IJAMML 4:1 (2016) 1-18 March 2016
ISSN: 2394-2258
Available at http://scientificadvances.co.in
DOI: http://dx.doi.org/10.18642/ijamml_7100121622

LEAST SQUARES SUPPORT MATRIX MACHINES BASED ON BILEVEL PROGRAMMING

Wenjing Xia and Liya Fan

School of Mathematics Sciences, Liaocheng University, 252059, P. R. China

Abstract

It is known that the classifications problems for matrix or more higher order tensor data are often met in many real-world applications. If using classical SVM-type methods for such problems, it needs to reshape matrix or tensor data into vectors, which may lead to the destruction of structure information contained in data. In order to overcome the limitation, this paper considers the classification problem with matrices as inputs directly and proposes a novel classification method named as least square support matrix machine (LSSMM). By means of bilevel programming (BP), an iteratively implemented algorithm (BP-LSSMM) for LSSMM is suggested. Experiment results indicate that BP-LSSMM is an effective and competitive classifier for matrix data classification.

Keywords: support matrix machine, least squares technology, matrix data classification, bilevel programming, iterative algorithm.

*Corresponding author.
E-mail address: fanliya63@126.com (Liya Fan).

2010 Mathematics Subject Classification: 68W40, 68Q25.
Submitted by Jose Luis Lopez-Bonilla.
Received January 13, 2016

1. Introduction

In the past decades, numerous SVM-type classification methods have been proposed and successfully applied to many fields. All these methods can be divided into two types. One is solving the Wolfe dual forms, such as support vector machine (SVM) [1, 2], twin support vector machine (TSVM) [3], improved TSVM (ITSVM) [4] and so on. Another is solving the primal problems, such as least square SVM (LS-SVM) [5], sparse least square SVM [6], least squares twin multi-class SVM [7], structural least square TSVM [8] and so on [9-11]. The use of least squares technology can reduce the computation time (training time and testing time) of classifiers by avoiding quadratic programming problems (QPPs).

However, in a wide range of real applications, such as image classification and electroencephalogram (EEG) classification, the input samples are represented as matrices naturally rather than vectors or scalars. In general, the structure information of the original matrix samples is useful and informative for data analysis tasks such as classification. One typical structure information is the correlation between columns or rows in the matrix samples. How to classify matrix data is an important research topic for pattern recognition and machine learning [12, 13]. Although vector-based classifiers and their variants have achieved satisfactory performance in many cases, they may lack efficiency in managing matrix data by simply reformulating them into vectors. The main reasons are as follows: (1) When we reformulate a matrix as a vector, the dimensionality of this vector is often very high, which may leads to the cause of dimensionality problem and the small sample size problem (the dimension of input data is much higher than the number of input data). (2) With the increase of dimensionality, the computation time will increase drastically. (3) When a matrix is collapsed as a vector, the spatial correlations of the matrix will be lost. In order to solve these problems, in recent years, a lot of researchers have been conducted on tensor-based approaches for image data analysis and some linear classifiers have been developed for pattern classification, for details, see [14-19].

Motivated by the above works, in this paper, we will study a matrix form generalization of SVM and simultaneously consider to seek a low-rank weighting matrix for matrix data classification. By means of bilevel programming (BP), we first propose a general framework for this generalization named as support matrix machine based on BP (BP-SMM) and then by means of least squares (LS) technology, we discuss the least squares solutions of BP-SMM and suggest a new classification method called as BP-LSSMM. The use of least squares technology aims to reduce the computation time of the proposed method. In order to verify the effectiveness of BP-LSSMM, a series of comparative experiments with SMM [19] are performed on Palm400, ORL [20], and Yale [21] three data sets.

The rest of the paper is organized as follows. In Section 2, background and related works are introduced. In Section 3, general framework of SMM based on bilevel programming (BP-SMM) is provided. In Section 4, the least square version of BP-SMM is considered and an iteratively algorithm (BP-LSSMM) is proposed with detailed derivation. Experiments and results analysis are performed in Section 5 and some conclusions are given in Section 6.

2. Preliminaries

This section recalls some basic concepts and basic results used in the sequel, for details, see [8, 22-24].

