A symmetric kernel partial least squares framework for speaker verification

Balaji Vasan Srinivasan, Yuancheng Luo, Daniel Garcia-Romero, Dmitry N. Zotkin, Ramani Duraiswami, Member, IEEE

***Abstract*—I-vectors are concise representations of speaker characteristics. Recent progress in i-vectors related research has utilized their ability to capture speaker and channel variability to develop efficient automatic speaker verification (ASV) systems. Inter-speaker relationships in the i-vector space are non-linear. Accomplishing effective speaker verification requires a good modeling of these non-linearities and can be cast as a machine learning problem. Kernel partial least squares (KPLS) can be used for discriminative training in the i-vector space. However, this framework suffers from training data imbalance and asymmetric scoring. We use "one shot similarity scoring" (OSS) to address this. The resulting ASV system (OSS-KPLS) is tested across several conditions of the NIST SRE 2010 extended core data set and compared against state-of-the-art systems: Joint Factor Analysis (JFA), Probabilistic Linear Discriminant Analysis (PLDA), and Cosine Distance Scoring (CDS) classifiers. Improvements are shown.**

***Index Terms*—One-shot similarity, kernel PLS, speaker verification, speaker recognition, discriminative classifier.**

I. Introduction

Speaker verification [1] deals with the task of verifying a speaker's claimed identity based on a sample utterance from the target speaker along with a number of training utterances from several non-target speakers. Apart from carrying the speaker-specific characteristics, the speech data also encapsulates phonemic content, channel variability, and inter-session variability. In addition, it is subject to degradations due to noise and reverberation, making the problem of speaker verification challenging. Over the past decade, the field has made substantial progress in addressing these issues.

State-of-the-art speaker verification systems use a Gaussian mixture model (GMM) to represent each speaker. To account for limited training data, the problem is cast into a framework in which differences from a universal background model (UBM) are used to adapt speaker-specific GMMs [2]. Recently, several approaches have been tested to make the GMM-based speaker verification robust to inter-session and channel variabilities, including the Joint Factor Analysis (JFA) technique [3] [4] and the i-vector framework [5]. The JFA learns two subspaces of maximal channel-related and speaker-related variabilities and projects any given supervector into these subspaces to separate channel-related and speaker-related components. On the other hand, i-vectors encapsulate the directions in a total variability space. The i-vectors are smaller in dimension compared to the GMM-supervectors and thus provide an abridged representation of the utterance.

Given the i-vector representation, the key problem is to develop learning techniques that distinguish target and non-target trials in the i-vector space. Generative PLDA models [6], discriminative SVMs [7], and CDS classifiers [8] have been studied for speaker verification using i-vectors. In this paper, we consider extending a linear partial least squares framework into its kernelized version to model the non-linear i-vector space.

Discriminative frameworks in speaker verification fall under two categories. The first class of approaches models each speaker independently against a set of background speakers [7]. While such an independent modeling is preferred in many practical ASV systems, there is usually an imbalance in the number of positive and negative examples available for training. Also, the scoring is generally asymmetric unlike most of generative approaches (e.g. JFA, PLDA, and others). Alternative methods attempt to learn a pair-wise similarities between a training utterance and a test utterance e.g. [8] [9]. While such an approach does not allow for explicit independent speaker modeling, it does not suffer from data imbalance and asymmetric scoring due to pair-wise considerations. In this paper, we combine both these approaches into a single hybrid framework by developing a KPLS-based discriminative "one-shot-similarity" framework.

The one-shot-similarity framework has been previously proposed in the context of face verification. Given two (feature) vectors, the one-shot-similarity reflects the likelihood that each vector belongs to the same class as the other vector and not in the class defined by a set of negative examples. The potential of this approach has been explored widely in computer vision [10] [11], and this paper applies it to speaker verification.

The paper is organized as follows. In Section II, GMM-UBM and the associated JFA frameworks are introduced and extended to the total variability framework with i-vectors. The partial least squares framework is introduced and kernelized to address the speaker verification problem with i-vectors in Section III. The one-shot similarity scoring is detailed in Section IV. Section V discusses the results obtained on NIST SRE 2010 data and includes comparisons against several state-of-the-art systems.

B. V. Srinivasan is with Adobe Research Bangalore Labs, Bangalore, India. Email: balsrini@adobe.com

Y. Luo, D. N. Zotkin, and R. Duraiswami are with the Perceptual Interfaces and Reality Laboratory, Department of Computer Science, University of Maryland, College Park, MD 20742 USA. Email: [yluo1,dz,ramani]@umiacs.umd.edu.

Daniel Garcia-Romero is with the Speech Communication Laboratory, Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742 USA. Email: dgromero@umd.edu.

II. Gaussian Mixtures, Joint Factor Analysis, and I-Vectors

Given a speech utterance, the GMM-UBM framework models the speaker with a mixture of N Gaussians in the d dimensional feature space. The speaker is recognized based on the likelihood ratio between the hypothesized speaker model and an imposter model. The imposter model is built by pooling speech from several speakers and training a single model, which is known as the Universal Background Model or UBM and denotes an "average" speaker. The UBM is also used to obtain individual speaker models via MAP adaptation [2]. It is a common practice to concatenate the centers of the MAP-adapted GMMs into a single $N \times d$ dimensional *supervector* and to learn discriminative speaker models in this space [12].

