Classification Based on Symmetric Maximized Minimal Distance in Subspace (SMMS)

Wende Zhang and Tsuhan Chen
Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA
{wendez, tsuhan}@andrew.cmu.edu

Abstract

We introduce a new classification algorithm based on the concept of Symmetric Maximized Minimal distance in Subspace (SMMS). Given the training data of authentic samples and imposter samples in the feature space, SMMS tries to identify a subspace in which all the authentic samples are close to each other and all the imposter samples are far away from the authentic samples. The optimality of the subspace is determined by maximizing the minimal distance between the authentic samples and the imposter samples in the subspace. We present a procedure to achieve such optimality and to identify the decision boundary. The verification procedure is simple since we only need to project the test sample to the subspace and compare it against the decision boundary. Using face authentication as an example, we show that the proposed algorithm outperforms several other algorithms based on support vector machines (SVM).

1. Introduction

With the evolving biometric applications, more and more biometric authentication algorithms will be applied to hand-held devices. Due to the size and power constraints of these devices, the computational complexity of the verification procedure will be the main consideration for choosing the right algorithms.

The subspace approach is suitable for biometric authentication applications because of its simplicity in the verification procedure. We only need to project the test sample onto the subspace. Most subspace matching algorithms determine the subspace by optimizing certain global measures such as the mean squared error (MSE) or the variance ratio. For example, the principle component analysis (PCA) [1] subspace is determined by minimizing the reconstruction errors of samples in the subspace using the MSE criterion. The Fisher's linear discriminant analysis (LDA) [2] projection is defined by maximizing the ratio between the intra-class variance and the inter-class variance in the subspace. The applications of these algorithms can be found in [3][4][5][6][7][8][9] for biometric recognition and authentication. These algorithms based on global optimization provide good performance when the data distribution is close to Gaussian.

Other matching algorithms determine their decision boundaries by supporting patterns [10]. Supporting patterns are typically training samples near the decision boundary. Support vector machines (SVM) [11][12][13][14][15] optimize the supporting patterns by maximizing the gap between the authentic samples and the imposter samples regardless of the data distribution. This "maxmin" approach outperforms the global optimization approach, when the data distribution is not Gaussian as shown by several face authentication and recognition applications [16][17][18].

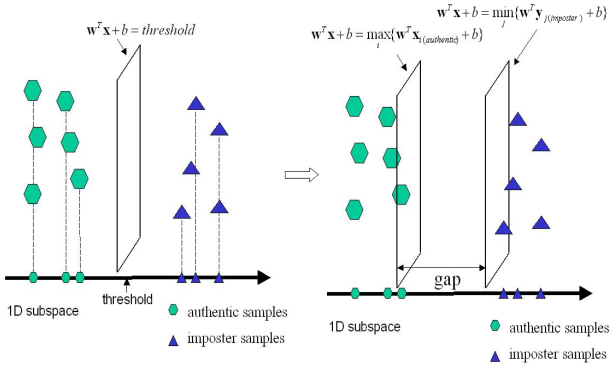
$\mathbf{w}^T\mathbf{x}+b = threshold$
1D subspace
threshold
authentic samples
imposter samples
$\mathbf{w}^T\mathbf{x}+b = \min_j\{\mathbf{w}^T\mathbf{y}_{j(imposter)}+b\}$
$\mathbf{w}^T\mathbf{x}+b = \max_i\{\mathbf{w}^T\mathbf{x}_{i(authentic)}+b\}$
gap
1D subspace
authentic samples
imposter samples

Figure 1. Linear SVM

A linear SVM [14] maximizes the separation gap between the authentic samples ($\mathbf{x}_{i(authentic)}$, hexagon pattern) and the imposter samples ($\mathbf{y}_{j(imposter)}$, triangle pattern) as shown in Figure 1. Alternatively, the optimal parameters **w** and *b* are optimized by

$$\max_{\mathbf{w},b}\{\min_j(\mathbf{w}^T\mathbf{y}_{j(imposter)}+b)-\max_i(\mathbf{w}^T\mathbf{x}_{i(authentic)}+b)\},$$
$$\mathbf{w}^T\mathbf{w}=1$$

