Comparison of Input and Feature Space Nonlinear Kernel Nuisance Attribute Projections for Speaker Verification

Xianyu Zhao [1], *Yuan Dong* [1, 2], *Jian Zhao* [2], *Liang Lu* [2], *Jiqing Liu* [2], *Haila Wang* [1]

[1] France Telecom/Orange Labs, Beijing, 100080, P. R. China
{xianyu.zhao, yuan.dong, haila.wang}@orange-ftgroup.com
[2] Beijing University of Posts and Telecommunications, Beijing, 100876, P. R. China
yuandong@bupt.edu.cn {michaeljianzhao, luliang07, robertbupt}@gmail.com

Abstract

Nuisance attribute projection (NAP) was an effective method to reduce session variability in SVM-based speaker verification systems. As the expanded feature space of nonlinear kernels is usually high or infinite dimensional, it is difficult to find nuisance directions via conventional eigenvalue analysis and to do projection directly in the feature space. In this paper, two different approaches to nonlinear kernel NAP are investigated and compared. In one way, NAP projection is formulated in the expanded feature space and kernel PCA is employed to do kernel eigenvalue analysis. In the second approach, a gradient descent algorithm is proposed to find out projection over input variables. Experimental results on the 2006 NIST SRE corpus show that both kinds of NAP can reduce unwanted variability in nonlinear kernels to improve verification performance; and NAP performed in expanded feature space using kernel PCA obtains slightly better performance than NAP over input variables.

Index Terms: speaker verification, session variability modeling, support vector machines, nuisance attribute projection

1. Introduction

Mismatch between training and testing conditions caused by intersession variability (due to microphones, acoustic environments, etc.) is one of the most critical factors which affect the performance of speaker verification systems. A number of techniques have been proposed recently to compensate or filter out session variability to improve speaker verification performance [1] – [6]. Among them, nuisance attribute projection (NAP) was shown to be an effective method to reduce session variability in the discriminative framework of support vector machines (SVMs) by doing projections to remove directions that cause unwanted variability in the underlying kernel of SVMs [1].

NAP was firstly developed for linear or generalized linear kernels [5] [7]. In this case, nuisance directions were identified via an eigenvalue problem in the SVM feature space (which has the same dimensionality as the input space in the linear case) and the NAP projection can be carried out over the feature vectors directly. For general nonlinear kernels, e.g. polynomial or Gaussian kernels, the feature space of SVM is derived by some nonlinear transformation over input variables, and the resultant dimension of transformed feature space might be very high, e.g. for higher order polynomial kernels, or be infinite, e.g. for Gaussian kernels. In this case, it is difficult to carry out conventional eigenvalue analysis in the transformed feature space. In [8], NAP was extended to general nonlinear kernels by doing kernel principal component analysis (kernel PCA) which makes it possible to formulate NAP in the high dimension SVM feature space without resort to explicit feature expansion. Such idea of using kernel PCA to handle high dimensional feature space effectively was also investigated in [9]. As kernel PCA relies on the storage of whole development set for representing nuisance directions and doing projection, its complexity depends on the size of the development set. This might raise storage and computational problems in cases involving large development sets.

In this paper, a different approach to NAP in general nonlinear kernels is proposed. In this method, NAP projection is formulated over the input variables, or in the input space of SVM instead of the high dimensional expanded feature space. And a gradient descent algorithm is developed to solve the nonlinear optimization problem which aims to reduce the session variability in the underlying nonlinear kernels. Advantages of this formulation lies in that the dimension of the input space is usually much lower than the feature space and the development set is no longer needed for future NAP projection after deriving the projection matrix.

These two methods for nonlinear kernel NAP are compared on the 2006 NIST Speaker Recognition Evaluation (NIST SRE) corpus [10]. Experimental results indicate that both can reduce unwanted variability in nonlinear kernels and improve speaker verification performance. The formulation of NAP in the expanded feature space using kernel PCA can extract more nuisance attributes and obtain slightly better performance than that in the input space.

This paper is organized as follows. In Section 2, we describe briefly the discriminative framework of SVMs and its application to speaker verification. In Section 3, we present two formulations of nonlinear kernel NAP. In Section 4, we report some experimental results on the 2006 NIST SRE corpus. And some conclusions and future work are given in Section 5.

