Expert Systems with Applications 39 (2012) 849–860

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Maximum Margin Multisurface Support Tensor Machines with application to image classification and segmentation

Zhao Zhang *, Tommy W.S. Chow

Department of Electronic Engineering, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong

ARTICLE INFO

Keywords:
Multisurface Support Tensor Machine
Maximum margin criterion
Eigen-decomposition
Image classification
Tensor representation
Image segmentation

ABSTRACT

Virtually all previous classifier models take vectors as inputs, performing directly based on the vector patterns. But it is highly necessary to consider images as matrices in real applications. In this paper, we represent images as second order tensors or matrices. We then propose two novel tensor algorithms, which are referred to as *Maximum Margin Multisurface Proximal Support Tensor Machine* (M^3PSTM) and *Maximum Margin Multi-weight Vector Projection Support Tensor Machine* (M^3VSTM), for classifying and segmenting the images. M^3PSTM and M^3VSTM operate in tensor space and aim at computing two proximal tensor planes for multisurface learning. To avoid the singularity problem, maximum margin criterion is used for formulating the optimization problems. Thus the proposed tensor classifiers have an analytic form of projection axes and can achieve the maximum margin representations for classification. With tensor representation, the number of estimated parameters is significantly reduced, which makes M^3PSTM and M^3VSTM more computationally efficient when handing the high-dimensional datasets than applying the vector representations based methods. Thorough image classification and segmentation simulations on the benchmark UCI and real datasets verify the efficiency and validity of our approaches. The visual and numerical results show M^3PSTM and M^3VSTM deliver comparable or even better performance than some state-of-the-art classification algorithms.

1. Introduction

In the machine learning community, high-dimensional image data with many attributes are often encountered in the real applications. The representation and selection of the features will have a strong effect on the classification performance. Thus, how to efficiently represent the image data is one of the fundamental problems in classifier model design. It is worth noting that most of existing classification algorithms are oriented to *vector space model* (VSM), e.g. *Support Vector Machines* (SVM) (Cristianini & Shawe-Taylor, 2000; Vapnik, 1995). SVM relies on a single dataset, takes vector data x in space R^n as inputs, and aims at finding a single linear (or nonlinear) function. Recently, *Multisurface Proximal Support Vector Machine classification algorithm via Generalized Eigenvalues* (GEPSVM) (Mangasarian & Wild, 2006) and *Multi-weight Vector Projection Support Vector Machines* (MVSVM) (Ye, Zhao, Ye, & Chen, 2010) are proposed for pattern classification. Different from SVM classifier, GEPSVM and MVSVM perform based on two datasets X, $Y \in R^n$ and aim at computing two proximal planes via solving two eigen-problems. In GEPSVM and MVSVM, each plane is generated such that it is closest to one of the two datasets and as far as possible from the other dataset. And each of the two datasets of different classes will be proximal to one of two distinct planes that were not parallel together. It is noted that GEPSVM and MVSVM are also based on the vector space model, like SVM. Thus, if GEPSVM and MVSVM are applied for image classification, images are commonly represented as long vectors in the high-dimensional vector space, in which each pixel of the images corresponds to a feature or dimension. Thus, when the VSM focused methods are applied, one is often confronted with an image space R^n with large n. Let x denote an image with 64×64 pixels, then image x will be represented as a long vector $\hat{x}$ with dimension $n = 4096$. In such cases, learning such a linear SVM function $g(\hat{x}) = \varpi^{\mathrm{T}}\hat{x} + b$ in vector space is time-consuming, where $\varpi \in R^n$ and b are the parameters to be estimated. It should be noticed that GEPSVM and MVSVM suffer from the same problem. That is, when GEPSVM is employed, the matrices to be eigen-decomposed are of some 4097×4097 symmetric matrices (Mangasarian & Wild, 2006). Similarly, the matrices to be eigen-decomposed in MVSVM are 4096×4096 symmetric matrices (Ye et al., 2010). Thus when facing high-dimensional features, GEPSVM and MVSVM may lose the property of efficiency and need more running time to complement

* Corresponding author.
E-mail addresses: itzzhang@ee.cityu.edu.hk, cszzhang@gmail.com (Z. Zhang).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.07.083

classification. And most importantly, such a vector representation fails to take into account the spatial locality of pixels in the images (Wang, Chen, Liu, & Zhang, 2008; Zhang & Ye, 2011).

