihood in (2) for the corresponding utterance. The estimated $\boldsymbol{\omega}$ is called iVector. Similar to the case of continuous features, the subspace multinomial model is used as a feature extractor and each iVector can be seen as a low-dimensional representation of the whole utterance. The model parameters $\mathbf{T}$ and $\boldsymbol{\omega}$ are estimated by means of an iterative numerical optimization algorithm. Detailed description of iVector extraction and parameter estimation can be found in [6].

3. CLASSIFIERS

Our classifiers are based on LR and SVM [10], trained in either of binary or multi-class flavour.

Both classifiers can be expressed as particular instances of a more general class of unconstrained (regularized) risk minimization problems of the form

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{2}\lambda\|\mathbf{w}\|^2 + \frac{1}{N}\sum_{i=1}^{N} l(\mathbf{w}, \mathbf{x_i}, y_i) \tag{5}$$

where $\mathbf{w}^*$ defines the class boundaries, λ is the regularization coefficient, $\mathbf{x}_i$ is the i–th training pattern with associated label y_i and l is a convex function of $\mathbf{w}$ representing the empirical loss.

3.1. One–versus–All training

One–versus–all training is a way to obtain multi-class scores from a set of binary classifiers. In particular, for each class C_i, a classifier is built to separate the patterns of class C_i from all the patterns belonging to the other classes. Multi-class scores are obtained as the scores of the test utterance for each of these classifiers.

3.1.1. SVM

Binary SVM separates classes according to a maximum–margin criterion [10], the margin being related to the norm of $\mathbf{w}$. When classes are not linearly separable, a soft-margin solution is built as a tradeoff between the margin and the misclassification cost, evaluated on training data. Using the coding scheme $y_i \in \{-1, +1\}$ to represent class labels for the binary problem, the misclassification cost corresponds to the hinge loss function

$$l(\mathbf{w}, \mathbf{x_i}, y_i) = \max(0, 1 - y_i \mathbf{w}^T \mathbf{x_i}). \tag{6}$$

3.1.2. Logistic Regression

Under rather general assumptions [10], class posterior probabilities can be expressed as a logistic sigmoid function applied to a linear function of the observed data $\mathbf{w}^T\mathbf{x}_i$. With such logistic regression model, value of $\mathbf{w}$ are estimated on a training data to maximize conditional likelihood of labels y_i given the corresponding observations $\mathbf{x}_i$ (i.e. to maximize the probability that all training examples are recognized correctly). This corresponds to minimizing objective function (5) with *cross entropy* loss function

$$l(\mathbf{w}, \mathbf{x}_i, y_i) = \log(1 + e^{-y_i \mathbf{w}^T \mathbf{x}_i}). \tag{7}$$

When training samples are completely separable by the logistic regression, the solution is no more unique and tends to largely over-fit the training data [10]. To solve this issue, a prior on $\mathbf{w}$ can be introduced, which is equivalent to adding the regularization term in objective function (5).

3.2. Multi-class SVM and Logistic Regression

For multi-class SVM and multi-class LR, the parameter $\mathbf{w}$ is actually a matrix $\mathbf{W} = [\mathbf{w}_1, \dots \mathbf{w}_n]$ representing a set of hyper-planes, one for each class.

The class scores for test patterns are then given by the projection of test patterns over set of hyper-planes $\mathbf{w}_i$. The process is similar to the One–versus–All technique, however, the training stage involves slightly different loss functions and the hyper-planes are jointly optimized.

3.2.1. Multi-class SVM

The loss function for multi-class SVM [11] is an extension of the binary loss function given by

$$l(\mathbf{W}, \mathbf{x}_i, y_i) = \max_{y'}(\mathbf{w}_{y'}^T \mathbf{x}_i - \mathbf{w}_{y_i}^T \mathbf{x}_i + \Delta(y', y_i)) \qquad (8)$$

where $\Delta(y_1, y_2)$ is the cost of misclassifying class y_1 for y_2. In our context, we use $\Delta(y_i, y_j) = 1 - \delta_{ij}$, where δ_{ij} is the Kronecker delta. Multi-class SVM can be interpreted as a joint optimization of N SVMs where the hyper-planes are trained as to maximize the margin between each class and all the remaining classes [10].