2. Support vector machines and its application to speaker verification

In the standard formulation, an SVM, $f(\mathbf{v})$, is given by [11],

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

where $k(\bullet,\bullet)$ is a kernel function and $\Phi(\bullet)$ is a feature transformation or expansion function from the input space (where $\mathbf{v}$ lives) to a feature space (where $\Phi(\mathbf{v})$ lives). The

inner product in the feature space can be evaluated through corresponding kernel function over input variables.

The b and $\{\alpha_i, \bar{\mathbf{v}}_i; i=1,\cdots,M\}$ 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 [12].

For SVM-based speaker verification, an SVM is trained for each target speaker using the speaker's enrollment utterances as positive examples, and the utterances in some development set as negative examples.

In this study, each utterance is represented by the following vector in the input space of SVMs [13] – [15],

$$\mathbf{v} = \left[l\left(\mathbf{x}\middle|\bar{\lambda}_1\right) \quad l\left(\mathbf{x}\middle|\bar{\lambda}_2\right) \quad \cdots \quad l\left(\mathbf{x}\middle|\bar{\lambda}_E\right) \right]^T, \tag{2}$$

where $l\left(\mathbf{x}\middle|\bar{\lambda}_i\right)$ is a normalized log-likelihood score of the speaker utterance data $\mathbf{x}$ (of T acoustic feature vectors) for the i-th reference speaker model (called *anchor models*), $\bar{\lambda}_i$, relative to a universal background model (UBM), $\bar{\lambda}_{UBM}$,

$$l\left(\mathbf{x}\middle|\bar{\lambda}_i\right) = \frac{1}{T}\log\left(p\left(\mathbf{x}\middle|\bar{\lambda}_i\right) \Big/ p\left(\mathbf{x}\middle|\bar{\lambda}_{UBM}\right)\right). \tag{3}$$

3. Nuisance attribute projection in nonlinear kernels

In this section, two kinds of nonlinear kernel NAP are presented. One is formulated in the feature space and the other is in the input space.

3.1. Nonlinear kernel NAP in the feature space

In this case, input variables are firstly transformed into the feature space. A projection, $P = I - U_m U_m^T$, is then found to filter out nuisance attributes (e.g. session/channel variability) in the feature vectors through optimizing the following criterion [5]:

$$\begin{aligned} &U_m^* = \arg\min \sum_{i,j} M_{i,j} \left\| P\cdot\Phi\left(\mathbf{v}_i^d\right) - P\cdot\Phi\left(\mathbf{v}_j^d\right)\right\|^2, \\ &s.t.\ P = I - U_m U_m^T \text{ and } U_m^T U_m = I. \end{aligned} \tag{4}$$

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, $M_{i,j}$, in this study are set to be

$$M_{i,j} = \begin{cases} 1, & \text{if } \mathbf{v}_i^d \text{ and } \mathbf{v}_j^d \text{ from the same speaker} \\ 0, & \text{otherwise} \end{cases}. \tag{5}$$

The constraint, $U_m^T U_m = I$, is used to guarantee P is a projection matrix, i.e. $P^2 = P$.

As shown in [5], U_m^* in (4) can be found to consist of m eigenvectors with largest eigenvalues of the symmetric eigenvalue problem:

$$AZ(M)A^T U_m = U_m \Lambda, \tag{6}$$

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

For nonlinear kernels, instead of doing eigenvalue analysis in the high (or infinite) dimensional transformed feature space directly, kernel PCA is employed as follows [16]. The eigenvectors in U_m^* are firstly represented as

$$U_m^* = AZ^{1/2}Y_m, \tag{7}$$

where $Z^{1/2} = \left(diag\left(M\cdot\mathbf{1}\right)\right)^{1/2} - \left(diag\left(M\cdot\mathbf{1}\right)\right)^{-1/2} M$ and the columns in the matrix Y_m represent how to construct the eigenvectors in U_m^* as combination of feature vectors in the development set.

