UNIFYING PLDA AND POLYNOMIAL KERNEL SVMS

Sibel Yaman, Jason Pelecanos, and Weizhong Zhu

IBM T. J. Watson Research Center, Yorktown Heights, NY, 10598 USA
{syaman, jwpeleca, zhuwe}@us.ibm.com

ABSTRACT

Probabilistic linear discriminant analysis (PLDA) is a generative model to explain between and within class variations. When the underlying latent variables are modelled by standard Gaussian distributions, the PLDA recognition scores can be evaluated as a dot product between a high dimensional PLDA feature vector and a weight vector. A key contribution of this paper is showing that the high dimensional PLDA feature vectors can be equivalently (in a non-strict sense) represented as the second-degree polynomial kernel induced features of the vectors formed by concatenating the two input vectors constituting a trial. This equivalence relationship paves the way for the speaker recognition problem to be viewed as a two-class support vector machine (SVM) training problem where higher degree polynomial kernels can give better discriminative power. To alleviate the large scale SVM training problem, we propose a kernel evaluation trick that greatly simplifies the kernel evaluation operations. In our experiments, a combination of multiple second degree polynomial kernel SVMs performed comparably to a state-of-the-art PLDA system. For the analysed test case, SVMs trained with third degree polynomial kernel reduced the EERs on average by 10% relative to that of the SVMs trained with second degree polynomial kernel.

Index Terms— speaker recognition, large scale SVMs, PLDA

1. INTRODUCTION

Originally proposed in the face recognition domain [1], probabilistic linear discriminant analysis (PLDA) is a generative model widely adopted in speaker verification [2]. It calculates the likelihood that the given observations share the same speaker identity variable. Under these assumptions, the j^{th} recording of the i^{th} speaker is represented as

$$\mathbf{a}_{ij} = \boldsymbol{\mu} + \mathbf{F}\mathbf{h}_i + \mathbf{G}\mathbf{w}_{ij} + \boldsymbol{\epsilon}_{ij} \quad (1)$$

where $\boldsymbol{\mu}$ is the overall mean of the training set i-vectors [3]. The columns of $\mathbf{F}$ and $\mathbf{G}$ represent a basis for the between and within speaker subspaces respectively. The term $\mathbf{h}_i$ is a latent variable that depends only on the speaker identity whereas the latent variable $\mathbf{w}_{ij}$ depends both the speaker identity and on the session. The remaining variability is explained with a zero-mean Gaussian noise term $\boldsymbol{\epsilon}_{ij}$ with a diagonal covariance matrix $\boldsymbol{\Sigma}$. The PLDA model parameters $\{\boldsymbol{\mu}, \mathbf{F}, \mathbf{G}, \boldsymbol{\Sigma}\}$ are estimated via an expectation-maximization (EM) algorithm with a maximum likelihood (ML) training criterion.

As proposed in [4, 5], the use of Gaussian distributions to model the latent variables reduces the PLDA scores to a dot-product between a PLDA feature vector that depends on the enrollment and test vectors, and a weight vector derived from the PLDA model parameters. Unlike generative PLDA, the parameters of this model are estimated via discriminative training techniques such as logistic regression [4] and support vector machines (SVMs) [5].

In this paper we build upon prior work related to discriminatively trained PLDA models [4]. One of the key contributions of our paper is showing the association between PLDA and polynomial kernel SVM approaches, which leads to formulating the speaker recognition problem as a two-class 2^{nd} degree polynomial kernel SVM training problem where each training sample is a concatenation of two i-vectors. Another key contribution is that this work presents the opportunity to significantly improve performance using higher degree polynomial kernels once the large scale SVM training problem is addressed (as we previously showed on a language recognition task [6]).

Unfortunately the scale of the task stands as a challenge, as the resulting SVM problem grows quadratically with the number of the training set samples. Training SVMs with millions of training examples is an active research area [7, 8]. Another key contribution is considerations for training large-scale non-linear kernel SVMs. To alleviate the large scale SVM training problem we further propose a kernel evaluation trick that simplifies kernel evaluations greatly.