The computation requirement for verification is low since only one inner product between the input vector **x** and the projection vector **w** is needed. We use the following criterion to decide whether the input vector **x** is authentic or not.

$$f(x) = \mathbf{w}^T\mathbf{x} + b \le threshold\,,\ \ \mathbf{x} \to \text{authentic sample}$$

This is equivalent to projecting **x** onto a one-dimensional subspace and comparing the result with the threshold. The threshold value can be chosen from the Receiver Operating Characteristic (ROC) curve to meet the false accept rate (FAR) vs. false reject rate (FRR) requirement in the design specifications.

A linear SVM may fail when the authentic samples (hexagon patterns) are surrounded by the imposter samples as seen in Figure 2. It is not possible to find a linear plane to separate authentic samples from all imposter samples in this case. One way to solve this problem is to seek a separation plane by mapping the training samples to a higher dimensional space [14]. After mapping, the samples may be well separated, but there are two potential problems in this approach. First, the advantage of simplicity is lost. When the samples are mapped to a higher dimensional space, we have to keep many support vectors to describe the separation boundary instead of only one projection vector **w**. In addition, the over fitting problem may be encountered, that is, the samples may be too few to train SVM in a high dimensional space.

In this paper, we propose a new approach called Symmetric Maximized Minimal distance in Subspace (SMMS) to perform classification. By allowing the imposter samples to surround the authentic samples, hence "symmetric", this algorithm tries to let the authentic samples cluster together and the imposter samples be far way from the authentic samples in the subspace. For example, in the case of using a one-dimensional subspace, the imposter samples are on both sides of the authentic samples and are separated 2 hyper-planes shown in Figure 2. The optimality of the subspace is determined by maximizing the minimal distance between the authentic samples and the imposter samples in the subspace. The samples that are not linearly separable by one plane can now be well separated by SMMS in Figure 2.

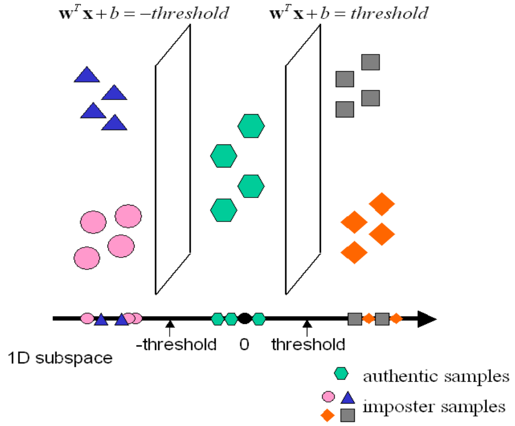
$\mathbf{w}^T\mathbf{x}+b=-threshold$
$\mathbf{w}^T\mathbf{x}+b=threshold$
1D subspace
-threshold
0
threshold
authentic samples
imposter samples

Figure 2. Samples separable by 2 hyper-planes

The paper is organized as follows. In the next section, we introduce the classification algorithm based on SMMS. In Section 3, we describe the face database used in the experiment and compare the authentication performance of our algorithm with SVM. The conclusion is given in Section 4.

2. Symmetric Maximized Minimal distance in Subspace (SMMS)

We will introduce the one-dimensional SMMS algorithm in details and then extend the algorithm to higher dimensions.

In the one-dimensional SMMS algorithm, we use the following criterion to decide whether the input vector **x** is authentic or not.

$$f(\mathbf{x}) = \left|\mathbf{w}^T\mathbf{x} + b\right| \le threshold \qquad (1)$$

To train the SMMS classifier, given M authentic samples, $\mathbf{x}_i$, and N imposter samples, $\mathbf{y}_j$, we try to maximize the minimal difference $f(\mathbf{x})$ between the authentic samples and the imposter samples as shown below:

$$\max_{\mathbf{w},b}\{\min_j\left|\mathbf{w}^T\mathbf{y}_j + b\right| - \max_i\left|\mathbf{w}^T\mathbf{x}_i + b\right|\}$$
$$j = 1, 2, \ldots, N;\ i = 1, 2, \ldots, M \qquad (2)$$

with the normalization constraint $\mathbf{w}^T\mathbf{w} + b^2 = 1$.