3.2.2. Multi-class LR

For multi-class variant of logistic regression, the multi-class cross entropy loss function

$$l(\mathbf{W}, \mathbf{x}_i, y_i) = \left(\log \sum_{y'} e^{\mathbf{w}_{y'}^T \mathbf{x}_i} \right) - \mathbf{w}_{y_i} \mathbf{x} \qquad (9)$$

is used, which again results in the objective function, where the probability of recognizing all training examples correctly is maximized.

3.3. Binary Classifiers with multi-class score projection

The last set of classifiers we explore are binary classifiers trained to produce pairwise scores. In this approach, one binary classifier is trained for each pair of languages. The resulting binary scores are then mapped into multi-class scores. The description of the loss functions is the same as the one in section 3.1.

3.3.1. Scores back-projection

Assuming that multi-class scores represent class–conditional log–likelihoods, log–likelihood ratios for two languages can be computed as the difference between the scores for the two languages. The mapping from multi-class scores to binary scores can then be expressed as a linear transformation represented by a rectangular $N \times \frac{1}{2}N(N-1)$ matrix whose entries are in $\{-1, 0, +1\}$. Each row of the matrix maps the multi-class scores to a binary score and thus has exactly one element valued $+1$ and one element valued -1.

To map binary scores scores to multi-class conditional log–likelihoods, we simply project the binary scores on the pseudo inverse of the multi-class–to–binary transformation matrix.

4. EXPERIMENTAL SETUP

We report our results on NIST LRE2009. The task comprises 23 languages. The evaluation dataset contains telephone data and narrowband broadcast data. [1].

4.1. Data

The training data is divided into two sets denoted as TRAIN and DEV. The TRAIN set comprises data from 23 languages corresponding to the target languages of the NIST LRE09 task [8]. The TRAIN set is filtered in order to keep at most 500 utterances per language as proposed in [12], resulting in 9763 segments (345 hours of recording). This allows to have almost balanced amounts of training data per language, thus avoiding biasing the classifiers toward languages with lots of training data. The DEV set contains 38469 segments from the same 23 languages and consists of data from the previous NIST LRE tasks plus some extra longer segments from the standard conversational telephone speech (CTS) databases (CallFriend, Switchboard, etc.) and voice of America (VOA). The TRAIN and the DEV sets contain disjoint sets of speakers. Full description of the dat used is given in [13]. The DEV set is used to choose the iVector extraction parameters and to calibrate the classifier scores.

4.2. Vector of n-gram counts

The n-gram counts were extracted using the Brno University of Technology (BUT) Hungarian (HU) phone recognizer, which is an ANN/HMM hybrid [14]. The HU phoneme list contains 61 phonemes. We map short and long variations of similar phones to the same token, obtaining 33 phonemes. This results in $33^3 = 35937$ 3-grams. Since neither 2-grams nor 1-grams improved the system performance we use only 3-gram counts. The 3-gram expected counts are extracted from phone lattices generated by the HU phone recognizer. We also computed square roots of the expected n-grams counts before going through other steps in all the systems. The square root compresses the dynamic range of the counts and slightly improves the performance over all systems.

5. SYSTEM EVALUATION & RESULTS

600 dimensional iVectors are extracted from expected 3-gram counts. The baseline system is a binary LR trained in a one–

[1] Detailed distribution of segments per language can be found on the official NIST LRE09 results page `http://www.itl.nist.gov/iad/mig/tests/lre/2009/lre09_eval_results/index.html`

Table 1. $C_{avg} \times 100$ for different systems on NIST LRE09 Evaluation task over 30s, 10s and 3s conditions.

System	Scores type	30s	10s	3s
LR	one-vs-all	2.98	8.25	21.37
SVM	one-vs-all	3.07	8.55	21.68
MC-LR	multi-class	3.16	8.66	21.82
MC-SVM	multi-class	3.89	10.60	23.92
MC-SP-LR	projection	**2.93**	**8.14**	**21.32**
MC-SP-SVM	projection	3.91	9.86	23.03

versus–all flavour. The iVectors are centered and whitened before the classifier is trained. The scores generated by 23 LR classifiers are calibrated on DEV data by means of a linear generative model followed by a multi-class LR as described in [13]. This setup is similar to the one used in [6], except for the presence of a regularization term and the whitening preprocessing step, which is necessary due to the presence of the regularizer.

Table 1 shows the results in terms of C_{avg} [8] using the classifiers described in Section 3. "MC" refers to the multi-class formulation of the classifiers and "SP" stands for binary pairwise systems with Score Projection.

