DISCRIMINATIVE TRAINING OF WEIGHTED POLYNOMIAL VECTOR FOR ACOUSTIC LANGUAGE RECOGNITION

Ce Zhang, Rong Zheng, Bo Xu

Digital Content Technology Research Center, Institute of Automation
Chinese Academy of Sciences, Beijing, China
{czhang, rzheng, xubo}@hitic.ia.ac.cn

ABSTRACT

In this paper, we propose a discriminative method for the acoustic feature based language recognizer, which is a modification of the polynomial expansion in generalized linear discriminant sequence (GLDS) kernel. It is inspired by the Gaussian mixture model-support vector machine (GMM-SVM) system which has been successfully used in both speaker and language recognition. Because of the restriction of calculations in our method, it is nearly impossible to stack component dependent polynomial expansion vectors as GMM-SVM system does. Thus we introduce a set of language dependent weights to fuse these expansion vectors and utilize maximum mutual information(MMI) criterion and logistic regression to estimate the model parameters. Finally, we evaluate our method on the close-set, 30 seconds test condition of NIST LRE 2007 and up to 30% relative improvement can be achieved comparing to the baseline GLDS system.

***Index Terms*—** Language recognition, weighted GLDS, GMM, maximum mutual information, multi-class logistic regression

1. INTRODUCTION

Language recognition from speech involves the algorithms and techniques that model and classify the language being spoken. Current state-of-the-art language recognition systems can be divided into two categories: spectral based (acoustic) and token based (phonotactic) [1]. Token based language recognizers segment the speech signals into logic units based on phoneme recognizer which is followed by language model or classifier such as SVM.

Unlike the token based systems, acoustic language recognizers extract cepstral features in fixed frame length, e.g. MFCC, PLP, and directly model these features for different languages. Two kinds of modeling techniques commonly used in language recognition are the generative model-GMM and the discriminative model-SVM. In [2] [3], the authors proposed GLDS kernel for SVM based speaker and language recognition system. The GLDS kernel is simply an inner product between two averaged high dimensional polynomial expansion vectors. Probably due to the GLDS kernel which reduces the computational load and preserves model complexity advantage, our experiments show that it performs worse than GMM-SVM system [4] in both speaker and language recognition. While GLDS kernel treats each frame of an utterance equally without discrimination, GMM-SVM maps different acoustic features to different Gaussian components so as to highlight the importance of different features.

Empirical evidence shows that some phonemes are more useful for us to distinguish specific language than the others. GMM-SVM system tends to strengthen those useful frames and weaken the useless ones through the component alignment. This behavior is consistent with that of human beings and however GLDS system seems not to be the case.

The contribution of this work is that we change the mean of polynomial expansion vectors in the original GLDS system to the weighted mean. Additionally, we propose two techniques to discriminatively train the weights. The first one assumes that the weighted polynomial vectors are normally distributed with different means and covariances conditioned on different languages. We use MMI criterion to maximize the posteriori of correctly classifying the training segments with respect to the weights and then train classifier for each language with SVM. In the second approach, a linear classifier is selected to optimize the weights and classification hyperplanes simultaneously via multi-class logistic regression.

The paper is organized as follows. Section 2 gives a brief introduction to the polynomial expansion and GMM super-vector. In section 3, we first introduce the form of weighted expansion vectors and then propose two discriminative training methods successively. We present our experiments configuration and results in section 4, 5 respectively. Finally, section 6 draws some conclusions.

2. THEORETICAL BACKGROUND

2.1. Polynomial expansion in GLDS

Given a sequence of F-dimensional acoustic feature vectors of an utterance, $\boldsymbol{S} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_T)$, the GLDS kernel first maps the variable length of sequence to a fixed length of polynomial expansion vector, $\overline{\boldsymbol{b}}_{\boldsymbol{s}}$. That is,

$$\boldsymbol{S} \rightarrow \overline{\boldsymbol{b}}_s = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{b}(\boldsymbol{x}_t) \tag{1}$$

where $\boldsymbol{b}(\boldsymbol{x})$ is the polynomial expansion vector of all monomials of the input feature $\boldsymbol{x}$ up to and including degree $K(K = 3$ in our experiment). The dimension of $\boldsymbol{b}(\boldsymbol{x})$, denoted as E(which means Expansion), is the same as binomial coefficient $\binom{F+K}{K}$,

$$E = \binom{F+K}{K} = \frac{(F+K)!}{F!K!} \tag{2}$$

GLDS kernel function between two sequences, $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, is defined as

$$K_{GLDS}(\boldsymbol{S}_1, \boldsymbol{S}_2) = \overline{\boldsymbol{b}}'_{s_1}\boldsymbol{R}^{-1}\overline{\boldsymbol{b}}_{s_2} \tag{3}$$