To simplify the formulation, we let $\mathbf{p}_i^T = \lfloor \mathbf{x}_i^T \quad 1 \rfloor$, $\mathbf{q}_j^T = \lfloor \mathbf{y}_j^T \quad 1 \rfloor$ and $\mathbf{a}^T = \lfloor \mathbf{w}^T \quad b \rfloor$. Then, we can rewrite (2) and normalization constraint as

$$\max_{\mathbf{a}}\{\min_j\left|\mathbf{a}^T\mathbf{q}_j\right| - \max_i\left|\mathbf{a}^T\mathbf{p}_i\right|\} \qquad (3)$$

$$\mathbf{a}^T\mathbf{a} = 1 \qquad (4)$$

Since we try to maximize the gap, ideally, the authentic samples should be all squeezed onto a plane, intuitively. We assume the authentic samples $\mathbf{p}_i$ can be fitted by one hyper plane as shown in Figure 3.

$$\mathbf{a}^T\mathbf{p} = 0 \qquad (5)$$

Typically, this is possible because M is less than the dimension of input data.

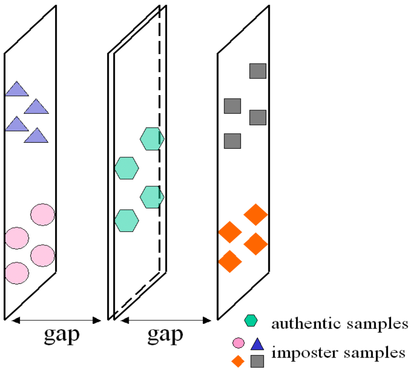

Figure 3. Modeling the authentic data

Therefore, we can simplify the optimization criterion (3) as follows:

$$\max_{\mathbf{a}}\{\min_{j}\left|\mathbf{a}^{T}\mathbf{q}_{j}\right|\} \tag{6}$$

along with the constraint (5).

To find **a** due to the constraint equation (5), we can limit the feasible solution region to the null space, $\mathbf{E}=[\mathbf{e}_{k+1,}\mathbf{e}_{k+2,}\ldots,\mathbf{e}_{d}]$, of the subspace spanned by the authentic samples, where k is the rank of the subspace of the authentic samples and d is the dimension of vector $\mathbf{q}_j$. The vector **a** can be expressed as follows.

$$\mathbf{a} = r_{k+1}\mathbf{e}_{k+1} + \ldots + r_{d}\mathbf{e}_{d} = \mathbf{Er}\,,\quad \mathbf{E}^{T}\mathbf{E}=\mathbf{I} \tag{7}$$

Now our optimization becomes solving the coefficient vector **r**. We can project the imposter sample $\mathbf{q}_j$ onto the subspace **E** to get the projected vector **q'**$_j$ as shown in Figure 4.

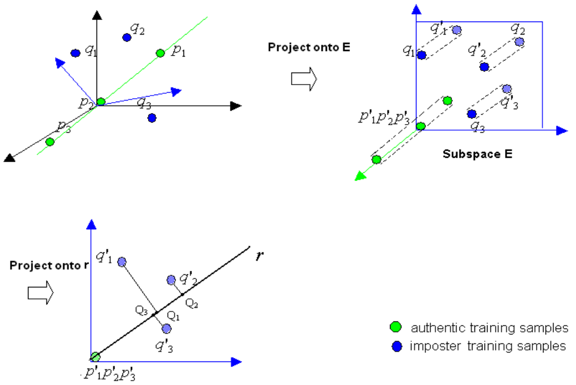

Figure 4. Projecting imposter samples into null space

Finding **r** is equivalent to finding a projection vector in **E** that optimizes (8)

$$\begin{aligned}&\max_{\mathbf{r}}\{\min_{j}\left|(\mathbf{Er})^{T}\mathbf{q}_{j}\right|\}=\max_{\mathbf{r}}\{\min_{j}\left|\mathbf{r}^{T}(\mathbf{E}^{T}\mathbf{q}_{j})\right|\}\\&=\max_{\mathbf{r}}\{\min_{j}\left|\mathbf{r}^{T}\mathbf{q'}_{j}\right|\}=\max_{\mathbf{r}}\{\min_{j}\left|\mathbf{r}^{T}\mathbf{q'}_{j}\right|^{2}\}\end{aligned} \tag{8}$$

with the following constraint equation $\mathbf{r}^{T}\mathbf{r}=\mathbf{a}^{T}\mathbf{a}=1$.

