NONLINEAR KERNEL NUISANCE ATTRIBUTE PROJECTION FOR SPEAKER VERIFICATION

Xianyu Zhao[1], Yuan Dong[1,2], Hao Yang[2], Jian Zhao[2], Liang Lu[2], Haila Wang[1]

[1]France Telecom R&D Center (Beijing), Beijing, 100080, P.R.China
[2]Beijing University of Posts and Telecommunications, Beijing, 100876, P.R.China
{xianyu.zhao, yuan.dong, haila.wang}@orange-ftgroup.com

ABSTRACT

Nuisance attribute projection (NAP) was successfully applied in SVM-based speaker verification systems to improve performance by doing projection to remove dimensions from the SVM feature space that cause unwanted variability in the kernel. Previous studies of NAP were focused mainly on linear and generalized linear kernel SVMs. In this paper, NAP in nonlinear kernel SVMs, e.g. polynomial or Gaussian kernels, are investigated. Instead of doing explicit feature expansion and projection in high-dimension feature space, kernel principal component analysis is employed to find nuisance dimensions; and, NAP is carried out implicitly by incorporating it into some compensated kernel functions. Experimental results on the 2006 NIST SRE corpus indicate the effectiveness of such nonlinear kernel NAP. Compared with linear NAP, nonlinear NAP with Gaussian kernel obtained about 11% relative improvement in Equal Error Rate (EER).

Index Terms— supporting vector machines, kernel principal component analysis, nuisance attribute projection, speaker location, speaker recognition.

1. INTRODUCTION

In recent years, support vector machines (SVMs) have become one of the most important and widely used classification techniques in the field of speaker recognition. In [1], Nuisance Attribute Projection (NAP) was developed to do projection to remove dimensions from the SVM feature space that are irrelevant to the classification problem. Through NAP, intersession or channel variability (due to microphones, acoustic environments, etc.) can be reduced to improve speaker verification performance [1]–[3].

Former studies of NAP were mainly focused on linear and generalized linear kernels, in which nuisance dimensions are found and projected out directly in the SVM feature space. In this paper, NAP in nonlinear kernels, e.g. polynomial or Gaussian kernels, are studied. For these nonlinear kernels, the feature space is derived by some nonlinear transformation of input variables, and the dimension of transformed feature space can be very high or even infinite, e.g. for Gaussian kernel. In these cases, finding nuisance attributes and doing NAP directly in the feature space would incur possibly expensive representational and computational cost. To solve these problems, in this paper nuisance attributes in nonlinear kernels are found out through kernel principal component analysis (kernel PCA [4]); and, NAP projection is carried out implicitly by incorporating it into some compensated kernel functions.

In this study, the input variables to SVMs are derived through anchor modeling [3] in which utterances are scored against a set of reference speaker models to determine its corresponding location vector in the space of reference speakers. The derived representation is called speaker location vector, and the SVMs used to classify these location vectors are referred as "location SVMs" in this paper.

This paper is organized as follows. In Section 2, we describe briefly the concepts of anchor models and speaker location. In Section 3, we discuss NAP and present how to do NAP in nonlinear kernels through kernel PCA. In Section 4, we report experimental results on the 2006 NIST speaker recognition evaluation (SRE) corpus. We end with conclusions and future work in Section 5.

2. ANCHOR MODELS AND SPEAKER LOCATION

In this study, speaker location vector derived from anchor modeling is fed into SVMs as input variables, which is represented by the following vector, $\mathbf{v}$, [3], [5],

$$\mathbf{v} = \left[l\left(\mathbf{x}\middle|\bar{\lambda}_1\right) \quad l\left(\mathbf{x}\middle|\bar{\lambda}_2\right) \quad \cdots \quad l\left(\mathbf{x}\middle|\bar{\lambda}_E\right) \right]^T, \qquad (1)$$

where $(\cdot)^T$ stands for vector transpose, $\left\{\bar{\lambda}_i; i = 1,2,\cdots,E\right\}$ is a set of well trained reference speaker models (called *anchor models*), which are modeled as Gaussian Mixture Models (GMMs) and MAP adapted from a Universal Background Model (UBM) [6]; $l\left(\mathbf{x}\middle|\bar{\lambda}_i\right)$ is the normalized log-likelihood of the speaker utterance data $\mathbf{x}$ (of T

acoustic feature vectors) for the i-th anchor model, $\bar{\lambda}_i$, relative to the UBM, $\bar{\lambda}_{UBM}$, i.e.,

$$l\left(\mathbf{x}\middle|\bar{\lambda}_i\right)=\frac{1}{T}\log\left(p\left(\mathbf{x}\middle|\bar{\lambda}_i\right)\middle/p\left(\mathbf{x}\middle|\bar{\lambda}_{UBM}\right)\right). \tag{2}$$