2.1. Notations

Let $A \in R^{p\times q}$ be a matrix with $\text{rank}(A) = r \le \min(p, q)$. The condensed singular value decomposition (SVD) [21, 22] of the matrix A is

$$A = U_A \sum\nolimits_A V_A^T = \sum\nolimits_{i=1}^{r} \sigma_i(A) u_i \nu_i^T,$$

where $U_A = [u_1, \cdots, u_r] \in R^{p\times r}$ and $V_A = [\nu_1, \cdots, \nu_r] \in R^{q\times r}$ are column orthogonal matrices, $\sum\nolimits_A = \text{diag}(\sigma_1(A), \cdots, \sigma_r(A))$ and $\sigma_1(A) \ge \cdots \ge \sigma_r(A) > 0$ are all nonzero singular values of the matrix A.

For any given $\tau > 0$, let

$$S_\tau[\sum\nolimits_A] = \mathrm{diag}((\sigma_1(A) - \tau)_+, \cdots, (\sigma_r(A) - \tau)_+),$$

$$D_\tau[A] = U_A S_\tau[\sum\nolimits_A] V_A^T,$$

where the plus function $(x)_+ = \max\{x, 0\}$ for all $x \in R$. $D_\tau[A]$ is often called a singular value thresholding (SVT) operator. In fact, if $0 < \tau \leq \sigma_r(A)$, then

$$S_\tau[\sum\nolimits_A] = \mathrm{diag}(\sigma_1(A) - \tau, \cdots, \sigma_r(A) - \tau),$$

$$D_\tau[A] = \sum\nolimits_{i=1}^{r} (\sigma_i(A) - \tau) u_i \nu_i^T.$$

If $\sigma_r(A) < \tau < \sigma_1(A)$, then there exists $1 < t < r$, such that

$$\sigma_1(A) \geq \cdots \geq \sigma_t(A) > \tau \geq \sigma_{t+1}(A), \cdots, \sigma_r(A).$$

In this case,

$$S_\tau[\sum\nolimits_A] = \mathrm{diag}(\sigma_1(A) - \tau, \cdots, \sigma_t(A) - \tau, 0, \cdots, 0),$$

$$D_\tau[A] = \sum\nolimits_{i=1}^{t} (\sigma_i(A) - \tau) u_i \nu_i^T.$$

If $\tau \geq \sigma_1(A)$, then $S_\tau[\sum\nolimits_A] = D_\tau[A] = 0_{r \times r}$.

The Frobenius norm of the matrix A is defined as $\|A\|_F = \sqrt{\sum\nolimits_{i=1}^{r} \sigma_i^2(A)}$. The nuclear norm of A is defined as $\|A\|_{nuc} = \sum\nolimits_{i=1}^{r} \sigma_i(A)$. The spectral norm of A is defined as $\|A\|_{spec} = \sigma_1(A)$, which is the largest singular value of A. The inner product of two same order matrices $A, B \in R^{p \times q}$ is defined as $< A, B > = \mathrm{tr}(A^T B)$, where $\mathrm{tr}(\cdot)$ denotes the trace of a matrix. It is evident that $\|A\|_F^2 = \mathrm{tr}(A^T A) = < A, A >$.

Due to the non-differentiability of the nuclear norm function $\|\cdot\|_{nuc} : R^{p\times q} \to R$, we consider its sub-differential in this paper:

$$\partial\|A\|_{nuc} = \{U_A V_A^T + Z : Z \in R^{p\times q}, U_A^T Z = 0, V_A Z = 0, \|Z\|_{spec} \le 1\}.$$

2.2. Support matrix machine (SMM)

This subsection briefly recalls SMM, for details, see [19]. Let $\{(X_i, y_i)\}_{i=1}^n \in R^{p\times q} \times \{\pm 1\}$ be a set of matrix data, where $X_i \in R^{p\times q}$ and $y_i \in \{\pm 1\}$ are the matrix input sample and class label of the i-th data, respectively. SMM aims to seek a classification hyperplane

$$f(X) = < W, X > + b = \mathrm{tr}(W^T X) + b = 0, \tag{1}$$

where $W \in R^{p\times q}$ is a weighting matrix and $b \in R$ is a threshold, by considering the following unconstrained optimization problem:

$$\min_{W,b} \frac{1}{2}\|W\|_F^2 + \tau\|W\|_{nuc} + C\sum_{i=1}^n \{1 - y_i[\mathrm{tr}(W^T X_i) + b]\}_+, \tag{2}$$

where $C, \tau > 0$ are trade-off parameters, $\sum_{i=1}^n \{1 - y_i[\mathrm{tr}(W^T X_i) + b]\}_+$ is the hinge loss function and $\frac{1}{2}\|W\|_F^2 + \tau\|W\|_{nuc}$ is a penalty function. Here the use of the nuclear norm $\|W\|_{nuc}$ aims to seek a low-rank weighting matrix. It is known that the hinge loss function enjoys the large margin principle and simultaneously embodies sparseness and robustness, which are two desirable properties for a good classifier.