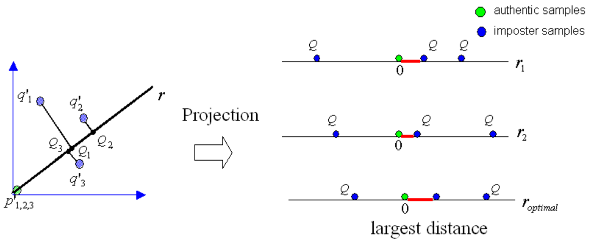

Figure 5. Maximizing the minimal distance to find the optimal projection basis

We approximate the minimization procedure as follows. We assign a weight s_j to each value $Q_j(\mathbf{r}) = \left|\mathbf{r}^{T}\mathbf{q'}_{j}\right|^{2}$. And we constrain the sum of all the weights to be 1.

$$\sum_{j} s_{j} = 1 \tag{9}$$

If we assign the non-zero weight to minimum value of Q_j and the zero weights to all the others, the minimum of $Q_j(\mathbf{r})$ will be equal to the weighted sum in equation (10).

$$\min_{j} Q_{j} \approx \sum_{j} s_{j}Q_{j} = m_{w} \tag{10}$$

The weights are adjusted iteratively to make the weighted mean close to the minimum value. The weights are increased for the small values of Q_j, while the other weights are decreased respectively. Now the optimization can be written as

$$J = \max_{\mathbf{r}}\{\sum_{j} s_{j}Q_{j}(\mathbf{r})\} = \max_{\mathbf{r}}\{\sum_{j} s_{j}\left|\mathbf{r}^{T}\mathbf{q'}_{j}\right|^{2}\} \tag{11}$$

Since both **r** and s_j are unknown, we try to solve them iteratively. We initialize s_j first. Given the updated s_j, we try to solve **r**. Then given current **r**, we try to update s_j.

The iterative procedure consists of four steps to obtain the optimal solution of **r**.

1. Initialize weights for s_j. Each s_j is set to the same value.

   $$s_{j} = 1/N\,,\qquad j=1,\ldots,N \tag{12}$$

2. Solve **r** by optimizing (11). The solution **r** is given by the eigenvector corresponding to the biggest eigenvalue of the following matrix

   $$\sum_{j} s_{j}\mathbf{q'}_{j}\,\mathbf{q'}_{j}^{\;T} \tag{13}$$

3. Update s_j with the projection vector **r**. We evaluate Q_j and compare it with the weighted mean m_w. We define the difference e_j as

   $$e_{j} = Q_{j} - m_{w} \tag{14}$$

   The difference e_j is used to determine how the weights are updated. If the value Q_j is smaller than the weighted mean m_w, e_j will be negative. In this case, Q_j is more likely to be closer to the origin on projection vector **r**; therefore, the corresponding weight will be increased. Otherwise, the weight remains unchanged.

   $$s'_{j_{(n+1)}} = \begin{cases} s_{j_{(n)}} + v(-e_{j})s_{j_{(n)}}\,, & e_{j} < 0 \\ s_{j_{(n)}}\,, & e_{j} \geq 0 \end{cases} \tag{15}$$

   where n is the index of iteration and v is a positive value to control the updating speed.

   The weights are normalized to satisfy the constraint of equation (9).

   $$s = \sum_{j} s'_{j_{(n+1)}} \tag{16}$$

   $$s_{j_{(n+1)}} = s'_{j_{(n+1)}} / s \tag{17}$$

4. Go back to *step 2* until the solution converges. If the solution converges, we choose **r** from the last step as the optimal solution for **r**. Otherwise, after the maximum number of iterations, we choose **r** that provides the minimum number of non-

positive terms e_j among all the iterations as the optimal solution for **r**.