While the MAP adaptation and supervectors work well for speaker verification, it fails in the presence of smaller training utterances where the data sparsity prevents some components of the UBM from being adapted. Further, it does not address any nuisance compensations in the supervector space. Joint Factor Analysis (JFA) [13] attempts to address this by correlating the various Gaussian components of the UBM. The key assumption in JFA is that the intrinsic dimension of the adapted supervector is much smaller than $N \times d$. JFA breaks down the speaker- and channel-dependent supervector M into two components:

$$M = s + c, \tag{1}$$

where s is the speaker-dependent part and c is the channel-dependent part given by

$$\begin{aligned} s &= m + Vy + Dz, \\ c &= Ux, \end{aligned} \tag{2}$$

where $\{m, U, V, D\}$ are the hyper-parameters of the JFA model, which are estimated via Expectation Maximization (EM). The rationale behind this equation is that aside from the offset m (UBM), the mean supervector is the superposition of three fundamental components with rather distinctive meaning. The component that lives in U is used to denote undesired variabilities in the observed vectors (e.g. channel-related variability). The component living in V is used to denote the speaker-related variability. U and V are termed the Eigen-channel and Eigen-voices respectively and typically have $Nd \times 400$ dimensions. D provides a mechanism to capture the residual variability not captured by U and V. Thus, the key idea in the JFA technique is to find two subspaces (V and U) that best capture the speaker and the channel variabilities in the feature space. The term joint factor analysis comes from the fact the three latent variables x, y and z are jointly estimated (unlike traditional factor analysis where an independent estimation is adopted).

Dehak et al. [5] observed that the channel subspace still contains information about the speaker and vice-versa. Therefore, a combined subspace was proposed to capture both variabilities called the *total variability space*. In this formulation, the speaker- and channel-dependent supervector s is modeled as,

$$s = m + Tw, \tag{3}$$

where m is a speaker- and channel-independent supervector (usually the UBM supervector), T is a matrix of dimensions $Nd \times 400$ representing the basis of the total variability space, and w is a normal-distributed vector representing the coordinates of the speaker in that space. The vector w is called the *i-vector*, short for "intermediate vectors" due to their intermediate representation between the acoustic and supervector representation or the "identity vectors" for their compact representation of a speaker's identity. The set $\{m, T\}$ represents the hyper-parameters of the total-variability framework. Typically, the number of dimensions of w is three orders of magnitude smaller than that of the supervectors (e.g. 400 vs 10^5). The i-vectors thus provide a concise representation of the high-dimensional supervectors.

A. Hyper-parameter training

A key difference between the training of total variability matrix T and the Eigen-voices V is that for V all utterances from a specific speaker are considered to belong to the same person, whereas T-training assumes each utterance to be independent regardless of the speaker identity. Otherwise, both training procedures use a similar EM-approach [4]. The Eigen-channel matrix U is estimated after estimating the Eigen-voices V, and the details are available in [14]. Both JFA and i-vectors are adept at modeling the intrinsic variabilities; however, i-vectors are increasingly being preferred due to their superior performance and compact representation.

B. Intersession compensation in i-vector space

Unlike JFA, the i-vector representation does not include explicit compensation for the intersession variabilities. However, standard intersession compensations have been proposed in the i-vector space [15]. The most successful method is a Linear Discriminant Analysis (LDA) projection followed by Within Class Covariance Normalization (WCCN).

Linear discriminant analysis (LDA) attempts to find orthogonal directions that simultaneously maximize the inter-speaker discrimination and minimize the intra-speaker variability. This is analogous to learning a subspace that removes nuisance directions from i-vectors. The idea behind WCCN is to scale the total variability space by a factor that is inversely proportional to an estimate of the within-class covariance. This deemphasizes the directions of high intra-speaker variability.

A LDA-based subspace is first learned on the i-vectors, and training and testing i-vectors are projected into this space. Let L denote the LDA projection matrix. Then, a within-class covariance normalization matrix W is learned on LDA-projected space. A compensated i-vector for a raw i-vector w is given by

$$\hat{w} = W^{-0.5}(Lw). \tag{4}$$

A detailed description of the nuisance compensation in the i-vector space is available in [15]. Given the compensated i-vectors, the key challenge in speaker verification is the design of appropriate learning techniques that can classify speakers in the i-vector space.

III. Partial least squares (PLS)

Partial least squares (PLS) is a subspace-based learning technique that has been used for dimensionality reduction as well as a regression. It was first developed by Herman Wold in the 1960s and 1970s to address problems in econometric path modeling [16] and was subsequently adapted in the 1980s to problems in chemometric and spectrometric modeling [17] [18]. In the late 1980s it attracted the attention of statisticians [19] due to its ability to handle learning where the data has a very low rank or a lot of redundancy, leading to the existence of a low-dimensional subspace. PLS exploits this subspace to effectively learn the target patterns. Recently, PLS has been successfully applied to problems in computer vision [20] [21] and also for speaker verification in the supervector space [22]. We shall describe PLS and its kernelization briefly here; readers are referred to [23] and [24] for rigorous discussion and further details.

Denote a d-dimensional set of independent variables (predictors) as x and the corresponding response variable by y. In the context of speaker verification, x could represent any feature representation (like a supervector or an i-vector) and y is the corresponding speaker identity (output variable that has to be learned). In all of our analysis and experiments, x represents the i-vector extracted from a speech utterance. Assume that the total number of speakers is N and denote the $N \times d$ matrix of i-vectors by X and the $N \times 1$ vector of labels (1 for speaker and -1 for imposter) by Y. Given the variable pairs $\{x_i, y_i\}, i = 1, \ldots, N$ $(x \in R^d, y \in R)$, PLS models the relationship between x and y using projection into latent spaces. PLS decomposes X and Y as

$$X = TP^T + E, \tag{5}$$

$$Y = UQ^T + F, \tag{6}$$

where T and U ($N \times p$, $p < d$) are the latent vectors, P ($d \times p$) and Q ($1 \times p$) are the loading vectors, and E ($N \times d$) and F ($N \times 1$) are residual matrices. PLS is usually solved via the *nonlinear iterative partial least squares (NIPALS) algorithm* [23] that constructs a set of weight vectors $W = \{w_1, w_2, \ldots, w_p\}$ such that

$$[\mathrm{cov}(t_i, u_i)]^2 = \max_{|w_i|=1} [\mathrm{cov}(Xw_i, Y)]^2, \tag{7}$$

where t_i and u_i are the i^{th} columns of T and U respectively and $\mathrm{cov}(t_i, u_i) = t_i^T u_i/n$ indicates the sample covariance between latent vectors t_i and u_i. In other words, the NIPALS algorithm learns weight w_i that maximizes the covariance of the corresponding latent vectors. After extraction of latent vectors t_i and u_i, the matrices X and Y are deflated by subtracting their rank-1 approximation based on t_i and u_i:

$$\begin{aligned} X &\leftarrow X - t_i p_i^T; \\ Y &\leftarrow Y - u_i q_i^T. \end{aligned} \tag{8}$$

This step removes any information captured by t_i and u_i from X and Y.