Substituting equation (7) into (6), we get

$$AZ^{1/2}Z^{1/2}A^T AZ^{1/2}Y_m = AZ^{1/2}Y_m\Lambda. \tag{8}$$

Multiplying both sides of equation (8) with $Z^{1/2}A^T$, we can deduce that Y_m can be derived through m eigenvectors with largest eigenvalues in the following eigenvalue analysis problem,

$$Z^{1/2}GZ^{1/2}Y_m = Y_m\Lambda, \tag{9}$$

where $G = A^T A$ is the Gram matrix for the development set with entries $G_{i,j}$:

$$G_{i,j} = k\left(\mathbf{v}_i^d, \mathbf{v}_j^d\right). \tag{10}$$

Comparing (6) and (9), we can see that through kernel PCA the eigenvalue analysis problem in the high dimensional feature space is reduced to the eigenvalue problem of $Z^{1/2}GZ^{1/2}$ whose size is determined by the number of features in the development set.

NAP in the feature space can be incorporated into a compensated kernel function without projecting high or infinite dimensional feature vector explicitly [8], i.e.,

$$\begin{aligned} k_{NAP}\left(\mathbf{v}_1,\mathbf{v}_2\right) &= \left\langle P\cdot\Phi\left(\mathbf{v}_1\right), P\cdot\Phi\left(\mathbf{v}_2\right)\right\rangle \\ &= \left\langle \Phi\left(\mathbf{v}_1\right), \Phi\left(\mathbf{v}_2\right)\right\rangle - \left\langle U_m^T\cdot\Phi\left(\mathbf{v}_1\right), U_m^T\cdot\Phi\left(\mathbf{v}_2\right)\right\rangle \\ &= k\left(\mathbf{v}_1,\mathbf{v}_2\right) - \left\langle U_m^T\cdot\Phi\left(\mathbf{v}_1\right), U_m^T\cdot\Phi\left(\mathbf{v}_2\right)\right\rangle. \end{aligned} \tag{11}$$

Using the representation in equation (7), the above compensated kernel function can be written to be:

$$k_{NAP}\left(\mathbf{v}_1,\mathbf{v}_2\right) = k\left(\mathbf{v}_1,\mathbf{v}_2\right) - \tilde{\mathbf{v}}_1^T\cdot Z^{1/2}Y_m Y_m^T Z^{1/2}\cdot\tilde{\mathbf{v}}_2, \tag{12}$$

where

$$\tilde{\mathbf{v}}_i = \left[k\left(\mathbf{v}_i,\mathbf{v}_1^d\right), k\left(\mathbf{v}_i,\mathbf{v}_2^d\right), \cdots, k\left(\mathbf{v}_i,\mathbf{v}_n^d\right)\right]^T. \tag{13}$$

Here, we note that equation (13) involve the evaluation of kernel function between input vector and every vector in the development set. This might cause some storage and computational problems under some circumstances which involve some large development sets.

3.2. Nonlinear kernel NAP in the input space

In this formulation, the projection, $P = I - U_m U_m^T$, is found in the input space through optimizing the following objective function,

$$\begin{aligned} &U_m^* = \arg\min \sum_{i,j} M_{i,j} \left\| \Phi\left(P\cdot\mathbf{v}_i^d\right) - \Phi\left(P\cdot\mathbf{v}_j^d\right)\right\|^2, \\ &s.t.\ P = I - U_m U_m^T \text{ and } U_m^T U_m = I. \end{aligned} \tag{14}$$

All settings in (14) are the same as those in (4) except we swap the order of projection, P, and feature transformation, Φ.

Using the relationship between feature transformation and kernel functions, we can get

$$\begin{aligned} &\left\| \Phi\left(P\cdot\mathbf{v}_i^d\right) - \Phi\left(P\cdot\mathbf{v}_j^d\right)\right\|^2 = k\left(P\cdot\mathbf{v}_i^d, P\cdot\mathbf{v}_i^d\right) \\ &+ k\left(P\cdot\mathbf{v}_j^d, P\cdot\mathbf{v}_j^d\right) - 2\cdot k\left(P\cdot\mathbf{v}_i^d, P\cdot\mathbf{v}_j^d\right). \end{aligned} \tag{15}$$

