A PARTIAL LEAST SQUARES FRAMEWORK FOR SPEAKER RECOGNITION

Balaji Vasan Srinivasan, Dmitry N. Zotkin, Ramani Duraiswami
Perceptual Interfaces and Reality Laboratory, Department of Computer Science & UMIACS,
University of Maryland, College Park, MD, USA.
{balajiv,dz,ramani}@umiacs.umd.edu

ABSTRACT

Modern approaches to speaker recognition (verification) operate in a space of "supervectors" created via concatenation of the mean vectors of a Gaussian mixture model (GMM) adapted from a universal background model (UBM). In this space, a number of approaches to model inter-class separability and nuisance attribute variability have been proposed. We develop a method for modeling the variability associated with each class (speaker) by using partial-least-squares – a latent variable modeling technique, which isolates the most informative subspace for each speaker. The method is tested on NIST SRE 2008 data and provides promising results. The method is shown to be noise-robust and to be able to efficiently learn the subspace corresponding to a speaker on training data consisting of multiple utterances.

Index Terms— Partial least squares, speaker recognition, latent vector, GMM supervectors

1. INTRODUCTION

Speaker recognition [1] deals with the task of verifying a speaker's claimed identity from a sample utterance based on a number of training utterances for which the speaker is known. Apart from carrying the speaker-specific characteristics, the speech data also encapsulates phonemic content, channel variability, and session variability. It is also often subject to noise and reverberation, making the problem of speaker recognition challenging. Over the past decade, the field has made substantial progress in addressing these issues. Variability in the phonemic content is removed by posing the problem of recognition over a collection of data spanning several utterances. The commonly used feature space is the set of mel-cepstral coefficients along with their deltas and double-deltas. More robust feature spaces have been considered but are not yet adopted as adding them to existing architectures might lead to excessive feature space dimensionality and therefore high computational load.

State-of-the-art speaker recognition systems use a Gaussian mixture model (GMM) to represent each speaker. To account for limited training data available, the problem is cast into a framework in which differences from a universal background model (UBM) are used to adapt speaker-specific GMMs [3]. More recently, the problem has been transformed into a task of learning the between-class separability in a supervector setting [2]. Substantial progress has also been made in rejecting channel/session variability in the supervector setting via joint factor analysis (JFA) technique [4] [5], nuisance attribute projection (NAP) [6] and i-vectors [7].

The objective in the supervector space is to discriminate between a speaker and imposters by accounting for the speaker variability while ignoring nuisance information. Commonly, only a few (often one) speech samples from a very large speech database belong to the target speaker, which necessitates use of the method capable of learning from a few samples in a very high dimensional space. Different approaches such as GMM likelihood ratios [3] and support vector machines [6] have been explored previously.

Several learning techniques have been used to tackle similar scenarios in other domains. One approach that was originally developed in chemometrics is partial least squares [8] and its kernelized version [9]. *Partial least squares (PLS)* [8] techniques are a wide class of methods for modeling relations between sets of observed variables by means of latent variables. These methods include regression, classification, and dimensionality reduction. The underlying assumption in PLS is that the observed data is generated by a system/process that is driven by a small number of latent variables.

PLS is often used as a dimensionality reduction technique and therefore draws comparisons with principal component analysis (PCA) and linear discriminant analysis (LDA). PCA is an unsupervised dimensionality reduction algorithm, which results in a single projection irrespective of the task. LDA is a supervised dimensionality reduction technique that results in different subspaces for different tasks. PLS is similar to LDA in this sense. But unlike LDA, PLS is not limited by a projection space dimension of $c-1$ (where c is the number of classes). A detailed comparison of PLS, PCA, and LDA is presented in [9]. PLS based techniques have been very successful in the fields of chemometrics and bioinformatics. Recently, PLS has been adapted to image processing and computer vision problems (such as human detection and face recognition) [10] and was shown to greatly improve the performance, especially for 2-class problems.