3. NUISANCE ATTRIBUTE PROJECTION IN SUPPORT VECTOR MACHINES

In the standard formulation, an SVM, $f\left(\mathbf{v}\right)$, is given by

$$\begin{aligned} f\left(\mathbf{v}\right)&=\sum_{i=1}^{M}\alpha_i k\left(\mathbf{v},\bar{\mathbf{v}}_i\right)+b \\ &=\sum_{i=1}^{M}\alpha_i\left\langle\Phi\left(\mathbf{v}\right),\Phi\left(\bar{\mathbf{v}}_i\right)\right\rangle+b, \end{aligned} \tag{3}$$

where $k\left(\mathbf{v}_1,\mathbf{v}_2\right)$ is a kernel function and $\Phi\left(\mathbf{v}\right)$ is a feature transformation function. The relationship between the feature transformation Φ and the kernel function k is that

$$\begin{aligned} k\left(\mathbf{v}_1,\mathbf{v}_2\right)&=\left\langle\Phi\left(\mathbf{v}_1\right),\Phi\left(\mathbf{v}_2\right)\right\rangle \\ &=\Phi\left(\mathbf{v}_1\right)^T\cdot\Phi\left(\mathbf{v}_2\right), \end{aligned} \tag{4}$$

i.e., the inner product in the transformed feature space can be realized by the kernel function over input variables. This property facilitates SVMs to do implicit feature transformation with kernel function.

The b and $\left\{\alpha_i,\bar{\mathbf{v}}_i;i=1,\cdots,M\right\}$ are obtained through a training process that maximizes the margin between two classes (positive vs. negative). SVMTorch is used as SVM trainer in our experiments [7].

In the application of SVMs for speaker verification by location in the space of reference speakers, an SVM is trained for each target speaker using the location vectors of the speaker's enrollment utterances as positive examples, and the location vectors of utterances in some development set as negative examples.

3.1. Nuisance Attribute Projection (NAP)

In our application with NAP for speaker verification, it aims to find a projection matrix, $P=I-U_m U_m^T$, which is able to filter out nuisance attributes (e.g. session/channel variability) in the feature vectors and to pull together features from the same speaker in the projected subspace. The matrix U_m used in NAP projection is found by minimizing the average cross-session distances [1]:

$$Q=\sum_{i,j}M_{i,j}\left\|P\cdot\Phi\left(\mathbf{v}_i^d\right)-P\cdot\Phi\left(\mathbf{v}_j^d\right)\right\|^2, \tag{5}$$

where $\left\{\Phi\left(\mathbf{v}_i^d\right);i=1,\cdots,n\right\}$ are n feature vectors derived from the development set, and M is a weight matrix whose elements, $M_{i,j}$, evaluate the cost of disparity between projected feature vectors. In this study, M is set to be:

$$M_{i,j}=\begin{cases}1, & \text{if } \mathbf{v}_i^d \text{ and } \mathbf{v}_j^d \text{, are from the same speaker}\\ 0, & \text{otherwise}\end{cases}. \tag{6}$$

As shown in [1], the objective function in (5) is minimized by m eigenvectors with largest eigenvalues of the symmetric eigenvalue problem:

$$AZ\left(M\right)A^T U_m=U_m\Lambda, \tag{7}$$

where the matrix $Z\left(M\right)=diag\left(M\cdot\mathbf{1}\right)-M$, $\mathbf{1}$ is the column vector of all ones, $diag\left(\cdot\right)$ is an operator of forming a diagonal matrix from a vector, and $A=\left[\Phi\left(\mathbf{v}_1^d\right),\Phi\left(\mathbf{v}_2^d\right),\cdots,\Phi\left(\mathbf{v}_n^d\right)\right]$.