Due to the existence of the nuclear norm function $\|\cdot\|_{nuc}$ and the plus function, the problem (2) is a non-smooth optimization problem. By using ADMM algorithm (see [19]), we can obtain (W^*, b^*). Consequently, for a new input sample $\widetilde{X} \in R^{p\times q}$, its class label $y_{\widetilde{X}}$ can be determined by

$$y_{\widetilde{X}} = \mathrm{sign}(< W^*, \widetilde{X} > + b^*) = \mathrm{sign}(\mathrm{tr}(W^{*T}\widetilde{X}) + b^*).$$

3. SMM Based on Bilevel Programming (BP-SMM)

In this section, we will extend and improve SMM by means of bilevel programming. The notations used in this section are same as in Section 2 unless special statements. The basic idea of bilevel programming is that the decision variables of the upper-level problem are the parameters of the lower-level problem and the (parameter) optimal solution of the lower-level problem is a response for the upper-level problem. To this end, let

$$H(W,\, b) = \frac{1}{2}\|W\|_F^2 + C\sum\nolimits_{i=1}^{n}\{1 - y_i[\mathrm{tr}(W^T X_i) + b]\}_+,$$

$$G(S) = \tau\|S\|_{nuc},$$

then the problem (2) can be transformed into the constrained optimization problem:

$$\min_{W,b,S} H(W,\, b) + G(S)$$

$$s.t.\ S - W = 0. \tag{3}$$

Considering the augmented Lagrange function of the problem (3)

$$L(W,\, b,\, S,\, \Lambda) = H(W,\, b) + G(S) + <\Lambda,\, (S - W)> + \rho\|S - W\|_F^2$$

$$= H(W,\, b) - \mathrm{tr}(\Lambda^T W) + \frac{\rho}{2}\|S - W\|_F^2$$

$$+ G(S) + \mathrm{tr}(\Lambda^T S) + \frac{\rho}{2}\|S - W\|_F^2,$$

and according to augmented Lagrange multiplier (ALM) method, we now that the problem (3) is equivalent to the following unconstrained optimization problem:

$$\min_{W,b,S,\Lambda} H(W, b) - \mathrm{tr}(\Lambda^T W) + \frac{\rho}{2}\|S - W\|_F^2 + G(S) + \mathrm{tr}(\Lambda^T S) + \frac{\rho}{2}\|S - W\|_F^2, \tag{4}$$

where $\rho > 0$ is an adjustable parameter and $\Lambda \in R^{p\times q}$ is a multiplier matrix.

If the matrices W and Λ are viewed as parameters and let

$$G_1(S) = G(S) + \mathrm{tr}(\Lambda^T S) + \frac{\rho}{2}\|S - W\|_F^2,$$

then the problem (4) can be transformed into the following bilevel programming problem:

$$\begin{aligned} &\min_{W,b,\Lambda} H(W, b) - \mathrm{tr}(\Lambda^T W) + \frac{\rho}{2}\|S - W\|_F^2 \\ &s.t. \min_S G_1(S). \end{aligned} \tag{5}$$

In addition, similar to the iterative thresholding (IT) method, we can get an iterative formula by considering the equality constraint $S - W = 0$:

$$\Lambda^{(k+1)} = \Lambda^{(k)} - \rho(S^{(k)} - W^{(k)}). \tag{6}$$

By means of the iterative formula (6), Λ can be viewed as a parameter of the upper-level problem of the problem (5) and then the problem (5) can be simplified as:

$$\begin{aligned} &\min_{W,b} H(W, b) - \mathrm{tr}(\Lambda^T W) + \frac{\rho}{2}\|S - W\|_F^2 \\ &s.t. \min_S G_1(S). \end{aligned} \tag{7}$$

4. Least Squares Solutions of BP-SMM (BP-LSSMM)

In this section, we mainly discuss the solving method of the problem (7). We firstly solve the lower-level problem of the problem (7) and then upper-level problem with least square technique. To this end, we need the following three results:

Theorem 1. *Let* $F : R^{p\times q} \to R$ *be a differentiable function and put*

$$\frac{dF(B)}{dB} = [\frac{dF(B)}{db_1}, \cdots, \frac{dF(B)}{db_q}], \quad \forall B \in R^{p\times q},$$

where $B = [b_1, \cdots, b_q]$ *and* $b_i \in R^p$, $i = 1, \cdots, q$. *Then*

(1) $\frac{d < A, B >}{dB} = \frac{d < B, A >}{dB} = A, \quad \forall A, B \in R^{p\times q}$;

(2) $\frac{d\|B\|_F^2}{dB} = 2B, \forall B \in R^{p\times q}$.