Our experiments show that third degree polynomial kernel SVMs reduce the EER by 10% relative to second degree polynomial kernel SVMs. We also found that the performance of a combination of multiple second degree polynomial kernel SVM system was comparable to our state-of-the-art PLDA based baseline system.

This paper is organized as follows: Section 2 presents a description of the related work on generative and discriminative PLDA approaches. Section 3 focuses on relating generative PLDA and second-degree polynomial kernel methods. Section 4 describes considerations for large scale problems. Finally, Section 5 reports our experimental results. Section 6 summarizes our concluding remarks.

2. BACKGROUND AND RELATED WORK

In this section we briefly review the generative [1] and discriminative PLDA [4, 5] approaches. The discriminative approach is based on a reformulation of the PLDA log-likelihood ratio as a dot product between a high dimensional vector and a weight vector.

Let us suppose that we are given a corpus of N recordings with each recording represented as an i-vector [3]. The PLDA model of Equation (1) decomposes each i-vector into a signal component, $\boldsymbol{\mu} + \mathbf{F}\mathbf{h}_i$, that depends on the speaker identity and a noise component, $\mathbf{G}\mathbf{w}_{ij} + \boldsymbol{\epsilon}_{ij}$, that represents the within speaker variability. In testing whether two recordings, $\mathbf{a}$ and $\mathbf{b}$, come from the same

This work was supported in part by Contract No. D11PC20192 DOI/NBC under the RATS program. The views, opinions, findings and recommendations contained in this article are those of the author(s) and should not be interpreted as representing the views or policies, either expressed or implied, of the DOI/NBC.

speaker, PLDA estimates a log-likelihood ratio

$$s_{PLDA} = \log \frac{p(\mathbf{a}, \mathbf{b}|\mathcal{H}_s)}{p(\mathbf{a}, \mathbf{b}|\mathcal{H}_d)}. \quad (2)$$

where $\mathcal{H}_s$ and $\mathcal{H}_d$ represent the same speaker and different speaker hypotheses respectively.

As shown previously in [4, 5], the PLDA score in Equation (2) can then be written as a dot product between a PLDA feature vector, $\mathbf{\Psi}_{PLDA}$, and a weight vector, $\mathbf{w}_{PLDA}$, i.e.,

$$s_{PLDA} =< \mathbf{w}_{PLDA}(\boldsymbol{\mu}, \mathbf{F}, \mathbf{G}, \mathbf{\Sigma}), \mathbf{\Psi}_{PLDA}(\mathbf{a}, \mathbf{b}) > . \quad (3)$$

The weight vector $\mathbf{w}_{PLDA}$ is initialized from the PLDA model parameters and then refined by discriminative training. The PLDA feature vector $\mathbf{\Psi}_{PLDA}$ is given by

$$\mathbf{\Psi}_{PLDA}(\mathbf{a}, \mathbf{b}) = \begin{bmatrix} vec(\mathbf{a}\mathbf{b}^T + \mathbf{b}\mathbf{a}^T) \\ vec(\mathbf{a}\mathbf{a}^T + \mathbf{b}\mathbf{b}^T) \\ \mathbf{a} + \mathbf{b} \\ 1 \end{bmatrix} \quad (4)$$

where $vec(.)$ stands for the vectorization of a matrix by stacking its columns into a vector. The dimensionality of a PLDA feature vector is given by $|\mathbf{\Psi}_{PLDA}(\mathbf{a}, \mathbf{b})| = 2 \cdot D^2 + D + 1$, which quickly grows large with increasing i-vector dimensionality D.

3. UNIFYING PLDA AND POLYNOMIAL KERNELS

In this section we focus our analysis on relating generative PLDA to second-degree polynomial kernel methods. Here we first review support vector machines (SVMs) and feature space representations of input vectors when a polynomial kernel function is used in SVM training.