We use the final projection vector $\mathbf{r}_{optimal}$ to determine the optimal projection vector $\mathbf{a}_{optimal} = \mathbf{E}\mathbf{r}_{optimal}$.

SMMS can be extended to higher dimensions. If we extend the subspace for verification to be *R*-dimensional, with the SMMS algorithm, the following criterion is used to decide whether **x** is authentic as shown in Figure 6.

$$f(\mathbf{x}) = \sum_{n=1}^{R} \left| \mathbf{l}_n{}^T \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} \right|^2 \le threshold \qquad (18)$$

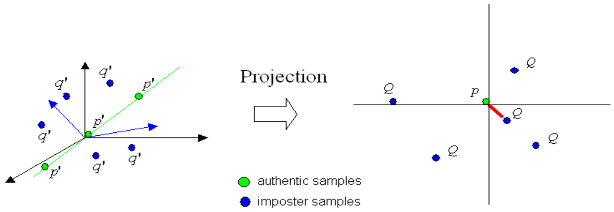

Figure 6. The optimal projection subspace in R=2 dimension

Here we try to maximize the minimal squared distance between the authentic samples and the imposter samples in a subspace $\mathbf{L} = [\mathbf{l}_1, \mathbf{l}_2, ..., \mathbf{l}_R]$ as follows.

$$\max_{\mathbf{L}}\{\min_{j} \sum_{n=1}^{R} \left| \mathbf{l}_n{}^T \begin{bmatrix} \mathbf{y}_j \\ 1 \end{bmatrix} \right|^2 - \max_{i} \sum_{n=1}^{R} \left| \mathbf{l}_n{}^T \begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix} \right|^2 \}$$

$$j = 1, 2, \ldots, N; \; i = 1, 2, \ldots, M \qquad (19)$$

with the normalization constraint (20).

$$\mathbf{l}_n{}^T \mathbf{l}_m{}^T = 1, \qquad m = n = 1, 2, ..., R$$

$$\mathbf{l}_n{}^T \mathbf{l}_m{}^T = 0, \qquad m \ne n \qquad (20)$$

The authentic samples are fitted to one hyper plane as before. Each imposter samples $\mathbf{q}_j$ is projected onto the null space **E** of the subspace spanned by the authentic samples to get the projected vector $\mathbf{q'}_j$. Instead of an optimal projection vector **r** in the one-dimensional subspace, we find an optimal projection subspace $\mathbf{H} = [\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_R]$ in the null space **E**.

The optimal $\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_R$ are determined by the following optimization

$$J = \max_{\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_R} \left\{ \min_j \left\{ \sum_{n=1}^{R} \left| \mathbf{r}_n{}^T \mathbf{q'}_j \right|^2 \right\} \right\} \qquad (21)$$

with the following constraints

$$\mathbf{r}_n{}^T \mathbf{r}_m{}^T = 1, \qquad m = n = 1, 2, ..., R$$

$$\mathbf{r}_n{}^T \mathbf{r}_m{}^T = 0, \qquad m \ne n \qquad (22)$$

We solve $\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_R$ by optimizing the (21) using the same procedure as before with the following exception. First, we redefine $Q_j = \sum_{n=1}^{R} \left| \mathbf{r}_n{}^T \mathbf{q'}_j \right|^2$.

In addition, we change the *Step 2* in the iterative optimization procedure. Since we try to solve $\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_R$ by optimizing (21), The solution $\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_R$ is given by the eigenvectors corresponding to the *R* biggest eigenvalues of the matrix in (13).

With $\mathbf{H}_{optimal} = [\mathbf{r}_{1_{optimal}}, \mathbf{r}_{2_{optimal}}, ..., \mathbf{r}_{R_{optimal}}]$, the optimal projection subspace is as follows

$$\mathbf{a}_{n_{optimal}} = \mathbf{E}\mathbf{r}_{n_{optimal}}, \quad n = 1, 2, ..., R \qquad (23)$$

3. Experiment

Our face database has 20 subjects. Each subject has 15 training images and 190 test images with different poses. Both the test and training images are of 64×64 grayscale images. In Figure 7, the sample images from six subjects in this database are shown in different rows. A lot of pose variations can be observed from this database. Thus, it is a very challenging database for face authentication.