Supported by the National Natural Science Foundation of China (Grant No. 90820303)

978-1-4673-0046-9/12/$26.00 ©2012 IEEE

where $\boldsymbol{R}$ is a correlation matrix derived from a large data set including multiple languages and is usually diagonal. The function of $\boldsymbol{R}^{-1}$ is variance normalization for the polynomial vectors.

Note that the input of SVM, $\bar{\boldsymbol{b}}_s$, is the mean of each polynomial vectors $\boldsymbol{b}(\boldsymbol{x}_t)$. We will introduce the weighted mean of $\boldsymbol{b}(\boldsymbol{x}_t)$ in section 3.

2.2. GMM super-vector

Given a well trained GMM universal background model(UBM) which consists of C mixtures, the zero and first order sufficient statistics are defined as,

$$n(c) = \sum_t \gamma_t(c) \tag{4}$$

$$\boldsymbol{f}(c) = \sum_t \gamma_t(c)\boldsymbol{x}_t \tag{5}$$

where $\gamma_t(c)$ is the posterior probability of the event that the feature vector $\boldsymbol{x}_t$ is emitted by component c, $c \in [1, C]$. The utterance dependent GMM is updated using (4) and (5) by relevance maximum a posteriori (MAP) adaptation [5] only for the means of each component. In GMM-SVM system [4], the key step is a mapping between sequence $\boldsymbol{S}$ and the super-vector which is a CF-dimensional vector formed by concatenating the means of each component. The super-vector is further taken as feature input to SVM.

3. DISCRIMINATIVE TRAINING

Comparing GLDS system and GMM-SVM system, we can find that in GLDS system each frame of utterance $\boldsymbol{S}$ has the same contribution to the final vector (1). However, the GMM-SVM system assigns different weights to each frame according to their position in the UBM space, which seems more reasonable because language specific information usually contained in separate components.

3.1. Weighted polynomial expansion

Inspired by the GMM-SVM system, we consider weighting the individual expansion vector, $\boldsymbol{b}(\boldsymbol{x})$, according to component alignment of $\boldsymbol{x}$. Mathematically, we define component dependent expansion vectors $\boldsymbol{g}_s(c)$ for sequence $\boldsymbol{S}$,

$$\boldsymbol{g}_s(c) = \frac{\sqrt{C}}{T} \sum_{t=1}^{T} \gamma_t(c)\boldsymbol{b}(\boldsymbol{x}_t) \tag{6}$$

where the normalization coefficient $\frac{\sqrt{C}}{T}$ is used to guarantee the consistency with original GLDS.

An intuitive thought is to concatenate $\boldsymbol{g}_s(c)$ of each component to form a single super-expansion vector just as the role of super-vector in GMM-SVM system. Unfortunately, this is computationally intractable because of the large scale classification problem. In our experiment, the number of training utterance is order of 10^4 and concatenating $\boldsymbol{g}_s(c)$ yields approximately 10^7 dimensional feature as input to classifiers. And therefore, we use a linear combination of $\boldsymbol{g}_s(c), c \in [1, C]$ which produces a vector of the same size as original GLDS system.

As mentioned before, the weights should be language dependent to distinguish the importance of different components for different languages. We define the C-dimensional weight $\boldsymbol{\alpha}_l$ for language $l \in [1, L]$, where L is the language number to be recognized. Given segment $\boldsymbol{S}$, let $\boldsymbol{G}_s$ be a $E \times C$ matrix whose columns are $\boldsymbol{g}_s(c), c \in [1, C]$ and $\boldsymbol{g}_s$ be the weighted polynomial expansion vector,

$$\boldsymbol{g}_s = \boldsymbol{G}_s \boldsymbol{\alpha}_{l(s)} \tag{7}$$

where $\boldsymbol{G}_s = [\boldsymbol{g}_s(1), \boldsymbol{g}_s(2), ..., \boldsymbol{g}_s(C)]$ and $l(s)$ denotes the language label of segment s.