Suppose that we are given $(\mathbf{x}_1, \ell_1), (\mathbf{x}_2, \ell_2), ..., (\mathbf{x}_M, \ell_M)$, where $\{\mathbf{x}_m\} \in \Re^K$ are K dimensional training samples and $\{\ell_m\} \in \{-1, +1\}$ are their corresponding labels. The training objective of SVMs [9] is a quadratic optimization problem that depends on the data only through feature space dot products, which can be replaced with any non-linear kernel function. A kernel function is evaluated for each pair of training examples as

$$K(\mathbf{x}, \mathbf{y}) =< \mathbf{\Psi}(\mathbf{x}), \mathbf{\Psi}(\mathbf{y}) > \quad (5)$$

where $< \mathbf{\Psi}(\mathbf{x}), \mathbf{\Psi}(\mathbf{y}) >$ represents a dot product in a high dimensional feature space.

3.1. Support Vector Machines with Polynomial Kernels

One of the strengths of SVMs is that the input vectors $\{\mathbf{x}_m\}$ can be *non-linearly* mapped onto vectors in a high dimensional feature space where the training task is formulated as estimating hyperplanes [10]. Through the use of the so-called kernel trick, the mapping is not explicitly needed. Although not needed in general, it is possible to construct feature vectors $\mathbf{\Psi}(\mathbf{x})$ for some kernel functions.

One standard class of kernel functions is the polynomial kernel function with degree d formulated as [11]

$$K_d(\mathbf{x}, \mathbf{y}) = \mathbf{\Psi}_d(\mathbf{x})^T \mathbf{\Psi}_d(\mathbf{y}) = (\mathbf{x}^T \mathbf{y} + 1)^d, \quad (6)$$

The polynomial kernel with $d = 1$ corresponds to a linear kernel. The feature vector for $d = 2$ consists of linear and quadratic terms, and can be represented as [11]

$$\mathbf{\Psi}_{d=2}(\mathbf{x}) = \begin{bmatrix} vec(Upper(\mathbf{x}\mathbf{x}^T)) \\ \mathbf{x} \\ 1 \end{bmatrix} \quad (7)$$

where the operator $Upper()$ extracts the upper triangular (including diagonal) entries of a matrix.

3.2. Association of the PLDA to the Second Degree Polynomial Kernel Induced Features

To show the association between the PLDA and the polynomial kernel induced feature vectors with $d = 2$, we decompose the PLDA feature vector in Equation (4) into its components as

$$\Psi_{PLDA}(\mathbf{a}, \mathbf{b}) = \begin{bmatrix} \{a_i b_j + b_i a_j\} \\ \{a_i a_j + b_i b_j\} \\ \{a_i + b_i\} \end{bmatrix}_{i,j=1}^{D} \quad (8)$$

Notice that the individual feature dimensions are conjunctions of the terms $a_i b_j$, $a_i a_j$, $b_i b_j$, a_i, and b_i for $i, j = 1, 2, ..., D$. The key in relating the PLDA model to the polynomial kernel induced feature vector lies in recognizing that the vector $\Psi_{PLDA}(\mathbf{a}, \mathbf{b})$ can be equivalently represented as

$$\mathbf{\Psi}(\mathbf{a}, \mathbf{b}) \equiv \begin{bmatrix} vec(Upper(\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}^T)) \\ \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} \\ 1 \end{bmatrix} \quad (9)$$

Equations (7) and (9) reveal that $\Psi(\mathbf{a}, \mathbf{b})$ is the second degree polynomial kernel induced feature vector of the input vector $\mathbf{x} = \begin{bmatrix} \mathbf{a}^T & \mathbf{b}^T \end{bmatrix}^T$.