It has been shown [23] that the NIPALS algorithm is equivalent to iteratively finding the dominant Eigenvectors of the problem

$$[X^T y y^T X] w_i = \lambda w_i. \tag{9}$$

The complete algorithm is described in Alg. 1.

Algorithm 1 *Nonlinear Iterative PArtial Least Squares (NIPALS) algorithm*

Given: $N \times d$ feature samples X and $N \times 1$ response variable Y
repeat
  Assign $u = Y$
  repeat
    $w = \frac{X^T u}{u^T u}$
    $\|w\| \rightarrow 1$
    $t = Xw$
    $c = \frac{Y^T t}{t^T t}$
    $\|c\| \rightarrow 1$
    $u = Yc$
  until convergence of t
  $p = X^T t$
  Deflate $X : X \leftarrow X - tp^T$
  Deflate $Y : Y \leftarrow Y - tc^T$
until Required number of factors are obtained

A. PLS Regression

The weight matrix W can be used for dimensionality reduction, and the resulting projection can be used with any standard regression/classifier to learn the speaker model. Alternately PLS can directly be used in a regression framework that implicitly utilizes the PLS weights W obtained from the NIPALS algorithm. This is described here.

Substituting the w from Equation (7) in Equation (5), we get

$$\begin{aligned} XW &= TP^T W + E \\ \Rightarrow T &= XW(P^T W)^{-1} + \bar{E}, \end{aligned} \tag{10}$$

where, $\bar{E}$ is the modified residue. Now, U can be written in terms of T [23] as

$$U = TD + H, \tag{11}$$

where the matrix D is diagonal (this is because estimation of $\{t_i, u_i\}$ is independent from the estimation of $\{t_j, u_j\}$ for $i \neq j$) and H is the residue. Equation (6) now becomes

$$Y = TDQ^T + HQ^T + F \tag{12}$$

$$= XW(P^T W)^{-1} DQ^T + \bar{F}, \tag{13}$$

and we get the PLS regression:

$$\begin{aligned} Y &= XB + G; \\ B &= W(P^T W)^{-1} DQ^T, \end{aligned} \tag{14}$$

where B is the set of PLS regression coefficients.

Once PLS training is done, a single testing i-vector x_t can be plugged into the regression equation to generate regression score s_{pls}:

$$s_{pls} = x_t X^T \tilde{B}, \qquad \tilde{B} = U(T^T X X^T U)^{-1} DQ^T. \tag{15}$$

In deriving this equation, we used the relationships $P = X^T T$ and $W = X^T U$ [23]. The matrix $\tilde{B}$ relates to matrix B as

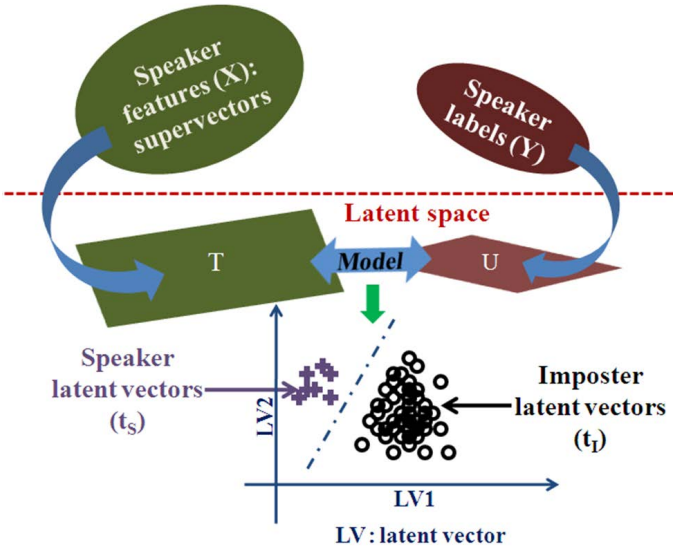

Fig. 1. PLS projects the input data into common subspace so that the speaker and the imposter latent vectors are well-separated.

$B = X^T\tilde{B}$; stripping of X^T from the front of B was done for a reason that will become clear in the next subsection.

As described before, the PLS procedure maximizes the covariance in the latent vector space. This is equivalent to maximizing discrimination in the same space. In other words, for a particular speaker KPLS learns a subspace in which the speaker latent vectors t_S and well-separated from the imposter latent vectors t_I as illustrated in Figure 1. Note that the regression matrix B (i.e., the learned latent subspace) is *unique for each speaker* and in fact constitute the speaker model if an explicit notion of speaker model is desired.

The PLS regression is linear by nature and has been used to model the speakers in the supervector space [22]; however, a direct extension to i-vectors does not suffice to model the non-linear i-vector space. We therefore consider kernelization of the PLS regression [24] to model the non-linear i-vector space.