Note that if $\forall l \in [1, L], \boldsymbol{\alpha}_l = (\frac{1}{\sqrt{C}}, \frac{1}{\sqrt{C}}, ..., \frac{1}{\sqrt{C}})'$, then $\boldsymbol{g}_s = \bar{\boldsymbol{b}}_s$ where $\bar{\boldsymbol{b}}_s$ is defined in (1). As a result, GLDS system is a special case of our weighted approach when $\boldsymbol{\alpha}_l, l \in [1, L]$ assign flat prior to each component.

The remaining problem is to estimate the language dependent parameters. We consider two discriminative training techniques, namely MMI and logistic regression.

3.2. MMI optimization followed by SVM

As an improvement of the GMM based language recognition system, the authors [6] retrained the language dependent GMMs using maximum mutual information estimation. Unlike the conventional maximum likelihood(ML) training which aims to maximize the overall likelihood of the training data, MMI objective is to maximize the posterior probability of correctly recognizing all training segments.

We follow the MMI criterion to train the language dependent weights $\boldsymbol{\alpha}_l, l \in [1, L]$. The purpose of this step is to improve the discrimination between classes which will be visualized in section 5.1. Specifically, instead of the GMM hypothesis of each language in [6], we assume the class conditional probability density of language l is multivariate normal with mean $\boldsymbol{\mu}_l$ and diagonal covariance $\boldsymbol{\Sigma}_l$ in the expansion vector space.

$$\begin{aligned} \log p(\boldsymbol{G}_s|l) = &-\frac{E}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}_l| \\ &-\frac{1}{2}\left(\boldsymbol{G}_s\boldsymbol{\alpha}_l - \boldsymbol{\mu}_l\right)' \boldsymbol{\Sigma}_l^{-1} \left(\boldsymbol{G}_s\boldsymbol{\alpha}_l - \boldsymbol{\mu}_l\right) \end{aligned} \tag{8}$$

where

$$\boldsymbol{\mu}_l = \frac{1}{N_l}\sum_{s \in l} \boldsymbol{g}_s \tag{9}$$

$$\boldsymbol{\Sigma}_l = \mathrm{diag}\left(\frac{1}{N_l}\sum_{s \in l} \boldsymbol{g}_s\boldsymbol{g}_s' - \boldsymbol{\mu}_l\boldsymbol{\mu}_l'\right) \tag{10}$$

N_l is the number of training segments of language l and $\sum_{l=1}^{L} N_l = N$. According to [6], the optimization problem can be formularized as follows:

$$\begin{aligned} \min_{\boldsymbol{\alpha}_l \in \Re^C} \quad & -\sum_s \log \frac{p\left(\boldsymbol{G}_s|l(s)\right)p(l(s))}{\sum_{\forall l} p(\boldsymbol{G}_s|l)p(l)} \\ \text{s.t.} \quad & \|\boldsymbol{\alpha}_l\|^2 = 1,\ l \in [1, L] \end{aligned} \tag{11}$$

where $p(l)$ is the prior probability of language l and $\|\boldsymbol{\alpha}_l\|^2 = \boldsymbol{\alpha}_l' \cdot \boldsymbol{\alpha}_l$. We add the constraint $\|\boldsymbol{\alpha}_l\|^2 = 1$ to avoid over-training. (11) is an optimization problem with equality constraints. We introduce the augmented Lagrange penalty function [7]

$$\begin{aligned} \mathcal{L}_{MMI} = &-\sum_s \log \frac{p(\boldsymbol{G}_s|l(s))p(l(s))}{\sum_{\forall l} p(\boldsymbol{G}_s|l)p(l)} \\ &-\sum_l \lambda_l(\|\boldsymbol{\alpha}_l\|^2 - 1) + \frac{\sigma}{2}\sum_l(\|\boldsymbol{\alpha}_l\|^2 - 1)^2 \end{aligned} \tag{12}$$

where $\boldsymbol{\lambda} = (\lambda_1, ..., \lambda_L)'$ are the Lagrange multipliers and σ is the penalty factor.