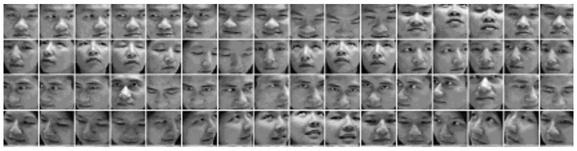

Figure 7. Samples from our pose database

We use all the training images (15*20) to construct a PCA subspace with the first 100 eigenvectors corresponding to the 100 biggest eigenvalues to perform dimension reduction for each training and test image.

EER	Linear SVM	SMMS -1D	SMMS -2D
Subject #01	10 %	20 %	**5.7** %
Subject #02	19 %	**11 %**	**16** %
Subject #03	7.4 %	**7.3 %**	**4.2** %
Subject #04	26 %	**15 %**	**15** %
Subject #08	9.5 %	13 %	**7.1** %
Subject #15	13 %	**10 %**	**11** %
Subject #16	28 %	**13 %**	**11** %
Subject #17	22 %	**6.1 %**	**4.2** %
Subject #18	19 %	**16 %**	**15** %
Subject #19	13 %	**4.1 %**	**4.3** %
Average of above 10 subjects	17 %	**12 %**	**9 %**
Average of all subjects	11.8 %	17 %	12.3 %

Table 1. EER on 100 PCA coefficients of pose database of 20 persons

Here we compare our proposed algorithms SMMS with one-dimensional subspace (SMMS-1D) and SMMS with two-dimensional subspace (SMMS-2D) with the linear SVM in our pose database as shown in Table 1. We show that the SMMS algorithms are better than linear SVM for some subjects in Table 1. When the linear SVM algorithm makes more errors for some subjects, such as subject 04, 16 and 17, the SMMS algorithm can reduce them to have fewer errors.

We perform another experiment by increasing the number of subspace dimensions for the proposed SMMS by average equal error rate (EER) of all subjects. We compare the result of EER of linear SVM and SMMS from 1D to 7D in Figure 8. The horizontal axis in Figure 8 is the number of dimensions for subspace and the vertical axis is EER of the algorithm. The SMMS outperforms the linear SVM in terms of average EER, when D = 4, 5, 6 and 7. The baseline algorithm, linear SVM, has on average EER=11.8%. The SMMS with a 7-dimensional subspace reduces EER to 9.1% on average with a total of 20 subjects.

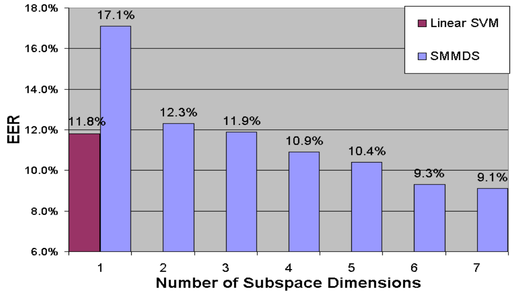

Figure 8. Comparison of EER between linear SVM and SMMS

We also test our algorithm on a pose variation subset of the PIE face database [19] with 68 subjects. We use 5 images for training and 4 images for testing for each subject. All the images are 64×64 grayscale cropped images. In Figure 9, the training and test face images from 3 subjects in this database are shown in different rows. This is also a very challenging database for face authentication because there are lots of pose variations.