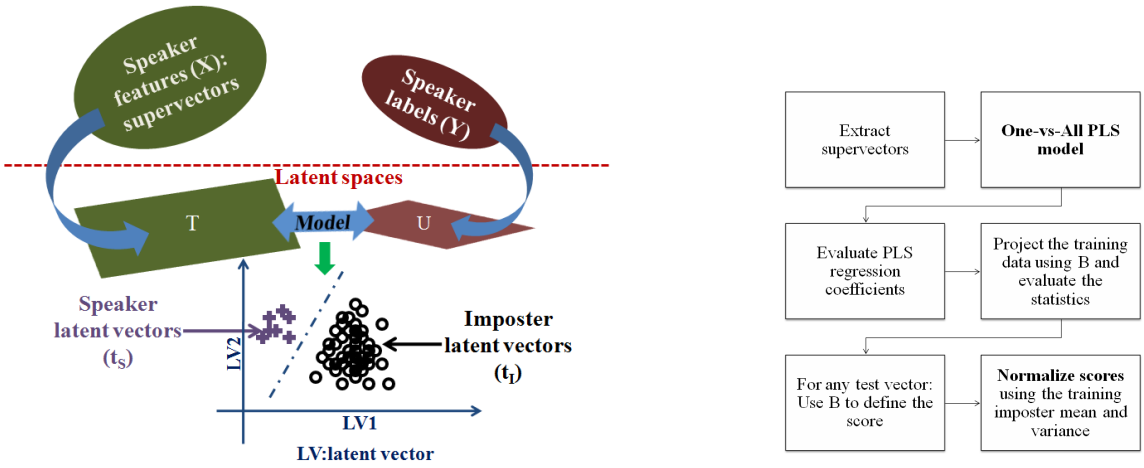
Motivated by this, we explore here a partial least squares based framework for speaker modeling and recognition in the supervector space. Extension to handling nuisance parameters is a subject of future work. This paper is organized as follows. In Section 2, we introduce the PLS framework and its adaptation to speaker recognition. We describe our experiments and discuss results in Section 3 and conclude the paper with future directions in Section 4.

2. PARTIAL LEAST SQUARES

Denote a d-dimensional supervector by x and the corresponding speaker label by y. Essentially, x is the feature (super)vector (input variable) and y is the speaker identity (output variable that has to be learned). Assume that the total number of speakers is N and denote the $N \times d$ matrix of supervectors 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 aims

This result was partially funded by the Office of the Director of National Intelligence (ODNI), 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 IARPA, the ODNI, or the U.S. Government.

We thank Drs. Larry Davis, William Schwartz, and Aniruddha Kembhavi for providing us with an introduction to partial least squares.



(a) PLS based subspaces

(b) PLS based speaker recognition

Fig. 1. *Schematic of the proposed Partial Least Squares (PLS) technique for speaker recognition.*

at modeling the relationship between x and y using projection into latent spaces. While a detailed analysis of PLS can be found in [8], we provide a brief overview here. PLS decomposes X and Y as

$$X = TP^T + E, \quad (1)$$
$$Y = UQ^T + F, \quad (2)$$

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* [8] that constructs a set of weight vectors $W = \{w_1, w_2, \ldots, w_p\}$ such that

$$\max[\text{cov}(t_i, u_i)]^2 = \max_{|w_i|=1} [\text{cov}(Xw_i, Y)]^2, \quad (3)$$

where t_i and u_i are the i^{th} columns of T and U respectively and $\text{cov}(t_i, u_i)$ indicates the sample covariance between latent vectors t_i and u_i. Maximizing the covariance in the latent vector space is equivalent to maximizing discrimination in the same space; in other words, for a particular speaker, PLS learns a subspace in which the speaker latent vectors t_S are well separated from the imposter latent vector t_I. This is illustrated in Fig. 1(a). Thus, PLS learns a unique latent space for each speaker. 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:

$$X \leftarrow X - t_i p_i^T; \qquad Y \leftarrow Y - u_i q_i^T. \quad (4)$$

This step removes any information captured by t_i and u_i from X and Y. The process is repeated till a sufficient number of latent vectors is obtained. This number is determined via standard cross-validation techniques [10].

It has been shown [8] that the NIPALS algorithm is equivalent to iteratively finding the dominant eigenvectors of the problem

$$[X^T y y^T X] w_i = \lambda w. \quad (5)$$

The weight matrix W can be used for dimensionality reduction, and the resulting projection can be used with any standard classifier to model a target speaker. However, it was observed that the performance was not as good as the alternative presented below. We instead use PLS in a regression framework that implicitly utilizes the PLS weights W obtained from the NIPALS algorithm.

PLS Regression: Substituting the w from Eq. (3) in Eq. (1), we get

$$XW = TP^T W + E \Rightarrow T = XW(P^T W)^{-1}. \quad (6)$$

Now, U can be written in terms of T [8] as $U = TD + H$, where D is a diagonal matrix and H is the residue. Eq. (2) now becomes

$$Y = TDQ^T + HQ^T + F = XW(P^T W)^{-1} DQ^T + \bar{F}, \quad (7)$$

and we get the PLS regression:

$$Y = XB + G; \qquad B = W(P^T W)^{-1} DQ^T, \quad (8)$$

where B is the set of PLS regression coefficients. This regression framework directly provides the way to compute the matching score for seamless speaker discrimination, eliminating the need for a separate classifier. It also utilizes the latent structure learnt by NIPALS algorithm better – the regression coefficients weight the supervector centers that discriminate the current speaker against imposters more than other centers. Hence, the regression coefficients are unique to each speaker. Note that, although PLS is used widely a dimensionality reduction technique, we use a PLS-based regression technique, and the dimensionality reduction is not used explicitly for speaker modeling.