Then the objective function in equation (14) can then be written as

$$\begin{aligned} Q &= \sum_{i,j} M_{i,j} \left\| \Phi\left(P\cdot\mathbf{v}_i^d\right) - \Phi\left(P\cdot\mathbf{v}_j^d\right) \right\|^2 \\ &= 2\cdot\sum_{i,j} M_{i,j}\left[k\left(P\cdot\mathbf{v}_i^d, P\cdot\mathbf{v}_i^d\right) - k\left(P\cdot\mathbf{v}_i^d, P\cdot\mathbf{v}_j^d\right)\right] \end{aligned} \quad (16)$$

For nonlinear kernels, this objective function is nonlinear. To find the optimum projection, a gradient descent algorithm proceeds as following:

1. Initialize randomly the projection matrix U_m;
2. Update U_m with

$$U_m \leftarrow U_m - \eta\frac{\partial Q}{\partial U_m}, \quad (17)$$

where $\eta > 0$ is the gradient step and

$$\frac{\partial Q}{\partial U_m} = 2\cdot\sum_{i,j} M_{i,j}\left[\frac{\partial k\left(P\cdot\mathbf{v}_i^d, P\cdot\mathbf{v}_i^d\right)}{\partial U_m} - \frac{\partial k\left(P\cdot\mathbf{v}_i^d, P\cdot\mathbf{v}_j^d\right)}{\partial U_m}\right]; \quad (18)$$

3. To impose the constraint on U_m in equation (14), orthonormalize U_m with

$$U_m \leftarrow U_m\left(U_m^T U_m\right)^{-1/2}; \quad (19)$$

4. Return to step 2 or terminate if the decrease of objective function in equation (16) falls below a predefined threshold ε.

In the following, we restrict ourselves to the Gaussian kernels,

$$k\left(\mathbf{v}_1,\mathbf{v}_2\right) = \exp\left(-\left\|\mathbf{v}_1-\mathbf{v}_2\right\|^2/\sigma^2\right), \quad (20)$$

and polynomial kernels,

$$k\left(\mathbf{v}_1,\mathbf{v}_2\right) = \left(s\cdot\mathbf{v}_1^T\mathbf{v}_2 + r\right)^d. \quad (21)$$

For Gaussian kernels, the gradient of kernel function with respect to the projection matrix in (18) can be calculated as

$$\begin{aligned} \frac{\partial k\left(P\cdot\mathbf{v}_1, P\cdot\mathbf{v}_2\right)}{\partial U_m} &= -\frac{1}{\sigma^2}k\left(P\cdot\mathbf{v}_1, P\cdot\mathbf{v}_2\right)\frac{\partial\left\|P\cdot\mathbf{v}_1 - P\cdot\mathbf{v}_2\right\|^2}{\partial U_m} \\ &= \frac{2}{\sigma^2}k\left(P\cdot\mathbf{v}_1, P\cdot\mathbf{v}_2\right)\left[\mathbf{v}_1\cdot\left(\mathbf{v}_1-\mathbf{v}_2\right)^T + \mathbf{v}_2\cdot\left(\mathbf{v}_2-\mathbf{v}_1\right)^T\right]\cdot U_m. \end{aligned} \quad (22)$$

Here, we use the following derivative,

$$\begin{aligned} \partial\mathbf{v}_1^T P\mathbf{v}_2 / \partial U_m &= \partial\mathbf{v}_1^T\left(I - U_m U_m^T\right)\mathbf{v}_2 / \partial U_m \\ &= -\mathbf{v}_1\cdot\mathbf{v}_2^T\cdot U_m - \mathbf{v}_2\cdot\mathbf{v}_1^T\cdot U_m \end{aligned} \quad (23)$$

Similarly, the gradient of polynomial kernels with respect to the projection matrix can be derived to be

$$\begin{aligned} \frac{\partial k\left(P\cdot\mathbf{v}_1, P\cdot\mathbf{v}_2\right)}{\partial U_m} &= d\cdot s\cdot\left(s\cdot\mathbf{v}_1^T P\mathbf{v}_2 + r\right)^{d-1}\frac{\partial\mathbf{v}_1^T P\mathbf{v}_2}{\partial U_m} \\ &= -d\cdot s\cdot\left(s\cdot\mathbf{v}_1^T P\mathbf{v}_2 + r\right)^{d-1}\left(\mathbf{v}_1\mathbf{v}_2^T + \mathbf{v}_2\mathbf{v}_1^T\right)\cdot U_m. \end{aligned} \quad (24)$$

For these two kinds of kernels, the optimization problem in equation (14) can be shown to be a convex optimization problem. Hence, the above gradient descent algorithm is guaranteed to find the global optimal solution in these cases.