Images are intrinsically matrices. To represent the images appropriately, it is important to consider transforming the vector patterns to the corresponding matrix patterns or second order tensors before classification. In recent years, some interests about tensor representation have been investigated. Specifically, some tensor representation based approaches (Fu & Huang, 2008; He, Cai, & Niyogi, 2005; Vasilescu & Terzopoulos, 2003; Xu & Yan, 2009; Yan et al., 2007; Yang, Zhang, Frangi, & Yang, 2004) are proposed for high-dimensional data analysis. Tensorface in Vasilescu and Terzopoulos (2003) is the most representative tensorized method, representing the images with a higher-order tensor by extending *Principal Component Analysis* (PCA) (Dempster, 1971) to the higher-order tensor decomposition. It is noted that, virtually all previously proposed tensor algorithms are presented for performing dimensionality reduction and feature extraction. We in this paper aim at proposing the tensorized algorithms and designing the classifier models for image classification and segmentation. The optimization problems of the proposed algorithms are modeled by extending the GEPSVM and MVSVM problems into tensorzied scenarios. We then propose two new supervised classification techniques, which we refer to the proposed methodologies as *Maximum Margin Multisurface Proximal Support Tensor Machine* (M^3PSTM) and *Maximum Margin Multi-weight Vector Projection Support Tensor Machine* (M^3VSTM). M^3PSTM and M^3VSTM each aim at obtaining four optimal transforming basis vectors and focus on structuring two proximal tensor planes for efficient image representation and classifier design. It is worth noting that each of the transforming basis vector can be easily obtained by the MATLAB command (e.g. *eig* MATLAB, 1994–2001) as the eigenvector corresponding to the biggest eigenvalue of a scale-reduced standard eigen-problem. Considering the computational efficiency, we represent each sample image with n pixels as a second order tensor in $R^{n_1} \otimes R^{n_2}$, where $n_1 \times n_2 \approx n$. As a result, an $n_1 \times n_2$ image can be identified with a data point in $R^{n_1} \otimes R^{n_2}$. A linear function in tensor space can be similarly represented as $h(x) = u^{\mathrm{T}} X v + b$, where $u \in R^{n_1}$ and $v \in R^{n_2}$. Let $X \in R^{n_1} \otimes R^{n_2}$ denote the image with 64×64 pixels, function $h(x)$ only involves $n_1 + n_2 + 1 = 129$ parameters (b, u_i, v_j, $i = 1,2,\ldots,n_1$, $j = 1,2,\ldots,n_2$), which is much less than $n + 1$(=4097) of a linear function in the vector space. Thus, compared with GEPSVM and MVSVM, the matrices to be decomposed in M^3PSTM and M^3VSTM are based on $n_1 \times n_1$ or (and) $n_2 \times n_2$ symmetric matrices. These properties will make M^3PSTM and M^3VSTM particularly applicable for the case such that the number of samples is smaller than the input dimensionality, because the size of matrices decomposed in M^3PSTM and M^3VSTM depends on the dimensionality, not on sample size. In solving the transforming vectors for constructing the tensor planes, the *maximum margin criterion* (MMC) (Li, Jiang, & Zhang, 2006) is employed for formulating the problems. As a result, the singularity problem can be effectively avoided in solving the M^3PSTM and M^3VSTM projection axes.

The outline of this paper can be organized as follows. In Section 2, we review the formulations of *Multisurface Proximal Support Vector Machine classification via Generalized Eigenvalues* (GEPSVM), *Multi-weight Vector Projection Support Vector Machines* (MVSVM) and *maximum margin criterion* mathematically. In Sections 3 and 4, we present the computational analysis of the proposed classification algorithms in detail. In Section 5, we numerically compare the performance of the proposed methods through image classification and segmentation tasks using the benchmark UCI and real-world datasets. Finally, we offer the concluding remarks in Section 6.