Proof. (1) Let $A = [a_1, \cdots, a_q]$ and $B = [b_1, \cdots, b_q]$, where $a_i, b_i \in R^p$, $i = 1, \cdots, q$. Then

$$< A, B > = \mathrm{tr}(A^T B) = \sum_{i=1}^{q} a_i^T b_i = \sum_{i=1}^{q} b_i^T a_i = \mathrm{tr}(B^T A) = < B, A > .$$

Consequently, $\frac{d < A, B >}{db_j} = \frac{d < B, A >}{db_j} = a_j$ for $j = 1, \cdots, q$, which indicates that the first conclusion is true.

(2) Let $B = [b_1, \cdots, b_q]$. By the definition of Frobenius norm, it can be deduced that

$$\|B\|_F^2 = \mathrm{tr}(B^T B) = \sum_{i=1}^{q} b_i^T b_i = \sum_{i=1}^{q} \|b_i\|^2.$$

Consequently, $\frac{d\|B\|_F^2}{db_j} = 2b_j$ for $j = 1, \cdots, q$, which shows that the second conclusion is true.

Theorem 2. *If a function* $F : R^{p\times q} \to R$ *is subdifferential and has a local minimum point at* $X^* \in R^{p\times q}$, *then* $0 \in \partial F(X^*)$.

Proof. This theorem can be proved by using the similar way of Theorem 10.1 in [25]. Theorem 2 indicates that $X^* \in R^{p\times q}$ can be viewed as an approximation of the local minimal solution if $0 \in \partial F(X^*)$.

Theorem 3. *For any given* $W, \Lambda \in R^{p\times q}$ *and* $\rho, \tau > 0$, *let* $S^* = \frac{1}{\rho} D_\tau(\rho W - \Lambda)$. *Then* $0 \in \partial G_1(S^*)$.

Proof. Considering the SVD of the matrix $\rho W - \Lambda$:

$$\rho W - \Lambda = U \sum V^T,$$

where $U \in R^{p\times t}$ and $V \in R^{q\times t}$ are column orthogonal matrices, $\sum = \text{diag}(\sigma_1, \cdots, \sigma_t)$, $\sigma_1 \geq \cdots \geq \sigma_t > 0$ and $t = \text{rank}(\rho W - \Lambda)$. Without loss of generality, we assume that $\sigma_1 \geq \cdots \geq \sigma_l > \tau \geq \sigma_{l+1} \geq \sigma_t > 0$. Put

$$\sum\nolimits_0 = \text{diag}(\sigma_1, \cdots, \sigma_l), \quad \sum\nolimits_1 = \text{diag}(\sigma_{l+1}, \cdots, \sigma_t),$$

$$U = [U_0, U_1], U_0 \in R^{p\times l}, \quad U_1 \in R^{p\times(t-l)},$$

$$V = [V_0, V_1], V_0 \in R^{q\times l}, \quad V_1 \in R^{q\times(t-l)}.$$

Then $U_0^T U_1 = 0$, $V_1^T V_0 = 0$ and

$$\rho W - \Lambda = [U_0, U_1] \begin{bmatrix} \sum_0 & \\ & \sum_1 \end{bmatrix} [V_0, V_1]^T = U_0 \sum\nolimits_0 V_0^T + U_1 \sum\nolimits_1 V_1^T,$$

$$S^* = \frac{1}{\rho} D_\tau(\rho W - \Lambda) = \frac{1}{\rho} U S_\tau(\rho W - \Lambda) V^T = U_0 [\frac{1}{\rho} (\sum\nolimits_0 - \tau I_l)] V_0^T,$$

which indicates that $\frac{1}{\rho} U_0 (\sum_0 - \tau I_l) V_0^T$ is the condensed SVD of the matrix S^*. Consequently,

$$\begin{aligned}\partial G_1(S^*) &= \partial G_1(S)\big|_{S=S^*} \\ &= \tau\partial\|S\|_{nuc}\big|_{S=S^*} + \rho S^* - (\rho W - \Lambda) \\ &= \tau\{U_0V_0^T + Z : Z \in R^{p\times q},\ U_0^TZ = 0,\ V_0Z = 0,\ \|Z\|_{spec} \le 1\} \\ &\quad + U_0(\sum\nolimits_0 - \tau I_l)V_0^T - (U_0\sum\nolimits_0 V_0^T + U_1\sum\nolimits_1 V_1^T) \\ &= \{\tau Z : Z \in R^{p\times q},\ U_0^TZ = 0,\ V_0Z = 0,\ \|Z\|_{spec} \le 1\} - U_1\sum\nolimits_1 V_1^T.\end{aligned}$$