Given fixed $\boldsymbol{\lambda}$ and σ, we use conjugate gradient method to solve (12). To obtain the solution of (11), we need to minimize (12) with respect to $\boldsymbol{\alpha}_l$ step by step with various $\boldsymbol{\lambda}$ and σ. The optimization process is listed in algorithm 1.

Algorithm 1 Augmented Lagrangian Method-Equality Constraints

```
Require: $\boldsymbol{\alpha}_l^{(1)} = [\frac{1}{\sqrt{C}}, ..., \frac{1}{\sqrt{C}}]', \lambda_l^{(1)} = 1, \sigma^{(1)} = 1, l \in [1, L], \epsilon > 0, k = 1$
loop
    Find an approximate minimizer $\boldsymbol{\alpha}_l^{(k)}, l \in [1, L]$ of (12)
    if $\sum_l \left( \|\boldsymbol{\alpha}_l^{(k)}\|^2 - 1 \right)^2 \leq \epsilon$ then
        return $\boldsymbol{\alpha}_l^{(k)}$
    end if
    $\lambda_l^{(k+1)} \leftarrow \lambda_l^{(k)} - \sigma^{(k)} \left( \|\boldsymbol{\alpha}_l^{(k)}\|^2 - 1 \right)$
    Choose new penalty parameter $\sigma^{(k+1)} \geq \sigma^{(k)}$
    $k \leftarrow k + 1$
end loop
```

Since $\boldsymbol{\alpha}_l$ is obtained, we utilize one-versus-all strategy to train SVM for each language. The training procedure is similar to GLDS system except that the feature as input to SVM is weighted polynomial expansion vectors (7) rather than (1). In recognition, the score for language l of test segment s is

$$\hat{\theta}_{sl} = \boldsymbol{w}_l'(\boldsymbol{G}_s \boldsymbol{\alpha}_l) - b_l \qquad (13)$$

where $\{\boldsymbol{w}_l, b_l\}$ are the parameters of language l trained by SVM.

3.3. Multi-class logistic regression

In the previous section, we have described a method to train language dependent models which is composed of two steps: optimization of the weights $\boldsymbol{\alpha}_l, l \in [1, L]$ using MMI criterion which is followed by SVM to obtain the separating hyperplanes. Next, we introduce how to train the hyperplane $\boldsymbol{w}_l$ and weight $\boldsymbol{\alpha}_l$ simultaneously. If we drop the offset term in (13), a linear classifier can be obtained

$$\begin{aligned} \theta_{sl} &= \boldsymbol{w}_l' \boldsymbol{G}_s \boldsymbol{\alpha}_l & (14) \\ &= \boldsymbol{\alpha}_l' \boldsymbol{G}_s' \boldsymbol{w}_l & (15) \end{aligned}$$

It is clear that $\boldsymbol{w}_l$ and $\boldsymbol{\alpha}_l$ are symmetrical in a bilinear form. This implies us to use multi-class logistic regression to optimize these parameters.

Logistic regression (LR) solves a classification task which assumes a sigmoid function acting on a linear model. Rather than one-versus-one or one-versus-all decomposition of a multi-class classification problem used in SVM, the multi-class logistic regression model can be trained simultaneously to discriminate between all classes [8]. There is a small change in our application that the score is a bilinear form (14) rather than linear. However, there is not much difference in the optimization procedure. The objective is to minimize

$$\mathcal{L}_{LR} = -\sum_s \log \frac{\exp(\theta_{sl(s)})}{\sum_l \exp(\theta_{sl})} + \frac{\delta}{2} \sum_{l=1}^{L} (\|\boldsymbol{w}_l\|^2 + \|\boldsymbol{\alpha}_l\|^2) \qquad (16)$$

where δ is a coefficient which controls the ratio between error function and regularizer. The parameter set $\Theta = \{\boldsymbol{\alpha}_l, \boldsymbol{w}_l\}$ can be optimized simultaneously by conjugate gradient method. In recognition phase, the score of test segment s for language l is obtained by (14) or (15) which gives log-likelihood interpretation.