Since the dimension of the input space is usually much lower than the expanded feature space, the projection over input variables can usually be carried out efficiently. And, in this case, the development set is no longer needed in the projection procedure once the projection matrix has been derived.

4. Experimental results

In this section, we report speaker verification experiments on the 2006 NIST SRE corpus with different configuration of nonlinear kernel NAP. Section 4.1 presents some general experiment setup information about the task, database, features and kernel configuration. The results of these experiments are discussed in Section 4.2.

4.1. Protocol

Speaker verification experiments were conducted on the 2006 NIST SRE corpus [10]. 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.

4.1.1. Database

A subset of the 2004 NIST SRE corpus (the single-side, 1 conversation train, single side, 1 conversation test part) is used as the development set. There are a total of 1790 utterances from 310 speakers. The development set is used for negative samples in SVM training; and it is also used to estimate parameters in NAP.

4.1.2. System configuration

For the cepstral features used for anchor modeling, 13 PLP coefficients are 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. A gender independent UBM with 2048 Gaussians is trained using about 40 hours of data from the Switchboard corpora (I, II and Cellular parts). Totally 500 reference speakers (230 male and 270 female speakers) in the data for UBM training are used as anchor models in our experiments.

For the Gaussian kernel used in following experiments, the parameter, σ, in equation (20) is set to be 2. The parameters, s, r and d in equation (21) for polynomial kernels are set to be 1, 1 and 2 respectively.

4.2. Results

In Figure 1 and 2, we summarize the Equal Error Rate (EER) results for Gaussian and polynomial kernels respectively. Detection cost function (DCF) results have similar trends and are not plotted here.

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 both NAP projection in feature space and that in input space could reduce session variability in the kernels; and verification performance gradually increases as more nuisance directions are projected out.

However, if too many directions are being left out, not only features from the same speaker would become closer but features from different speakers would also be pulled together which might cause the discrimination across different speakers become more difficult. This situation is more serious in the low dimensional input space. Under current experimental setting, the dimension of input space is 500. For NAP in the input space, verification performance degraded seriously when the number of directions projected out exceeded 64. In these cases, as there are many more higher

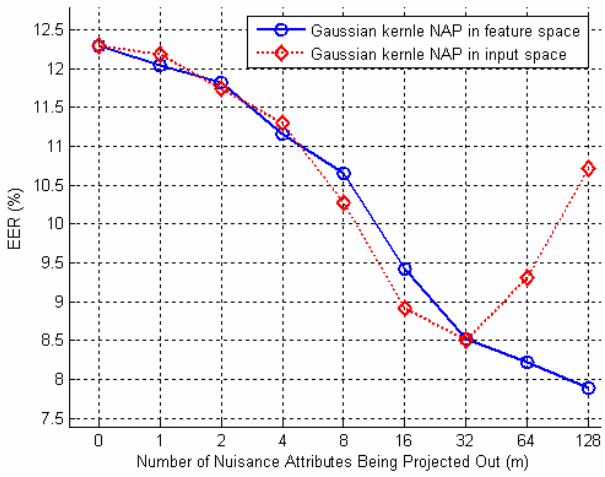

Figure 1: *EER vs. number of nuisance attributes projected out for Gaussian kernel NAP in feature and input spaces.*