Taking $Z = \frac{1}{\tau}U_1\sum_1 V_1^T$, it has $U_0^TZ = 0$, $ZV_0 = 0$, $\|Z\|_{spec} = \frac{1}{\tau}\sigma_{l+1} \le 1$ and then $\tau Z - U_1\sum_1 V_1^T = 0 \in \partial G_1(S^*)$. So, the conclusion of the theorem is true.

According to Theorems 2 and 3, $S^* = \frac{1}{\rho}D_\tau(\rho W - \Lambda)$ can be viewed as an approximate parameter optimal solution of the lower-level problem of the problem (7) for given $W, \Lambda \in R^{p\times q}$ and $\rho, \tau > 0$. Proceeding to the next step, we have the following iterative formula:

$$S^{(k+1)} = \frac{1}{\rho}D_\tau(\rho W^{(k)} - \Lambda^{(k)}). \tag{8}$$

Substituting $S^* = \frac{1}{\rho}D_\tau(\rho W - \Lambda)$ into the problem (7), it has the following single level optimization problem:

$$\min_{W,b}\frac{1}{2}\langle W, W\rangle + C\sum\nolimits_{i=1}^{n}(1 - y_i(\langle W, X_i\rangle + b))_+ - \langle\Lambda, W\rangle + \frac{\rho}{2}\|S^* - W\|_F^2. \tag{9}$$

Next, we solve the problem (9) by means of least squares technique. To this end, we use smooth function $\sum_{i=1}^{n}(1 - y_i(\langle W, X_i\rangle + b))^2$ to approximate the hinge loss function and get the following smooth unconstrained optimization problem:

$$\min_{W,b} \frac{1}{2}\langle W, W\rangle + \frac{C}{2}\sum_{i=1}^{n}(1 - y_i(\langle W, X_i\rangle + b))^2 - \langle \Lambda, W\rangle + \frac{\rho}{2}\|S^* - W\|_F^2. \tag{10}$$

It is well known that $\langle A, B\rangle = \langle vec(A), vec(B)\rangle$ for any $A, B \in R^{p\times q}$, where $vec(\cdot)$ denotes the vectorization of a matrix. In order to facilitate theoretical deduction, we transform the problem (10) into the vector version. Let

$$w = vec(W),\ s = vec(S^*),\ \gamma = vec(\Lambda),\ x_i = vec(X_i) \in R^{pq},\ i = 1, \cdots, n,$$

$$X = [x_1, \cdots, x_n] \in R^{pq\times n},\ D = \mathrm{diag}(y_1, \cdots, y_n) \in R^{n\times n},$$

$$e = (1, \cdots, 1)^T \in R^n,\ y = (y_1, \cdots, y_n)^t \in R^n,$$

then the problem (10) can be equivalently written as

$$\min_{w,b} F(w, b) = \frac{1}{2}\|w\|^2 + \frac{C}{2}\|e - D(X^T w + be)\|^2 - \langle \gamma, w\rangle + \frac{\rho}{2}\|s - w\|^2.$$

Letting $\frac{\partial F(w, b)}{\partial w} = \frac{\partial F(w, b)}{\partial b} = 0$, it can be deduced that

$$\left(\begin{bmatrix}(1+\rho)I & 0\\ 0 & 0\end{bmatrix} + C\begin{bmatrix}XX^T & Xe\\ e^TX^T & e^Te\end{bmatrix}\right)\begin{bmatrix}w\\ b\end{bmatrix} = \begin{bmatrix}\rho s + \gamma\\ 0\end{bmatrix} + C\begin{bmatrix}Xy\\ e^Ty\end{bmatrix}. \tag{11}$$

Put

$$N = (1+\rho)\begin{bmatrix}I & 0\\ 0 & 0\end{bmatrix} \in R^{(pq+1)\times(pq+1)},\ M = [X^T, e] \in R^{n\times(pq+1)},$$

$$P = \begin{bmatrix}\rho s + \gamma\\ 0\end{bmatrix} \in R^{(pq+1)},$$

then (11) can be rewritten as

$$\left(N + CM^TM\right)\begin{bmatrix}w\\ b\end{bmatrix} = P + CM^Ty. \tag{12}$$