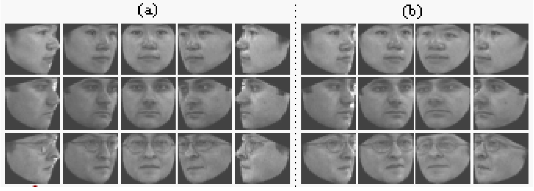

Figure 9. Sample training (a) images and test (b) images from PIE database

Six algorithms have been tested on this database. Three of them are algorithms based on support vector machines with different kernels such as linear, Radial Basis Function (RBF) and polynomial [14]. The other three are the proposed SMMS algorithm with 1D, 2D and 3D subspaces. We show the experimental results on this PIE pose database with different number of PCA coefficients separately. For dimension reduction, we use all the training images (5*68) to construct a PCA subspace with 100, 50 or 35 coefficients. The experimental data in the Table 2, Table 3 and Table 4 and are the numbers of the subjects, for which the corresponding SMMS algorithm outperforms the corresponding SVM algorithm in the test, out of a total of 68 subjects. The EER for 68 subjects are shown in Table 5, Table 6 and Table 7.

Numbers of better subjects	SVM-linear	SVM-RBF	SVM-polynomial (degree=2)
SMMS-1D	6	6	35
SMMS-2D	12	12	39
SMMS-3D	13	15	47

Table 2. Numbers of better subjects on 100 PCA coefficients of PIE pose database of 68 persons

Numbers of better subjects	SVM-linear	SVM-RBF	SVM-polynomial (degree=2)
SMMS-1D	5	10	41
SMMS-2D	8	14	50
SMMS-3D	11	11	53

Table 3. Numbers of better subjects on 50 PCA coefficients of PIE pose database of 68 persons

Numbers of better subjects	SVM-linear	SVM-RBF	SVM-polynomial (degree=2)
SMMS-1D	4	6	39
SMMS-2D	14	10	49
SMMS-3D	9	13	50

Table 4. Numbers of better subjects on 35 PCA coefficients of PIE pose database of 68 persons

100 PCA coefficients	SVM-linear	SVM-RBF	SVM-polynomial (degree=2)
EER	20 %	18 %	38 %
100 PCA coefficients	SMMS-1D	SMMS-2D	SMMS -3D
EER	36 %	31 %	27 %

Table 5. Average EER on 100 PCA coefficients of PIE pose database of 68 persons

50 PCA coefficients	SVM-linear	SVM-RBF	SVM-polynomial (degree=2)
EER	19 %	20 %	40 %
50 PCA coefficients	SMMS-1D	SMMS-2D	SMMS -3D
EER	36 %	30 %	27 %

Table 6. Average EER on 50 PCA coefficients of PIE pose database of 68 persons

35 PCA coefficients	SVM-linear	SVM-RBF	SVM-polynomial (degree=2)
EER	20 %	20 %	40 %
35 PCA coefficients	SMMS-1D	SMMS-2D	SMMS -3D
EER	36 %	32 %	27 %

Table 7. Average EER on 35 PCA coefficients of PIE pose database of 68 persons

We have shown that the SMMS algorithms are better than SVM algorithms for some subjects in the PIE pose database. So we propose to use SMMS to authenticate those subjects instead of SVM algorithms, because SMMS algorithms are simple and yield better performance. As shown in Table 5, Table 6 and Table 7, the average result of SMMS is better than SVM with polynomial kernel but worse than SVM with linear or RBF kernels. It is good to apply different algorithms to authenticate different subjects. We can choose from the SVM and SMMS algorithms to perform authentication on different subjects using some offline model selection method, such as taking a validation set to choose different algorithms. This kind of selection approach will improve the overall authentication performance. In Table 8, we show the algorithm complexity of SMMS of *R*-dimensional subspace with *d*-dimensional input data.

	Storage unit	Authentication time unit	Verification time unit
SMMS-*R*D	*R*	$O(d^3)$	$O(Rd)$

Table 8. Algorithm complexity

4. Conclusions

In this paper, we proposed a new algorithm based on the concept of Symmetric Maximized Minimal distance in Subspace (SMMS). In the training stage, SMMS tries to identify a subspace in the feature space where all the authentic samples are clustered together and all the imposter samples surrounding the authentic data are as far as possible. The optimality of the subspace is determined by maximizing the minimal distance between two classes in the subspace to identify the decision boundary. A new procedure to achieve such “maxmin” optimality is proposed. In the verification stage, the test sample is simply projected onto the subspace and compared against the decision boundary. Based on face authentication as an example, we show that the proposed algorithm outperforms several algorithms based on support-vector machines (SVM) on those challenging data patterns from several databases. SMMS can provide better performance for those subjects in our pose database, while SVM has a relatively large error. So we propose SMMS to perform authentication for those subjects instead of SVM.