In our work here, we first train the GMM UBM using a large amount of data. Then, we create a specific GMM for each speaker in the database by adapting the UBM using the speaker (training) utterances. Then, the speaker supervector [2] is created by concatenating the means of the speaker GMM. Note that the whole training utterance is represented by one point in the supervector space. We then learn the PLS regression model using a one-vs-all scheme. Finally, we perform the scoring and normalize output scores in a T-norm sense [1] using a large number of non-target speakers (imposters). These steps are summarized in Fig. 1(b).

The beneficial properties of the proposed PLS framework for speaker recognition can be summarized as follows:

1. It is a discriminative technique (like SVM); hence, the performance should improve as the amount of speaker training data increases.
2. SVM learns a separating hyperplane between speaker and imposter supervectors, whereas PLS learns discriminative projection that maximizes the covariance of supervectors and speaker labels in the projected space. PLS regression weights the supervectors based on these projections to score each utterance.

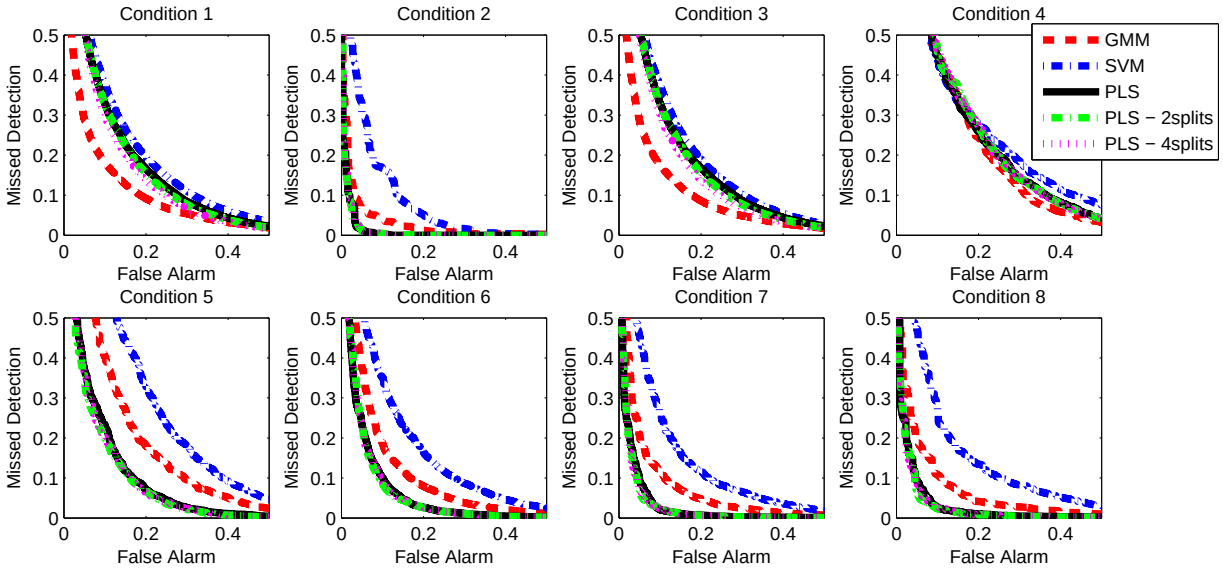


Fig. 2. *Performance of PLS against SVM and GMM baseline systems on the NIST 2008 core set.*

3. The computational cost of PLS is O(Nd) against O(N^2d) for SVM, where d is the supervector dimension and N is the number of supervectors.
4. The PLS technique used here is linear. Non-linear PLS can potentially be done by using the kernel-trick [9]. However, this direction was not pursued here.

Accelerating PLS: Despite the success of PLS, its O(Nd) computational cost does not scale well for large sample sizes and large number of features. We have already addressed this scalability issue via use of graphical processors [11] and achieved $\sim$ 30X speedups against standard CPU-based implementations.