It is evident that the matrix $N + CM^T M$ is a symmetric nonnegative definite matrix. Without loss of generality, we assume that $N + CM^T M$ is nonsingular; otherwise, it can be regularized by using $N + CM^T M + \delta I$ instead of $N + CM^T M$, where I is an identity matrix of appropriate dimensions and $\delta > 0$ is a sufficiently small number. Consequently, it can be deduced from (12) that

$$\begin{bmatrix} w \\ b \end{bmatrix} = \left(N + CM^T M\right)^{-1}(P + CM^T y),$$

and then an iterative formula can be obtained

$$\begin{bmatrix} w^{(k+1)} \\ b^{(k+1)} \end{bmatrix} = \left(N + CM^T M\right)^{-1}(P^{(k)} + CM^T y). \tag{13}$$

By using the iterative procedure described above, we can get the class label of a new matrix input. Specific algorithm is as follows.

Algorithm 1. BP-LSSMM

Step 1. Initialization. Given $\varepsilon > 0$ and parameters C, ρ, $\tau > 0$. Let T be the maximum number of iterations and $k = 0$. Take arbitrarily Λ^k, W^k, $S^k \in R^{p\times q}$ and $b^k \in R$.

Step 2. Calculate the matrices N and M.

Step 3. Update $\Lambda^{(k)}$ by (6).

Step 4. Update $S^{(k)}$ by (8).

Step 5. Vectorization. Calculate $w^k = vec(W^k)$, $\lambda^{(k)} = vec(\Lambda^{(k)})$ and $s^{(k)} = vec(S^{(k)})$.

Step 6. Calculate the matrix $P^{(k)} = \begin{bmatrix} \rho s^{(k)} + \gamma^{(k)} \\ 0 \end{bmatrix}$.

Step 7. Update $w^{(k)}$ and $b^{(k)}$ by (13).

Step 8. Calculate the matrix W^{k+1}.

Step 9. If $\left\| S^{(k+1)} - W^{(k+1)} \right\|_{\infty} < \varepsilon$ or T is achieved, stop iteration and put $W^* \leftarrow W^{(k+1)}$ and $b^* \leftarrow b^{(k+1)}$; otherwise, put $k \leftarrow k+1$ and return to Step 3.

Step 10. For a new matrix input $\widetilde{X}$, its class label $y_{\widetilde{X}}$ can be obtained by $y_{\widetilde{X}} = \text{sign}(\text{tr}(W^{*T}\widetilde{X}) + b^*)$.

5. Experiments

In order to demonstrate the effectiveness of the proposed BP-LSSMM, in this section, we will perform a series of comparative experiments with SMM on Palm400, ORL, and Yale three datasets. All the experiments are implemented by using 5-fold cross-validation method and in MATLAB (R2013a) running on a PC with system configuration Intel(R) Core(TM) i3 (2.53GHz) with 2GB of RAM.

Palm400 dataset includes 8000 palm pictures of 400 individuals, each individual has 20 palm images, the first ten copies and the last ten copies are taken at different time. ORL face dataset contains 400 face images of 40 individuals taken between April 1992 and April 1994 at different times, light and facial expressions. Each individual has 10 face images. Yale dataset contains 165 face images of 15 individuals with 11 images for each one. We choose randomly three pairs for each dataset with the original number according to different details and list them into Table 1. In addition, we crop Palm100 images into 16×16 pixels, ORL images into 14×11.5 pixels, and Yale images into 15×12 pixels.