B. PLS Kernelization

Kernel PLS considers the mapping of the features X to a higher dimensional space given by $\Phi : R^d \Rightarrow R^{\bar{d}}$ and learns a PLS subspace for $\{\Phi(X), Y\}$. Assume momentarily that such a Φ is defined and known. The covariance that is maximized in linear PLS is modified for the kernel PLS as

$$[\mathrm{cov}(t_i, u_i)]^2 = \max_{|w_i|=1} [\mathrm{cov}(\Phi(X)w_i, Y)]^2, \tag{16}$$

where t_i and u_i are similar to those in Equation 7. The Eigenvalue problem of the linear PLS becomes

$$[\Phi(X)^T yy^T \Phi(X)]w_i = \lambda w. \tag{17}$$

The $\Phi(X)$-scores t_i are then obtained as $t_i = \Phi(X)w_i$. Rosipal et al. [24] modify this Eigenproblem as

$$[\Phi(X)\Phi(X)^T yy^T]t = \gamma t. \tag{18}$$

Using the "kernel" trick [25], $\Phi(X)\Phi(X)^T$ can be defined as a kernel matrix K leading to the final Eigenproblem

$$[Kyy^T]t = \gamma t. \tag{19}$$

A key advantage of this kernelization is that (similarly to SVM) an explicit definition of the mapping function Φ is not required, and it suffices to define a kernel function between pairs of feature vectors. After extraction of latent vectors t_i and u_i, the kernel matrix K is deflated (similar to the Equation 8 for linear PLS) by removing any information captured by t_i and u_i from K:

$$K \leftarrow (I_n - tt^T)K(I_n - tt^T). \tag{20}$$

The entire process is repeated till a sufficient number (determined via standard cross-validation) of latent vectors is obtained. More detailed description of KPLS is available in [24].

After KPLS training is performed, the KPLS score s_{kpls} is computed as

$$s_{kpls} = \Phi(x_t)\Phi(X)^T\tilde{B}, \quad \tilde{B} = U(T^T\Phi(X)\Phi(X)^T U)^{-1}DQ^T. \tag{21}$$

The corresponding derivations are omitted as they duplicate the same for linear PLS. In kernelized form, the equations become

$$s_{kpls} = K_t\tilde{B}, \quad \tilde{B} = U(T^T KU)^{-1}DQ^T, \tag{22}$$

where $K_t = \Phi(x_t)\Phi(X)^T$ is the kernel between the testing vector and all training vectors and $K = \Phi(X)\Phi(X)^T$ is the pairwise kernel between all training vectors. Note that (as desired) the explicit computation of $\Phi(X)$ is never required.

IV. One-shot similarity scoring

One-shot similarity (OSS) draws its motivation from the class of one-shot learning techniques that learn from one or few training examples [10]. It has been explored in the contexts of insect species identification [10] and face verification [11]. OSS compares two vectors by building two models (one for each of them) against a *common background data set* A. Given A, OSS first computes a model using vector x as a single positive examples and all vectors in A as negative examples and then uses the model to score y. Intuitively, this score would give the likelihood of y belonging to the same class as x. A similar score for x based on a model built with y as a positive example and the same negative example set is generated. The OSS score is then obtained by averaging these two scores. These steps are illustrated in Figure 2.

In the context of the current work, the background data set A is the set of i-vectors from background speakers, x is the target speaker's training i-vector, y is the test i-vector, and KPLS is used to build these "one-against-all" models.

In our experiments, we use gender-dependent sets A. For each target speaker, the corresponding i-vector is assigned a label of $+1$; samples in set A are assigned a label of -1; the KPLS model is trained; and the speaker-specific regression coefficients $\tilde{B}$ are learned according to Equation (22). This is repeated for both train and test i-vectors. KPLS output scores in each case are Z-normalized [1]. The scores are then

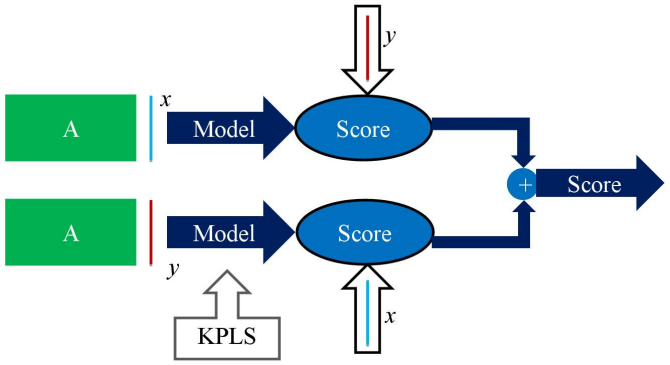

Fig. 2. [color] One Shot Similarity scoring.

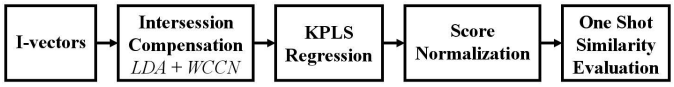

Fig. 3. A schematic view of the speaker verification system implemented.

combined into a one-shot similarity score. These steps are summarized in Figure 3.

V. Experiments

We performed experimental evaluation of the proposed method on the *extended core set* of the NIST SRE 2010 evaluation data set, which is grouped into 9 trial conditions[1]. Our development data consisted of NIST SRE 2004, 2005, 2006, and 2008 data; Switchboard data set, phases 2 and 3; Switchboard-Cellular data set, parts 1 and 2; and Fisher data set (total of 17319 male and 22256 female utterances). A gender-dependent 2048-center UBM with diagonal covariance was trained using the standard 57 MFCC features (extracted from the voice frames identified using the energy-based VAD in [26]), and the gender-dependent total variability matrix T of dimension 400 was also learned.

A. Parameters of OSS-KPLS

Two main parameters of the OSS-KPLS system are the kernel function and the background data set size.

1) Choice of kernel: There are several kernels used with kernel methods. We explored two of them – Gaussian kernel and cosine kernel – after fixing the size of A to 6000.

$$k_{\text{Gaussian}}(\hat{w}_1, \hat{w}_2) = \exp\left[\frac{\|\hat{w}_1 - \hat{w}_2\|^2}{2}\right], \quad (23)$$

$$k_{\text{Cosine}}(\hat{w}_1, \hat{w}_2) = \frac{\left(\hat{w}_1^T \hat{w}_2\right)}{\left(\hat{w}_1^T \hat{w}_1\right)\left(\hat{w}_2^T \hat{w}_2\right)}. \quad (24)$$

The performance of KPLS on the SRE 2010 extended-core dataset based on these two kernels is shown in Figure 4. The results are comparable, though the cosine kernel is marginally better. Hence, it was used in all subsequent experiments.