The 2^{nd} degree polynomial kernel induced feature vector and the PLDA feature vector share the terms $vec(Upper(\mathbf{x}\mathbf{x}^T))$. The only difference is that the former has terms of the form $\mathbf{a}$ and $\mathbf{b}$ whereas the latter has $\mathbf{a} + \mathbf{b}$. Therefore the two are equivalent in a non-strict sense, i.e., except for a rather insignificant difference.

3.3. Use of High Degree Polynomial Kernels

Having shown the equivalence of the PLDA feature vector and the 2^{nd} degree polynomial kernel induced feature vector, it is natural to ask whether the classifier performance would benefit from using higher degree polynomial kernels. In our earlier work in a language recognition task [6], we observed a relative reduction of more than 20% in EER when the polynomial degree was raised from 2 to 3. We furthermore obtained the lowest EER when the degree of the polynomial kernel was increased to $d = 5$ (with 37% relative improvement in EER over $d = 2$). While this is a challenging problem for speaker recognition, we explore the opportunity for using higher degree polynomial kernels for speaker recognition.

The 2^{nd} degree polynomial kernel induced feature vectors are composed of monomials of degree 0, 1, and 2. When the degree of the polynomial kernel is raised to $d = 3$, the kernel induced feature vectors also includes monomials of degree 3, i.e., terms in the form of $a_i a_j a_k, a_i a_j b_k, a_i b_j b_k$, and $b_i b_j b_k$. Increasing the model complexity this way may yield performance gains provided sufficient training data is available.

4. CONSIDERATIONS FOR LARGE SCALE TASKS

The discriminative PLDA approaches of [4, 5] compute high dimensional feature vectors for a total of N^2 i-vector pairs. The dimensionality of these feature vectors grow quadratically with the dimension of the i-vectors. However, the technique proposed in this paper

operates in the low-dimensional space: The equivalence relationship of the previous section makes it possible to concatenate the two i-vectors constituting a trial and train SVMs using these.

4.1. A Kernel Evaluation Simplification Trick

Although there are techniques to train linear SVMs using millions of samples within a few seconds, it is highly challenging to develop efficient nonlinear SVM training algorithms that can handle millions of input samples. We propose a *kernel evaluation trick* that reduces the computations in kernel evaluations to two look-ups, two additions, and a power operation. It thereby avoids the need to explicitly generate concatenated vectors from i-vectors.

The kernel $K(\mathbf{x},\mathbf{y}) = \mathbf{\Psi}^T(\mathbf{x})\mathbf{\Psi}(\mathbf{y})$ for two trials $\mathbf{x} = \begin{bmatrix}\mathbf{a}_i^T & \mathbf{b}_j^T\end{bmatrix}^T$ and $\mathbf{y} = \begin{bmatrix}\mathbf{a}_k^T & \mathbf{b}_m^T\end{bmatrix}^T$ can be evaluated as

$$\begin{aligned} K(\mathbf{x},\mathbf{y}) &= (<\mathbf{a}_i,\mathbf{a}_k> + <\mathbf{b}_j,\mathbf{b}_m> +1)^d \\ &= (\mathbf{Q}_{ij} + \mathbf{Q}_{km} + 1)^d \end{aligned} \quad (10)$$

where $\mathbf{Q}$ is the N-by-N i-vector Gram matrix and where the subscript ij refers to the element of $\mathbf{Q}$ in the i^{th} row and the j^{th} column. With this simplification the proposed technique can be made scalable to problems with millions of trials.

4.2. Further Considerations

The kernel evaluation simplification trick of the previous section alleviates the SVM training problem with millions of trials. However, current speaker recognition systems with tens of thousands of input recordings require training with hundreds of millions of trials.

One common approach adopted in large scale tasks is to train SVMs with manageable training set sizes and to combine them in a second phase. As we found out in our experiments, although the average performance of the individual SVMs is not outstanding, a combination of them can achieve a performance comparable to state-of-the-art systems. As our experiments in Section 5 show, an equal-weight combination of the top $B\%$ of the SVMs based on a held-out set performed comparably to a generative PLDA system.