3.2. Nonlinear Kernel NAP

For nonlinear kernels, to solve (7) directly in the transformed feature space would involve matrix factorization in high-dimensional space (e.g. for high order polynomial kernels which take into account the correlation among input variables) or integral equation in infinite dimension space (e.g. for Gaussian kernel). Hence, instead of doing eigenvalue analysis in the high (or infinite) dimensional transformed feature space directly, kernel PCA is employed as follows. We begin by factoring $Z\left(M\right)$ as,

$$Z\left(M\right)=Z^{1/2}Z^{1/2}, \tag{8}$$

where $Z^{1/2}=\left(diag\left(M\cdot\mathbf{1}\right)\right)^{1/2}-\left(diag\left(M\cdot\mathbf{1}\right)\right)^{-1/2}M$. Then, equation (7) becomes

$$AZ^{1/2}Z^{1/2}A^T U_m=U_m\Lambda. \tag{9}$$

It can be observed that U_m lie in the span of the columns of $AZ^{1/2}$; i.e., there exists matrix $n\times m$ Y_m such that

$$U_m=AZ^{1/2}Y_m. \tag{10}$$

After substituting (10) into (9), we get

$$AZ^{1/2}Z^{1/2}A^T AZ^{1/2}Y_m=AZ^{1/2}Y_m\Lambda. \tag{11}$$

Multiplying both sides with $Z^{1/2}A^T$, we can deduce from (11) that Y_m can be found by m eigenvectors with largest eigenvalues of the symmetric matrix, $Z^{1/2}A^T AZ^{1/2}$, i.e,

$$Z^{1/2}A^T AZ^{1/2}Y_m=Y_m\Lambda. \tag{12}$$

For calculation of the above symmetric matrix, $Z^{1/2}A^T AZ^{1/2}$, the elements in $A^T A$ are the inner products of the feature vectors in the development corpus and can be calculated through the kernel function, i.e.,

$$\begin{aligned} \left(A^T A\right)_{i,j}&=\left\langle\Phi\left(\mathbf{v}_i\right),\Phi\left(\mathbf{v}_j\right)\right\rangle \\ &=k\left(\mathbf{v}_i,\mathbf{v}_j\right), \end{aligned} \tag{13}$$

where $\left(A^T A\right)_{i,j}$ stands for the entry in row i and column j of $A^T A$.

We normalize columns in Y_m to ensure that corresponding eigenvectors in U_m are orthonormal in the feature space. Combining (10) and (12), this requirement translates into the following normalization condition,

$$Y_m^T Y_m \Lambda = I. \quad (14)$$

With kernel PCA, the eigenvalue analysis problem in the high-dimensional feature space is reduced to the eigenvalue problem of $Z^{1/2} A^T A Z^{1/2}$ whose size is n, i.e. the number of features in the development set; and, calculation can be done through kernel functions over input variables without resort to explicit feature transformation in the high dimensional feature space.

After obtaining the nuisance dimensions, the original nonlinear kernel function is compensated by incorporating NAP as

$$\begin{aligned} k_{NAP}(\mathbf{v}_1, \mathbf{v}_2) &= \langle P \cdot \Phi(\mathbf{v}_1), P \cdot \Phi(\mathbf{v}_2) \rangle \\ &= \langle \Phi(\mathbf{v}_1), \Phi(\mathbf{v}_2) \rangle - \langle U_m^T \cdot \Phi(\mathbf{v}_1), U_m^T \cdot \Phi(\mathbf{v}_2) \rangle \\ &= k(\mathbf{v}_1, \mathbf{v}_2) - \langle U_m^T \cdot \Phi(\mathbf{v}_1), U_m^T \cdot \Phi(\mathbf{v}_2) \rangle. \end{aligned} \quad (15)$$

Substituting (10) into (15), we get that

$$k_{NAP}(\mathbf{v}_1, \mathbf{v}_2) = k(\mathbf{v}_1, \mathbf{v}_2) - \tilde{\mathbf{v}}_1^T \cdot Z^{1/2} Y_m Y_m^T Z^{1/2} \cdot \tilde{\mathbf{v}}_2, \quad (16)$$

where $\tilde{\mathbf{v}}_i$ can be calculated through kernel function as

$$\begin{aligned} \tilde{\mathbf{v}}_i^T &= \Phi(\mathbf{v}_i)^T \cdot A \\ &= \Phi(\mathbf{v}_i)^T \cdot \left[\Phi(\mathbf{v}_1^d), \Phi(\mathbf{v}_2^d), \cdots, \Phi(\mathbf{v}_n^d) \right] \\ &= \left[k(\mathbf{v}_i, \mathbf{v}_1^d), k(\mathbf{v}_i, \mathbf{v}_2^d), \cdots, k(\mathbf{v}_i, \mathbf{v}_n^d) \right]. \end{aligned} \quad (17)$$

Again, we find that the NAP can be carried out implicitly by incorporating it into some compensated kernel functions without projecting feature vectors in the high-dimensional feature space directly. We note that a similar idea was also explored for processing some high-dimensional features efficiently with NAP in high-level speaker recognition [9].