2. Preliminaries

2.1. Multisurface Proximal Support Vector Machine via Generalized Eigenvalues

A new classification algorithm called GEPSVM (Mangasarian & Wild, 2006) is recently proposed wherein each of two sets are proximal to one of two distinct planes that are not parallel to each other (Mangasarian & Wild, 2006). Each plane is generated such that it is closest to one of the two sets and as far as possible from the other sets. Given m training data points in an n-dimensional space R^n denoted by $n \times m_1$ matrix X in class C_1 and $n \times m_2$ matrix Y in class C_2, satisfying $m_2 + m_1 = m$. The main focus of GEPSVM is to find two nonparallel hyperplanes in an n-dimension space, that is

$$X^{\mathrm{T}}\varpi^{(X)} - r^{(X)} = 0, \quad Y^{\mathrm{T}}\varpi^{(Y)} - r^{(Y)} = 0, \tag{1}$$

where the first plane is closest to the points belonging to class C_1 and furthest from the points belonging to class C_2, while the second plane is closest to the sample points belonging to class C_2 and furthest from the sample points in class C_1. For a nonnegative parameter δ, the objective functions of GEPSVM can be defined as follows:

$$\begin{aligned}
&\underset{(\varpi^{(X)},r^{(X)})\neq 0}{Min} \frac{\|X^{\mathrm{T}}\varpi^{(X)} - e^{(X)}r^{(X)}\|^2 + \delta\left\|\begin{bmatrix}\varpi^{(X)}\\ r^{(X)}\end{bmatrix}\right\|^2}{\|Y^{\mathrm{T}}\varpi^{(X)} - e^{(X)}r^{(X)}\|^2},\\
&\underset{(\varpi^{(Y)},r^{(Y)})\neq 0}{Min} \frac{\|Y^{\mathrm{T}}\varpi^{(Y)} - e^{(Y)}r^{(Y)}\|^2 + \delta\left\|\begin{bmatrix}\varpi^{(Y)}\\ r^{(Y)}\end{bmatrix}\right\|^2}{\|X^{\mathrm{T}}\varpi^{(Y)} - e^{(Y)}r^{(Y)}\|^2},
\end{aligned} \tag{2}$$

where the left formula in Eq. (2) defines the optimality computational rule that is able to enable GEPSVM obtaining the plane which is closest to the data points of class C_1 and, meanwhile, furthest from another set. Similarly, the right formula show the optimality computational rule enables GEPSVM obtaining the plane which is closest to the data points of class C_2 and furthest from the data points for set C_1. In Eq. (2), $\|\cdot\|$ denotes the two-norm and it is implicitly assumed that $(\varpi^{(X)}, r^{(X)}) \neq 0 \Rightarrow Y^{\mathrm{T}}\varpi^{(X)} - e^{(X)}r^{(X)} \neq 0$ and $(\varpi^{(Y)}, r^{(Y)}) \neq 0 \Rightarrow \|X^{\mathrm{T}}\varpi^{(Y)} - e^{(Y)}r^{(Y)}\|^2 \neq 0$. Define

$$\begin{aligned}
G^{(X)} &= [X^{\mathrm{T}} \quad -e^{(X)}]^{\mathrm{T}}[X^{\mathrm{T}} \quad -e^{(X)}] + \delta I,\\
H^{(X)} &= [Y^{\mathrm{T}} \quad -e^{(X)}]^{\mathrm{T}}[Y^{\mathrm{T}} \quad -e^{(X)}],\\
G^{(Y)} &= [Y^{\mathrm{T}} \quad -e^{(Y)}]^{\mathrm{T}}[Y^{\mathrm{T}} \quad -e^{(Y)}] + \delta I,\\
H^{(Y)} &= [X^{\mathrm{T}} \quad -e^{(Y)}]^{\mathrm{T}}[X^{\mathrm{T}} \quad -e^{(Y)}],\\
Z^{(X)} &= \begin{bmatrix}\varpi^{(X)}\\ r^{(X)}\end{bmatrix}, \quad Z^{(Y)} = \begin{bmatrix}\varpi^{(Y)}\\ r^{(Y)}\end{bmatrix},
\end{aligned} \tag{3}$$

where $^{\mathrm{T}}$ denotes the transpose of a matrix or a vector, $G^{(X)}$, $G^{(Y)}$, $H^{(X)}$ and $H^{(Y)}$ are symmetric matrices in space $R^{(n+1)} \times (n+1)$. Then, each of the two proximal planes can be respectively obtained by the eigenvectors corresponding to the smallest eigenvalues of the following two generalized eigenvalue problems:

$$\underset{Z^{(X)}\neq 0}{Min}\, p(Z^{(X)}) = \frac{(Z^{(X)})^{\mathrm{T}}G^{(X)}Z^{(X)}}{(Z^{(X)})^{\mathrm{T}}H^{(X)}Z^{(X)}}, \quad \underset{Z^{(Y)}\neq 0}{Min}\, q(Z^{(Y)}) = \frac{(Z^{(Y)})^{\mathrm{T}}G^{(Y)}Z^{(Y)}}{(Z^{(Y)})^{\mathrm{T}}H^{(Y)}Z^{(Y)}}. \tag{4}$$

It is noted that the problems in Eq. (4) are known as the *Rayleigh quotient*, thus we can solve the problems by the useful properties of *Rayleigh quotient* (Parlett, 1998) effectively, i.e., solve $G^{(X)}Z^{(X)} = \lambda^{(X)}H^{(X)}Z^{(X)}$ and $G^{(Y)}Z^{(Y)} = \lambda^{(Y)}H^{(Y)}Z^{(Y)}$.

2.2. Multi-weight Vector Projection Support Vector Machines

Another effective multisurface support vector machine classifier is called *Multi-weight Vector Projection Support Vector Machines* (MVSVM) (Ye et al., 2010), which is originally addressed for handling two-class classification problem. In MVSVM, the intra-class scatter difference is defined by minimizing distances between the intra-class points and the corresponding sample means. Meanwhile, MVSVM aims at optimizing the inter-class scatter by maximizing the difference between the class means from different classes. Let vector X_i, Y_i be the ith sample point of the $n \times m_1 X$, $n \times m_2 Y$ dataset belonging to different classes, then the criteria of the MVSVM problems are given as

$$\begin{aligned}\underset{\varpi^{(X)}}{Max} &\left\| \varpi^{(X)\mathrm{T}} \frac{1}{m_2}\sum_{i=1}^{m_2} Y_i - \varpi^{(X)\mathrm{T}} \frac{1}{m_1}\sum_{j=1}^{m_1} X_j \right\|^2 - \beta \sum_{i=1}^{m_1} \left\| \varpi^{(X)\mathrm{T}} X_i \right. \\ &\left. - \varpi^{(X)\mathrm{T}} \frac{1}{m_1}\sum_{j=1}^{m_1} X_j \right\|^2, \quad s.t.\ \|\varpi^{(X)}\|^2 - 1 = 0, \end{aligned} \tag{5}$$

$$\begin{aligned}\underset{\varpi^{(Y)}}{Max} &\left\| \varpi^{(Y)\mathrm{T}} \frac{1}{m_1}\sum_{i=1}^{m_1} X_i - \varpi^{(Y)\mathrm{T}} \frac{1}{m_2}\sum_{j=1}^{m_2} Y_j \right\|^2 - \beta \sum_{i=1}^{m_2} \left\| \varpi^{(Y)\mathrm{T}} Y_i \right. \\ &\left. - \varpi^{(Y)\mathrm{T}} \frac{1}{m_2}\sum_{j=1}^{m_2} Y_j \right\|^2, \quad s.t.\ \|\varpi^{(Y)}\|^2 - 1 = 0, \end{aligned} \tag{6}$$

where vectors $\varpi^{(X)}$ and $\varpi^{(Y)}$ denote the two one-dimensional projection directions that are required to optimize, $(1/m)\sum_{j\in C_2} Y_j = M^{(Y)}$ and $(1/m)\sum_{j\in C_1} X_j = M^{(X)}$ are mean vectors of the sample data in the datasets X and Y, and β is a trade-off parameter, which is used for balancing the functional value. It is noted that MVSVM is based on the maximum margin criterion. Let $\wp^{(1)}$, $\wp^{(2)}$ and $\wp^{(3)}$ satisfy