3. EXPERIMENTS

We performed experimental evaluation of the proposed method on the *core (short2-short3) test set* and *8conv-short3 test set* in the NIST SRE 2008[1] evaluation dataset. The dataset is grouped into 8 trial conditions: C1: interview speech (IS) both for training and testing (BTT); C2: IS, using the same microphone for training and testing; C3: IS, using different microphones for training and testing; C4: IS for training, telephone speech (TS) for testing; C5: TS for training, noninterview microphone speech for testing; C6: TS BTT; C7: English TS BTT; and C8: English TS BTT by native English speakers. For all experiments, we used 19 MFCC features along with their deltas.

We compared performance of the PLS-based approach against the GMM/UBM based system [3] and GMM-supervector-kernel based SVM [6]. The libSVM package was used for our SVM runs. The GMM/UBM code was developed in house and validated against results reported in NIST SRE 2006. Note that since nuisance attributes are not being modeled, the GMM/UBM EER is relatively high compared to SRE 2008 results (where nuisance corrections based on JFA were applied).

Supervector dimensions: It was observed that 4096-center GMM gave the best performance with core set, while 2048-center model was best for 8conv-short3 set. With SVM, these numbers are 1024 and 512, respectively; and with PLS, 512 and 256, respectively. While a larger number of GMM centers leads to severe over-fitting to the background data (which helps GMM capture background characteristics better but does not provide room for supervector based discrimination), very few centers lead to severe under-fitting and the resulting GMM models do not generalize well to test conditions. This is the reason PLS works best with moderate number of GMM centers, which also reduces the computational load by an order of magnitude.

[1] www.itl.nist.gov/iad/mig/tests/sre/2008/

	GMM	SVM	PLS	PLS 2-splits	PLS 4-splits
C1	**13.84**	19.73	18.43	18.02	17.13
C2	6.15	12.73	**3.38**	**3.38**	3.64
C3	**13.54**	19.49	18.39	18.08	17.13
C4	**21.31**	23.63	22.48	22.79	22.79
C5	19.03	24.34	13.90	**13.64**	13.90
C6	13.48	18.44	10.13	**9.60**	9.77
C7	10.69	15.41	6.52	**5.51**	5.92
C8	10.42	16.34	6.61	**5.47**	5.98

Table 1. *Equal-error-rates obtained with PLS (with/without data splitting), SVM, and GMM across various condition for the NIST 2008 core set. Note: there is no nuisance attribute compensation.*

Single training utterance: In the core set, there is only one training utterance per speaker. There are 1270 male and 1993 female speakers (3263 total) and 98776 trials. Each trial belongs to one or more of 8 conditions outlined above. The DET curves for all 8 conditions are shown in Fig. 2 and the equal error rates are listed in Table 1.

Note that having only one training utterance per speaker does not provide enough data for discriminative approaches like SVM and PLS; the GMM/UBM system is likely to perform better in this case. In spite of that, the PLS framework outperforms GMM/UBM in conditions $2, 5-8$ (5 out of 8) and is comparable for condition 4.

As mentioned, the PLS framework makes use of intra-speaker variability. Therefore, the performance is expected to improve if more supervectors belonging to the target speaker are available. Ideally, speaker utterances should be recorded across various nuisance conditions, which will enable PLS to truly capture speaker-related information and reject channel-related one. Alternatively, we explored simple mechanism of splitting the training data to create multiple supervectors per utterance. Note that this does not guarantee the availability of training vectors across nuisance conditions. However, it was observed that the PLS performance indeed improved significantly with 2-way split of the training data, although there is no further improvement with 4-way split. The DET curves for these

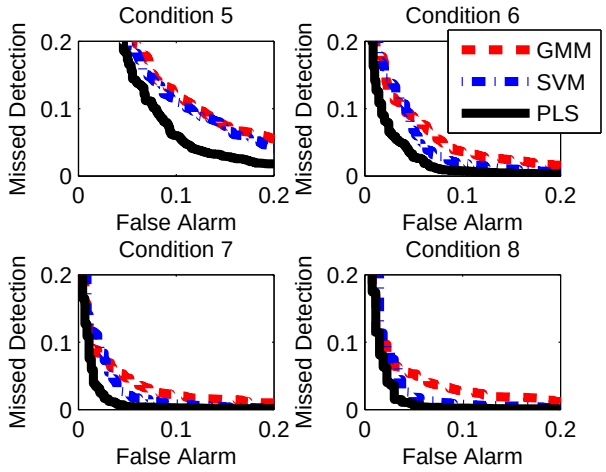


Fig. 3. *Performance of PLS against SVM and GMM baselines on 8conv-short3 set.*

	GMM	SVM	PLS
C5	10.98	10.52	**8.63**
C6	6.66	5.62	**4.30**
C7	4.82	3.94	**2.41**
C8	5.27	3.76	**3.02**

Table 2. *Equal-error-rates obtained with PLS, SVM and GMM across various condition for the 8conv-short3 set. Note: there is no nuisance attribute compensation.*

cases are also shown in Fig. 2.