[1]http://www.itl.nist.gov/iad/mig/tests/sre/2010/

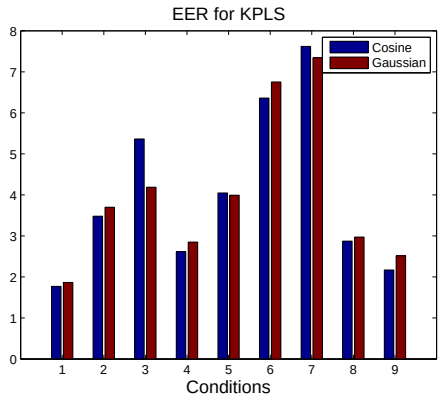

Fig. 4. [color] Performance of KPLS on the SRE 2010 extended core dataset based on the Gaussian and the Cosine kernels.

2) Background data set size: The common negative example data set determines the distribution of the negative examples. The samples in this set should be chosen to maximize the number of distinct speakers recorded on various channels, which would provide most information for the discriminative framework. The effect of background set size is shown in Figure 5 in terms of EER for condition 2 (interview speech from different microphone for training and testing) in SRE 2010 extended-core task.

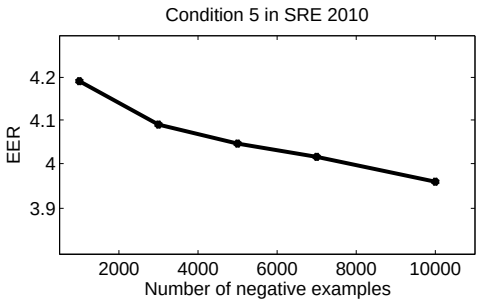

Fig. 5. Effect of the negative examples A set size on the performance: EER for condition 2 in the SRE 2010 extended core task with OSS-KPLS.

The drop in EER when the negative example set size is increased from 1000 to 5000 is considerable ($4.2 \rightarrow 4.0$). However, the same drop when the size is increased further to 10000 is not equally significant ($4.0 \rightarrow 3.95$) considering the increased computational cost associated with the increase. Therefore, the number of negative examples was set to 6000 in our experiments.

B. Systems compared

The proposed OSS-KPLS-based speaker verification was compared against several state-of-the-art systems, specifically JFA [3] [4], PLDA [6], and CDS [27]. We discussed JFA in Sec. II; we describe the other systems briefly here.

1) Joint Factor Analysis: In our experiments, we use the JFA as described in [14]. The U and V matrices are learned with 300 and 100 dimensions respectively.The final JFA scores are ZT-normalized [1].

2) Probabilistic Linear Discriminant Analysis: PLDA facilitates the comparison of i-vectors in a verification trial. A special *two-covariance* PLDA model is generally used for speaker verification in the i-vector space. The speaker variability and session variability are modeled using across-class and within-class covariance matrices (Σ_{ac} and Σ_{wc} respectively) in the PLDA setup. A latent vector y representing the speakers is assumed to be normally distributed $\mathcal{N}(y; \mu, \Sigma_{ac})$, and for a given speaker represented by this latent vector, the i-vector distribution is assumed to be $p(w|y) = \mathcal{N}(w; y, \Sigma_{wc})$.

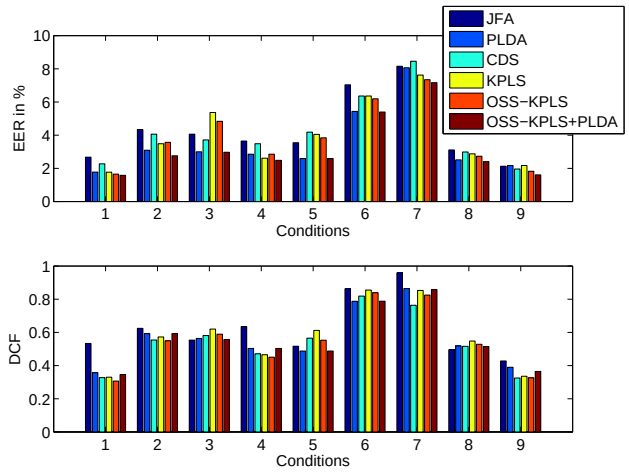

Fig. 6. [color] Performance of JFA, PLDA, CDS, KPLS, and OSS-KPLS classifiers on the NIST SRE 2010 extended core data set: EER and DCF metrics.

Given two i-vectors w_1 and w_2, PLDA defines two hypotheses $\mathcal{H}_s$ and $\mathcal{H}_d$ indicating that they belong to the same speaker or to different speakers respectively. The score is then defined as $\log \frac{p(w_1,w_2|\mathcal{H}_s)}{p(w_1,w_2|\mathcal{H}_d)}$. Marginalization of the two distributions with respect to the latent vectors leads to

$$\text{score}_{\text{PLDA}} = \log \frac{\mathcal{N}\left(\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}; \begin{bmatrix} \mu \\ \mu \end{bmatrix}, \begin{bmatrix} \Sigma_{tot}\Sigma_{ac} \\ \Sigma_{ac}\Sigma_{tot} \end{bmatrix}\right)}{\mathcal{N}\left(\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}; \begin{bmatrix} \mu \\ \mu \end{bmatrix}, \begin{bmatrix} \Sigma_{tot}0 \\ 0\Sigma_{tot} \end{bmatrix}\right)}$$

In our experiments, we used PLDA from [28] and found that using a Σ_{ac} of rank 200 along with a full-rank (rank 400) matrix Σ_{wc} produced the best results. The scores were S-normalized only for those conditions that involve telephone speech (all except C1, C2 and C4, where S-norm was found to be detrimental for both EER and DCF). The S-norm is defined in [6] and can be interpreted loosely as a symmetric version of Z-norm [1].