$$\begin{aligned}\wp^{(1)} &= \left\| \frac{1}{m_2}\sum_{i=1}^{m_2} Y_i - \frac{1}{m_1}\sum_{j=1}^{m_1} X_j \right\|^2, \\ \wp^{(2)} &= \sum_{i=1}^{m_1} \left\| X_i - \frac{1}{m_1}\sum_{j=1}^{m_1} X_j \right\|^2, \\ \wp^{(3)} &= \sum_{i=1}^{m_2} \left\| Y_i - \frac{1}{m_2}\sum_{j=1}^{m_2} Y_j \right\|^2, \end{aligned} \tag{7}$$

then $\varpi^{(X)}$ and $\varpi^{(Y)}$ can be obtained by solving the following equivalent problems to Eqs. (5) and (6):

$$\underset{\varpi^{(X)}}{Max}\ \varpi^{(X)\mathrm{T}}\wp^{(1)}\varpi^{(X)} - \beta\varpi^{(X)\mathrm{T}}\wp^{(2)}\varpi^{(X)}, \quad s.t.\ \varpi^{(X)\mathrm{T}}\varpi^{(X)} - 1 = 0, \tag{8}$$

$$\underset{\varpi^{(Y)}}{Max}\ \varpi^{(Y)\mathrm{T}}\wp^{(1)}\varpi^{(Y)} - \beta\varpi^{(Y)\mathrm{T}}\wp^{(3)}\varpi^{(Y)}, \quad s.t.\ \varpi^{(Y)\mathrm{T}}\varpi^{(Y)} - 1 = 0, \tag{9}$$

where the term $\wp^{(1)}$ can be expressed in a matrix interpretation as the following formulation:

$$\begin{aligned}\wp^{(1)} &= \left[\frac{1}{m_2}\sum_{i=1}^{m_2} Y_i - \frac{1}{m_1}\sum_{j=1}^{m_1} X_j\right]\left[\frac{1}{m_2}\sum_{i=1}^{m_2} Y_i - \frac{1}{m_1}\sum_{j=1}^{m_1} X_j\right]^{\mathrm{T}} \\ &= \frac{1}{m_2^2}\sum_{i=1}^{m_2}\sum_{j=1}^{m_2} Y_i Y_j^{\mathrm{T}} + \frac{1}{m_1^2}\sum_{i=1}^{m_1}\sum_{j=1}^{m_1} X_i X_j^{\mathrm{T}} - \frac{1}{m_1 m_2}\sum_{j=1}^{m_1}\sum_{i=1}^{m_2} X_j Y_i^{\mathrm{T}} - \frac{1}{m_2 m_1}\sum_{i=1}^{m_2}\sum_{j=1}^{m_1} Y_i X_j^{\mathrm{T}}, \\ &= \frac{1}{m_2^2} Y e_2^{\mathrm{T}} e_2 Y^{\mathrm{T}} + \frac{1}{m_1^2} X e_1^{\mathrm{T}} e_1 X^{\mathrm{T}} - (X P^{(XY)} Y^{\mathrm{T}} + Y Q^{(YX)} X^{\mathrm{T}}) \end{aligned} \tag{10}$$

where e_1 is a $1 \times m_1$ vector of all ones, e_2 is a $1 \times m_2$ vector of all ones, $m_1 \times m_2$ $P^{(XY)}$ and $m_2 \times m_1$ $Q^{(YX)}$ are matrices satisfying $P^{(XY)} = (1/m_1 m_2) e_1^{\mathrm{T}} e_2$, $Q^{(YX)} = (1/m_2 m_1) e_2^{\mathrm{T}} e_1$. Obviously, $\wp^{(1)}$ is symmetric. Then terms $\wp^{(2)}$ and $\wp^{(3)}$ can be similarly represented as

$$\begin{aligned}\wp^{(2)} &= \sum_{i=1}^{m_1}\left[X_i - \frac{1}{m_1}\sum_{j=1}^{m_1} X_j\right]\left[X_i - \frac{1}{m_1}\sum_{j=1}^{m_1} X_j\right]^{\mathrm{T}} = \sum_{i=1}^{m_1} X_i X_i^{\mathrm{T}} - \frac{1}{m_1}\sum_{ij=1}^{m_1} X_j X_i^{\mathrm{T}} = X\left(I - \frac{1}{m_1} e_1^{\mathrm{T}} e_1\right) X^{\mathrm{T}}, \\ \wp^{(3)} &= Y\left(I - \frac{1}{m_2} e_2^{\mathrm{T}} e_2\right) Y^{\mathrm{T}}, \end{aligned} \tag{11}$$

where I is an identity matrix,

According to Ye et al. (2010), MVSVM has been applied to classify the real handwritten digits, but most real data, including handwritten digits, are usually stored as form of images (or, matrices), but MVSVM performs in vector space, like GEPSVM. As a result, when a pattern itself is an image, the image first has to be transformed to a long vector pattern by concatenating its pixels and then MVSVM can perform classification on it.