3.4. Analysis of the training procedure

In mathematics, the bilinear form $\boldsymbol{x}'\boldsymbol{A}\boldsymbol{y}$ can be expressed in terms of Frobenius inner product. That is,

$$\begin{aligned} \boldsymbol{x}'\boldsymbol{A}\boldsymbol{y} &= \sum_{ij} \boldsymbol{A}_{ij} \boldsymbol{x}_i \boldsymbol{y}_j & (17) \\ &= vec(\boldsymbol{x}\boldsymbol{y}')' \cdot vec(\boldsymbol{A}) \end{aligned}$$

where $vec(\cdot)$ concatenates the columns of a matrix into a vector. And therefore, the score (14) can be rewritten as an inner product between a decision vector and a super-expansion vector which is formed by concatenating the component dependent polynomial vectors (6)

$$\begin{aligned} \theta_{sl} &= vec(\boldsymbol{w}_l \boldsymbol{\alpha}_l')' \cdot vec(\boldsymbol{G}_s) & (18) \\ &= \varphi_l' \cdot vec(\boldsymbol{G}_s) \end{aligned}$$

In section 3.1 we mentioned that it is unmanageable to directly perform classifier training for the big super-expansion vector $vec(\boldsymbol{G}_s)$ to obtain φ_l. In fact, (18) indicates that our weighted polynomial method, which optimizes the different components of parameter separately, is indeed an alternative solution to the super-expansion vector classification.

4. EXPERIMENT

4.1. Data set

We evaluate our algorithms on the 14 languages of the close-set language detection task of the NIST 2007 Language Recognition Evaluation(LRE07). Only those test segments with 30 seconds duration are selected to measure the performance. All the development and training data are distributed before LRE07 and do not overlap with LRE07 evaluation data.

4.2. Configuration

Common feature set of SDC 7-1-3-7 concatenated with 7 static PLP coefficients are calculated to produce 56 dimensional feature vectors. The feature streams are processed through energy-based speech activity detection to eliminate low-energy speech vectors. Mean variance normalization, feature warping and feature domain intersession compensation (FDIC) [9] are then applied to reduce session variability during the training and testing procedures.

We pool all the language training data to estimate the UBM with 256 components for the calculation of component posteriori $\gamma_t(c)$. Although a larger UBM(e.g. 1024 mixtures) is likely to capture more details of acoustic distribution, it will increase considerable computations in the weighted polynomial extraction process (6).

4.3. Evaluation criteria

We use two criteria to judge our language recognizer. The first one is multi-class classification error rate. Although it does not take the costs and priors into consideration, it reflects the discrimination of the recognizer. The second one is average cost performance, C_{avg}, as defined in the LRE07 evaluation plan [10].

5. RESULTS

5.1. Visualization of MMI

In order to examine the effects of MMI training (11), we use principal components analysis (PCA)[11] to visualize what happened. For simplicity, only 3 languages(Chinese, Arabic, Russian) are selected and we depict the first two principal directions of the polynomial vectors in figure 1. Each star in figure 1 represents an individual segment. The left panel shows the original expansion vectors (1) while the right shows the weighted expansion vectors (7) which are optimized via MMI. It is clear that the stars in the right panel are more separable between languages than those in the left. MMI training procedure described in section 3.2 not only reduce the number of misclassified samples in the marginal region, but also increase the distances between each pair of language centers.

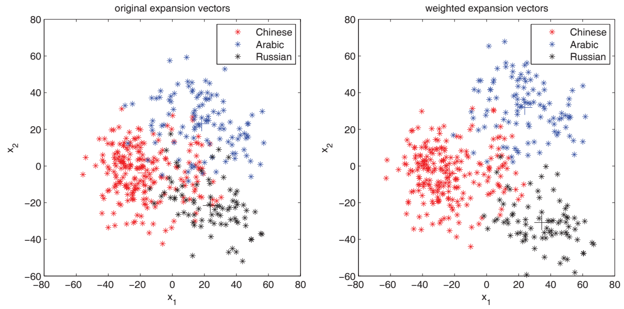

Fig. 1. Left panel: original expansion vectors defined by equation(1). Right panel: weighted expansion vectors defined by equation(7).