3) Cosine Distance Scoring: The CDS classifier has been used by Dehak et al. [5] and Senoussaoui et al. [8]. Improved performance has been reported over the corresponding SVM-based approach. The CDS classifier defines the score for the trial as a cosine similarity function between two i-vectors after compensating for intersession variabilities via LDA and WCCN projections. If w_1 and w_2 are the compensated i-vectors, the CDS score is given by

$$\text{score}_{\text{CDS}} = \frac{(w_1)^T(w_2)}{\sqrt{(w_1)^T(w_1)}\sqrt{(w_2)^T(w_2)}}. \quad (25)$$

In our experiments, the CDS scores were Z-normalized [1].

C. Results

We compared the performance of the OSS-KPLS based speaker verification against the JFA, PLDA, CDS, and KPLS systems. The corresponding equal error rate (EER) and the detection cost function (DCF) for each condition are tabulated in Table 1 and are shown graphically in Figure 6. DCF here is a "new-DCF" version as defined in NIST SRE 2010 "core" test. The corresponding DET curves are plotted in Figure 7. Table 1 also shows the "average" EER and DCF, which are obtained by averaging them across conditions weighted by the total number of trials in each condition. For reference, the breakdown of trials per condition is listed in Table 2. Because of a large number of trials in condition 2, the average EER and DCF are heavily dominated by this condition.

Both JFA and PLDA belong to the class of generative methods for speaker verification. Between them, the PLDA shows consistently better performance. In contrast, CDS, KPLS, and OSS-KPLS belong to the class of discriminative methods, and OSS-KPLS outperforms CDS in most of the conditions (in terms of EER). The problem of speaker verification typically suffers from class imbalance due to the fact that only a few utterances per target speaker are available for training. In such scenario, generative models generally perform better than discriminative ones, which is indeed the case as can be seen from the table. However, OSS-KPLS is the best system among all tested in terms of average DCF (with CDS being a very close second) and is the second best in terms of average EER.

Given the complementary nature of generative and discriminative representations, we explored the possibility of score fusion to leverage the combined power of generative and discriminative models. We chose to fuse PLDA with OSS-KPLS, as these are best representatives of each class. We have learned the linear fusion weights on a small development data set having EER as the objective function. Then, the fused score on the same SRE 2010 extended-code data set was computed by simple weighted averaging of PLDA and OSS-KPLS output scores (a more sophisticated fusion strategy is a subject of further research). The results are included in Table 1 (under the title of "FUSION") and in Figure 6; the EER obtained with the fusion approach is the best for all conditions.

For further analysis, Table 2 shows the relative EER improvement obtained through OSS-KPLS+PLDA fusion using PLDA system as the baseline. The improvement ranges from none in condition 5 to 25.8% in condition 9; however, this particular number is not indicative of the performance as the number of trials in condition 9 is very small.

Generally conditions 2 and 5 are considered most important for evaluation of the ASV system performance. Condition 2 is a generic microphone-recorded (interview) speech in testing and training; condition 5 is a generic telephone speech in testing and training. The presented results clearly demonstrate that

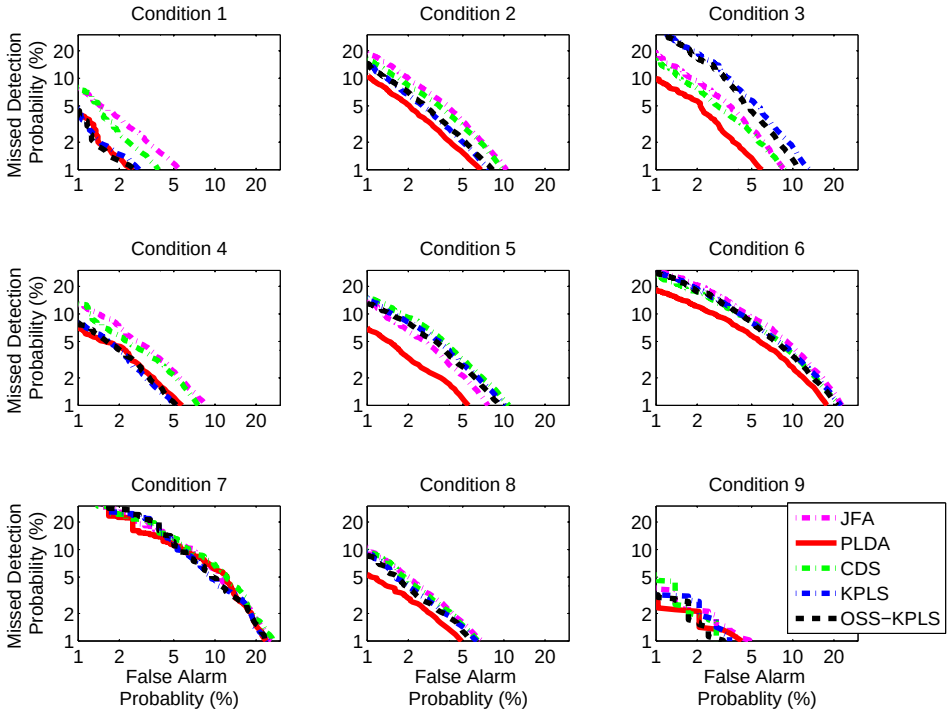

Fig. 7. [color] Performance of JFA, PLDA, CDS, KPLS, and OSS-KPLS classifiers on the NIST SRE 2010 extended-core data set: DET metric.