Table 1. Selected parameters for classifiers

Dataset	Classifier	Parameters	Parameters values	
	ORL (3,8)	SMM	c, ρ, τ	2^3, 1.1, 0.1
	(10^*2)	BP-LSSMM	c, ρ, τ	2^3, 1.1, 0.1
	ORL (5,1)	SMM	c, ρ, τ	2^3, 1.1, 0.1
	(10^*2)	BP-LSSMM	c, ρ, τ	2^3, 1.1, 0.1
	ORL (5,4)	SMM	c, ρ, τ	2^3, 1.1, 0.1
	(10^*2)	BP-LSSMM	c, ρ, τ	2^3, 1.1, 0.1
	Palm400 (7,37)	SMM	c, ρ, τ	2^3, 1.1, 3
	(20^*2)	BP-LSSMM	c, ρ, τ	2^3, 1.1, 3
	Palm400 (1,37)	SMM	c, ρ, τ	2^3, 1.1, 3
	(20^*2)	BP-LSSMM	c, ρ, τ	2^3, 1.1, 3
	Palm400 (5,28)	SMM	c, ρ, τ	2^3, 1.1, 3
	(20^*2)	BP-LSSMM	c, ρ, τ	2^3, 1.1, 3
	Yale (1,8)	SMM	c, ρ, τ	2^{-4}, 1.1, 1
	(11^*2)	BP-LSSMM	c, ρ, τ	2^{-4}, 1.1, 1
	Yale (3,7)	SMM	c, ρ, τ	1, 1.1, 1
	(11^*2)	BP-LSSMM	c, ρ, τ	1, 1.1, 1
	Yale (6,8)	SMM	c, ρ, τ	2^6, 1.1, 1
	(11^*2)	BP-LSSMM	c, ρ, τ	2^6, 1.1, 1

It is known that the performance of classifiers seriously depends on the choice of parameters. In order to facilitate the comparison, take $\varepsilon = 10^{-3}$, $\delta = 10^{-8}$ in all experiments and select the other parameters involved in classifiers from 2^{-8} to 2^8 by grid search. The selected results

are listed in Table 1, where (a^*2) denotes that there are a photos for each class of two classes. The experiment results are listed in Table 2, where Acc, Error, and Time denote the classification accuracy, the accuracy error, and the running time of classifiers, respectively.

Table 2. Comparative results on nine experiment datasets

Datasets	Classifiers	Acc	Error	Time(s)
ORL (3,8)	SMM	**98.125** ± 1.88	0.00	1.0797
	BP-LSSMM	90.00 ± 0.00	0.00	**0.6689**
ORL (5,1)	SMM	96.25 ± 2.50	0.003	1.3931
	BP-LSSMM	**100.00** ± 0.00	0.00	**0.6254**
ORL (5,4)	SMM	**97.375** ± 1.625	0.007	1.0152
	BP-LSSMM	80.00 ± 0.00	0.05	**0.6025**
Palm400 (7,37)	SMM	97.51 ± 1.875	0.0091	2.4662
	BP-LSSMM	**100.00** ± 0.00	0.00	**0.4067**
Palm400 (1,37)	SMM	93.75 ± 6.25	0.0195	3.0333
	BP-LSSMM	**100.00** ± 0.00	0.01	**0.4054**
Palm400 (5,28)	SMM	96.25 ± 3.75	0.0125	2.7071
	BP-LSSMM	**97.50** ± 2.50	0.0031	**0.4118**
Yale (1,8)	SMM	**85.625** ± 5.625	0.0051	2.7610
	BP-LSSMM	85.00 ± 0.14	0.050	**0.4426**
Yale (3,7)	SMM	**93.75** ± 1.25	7.8125e-04	2.8663
	BP-LSSMM	90.00 ± 0.00	0.0187	**0.4874**
Yale (6,8)	SMM	**82.50** ± 6.25	0.0324	3.2894
	BP-LSSMM	80.00 ± 0.00	0.0313	**0.4514**

We can see from Table 2 that for Palm400 experiment datasets, the classification accuracy of BP-LSSMM achieves 100% for pairs (7,37) and (1,37) and is higher than that of SMM for pair (5,38). At the same time, the running time of BP-LSSMM is faster than that of SMM at least 6 times. For Yale experiment datasets, although the classification accuracy of BP-LSSMM is lower than that of SMM 3.75% the maximum and 0.625% the minimum, the running time of BP-LSSMM is faster than

that of SMM at least 5.8 times. For ORL experiment datasets, the classification accuracy of BP-LSSMM achieves 100% for pair (5,1) and is lower than that of SMM for pairs (3,8) and (5,4). But the running time of BP-LSSMM is faster than that of SMM at least 1.6 times.

According to the above analysis, we can conclude that the proposed BP-LSSMM is an feasibly and competitively quick classification method for matrix data.

6. Conclusion

In this paper, a novel quick classification method BP-LSSMM for matrix data is proposed by means of bilevel programming and least squares technique. The main advantage of the proposed method is to reduce the running time of classifiers and meanwhile protect the structure information of matrix samples. Experiment results show that the proposed BP-LSSMM is a feasibly and competitively quick classifier for matrix data classification. But, we only consider linear BP-LSSMM in this paper, not involve nonlinear case, which is the next step in our work.