4. EXPERIMENTAL RESULTS

In this section, we report speaker verification experiments by location SVMs with linear and nonlinear kernel NAP. Section 4.1 presents some general experiment setup information about the task, corpora, features and kernel configuration. The results of these experiments are discussed in Section 4.2.

4.1. Experiment Setup

Speaker verification experiments were conducted on the 2006 NIST SRE corpus [8]. We focused on the single-side 1 conversation train, single-side 1 conversation test task. This task involves 3,612 true trials and 47,836 false trials. Enrollment and testing utterances contain about 2 minutes of pure speech after some voice activity detection.

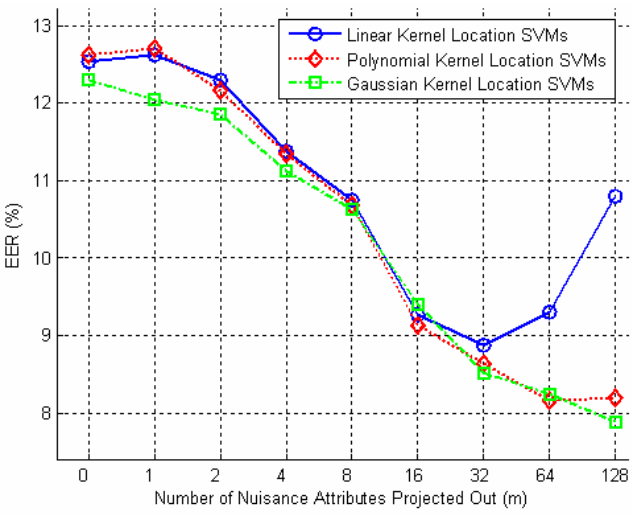

Fig.1 EER vs. number of nuisance attributes projected out in linear, polynomial and Gaussian kernels

Totally 500 reference speakers (230 male and 270 female speakers) are selected from Switchboard I and II corpora. They are used as anchor models in our experiments. A subset of the 2004 NIST SRE corpus is used as the development set. Location vectors of utterances in the development set are calculated as negative examples in SVM training; there are a total of 1790 background location vectors from 321 speakers. These background location vectors are also used as development set for NAP. For the cepstral features used for anchor modeling, PLP is calculated every 10 ms using a 25ms Hamming window. HLDA, RASTA, feature mapping and histogram equalization (HEQ) are applied to improve channel/noise robustness of feature and to facilitate GMM modeling [3].

Besides linear kernel, two nonlinear kernels are investigated:

- ✓ In the first case, polynomial kernel is used, i.e.,

$$k(\mathbf{v}_1, \mathbf{v}_2) = (s \cdot \mathbf{v}_1^T \mathbf{v}_2 + r)^d. \quad (18)$$

  In the following experiments, s, r and d are set to be 1, 1 and 2 respectively.

- ✓ In the second case, Gaussian kernel is chosen, i.e.,

$$k(\mathbf{v}_1, \mathbf{v}_2) = \exp\left(-\|\mathbf{v}_1 - \mathbf{v}_2\|^2 / \sigma^2\right). \quad (19)$$

  In the following experiments, σ is set to be 2.

4.2. Results

In Fig.1 and Fig.2, we summarize the Equal Error Rate (EER) and NIST Detection Cost Function (DCF) results of location SVMs with linear, polynomial and Gaussian kernels respectively. The baseline systems for each kernel configuration correspond to those without NAP, i.e., the number of nuisance attributes projected out is zero. From these figures, we can see that nonlinear kernel NAP could filter out some nuisance attributes in nonlinear kernels to

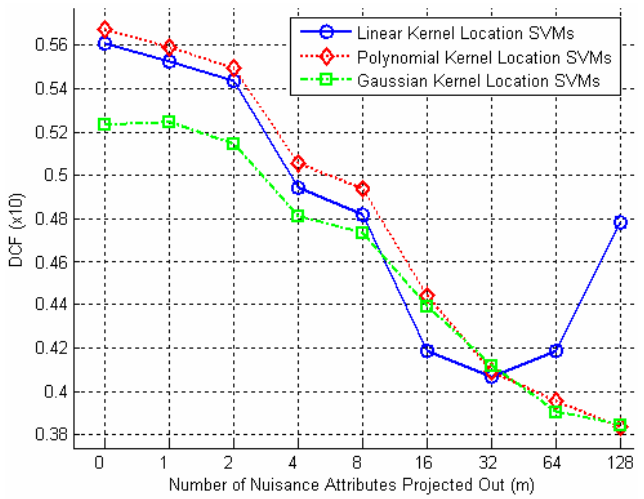

Fig.2 DCF vs. number of nuisance attributes projected out in linear, polynomial and Gaussian kernels