5. EXPERIMENTS

The task we focus on in our experiments is the speaker recognition track of the DARPA sponsored Robust Automatic Transcription of Speech (RATS) program. The training data used in the experiments consist of 10,143 recordings from 1,085 speakers from the first three data releases. The evaluation dataset consists of 5,593 recordings from 68 speakers. Each audio recording is 120 seconds in duration and is obtained by passing a clean source recording through one of the eight highly degraded and/or noisy high frequency communication channels. The evaluation set consists of 6,028 trials where each evaluation trial scores a given test speech segment against enrollment models consisting of 6 speech segments.

We conducted speaker recognition experiments with a standard MFCC-based system. We generated 800 dimensional i-vectors and performed an LDA transform on them to reduce the dimensionality to 500. We found that within-class covariance normalization and unit length normalization steps improved the PLDA performance but they were detrimental to SVM performance. [1]

[1]Through personal communication, we were informed that the relative gains reported in [4, 5] are not significant over generative PLDA when the i-vectors are unit length normalized before applying generative PLDA.

Table 1. The performance of the generative PLDA systems.

	min DCF(x1000)	EER(%)
Generative MultiPLDA	25.0	4.7
Generative AvgPLDA	30.2	4.8

We built two generative PLDA systems as reference baselines. The first one, which we refer to as multi-session enrollment PLDA (denoted as MultiPLDA), scores each test recording against an enrollment model trained from 6 sessions. The second one, which we refer to as average PLDA (denoted as AvgPLDA), scores the given test recording against each of the 6 enrollment recordings independently and averages the resulting 6 pairwise scores. We report the baseline performances in Table 1.

5.1. The Effect of the Degree of the Polynomial Kernel

We trained SVMs with a varying number of randomly selected i-vectors. We analysed trends in the performance (i) as we increased the number of training set i-vectors and (ii) as we increased the degree of the polynomial kernel.

The performance statistics in terms of EER (%) are shown using box plots in Figure 1. On each box, the central mark is the median, and the edges of the box are the 25^{th} and 75^{th} percentiles. The tick marks joined by dotted lines represent the lower and upper bounds on the performance of the set of systems tested.

We first note that, for all degrees of polynomial kernels, the performance improves as the number of randomly selected i-vectors increases. Unfortunately, doubling the number of training set i-vectors beyond $2,000$ resulted in training times of a week or more and therefore we do not consider these as useful. We expect the improving performance trend to continue once efficient nonlinear kernel SVM algorithms become available.

Secondly we note that, for all training set sizes, the performance improves when the degree of the polynomial kernel is increased from $d = 2$ to $d = 3$. In particular, when $N = 2,000$ i-vectors are used in SVM training, the median EER dropped from 7.5% to 6.7% (a relative EER improvement of 10%).

We observe from Figure 1 that the performance may degrade when the polynomial kernel degree is increased to $d = 4$ with small training set sizes. However, when the training set is sufficiently large (e.g., $N = 2000$) we start to see performance improvements. Based on the trends with $N = 2000$ we expect higher degree polynomial kernel SVMs trained with sufficiently large training sets to provide

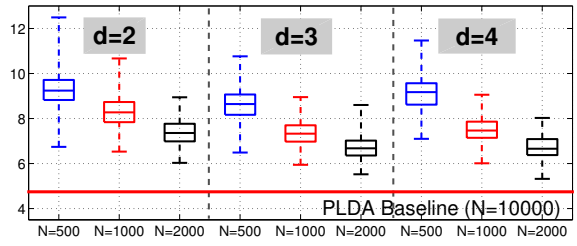

Fig. 1. The effect of the number of training set i-vectors and the degree of the polynomial kernel on EER (%)

Table 2. Performance of linear score combination of SVMs trained with randomly selected i-vectors.