References

[1] V. N. Vapnik, Statistical Learning Theory, Springer, 1998.

[2] V. Vapnik, The Nature of Statistical Learning Theory, New York, NY, USA: Springer-Verlag, 1995.

[3] Jayadeva, R. Khemchandani and S. Chandra, Twin support vector machines for pattern classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 29(5) (2007), 905-910.

[4] Y. H. Shao, C. H. Zhang, X. B. Wang and N. Y. Deng, Improvements on twin support vector machines, IEEE Transactions on Neural Networks 22(6) (2011), 962-968.

[5] M. Arun Kumar and M. Gopal, Least squares twin support vector machines for pattern classification, Expert Systems with Applications 36 (2009), 7535-7543.

[6] Danilo Avilar Silva, Juliana Peixoto Silva and Ajalmar R. Rocha Neto, Novel approaches using evolutionary computation for sparse least square support vector machines, Neurocomputing 168(30) (2015), 908-916.

[7] Jalal A. Nasiri, Nasrollah Moghadam Charkari and Saeed Jalili, Least squares twin multi-class classification support vector machine, Pattern Recognition 48(3) (2015), 984-992.

[8] Yitian Xu, Xianli Pan, Zhijian Zhou, Zhiji Yang and Yuqun Zhang, Structural least square twin support vector machine for classification, Applied Intelligence 42(3) (2015), 527-536.

[9] Xiaopeng Hua and Shifei Ding, Weighted least squares projection twin support vector machines with local information, Neurocomputing 160(21) (2015), 228-237.

[10] Shifei Ding and Xiaopeng Hua, Recursive least squares projection twin support vector machines for nonlinear classification, Neurocomputing 130(23) (2014), 3-9.

[11] Jianhui Guo, Ping Yi, Ruili Wang, Qiaolin Ye and Chunxia Zhao, Feature selection for least squares projection twin support vector machine, Neurocomputing 144(20) (2014), 174-183.

[12] J. Q. Gao, L. Y. Fan, L. Li and L. Z. Xu, A practical application of kernel-based fuzzy discriminant analysis, Int. J. Appl. Math. Comput. Sci. 23(4) (2013), 887-903.

[13] J. Q. Gao, L. Z. Xu, A. Shi and F. C. Huang, A kernel-based block matrix decomposition approach for the classification of remotely sensed images, Applied Mathematics and Computation 228 (2014), 531-545.

[14] C. Hou et al., Multiple rank multi-linear SVM for matrix data classification, Pattern Recognition 47 (2014), 454-469.

[15] R. Khemchandani, A. Karpatne and S. Chandra, Proximal support tensor machines, International Journal of Machine Learning and Cybernetics 4(6) (2013), 703-712.

[16] D. Tao, X. Li, W. Hu, S. J. Maybank and X. Wu, General tensor discriminant analysis and Gabor features for gait recognition, IEEE Trans. Pattern Anal. Mach. Intell. 29(10) (2007), 1700-1715.

[17] D. Tao, M. Song, X. Li, J. Shen, J. Sun, X. Wu, C. Faloutsos and S. J. Maybank, Bayesian tensor approach for 3-D face modeling, IEEE Trans. Circuits Syst. Video Technol. 18(10) (2008), 1397-1410.

[18] D. Tao, J. Sun, J. Shen, X. Wu, X. Li, S. J. Maybank and C. Faloutsos, Bayesian tensor analysis, IEEE International Joint Conference on Date of Conference Neural Networks (2008), 1-8.

[19] Luo Luo, Yubo Xie, Zhihua Zhang and Wu-Jun Li, Support Matrix Machines, Proceedings of the 32nd International Conference on Machine Learning (2015), 938-947.

[20] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

[21] http://cvc.yale.edu/projects/yalefaces/yalefaces.html.

[22] Zi Qiang Shi, Ji Qing Han and TieRan Zheng, Soft margin based low-rank audio signal classification, Neural Processing Letters 42(2) (2015), 291-299.

[23] Jian-Feng Cai, J. Emmanuel Cand‘es and Zuowei Shen, A singular value thresholding algorithm for matrix completion, SIAM Journal on Optimization 20(4) (2010), 1956-1982.

[24] Pan-Pan Zheng, Jun Feng, Zhan Li and Ming-quan Zhou, A novel SVD and LS-SVM combination algorithm for blind watermarking, Neurocomputing 142(22) (2014), 520-528.

[25] R. Tyrrell Rochafellar and Roger S-B Wets, Variational Analysis, New York, Springer, 1998.

■