2.3. Maximum margin criterion (MMC)

Inspired by Support Vector Machines (SVMs) (Vapnik, 1995), Li et al. (2006) proposed an efficient and robust learning method called maximum margin criterion (MMC) for feature extraction. When the class label $\eta \in \{C_1, C_2, \ldots, C_c\}$ of some pattern x is available, the margin is then defined to characterize the discriminant ability of the features, where c is the number of classes. MMC maximizes the distances, which are used to measure the similarity or dissimilarity, between classes after the transformation by optimizing the following feature extraction criterion:

$$J = \frac{1}{2}\sum_{i=1}^{c}\sum_{j=1}^{c} p_i p_j d(C_i, C_j), \tag{12}$$

which is actually the summation of all pair interclass margins. p_i denotes *a priori* probability of class i and $d(C_i, C_j)$ is the distance metric between class C_i and C_j. Let χ_i and χ_j be the mean vectors of the class C_i and C_j, the interclass distance (or margin) can be defined as

$$d(C_i, C_j) = d(\chi_i, \chi_j) - \rho(C_i) - \rho(C_j), \tag{13}$$

where $\rho(C_i)$ is some measure of the scatter of class C_i. Let S_i be the covariance matrix of class C_i, if overall variance $tr(S_i)$ is used to measure the scatter of data, then Eq. (13) measures the "average margin" between two classes while the minimum margin is used in SVM, where $tr(S_i)$ denotes the matrix trace of the matrix S_i. With Eq. (13), $\rho(C_i)$ and $tr(S_i)$, Eq. (12) can be rewritten as

$$\begin{aligned} J &= \frac{1}{2}\sum_{i=1}^{c}\sum_{j=1}^{c} p_i p_j d(\chi_i, \chi_j) - tr(S_i) - tr(S_j) \\ &= \frac{1}{2}\sum_{i=1}^{c}\sum_{j=1}^{c} p_i p_j d(\chi_i, \chi_j) - \frac{1}{2}\sum_{i=1}^{c}\sum_{j=1}^{c} p_i p_j (tr(S_i) + tr(S_j)). \end{aligned} \tag{14}$$

Let $S^{(wc)}$ and $S^{(bc)}$ be the intra-class scatter and inter-class scatter matrices of *Fisher Linear Discriminant Analysis* (LDA) (Martinez & Kak, 2001) and χ denote the overall mean vector. Because $\sum_{j=1}^{c} p_j(\chi - \chi_j) = 0$, by employing the Euclidean distance metric, the former part of Eq. (14) can be simplified to $tr(S^{(bc)})$:

$$\begin{aligned} &\frac{1}{2}\sum_{i=1}^{c}\sum_{j=1}^{c} p_i p_j d(\chi_i, \chi_j) \\ &= \frac{1}{2}\sum_{i=1}^{c}\sum_{j=1}^{c} p_i p_j (\chi_i - \chi_j)^{\mathrm{T}}(\chi_i - \chi_j) \\ &= \frac{1}{2}\sum_{i=1}^{c}\sum_{j=1}^{c} p_i p_j (\chi_i - \chi + \chi - \chi_j)^{\mathrm{T}}(\chi_i - \chi + \chi - \chi_j) \\ &= tr\left[\sum_{i=1}^{c} p_i (\chi_i - \chi)(\chi_i - \chi)^{\mathrm{T}}\right] = tr(S^{(bc)}). \end{aligned} \tag{15}$$

Similarly, the latter part of Eq. (14) can be simplified to $tr(S^{(wc)})$:

$$\frac{1}{2}\sum_{i=1}^{c}\sum_{j=1}^{c}p_i p_j(tr(S_i)+tr(S_j))=\sum_{i=1}^{c}p_i tr(S_i)=tr\left[\sum_{i=1}^{c}p_i S_i\right]=tr(S^{(wc)}). \tag{16}$$