	JFA		PLDA		CDS		KPLS		OSS-KPLS		FUSION	
	EER	DCF	EER	DCF	EER	DCF	EER	DCF	EER	DCF	EER	DCF
C1	2.67	0.533	1.77	0.356	2.27	0.327	1.77	0.330	**1.65**	**0.307**	**1.58**	0.346
C2	4.34	0.624	**3.09**	0.592	4.06	0.554	3.48	0.573	3.57	**0.549**	**2.76**	0.592
C3	4.06	**0.553**	**3.00**	0.563	3.71	0.581	5.36	0.620	4.84	0.589	**2.97**	0.557
C4	3.65	0.635	2.85	0.503	3.48	0.471	**2.61**	0.465	2.85	**0.450**	**2.48**	0.503
C5	3.55	0.517	**2.59**	**0.487**	4.18	0.565	4.05	0.612	3.84	0.553	**2.59**	**0.487**
C6	7.04	0.863	**5.43**	**0.787**	6.36	0.819	6.36	0.855	6.19	0.839	**5.39**	0.788
C7	8.16	0.958	8.06	0.864	8.46	**0.763**	7.62	0.853	**7.34**	0.825	**7.16**	0.857
C8	3.11	**0.495**	**2.51**	0.518	2.99	0.516	2.87	0.548	2.73	0.528	**2.40**	0.514
C9	2.13	0.428	2.17	0.390	1.96	**0.325**	2.17	0.335	**1.82**	0.326	**1.61**	0.364
AVG	4.11	0.611	**3.04**	0.553	3.87	0.536	3.60	0.557	3.56	**0.532**	**2.80**	0.551

TABLE I
EQUAL ERROR RATE (EER) AND DETECTION COST FUNCTION (DCF) VALUES OBTAINED USING JFA, PLDA, CDS, KPLS, OSS-KPLS, AND FUSION (OSS-KPLS+PLDA) CLASSIFIERS ON THE NIST SRE 2010 EXTENDED-CORE DATA SET.

the inclusion of OSS-KPLS scores in fusion substantially (by 10.7%) increases the performance in condition 2. On the other hand, no improvement is seen on condition 5. This is likely due to substantially narrower telephone speech bandwidth, for which little additional information is available beyond that already used by PLDA. Overall, the average EER improvement across all conditions is 8.4% (this number is computed in the same manner as average EER and average DCF in Table 1), which suggests that generative and discriminative models indeed capture complementary speaker characteristics.

Note that we have used diagonal covariance GMMs for our experiments. Use of full-covariance GMMs have been shown to improve the distinction capabilities of the supervectors and i-vectors [29]. Use of this will push the performance of all the systems used here.

D. *Effect of Noise*

In order to test the noise sensitivity of the OSS-KPLS system, babble noise of various levels were added to all test utterances and the performances of the individual systems were evaluated. The results for Condition 2 of the SRE 2010 extended-core is shown in Figure 8. It can be seen that the additive noise deteriorates the performance of all the systems approximately evenly and the results described above hold also in noisy condition; that is, OSS-KPLS is still the best system in terms of DCF and is the second best in terms of EER.

VI. CONCLUSIONS

In this paper, we have used kernel PLS technique to produce a symmetric one-shot similarity framework for speaker verification. The developed framework was compared against

	Target Trials	Non-Target Trials	Total Trials	Relative EER Improvement
C1	4304	795995	800299	10.7%
C2	15084	2789534	2804618	10.7%
C3	3989	637850	641839	1.0%
C4	3637	756775	760412	13.0%
C5	7169	408950	416119	0.0%
C6	4137	461438	465575	0.7%
C7	359	82551	82910	11.2%
C8	3821	404848	408669	4.4%
C9	290	70500	70790	25.8%
AVG	—	—	—	8.4%

TABLE II
THE BREAKDOWN OF TRIALS IN NIST SRE 2010 EXTENDED-CORE DATA SET AND THE RELATIVE EER IMPROVEMENT OBTAINED PER CONDITION. THE EER IMPROVEMENT SHOWN CORRESPONDS TO USE OF OSS-KPLS+PLDA ("FUSION") SYSTEM VERSUS PLAIN PLDA SYSTEM.

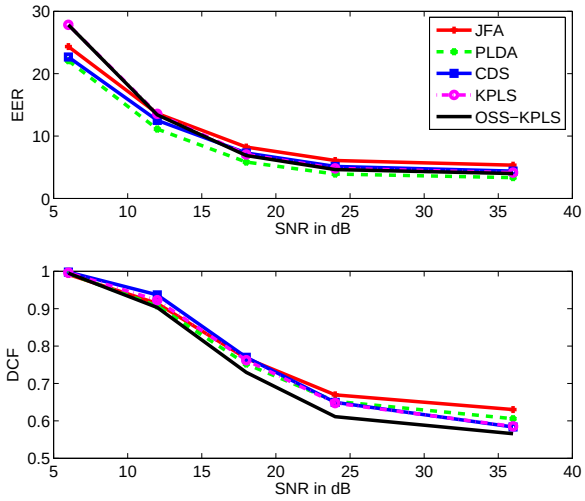

Fig. 8. [color] Sensitivity of JFA, PLDA, CDS, KPLS, and OSS-KPLS to additive babble noise on the Condition 2 of SRE 2010 extended core dataset.

several state-of-the-art systems on the NIST SRE 2010 data set. The OSS-KPLS was the best discriminative technique among those tested, and its combination with the best generative technique (PLDA) via score fusion results in 8.4% performance improvement (relative) in terms of EER, which shows the potential of using complementary information provided by generative and discriminative systems. Performance indications on noisy data are consistent with the same in clean case. Further research is planned on improving fusion strategy to target metrics other than EER and on using larger data sets to increase accuracy of comparisons of different ASV methods.

ACKNOWLEDGMENT

This research was partially funded by the Office of the Director of National Intelligence (ODNI) and Intelligence Advanced Research Projects Activity (IARPA) through the Army Research Laboratory (ARL). All statements of fact, opinion, or conclusions contained herein are those of the authors and should not be construed as representing the official views or policies of ODNI, the IARPA, or the U. S. Government.

We also acknowledge NSF award 0403313 and NVIDIA support for the Chimera cluster at the CUDA Center of Excellence at UMIACS.

REFERENCES

[1] F. Bimbot, J. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-García, D. Petrovska-Delacrétaz, and D. Reynolds, "A tutorial on text-independent speaker verification," *EURASIP Journal on Applied Signal Processing*, vol. 4, pp. 430–451, 2004.

[2] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker verification using adapted Gaussian mixture models," Digital Signal Processing, vol. 10, pp. 19–41, 2000.