significantly improve speaker verification performance as well as conventional NAP for linear/generalized linear kernels. These figures also show that if too many dimensions are identified as nuisance dimensions and projected out, the discriminative capability of resultant feature would be reduced which would degrade speaker verification performance. In our experiments, the optimal number of nuisance attributes for two nonlinear kernels is 128 while it is 32 for the linear kernel. Under their respective optimal settings, the nonlinear NAP with Gaussian kernel obtained about 11% relative improvement in EER over the linear NAP.

In the second set of experiments, we compared doing nonlinear kernel NAP directly against a two-stage approach to applying NAP in nonlinear kernels. In the two-stage approach [10], NAP was carried out linearly over the original input variables and the NAP projected variables were fed into nonlinear kernel SVMs for classification purpose. The EER and DCF results were summarized in Table 1. We can see that although the two-stage way could improve verification performance over the baseline nonlinear kernel SVMs, the proposed direct nonlinear kernel NAP is more effective by closely matching the nuisance attributes projected out with the underlying nonlinear kernel. We also conducted experiments, in which linear NAP over input variables and nonlinear kernel NAP were cascaded, but did not see further improvements over using nonlinear kernel NAP alone.

5. CONCLUSION

In this paper, nuisance attribute projection for general nonlinear kernel SVMs is presented. Through kernel PCA, the nuisance dimensions can be identified without resort to explicit feature transformation; and, the NAP projection can also be done implicitly through incorporating it into the compensated kernel functions instead of doing high-dimensional feature projection explicitly. The complexity of the proposed method depends only on the size of development data. Speaker verification experiments on the 2006 NIST SRE corpus demonstrate that such kind of nonlinear NAP can filter out session or channel variability in nonlinear kernels (e.g., polynomial or Gaussian) and significantly improve verification performance. Comparisons of linear and nonlinear NAP with more extensive data and in other tasks are planned in future work.

Table 1. Comparison of direct nonlinear kernel NAP against combining linear NAP and nonlinear kernels in a two-stage way

Kernel Type	EER(%)	DCF(x10)
Polynomial Kernel	12.62	0.5671
Two stage: Linera NAP (m = 32) and Polynomial Kernel	8.78	0.4047
Polynomial Kernel NAP (m = 128)	8.19	0.3840
Gaussian Kernel	12.29	0.5235
Two stage: Linera NAP (m = 32) and Gaussian Kernel	8.61	0.3970
Gaussian Kernel NAP (m = 128)	7.89	0.3851

6. REFERENCES

[1] A. Solomonoff, W. M. Campbell and I. Boardman, "Advances in channel compensation for SVM speaker recognition," in *Proc. ICASSP'2005*, 2005.

[2] S. Kajarekar and A. Stolcke, "NAP and WCCN: comparison of approaches using MLLR-SVM speaker verification system," in *Proc. ICASSP'2007*, 2007.

[3] X. Zhao, Y. Dong, H. Yang, J, Zhao and H. Wang, "SVM-based speaker verification by location in the space of reference speakers," in *Proc. ICASSP'2007*, 2007.

[4] B. Scholkopf, A. Smola and K.-R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299-1319, 1998.

[5] D. Sturim, D. Reynolds, E. Singer, and J. P. Campbell, "Speaker indexing in large audio database using anchor models," in *Proc. ICASSP'2001*, pp. 429-432, 2001.

[6] D. Reynolds, T. Quatieri and R. Dunn, "Speaker verification using adapted Gaussian Mixture Models," *Digital Signal Processing*, vol.10, pp. 19-41, 2000.

[7] R. Collobert, S. Bengio, "SVMTorch: Support vector machines for large–scale regression problems," *Journal of Machine Learning Research*, vol. 1, pp. 143-160, 2001.

[8] "The NIST 2006 speaker recognition evaluation plan," http://www.nist.gov/speech/tests/spk/spk/2006/ .

[9] W. M. Campbell, "Compensating for Mismatch in High-Level Speaker Recognition," in Proc. IEEE Odyssey 2006, June 2006.

[10] R. Dehak, N. Dehak, P. Kenny and P. Dumouchel, "Linear and nonlinear kernel GMM supervector machines for speaker verification," in *Proc. INTERSPEECH'2007*, 2007.