This paper shows that the subspace classification with the symmetric maximizing minimal criterion is a promising way to enhance the performance of subspace-based authentication algorithms. Along this direction, our future work is to select the subspace without constraining the authentic samples on a hyper plane.

Acknowledgements

This research is supported in part by General Motors Corporation.

References

[1] I.T. Jolliffe, *Principle Component Analysis*, Spring-Verlag, New York, 1986.

[2] R.A. Fisher, “The use of multiple measurements in taxonomic problems”, *Annuals of Eugenics*, v. 7, 1936, pp. 111-132.

[3] M. Turk and A. Pentland, “Eigenfaces for recognition”, *Journal of Cognitive Neuroscience*. v. 3, 1991, pp. 71-86.

[4] X. M. Liu, T. Chen and B.V.K.V. Kumar, “Face Authentication for Multiple Subjects Using Eigenflow”, *Pattern Recognition, Special issue on Biometric*, Nov 2001.

[5] P.N. Belhumeur, J.P. Hespanha and D.J. Kriegman, “Eigenfaces vs Fisherfaces: recognition using class specific linear projection”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 20, 1997, pp. 711 –720.

[6] H. Yu, J. Yang, “A Direct LDA Algorithm for High-Dimensional Data -- with Application to Face Recognition”, *Pattern Recognition*, vol. 34, no. 10, 2001, pp. 2067-2070.

[7] L. Chen, H. Liao, M. Ko, J. Lin, and G. Yu, “A new LDA-based face recognition system which can solve the small sample size problem”, *Pattern Recognition*, v. 33, no.10, 2000, pp.1713-1726.

[8] B. Yu, L. Jin, P. Chen, “A New LDA-Based Method for Face Recognition”, in *Proceedings of the IEEE International Conf. on Pattern Recognition*, v. 1, 2002.

[9] R. Huang, Q. Liu, H. Lu, S. Ma, “Solving the Small Sample Size Problem of LDA”, in *Proceedings of the IEEE International Conf. on Pattern Recognition*, v. 3, 2002.

[10] V.N. Vapnik, *Estimation of dependences based on empirical data*, Springer-Verlag, Berlin, 1982.

[11] B.E Boser, I.M. Guyon and V.N. Vapnik, “A training algorithm for optimal margin classifiers”, in *Proceedings of the 5th annual ACM Workshop on Computational Learning Theory*, D. Haussler, Ed., 1992, pp. 144-152.

[12] C. Cortes and V.N. Vapnik, “Support vector networks”, *Machine Learning*, vol. 20, 1995, pp. 273-297.

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

[14] C.J.C. Burges, “A tutorial on support vector machines for pattern recognition”, *Knowledge Discovery and Data Mining*, vol. 2, no. 2, 1998, pp. 121-167.

[15] K.-R. Müller, S. Mika, G. Rätsch, , K. Tsuda, B. Schölkopf, “An Introduction to kernel-based learning algorithms”, *IEEE Transactions on Neural Networks*, Vol. 12, 2001, pp.181 –201.

[16] K. Jonsson, J. Kittler, Y. P. Li, and J. Matas “Support vector machines for face authentication”, in *Proceedings of British Machine Vision Conference*, T. Pridmore and D. Elliman, Ed., 1999, pp. 543–553.

[17] P.J. Phollips, “Support vector machines applied to face recognition”’ in *Proceedings of Advances in Neural Information Processing Systems 11*, M.S Keans, S.A. Solla, and D.A. Cohn, Ed., 1998.

[18] G. Guo, S. Li, and K. Chan, “Face recognition by support vector machines”, in *Proceedings of the International Conferences on Automatic Face and Gesture Recognition*, 2000, pp.196-201.

[19] T. Sim, S. Baker, and M. Bsat, “The CMU Pose, Illumination, and Expression (PIE) Database”, in *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, 2002.