[3] L. Burget, P. Matejka, P. Schwarz, O. Glembek, and J. Cernocky, "Analysis of feature extraction and channel compensation in a GMM speaker recognition system," IEEE Transactions on Audio, Speech, and Language Processing, vol. 15(7), pp. 1979–1986, 2007.

[4] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," IEEE Transactions on Audio, Speech, and Language Processing, vol. 16(5), pp. 980–988, 2008.

[5] N. Dehak, R. Dehak, P. Kenny, N. Brummer, P. Ouellet, and P. Dumouchel, "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification," Proc. INTERSPEECH 2009, Brighton, U.K., September 2009, pp. 1559–1562.

[6] P. Kenny, "Bayesian speaker verification with heavy tailed prior," Proc. Odyssey 2010: The Speaker and Language Recognition Workshop, Brno, Czech Republic, June 2010.

[7] W. Campbell, J. Campbell, D. Reynolds, E. Singer, and P. Torres-Carrasquillo, "Support vector machines for speaker and language recognition," Computer Speech and Language, vol. 20, pp. 210–229, 2006.

[8] M. Senoussaoui, P. Kenny, N. Dehak, and P. Dumouchel, "An i-vector extractor suitable for speaker recognition with both microphone and telephone speech," Proc. Odyssey 2010: The Speaker and Language Recognition Workshop, Brno, Czech Republic, June 2010.

[9] L. Burget, S. Cumani, O. Glembak, P. Matejka, and N. Brummer, "Discriminatively trained probabilistic linear discriminant analysis for speaker verification," Proc. IEEE ICASSP 2011, Prague, Czech Republic, May 2011, pp. 4832–4835.

[10] L. Wolf, T. Hassner, and Y. Taigman, "The one-shot similarity kernel," Proc. IEEE ICCV 2009, Kyoto, Japan, September 2009, pp. 897–902.

[11] L. Wolf, T. Hassner, and Y. Taigman, "Descriptor-based methods in the wild," Proc. Faces in Real-Life Images Workshop (in assoc. with ECCV 2008), Marseille, France, October 2008.

[12] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer, and P. A. Torres-Carrasquillo, "Support vector machines for speaker and language recognition," Computer Speech and Language, vol. 20, pp. 210–229, 2006.

[13] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Speaker and session variability in GMM-based speaker verification," IEEE Transactions on Audio, Speech, and Language Processing, vol. 15(4), pp. 1448–1460, 2007.

[14] D. Garcia-Romero and C. Espy-Wilson, "Joint factor analysis for speaker recognition reinterpreted as signal coding using overcomplete dictionaries," Proc. Odyssey 2010: The Speaker and Language Recognition Workshop, Brno, Czech Republic, June 2010.

[15] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," IEEE Transactions on Audio, Speech, and Language Processing, vol. 19(4), pp. 788–798, 2011.

[16] H. Wold, "Estimation of principal components and related models by iterative least squares," in *Multivariate Analysis*, ed. by P. R. Krishnaiah, Academic Press, New York, NY, pp. 391–420, 1966.

[17] S. Wold, E. Johansson, and M. Cocch, "PLS – Partial Least Squares projections to latent structures," in *Three-Dimensional Quantitative Structure Activity Relationships, Volume 1: Theory, Methods, and Applications*, ed. by H. Kubinyi, ESCOM Science Publishers, Leiden, The Netherlands, pp. 523–548, 1993.

[18] S. Wold, A. Ruhe, H. Wold, and W. J. Dunn, "The collinearity problem in linear regression. the partial least squares PLS approach to generalized inverses," SIAM Journal on Scientific and Statistical Computing, vol. 5(3), pp. 735–743, 1984.

[19] P. Garthwaite, "An interpretation of partial least squares," Journal of the American Statistical Association, vol. 89, pp. 122–127, 1994.

[20] W. Schwartz, A. Kembhavi, D. Harwood, and L. Davis, "Human detection using partial least squares analysis," Proc. IEEE ICCV 2009, Kyoto, Japan, September 2009, pp. 24–31.
[21] W. Schwartz, H. Guo, and L. Davis, "A robust and scalable approach to face identification," Proc. ECCV 2010, Heraklion, Crete, Greece, September 2010, vol. 6, pp. 476–489.
[22] B. Srinivasan, D. N. Zotkin, and R. Duraiswami, "A partial least squares framework for speaker recognition," Proc. IEEE ICASSP 2011, Prague, Czech Republic, May 2011, pp. 5276–5279.
[23] R. Rosipal and N. Kramer, "Overview and recent advances in partial least squares," in *Subspace, Latent Structure and Feature Selection*, ed. by C. Saunders, M. Grobelnik, S. Gunn, and J. Shawe-Taylor, Springer-Verlag, Berlin, Germany, 2006, pp. 34–51.
[24] R. Rosipal and L. Trejo, "Kernel partial least squares regression in reproducing kernel Hilbert space," Journal of Machine Learning Research, vol. 2, pp. 97–123, 2002.
[25] C. Bishop, "Pattern Recognition and Machine Learning," Springer, New York, NY, 2006.
[26] D. Reynolds, W. Campbell, T. Gleason, C. Quillen, D. Sturim, P. Torres-Carrasquillo, and A. Adami, "The 2004 MIT Lincoln Laboratory Speaker Recognition System," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005, pp. 177 – 180.
[27] N. Dehak, R. Dehak, J. Glass, D. Reynolds, and P. Kenny, "Cosine similarity scoring without score normalization techniques," Proc. Odyssey 2010: The Speaker and Language Recognition Workshop, Brno, Czech Republic, June 2010.
[28] D. Garcia-Romero and C. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *12th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2011.
[29] P. Matejka, O. Glembek, F. Castaldo, M. Alam, O. Plchot, P. Kenny, L. Burget, and J. Cernocky, "Full-covariance ubm and heavy-tailed plda in i-vector speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 4828 – 4831.