We can observe that regularized logistic regression–based models give better results than SVM–based techniques. Surprisingly, the one-versus-all approach shows better accuracy than the multi-class system. Another set of experiments was performed adding an iVector post-processing step based on Heteroscedastic Discriminant Analysis (HDA) [15] to reduce dimensionality to 22 (number of languages minus one). Within Class Covariance Normalizatioin followed by length normalization was then applied to the 22-dimensional features. The choice for these steps was dictated by the success of Linear Discriminant Analysis and WCCN in acoustic iVector post-processing for speaker recognition [9].

The results are shown in table 2. We can observe that the different classifiers, trained in this 22-dimensional subspace, achieve almost the same performance, with multi-class systems giving slightly better results than binary systems. iVector post-processing proves to be valuable to improve the system performance.

6. DISCUSSIONS & CONCLUSIONS

We studied performance of two discriminative classifiers based on iVectors for the NIST LRE2009 task. Without any post-processing of iVectors, LR performs slightly better than SVMs, and binary classification allows to achieve slightly better results than multi-class systems. iVector post-processing by HDA, WCCN and length normalization allows to achieve 5% relative improvement over the baseline systems. In the latter case, the choice of the classifier makes very little difference.

Table 2. $C_{avg} \times 100$ for different systems on NIST LRE09 Evaluation task over 30s, 10s and 3s conditions after HDA, WCCN and length normalization.

System	Scores type	30s	10s	3s
LR	one-vs-all	2.83	8.09	21.34
SVM	one-vs-all	2.83	8.09	21.34
MC-LR	multi-class	2.79	8.06	21.35
MC-SVM	multi-class	2.81	8.05	21.33
SP-LR	projection	2.80	8.09	21.31
SP-SVM	projection	2.82	8.04	21.33

7. REFERENCES

[1] J. Navratil, "Recent advances in phonotactic language recognition using binary-decision trees," in *Proc. of Interspeech*, Pittsburgh, PA, USA, 2006.

[2] W.M. Campbell, F. Richardson, and D.A. Reynolds, "Language recognition with word lattices and support vector machines," in *Proc. of ICASSP*, Honolulu, Hawaii, USA, 2007.

[3] S. Cumani et al., "Comparison of large-scale SVM training algorithms for language recognition," in *Proc. of Odyssey 2010*, 2010.

[4] F.S. Richardson and W.M. Campbell, "Language recognition with discriminative keyword selection," in *Proc. of ICASSP*, Las Vegas, USA, 2008.

[5] T. Mikolov et al., "PCA-based feature extraction for phonotactic language recognition," in *Proc. of Odyssey*, Brno, CZ, 2010.

[6] M. Soufifar et al., "iVector approach to phonotactic language recognition," in *Proc. of Interspeech*, Florence, IT, 2011.

[7] M. Kockmann et al., "Prosodic speaker verification using subspace multinomial models with intersession compensation," in *Proc. of Interspeech*, Makuhari, Chiba, Japan, 2010.

[8] "NIST language recognition evaluation plan," 2009, http://www.itl.nist.gov/iad/mig/tests/lre/2009/.

[9] N. Dehak et al., "Front-end factor analysis for speaker verification," *ASLP, IEEE Transactions on*, vol. 19, no. 4, pp. 788 –798, 2011.

[10] Christopher M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer, 1st ed. 2006. corr. 2nd printing edition, Oct. 2007.

[11] Ben Taskar, Carlos Guestrin, and Daphne Koller, "Max-margin Markov networks," 2003, MIT Press.

[12] Niko Brummer et al., "BUT-AGNITIO system description for NIST LRE 2009," http://www.fit.vutbr.cz/research/view_pub.php.en?id=9551.

[13] Z. Jancik et al., "Data selection and calibration issues in automatic language recognition - investigation with BUT-Agnitio NIST LRE 2009 system," in *Proc. of Odyssey*, Brno, CZ, 2010.

[14] P. Schwarz, P. Matejka, and J. Cernocky, "Hierarchical structures of neural networks for phoneme recognition," in *Proc. of ICASSP*, Toulouse, FR, 2006.

[15] G. Saon et al., "Maximum likelihood discriminant feature spaces," in *Proc. ICASSP*, 2000, pp. 1129–1132.