5.2. Comparative results

Experimental results are shown in table 1 in terms of C_{avg} and classification error rate. All the results are calibrated by focal multiclass toolkit[1] using a small set of development data. The first line corresponds to the baseline GLDS system as described in section 2. Results of the proposed method are shown in the second and third line. Generally speaking, improvements can be obtained by involving the weights $\boldsymbol{\alpha}_l$ compared to the baseline. Comparing line 2 and line 3, we can find that LR further improves the performance based on MMI+SVM. As mentioned in section 3.2 and 3.3, a main difference between the two methods is that LR optimizes the parameters simultaneously while MMI+SVM decouples the estimate of $\boldsymbol{\alpha}_l$ and $\boldsymbol{w}_l$ which may lead to a solution that is far from the global optimum.

Table 1. *Comparison of results on LRE07. Close-set condition, 30 seconds test segment*

	C_{avg}(%)	Error Rate(%)
baseline GLDS	6.19	19.5
Weighted GLDS:MMI+SVM	4.35	15.9
Weighted GLDS:LR	**3.45**	**13.8**

6. CONCLUSION

A weighted polynomial expansion method is proposed to distinguish different units in a speech segment. This method is inspired by GMM-SVM system where the super-vector is the core connection between GMM and SVM. We map the polynomial vectors to UBM components frame by frame according to component posteriori. However, the idea of stacking all component means can not be directly used in our case because of the computational issues. Instead of stacking, we fuse the component dependent vectors by involving a set of weights and introduce two discriminative training techniques to estimate the weights and classification boundaries. Experimental results on LRE07 show consistently improvements compared to the baseline GLDS system. The drawback of our approach is computation demanding especially during the extraction of the component dependent vector (6). We will focus mainly on the computational simplification in the future work.

7. ACKNOWLEDGEMENTS

The authors would like to thank Jiaen Liang for a series of useful advice and discussion.

8. REFERENCES

[1] Douglas A. Reynolds, William M. Campbell, Wade Shen, and Elliot Singer, "Automatic language recognition via spectral and token based approaches," in *Springer Handbook of Speech Processing*, Jacob Benesty, M. Mohan Sondhi, and Yiteng (Arden) Huang, Eds., pp. 811–824. Springer Berlin Heidelberg, 2008.

[2] W. M. Campbell, "Generalized linear discriminant sequence kernels for speaker recognition," in *Proc.ICASSP*. IEEE, 2002, pp. 161–164.

[3] W.M. Campbell, J.P. Campbell, D.A. Reynolds, E. Singer, and PA Torres-Carrasquillo, "Support vector machines for speaker and language recognition," *Computer Speech & Language*, vol. 20, no. 2-3, pp. 210–229, 2006.

[4] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using gmm supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, pp. 308–311, 2006.

[5] D.A. Reynolds, T.F. Quatieri, and R.B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[6] P. Matejka, L. Burget, P. Sckwarz, and J. Cernocky, "Brno university of technology system for nist 2005 language recognition evaluation," in *Speaker and Language Recognition Workshop, 2006. IEEE Odyssey 2006: The*. IEEE, 2006, pp. 1–7.

[7] J. Nocedal and S.J. Wright, *Numerical optimization*, vol. 17, Springer verlag, the 2nd edition, 2006.

[8] D.A. van Leeuwen and N. Brummer, "Channel-dependent gmm and multi-class logistic regression models for language recognition," in *Speaker and Language Recognition Workshop, IEEE Odyssey*. IEEE, 2006, pp. 1–8.

[9] F. Castaldo, D. Colibro, E. Dalmasso, P. Laface, and C. Vair, "Compensation of nuisance factors for speaker and language recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 7, pp. 1969–1978, 2007.

[10] "The 2007 NIST Language Recognition Evaluation Plan (LRE07)," 2007, `http://www.itl.nist.gov/iad/mig/tests/lre/2007`.

[11] L.I. Smith, "A tutorial on principal components analysis," *Cornell University, USA*, vol. 52, 2002.

[1] https://sites.google.com/site/nikobrummer/focalmulticlass