Thus MMC aims at maximizing the following criterion as discriminant criterion instead of utilizing the ratio form $J(\varpi)=tr(\varpi^{\mathrm{T}}S^{(bc)}\varpi/\varpi^{\mathrm{T}}S^{(wc)}\varpi)$, which is called a *margin* in

$$J(\varpi)=tr(\varpi^{\mathrm{T}}(S^{(bc)}-S^{(wc)})\varpi). \tag{17}$$

Compared with the LDA criterion, MMC is more efficient and easier to be implemented. Most importantly, the small sample size problem does not exist because it needs not to calculate the inverse within-class scatter matrix. The canonical features extracted by margin based methods can achieve better results in classification (Liu, Chen, Tan, & Zhang, 2007; Wang, Zheng, Hu, & Chen, 2007; Zheng, Zou, & Zhao, 2005) MMC has the advantages of reasonable motivation in principle and simplicity (Li et al., 2006).

3. Maximum margin criterion based Multisurface Proximal Support Tensor Machine (M^{3}PSTM)

3.1. Tensor representation of data

The general formulation of the linear subspace learning in the second order tensor space (He et al., 2005) is given as follows. Let $X,Y\in R^{n_1\times n_2}$ represent an image of size $n_1\times n_2$, accompanied with two class labels. Mathematically, X and Y can be regarded as the second order tensor (or, 2-tensor) in the tensor space $R^{n_1}\otimes R^{n_2}$. Let $(u_1,u_2,\ldots,u_{n_1})$ denote a set of the orthonormal basis functions in R^{n_1} and let $(v_1,v_2,\ldots,v_{n_2})$ be a set of the orthonormal basis functions in R^{n_2} (He et al., 2005), then a second order tensor of X and Y can be mathematically formulated as

$$X=\sum_{ij}(u_i^{\mathrm{T}}Xv_j)u_iv_j^{\mathrm{T}},\quad Y=\sum_{ij}(u_i^{\mathrm{T}}Yv_j)u_iv_j^{\mathrm{T}}, \tag{18}$$

which indicates that $\left\{u_iv_j^{\mathrm{T}}\right\}$ forms a basis of the tensor space $R^{n_1}\otimes R^{n_2}$. Let $U^{(X)}$ (or, $U^{(Y)}$) represent a subspace of R^{n_1} spanned by using $\{u_i\}_{i=1}^{d_1}$ and $V^{(X)}$ (or, $V^{(Y)}$) denote a subspace of R^{n_2} spanned by using $\{v_i\}_{i=1}^{d_2}$. Thus the tensor product $U^{(X)}\otimes V^{(X)}$ (or, $U^{(Y)}\otimes V^{(Y)}$) is a subspace of $R^{n_1}\otimes R^{n_2}$. Thus, by projecting vectors of $X,Y\in R^{n_1\times n_2}$ onto the subspaces $U^{(X)}\otimes V^{(X)}$ and $U^{(Y)}\otimes V^{(Y)}$, we can then obtain tensors $\widetilde{X}_i=U^{(X)\mathrm{T}}X_iV^{(X)}\in R^{d_1\times d_2}$ and $\widehat{Y}_i=U^{(Y)\mathrm{T}}Y_iV^{(Y)}\in R^{d_1\times d_2}$, respectively. It should be noted that, the number of parameters estimated in the tensor space problem is much smaller than that in the vector space problems. This property will make tensor representation particularly applicable for dealing with the high-dimensional small sample size problems (He et al., 2005).