Multiple training utterances: The 8conv-short3 set consists of 8 training utterances per speaker. There are 240 male and 395 female speakers (635 total) and 16570 trials. There are no trials corresponding to conditions C1 through C4, as all training data is telephone speech.

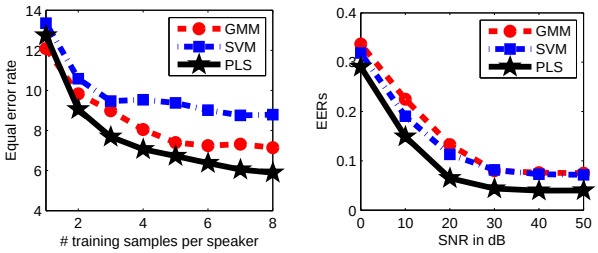
We compared the performance of PLS-based speaker recognition against GMM/UBM and SVM baseline systems, and the DET curves are shown in Fig. 3 with the corresponding equal error rates in Table 2. It can be seen that PLS outperforms other systems in all conditions.

Effect of training sample size per speaker: Because the 8conv-short3 set contains 8 training utterances (and therefore 8 supervectors) per speaker, it also provides a good framework for evaluation of the recognizer performance dependence on the amount of training data. We have trained each of our recognition systems with $1, 2, \ldots 8$ utterances per speaker; the corresponding results are shown in Fig. 4(a). It can be seen that all 3 system show improved performance with the increase in the number of training speaker supervectors. However, unlike GMM and SVM, PLS performance does not saturate but instead continues to decrease. This is because PLS relies on the intra-class variance to determine the projection; therefore, having more training data implies better intra-speaker variance estimate and better performance.

Noise robustness of PLS: To evaluate PLS robustness to noise, we added Gaussian noise to test samples in the 8conv-short3 set (male only) and evaluated the performance of all three recognition systems. The results are shown in Fig. 4(b). It can be seen that additive noise decreases the performance for all systems, but PLS still outperforms both SVM and GMM.

4. CONCLUSION

We have applied a PLS latent vector framework to the GMM supervectors in speaker recognition and have shown that it outperforms the baseline GMM/UBM and SVM systems on NIST 2008 SRE dataset in most conditions. The PLS system we currently have does not account for nuisance parameters (channel and session variability); therefore, our baseline systems also did not include nuisance parameter elimination for fair comparison. The PLS approach proposed here is currently being extended to address channel/session variability issues.



(a) *Number of training points* (b) *Noise robustness*

Fig. 4. *EER for PLS, SVM, and GMM/UBM systems in various conditions.*

5. REFERENCES

[1] F. Bimbot et al. (2004). "A tutorial on text-independent speaker verification", EURASIP Journal on Applied Signal Processing, vol. 2004, pp. 430-451.

[2] T. Kinnunen and H. Li (2010). "An overview of text-independent speaker recognition: From features to supervectors", Speech Communication, vol. 52, no. 1, pp. 12-40.

[3] D. A. Reynolds, T. Quatieri, and R. Dunn (2000). "Speaker verification using adapted Gaussian mixture models", in Digital Signal Processing.

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

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

[6] W. M. Campbell, D. E. Sturim, D. A. Reynolds, and A. Solomonoff (2006). "SVM based speaker verification using a GMM supervector kernel and NAP variability compensation", Proc. IEEE ICASSP 2006, vol. 1. pp. 97-100.

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

[8] R. Rosipal and N. Krámer (2006). "Overview and recent advances in partial least squares", in Subspace, Latent Structure, and Feature Selection Techniques: Lecture Notes in Computer Science, pp. 34-51, Springer.

[9] R. Rosipal and L. J. Trejo (2003). "Kernel PLS-SVC for linear and nonlinear classification", Proc. ICML 2003, Washington, DC, pp. 640-647.

[10] W. R. Schwartz, A. Kembhavi, D. Harwood, and L. S. Davis (2009). "Human detection using partial least squares analysis", Proc. IEEE ICCV 2009, Kyoto, Japan.

[11] B. V. Srinivasan, W. R. Schwartz, R. Duraiswami, and L. S. Davis (2010). "Partial least squares on graphical processor for efficient pattern recognition", CS-TR-4968, Department of Computer Science, University of Maryland, College Park. `http://hdl.handle.net/1903/10975`