	Equal Weight		Top 30%	
	min DCF(x1000)	EER(%)	min DCF(x1000)	EER(%)
$d = 2, N = 1,000$	27.8	5.5	26.8	5.1
$d = 3, N = 1,000$	30.5	5.6	30.2	5.7
$d = 4, N = 1,000$	34.7	5.6	34.5	5.6
$d = 2, N = 2,000$	26.5	5.0	25.8	5.1
$d = 3, N = 2,000$	28.2	5.6	27.8	5.3
$d = 4, N = 2,000$	32.4	5.6	32.5	5.4

greater relative improvements.

The trends in Figure 1 highlight opportunities to significantly improve performance using higher degree polynomial kernels once the issue of scale is addressed. In the LID task [6] we found that there was a significant performance improvement when the degree of the polynomial kernel was changed from $d = 2$ to $d = 3$ to $d = 4$. We furthermore found that these improvements become more significant when the amount of training size was increased.

5.2. Combining SVMs

We investigated methods to combine multiple SVMs trained with randomly selected i-vectors. We found that linear score combination provided significant gains. In our experiments we explored in the combination performance (i) as we increased the number of randomly selected i-vectors and (ii) as we increased the degree of the polynomial kernel. We investigated two kinds of score combination: Combining all SVMs with equal weight and combining only top performing SVMs, which are selected based on their performance on a separate held-out set.

We report the results of our score combination experiments in Table 2. We first observe that the proposed SVM training technique performed comparably with our reference baselines with both of the combination strategies. A comparison of the results in the $d = 2, N = 1,000$ and $d = 2, N = 2,000$ rows of Table 2 indicate that the combination performance improves if the individual SVMs see more training i-vectors. A comparison of the results reported in the "Equal Weight" columns of Table 2 with their counterparts in the "Top 30%" columns shows that using best SVMs slightly improves the combination performance.

The results in the $d = 3$ and $d = 4$ rows of Table 2 also indicate significant improvement over individual SVM performances. However, we observe that linear score combination does not work well for higher degree polynomial kernel SVMs. Although the individual SVMs performances with $d = 3$ are better than those with $d = 2$, linear score combination gives better results when $d = 2$.

6. CONCLUSIONS AND FUTURE WORK

In this paper we showed that the PLDA feature vectors can be equivalently (in a non-strict sense) represented as second degree polynomial kernel induced features of the vectors formed by concatenating the two input vectors constituting a trial. This reduced the speaker recognition problem to a two-class SVM training problem with improved discriminative power using higher degree polynomial kernels. Furthermore, a kernel evaluation trick is proposed to avoid forming N^2 concatenated vectors from N input vectors in large scale tasks.

Based on the trends that we observed in our experiments, we hypothesize that higher degree polynomial kernel SVMs trained with sufficiently large training sets can provide greater relative improvements. Our ongoing research is focused on an investigation of principled SVM combination approaches to maximize the diversity as well as the generalization capabilities of the combined SVM system.

7. REFERENCES

[1] S. J. D. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *International Conference on Computer Vision*, 2007.

[2] P. Matejka et al., "Full-covariance UBM and heavy-tailed PLDA in i-vector speaker verification," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2011.

[3] N. Dehak et al., "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, 2010.

[4] L. Burget et al., "Discriminatively trained probabilistic linear discriminant analysis for speaker verification," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2011.

[5] S. Cumani et al., "Fast discriminative speaker verification in the i-vector space," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2011.

[6] S. Yaman, J. Pelecanos, and M. Omar, "On the use of non-linear polynomial kernel svms in language recognition," in *IEEE Interspeech*, 2012.

[7] Cho jui Hsieh, Kai wei Chang, Chih jen Lin, and S. Sathiya Keerthi, "A dual coordinate descent method for large-scale linear svm," in *International Conference on Machine Learning*, 2008.

[8] Thorsten Joachims, "Training linear svms in linear time," in *ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006.

[9] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, 1998.

[10] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, 1995.

[11] J. Shawe Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, 2004.