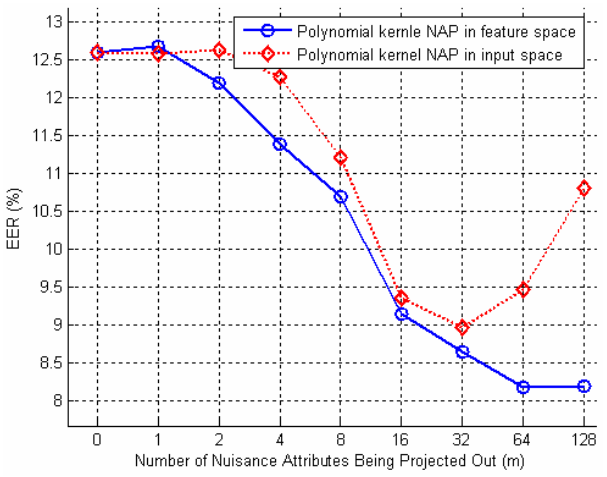

Figure 2: *EER vs. number of nuisance attributes projected out for polynomial kernel NAP in feature and input spaces.*

order features in the expanded feature space, NAP in the feature space using kernel PCA could extract more nuisance directions and obtain better verification performance than NAP in the input space.

When the number of nuisance directions was set to be 32, the evaluation of NAP projection in the input space for each trial takes about 5ms on a machine with Pentium 4, 3 Ghz. While for NAP in the feature space, the calculation of $Y_m^T Z^{1/2}\tilde{\mathbf{v}}$ in (12) takes about 35ms (for the Gaussian kernel configuration) on the same machine as it requires evaluation of kernel function against all 1790 features in the development set. Hence, NAP in the input space is computationally more efficient that that in the feature space.

5. Conclusions

In this study, two different approaches to nonlinear kernel nuisance attribute projection are compared for SVM-based text-independent speaker verification. For NAP in the feature space, kernel PCA was employed to do kernel eigenvalue analysis in the high or infinite dimensional expanded feature space; and the kernel function was compensated accordingly to incorporate NAP without explicit feature projection. For NAP in the input space, although it is carried out in the input space, the projection is still aiming to reduce unwanted variability in the underlying nonlinear kernel; and a gradient descent algorithm was proposed to solve corresponding nonlinear optimization problem. Experimental results on the 2006 NIST SRE corpus show that both approaches could reduce session variability in nonlinear kernels and improve speaker verification performance effectively. Comparing the two approaches, NAP in the feature space provides the possibility to extract more nuisance attributes after exploiting higher order correlation among input variables. On the other hand, NAP in the input space is less expensive with respect to computational and storage requirements.

6. References

[1] D. Reynolds, "Channel robust speaker verification via feature mapping," in *Proc. ICASSP*, 2003.

[2] P. Kenny, G. Boulianne, P. Ouellet and P. Dumouchel, "Speaker and session variability in GMM-based speaker verification," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 15, no. 4, pp. 1448-1460, May, 2007.

[3] R. Vogt, B. Baker and S. Sridharan, "Modeling session variability in text-independent speaker verification," in *Proc. INTERSPEECH*, 2005.

[4] C. Vair, D. Colibro, F. Castaldo, E. Dalmasso and P. Laface, "Channel factors compensation in model and feature domain for speaker recognition," in *Proc. IEEE Odyssey 2006*, 2006.

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

[6] A. O. Hatch, S. Kajarekar and A. Stolcke, "Within-Class Covariance Normalization for SVM-based Speaker Recognition," in *Proc. ICSLP*, 2006.

[7] W. M. Campbell, D. E. Sturim, D. A. Reynolds and A. Solomonoff, "SVM based speaker verification using a GMM supervector kernel and NAP variability compensation," in *Proc. ICASSP*, 2006.

[8] X. Zhao, Y. Dong, H. Yang, J. Zhao, L. Lu and H. Wang, "Nonlinear kernel nuisance attribute projection for speaker verification," in *Proc. ICASSP*, 2008.

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

[10] National Institute of Standards and Technology, "The NIST 2006 speaker recognition evaluation plan," http://www.nist.gov/speech/tests/spk/spk/2006/ .

[11] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

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

[13] 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.

[14] Y. Mami, D. Charlet, "Speaker recognition by location in the space of reference speakers," *Speech Communication*, vol.48, pp. 127-141, 2006.

[15] 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.

[16] 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.