3.2. The objective function

The proposed M^{3}PSTM classifier is fundamentally formulated based on the optimization problems of the regular GEPSVM method. The linear classifiers of M^{3}PSTM in the tensor space can be represented as follows:

$$f(X)=(U^{(X)})^{\mathrm{T}}XV^{(X)}-r^{(X)},\quad f(Y)=(U^{(Y)})^{\mathrm{T}}YV^{(Y)}-r^{(Y)}. \tag{19}$$

It is noted that the criterions of Eq. (19) can be mathematically formulated through matrix inner product and are then defined as the following criterions:

$$f(X)=\langle X,U^{(X)}(V^{(X)})^{\mathrm{T}}\rangle-r^{(X)},\quad f(Y)=\langle Y,U^{(Y)}(V^{(Y)})^{\mathrm{T}}\rangle-r^{(Y)}. \tag{20}$$

In our work, we also drop the parallelism condition on the proximal planes, just like GEPSVM. We also require that each computed tensor plane to be as close as possible to one of the datasets and as far as possible from the other one. Then, we aim at finding the following two proximal tensor planes in the tensor spaces:

$$(U^{(X)})^{\mathrm{T}}XV^{(X)}-r^{(X)}=0,\quad (U^{(Y)})^{\mathrm{T}}YV^{(Y)}-r^{(Y)}=0, \tag{21}$$

where the first tensor plane is defined to be closest to the data points belonging to dataset X and furthest from the points in dataset Y. In contrast, the second tensor plane is defined to be closest to the points in the dataset Y and furthest from the points in dataset X. To achieve the tensor planes in Eq. (21), we aim to minimize the sum of the squares of two-norm distances between each sample points of the dataset X (or, Y) to the first (or, second) tensor plane in Eq. (21) and maximizing the sum of the squares of two-norm distances between each sample point of dataset Y (or, X) to the same tensor plane (Mangasarian, 1999). All of these definitions can lead to the following formulations.

Similar to the objective function of GEPSVM, the optimization problems of the binary M^{3}PSTM classification algorithm in tensor space can be described as the following two maximization problems:

$$\underset{((U^{(X)},V^{(X)}),r^{(X)})\neq 0}{Max}\|(U^{(X)})^{\mathrm{T}}YV^{(X)}-e^{(X)}r^{(X)}\|^2/\|(U^{(X)})^{\mathrm{T}}XV^{(X)}-e^{(X)}r^{(X)}\|^2, \tag{22}$$

$$\underset{((U^{(Y)},V^{(Y)}),r^{(Y)})\neq 0}{Max}\|(U^{(Y)})^{\mathrm{T}}XV^{(Y)}-e^{(Y)}r^{(Y)}\|^2/\|(U^{(Y)})^{\mathrm{T}}YV^{(Y)}-e^{(Y)}r^{(Y)}\|^2, \tag{23}$$

where notation $\|\cdot\|^2$ denotes the squares of the two-norm distances and it is also implicitly assumed that $((U^{(X)},V^{(X)}),\ r^{(X)})\neq 0\Rightarrow(U^{(X)})^{\mathrm{T}}XV^{(X)}-e^{(X)}r^{(X)}\neq 0$ and $((U^{(Y)},V^{(Y)}),\ r^{(Y)})\neq 0\Rightarrow(U^{(Y)})^{\mathrm{T}}YV^{(Y)}-e^{(Y)}r^{(Y)}\neq 0$. To achieve large margin classification, the numerator of the problem in Eq. (22) maximizes the sum of the squares of the two-norm distances in the $((U^{(X)},V^{(X)}),r^{(X)})$-space of the sample data points belonging to the dataset Y to the plane $(U^{(X)})^{\mathrm{T}}XV^{(X)}-r^{(X)}=0$, while the denominator of Eq. (22) minimizes the sum of the squares of generalized two-norm distances in the $((U^{(X)},V^{(X)}),r^{(X)})$-space of the data points of dataset X to the same tensor plane. All of these definitions and formulations are suitable for the maximization optimization problem in Eq. (23).

3.3. Computational analysis

To compute the transforming basis vectors for constructing the tensor planes, we can adopt the similar optimization approach of GEPSVM to formulate the problems. We analogously define the following formulations:

$$\begin{aligned}
G^{(X)}&=[\,Y^{\mathrm{T}}\quad -e^{(Y)}\,]^{\mathrm{T}}[\,Y^{\mathrm{T}}\quad -e^{(Y)}\,],\\
H^{(X)}&=[\,X^{\mathrm{T}}\quad -e^{(X)}\,]^{\mathrm{T}}[\,X^{\mathrm{T}}\quad -e^{(X)}\,],\\
G^{(Y)}&=[\,X^{\mathrm{T}}\quad -e^{(X)}\,]^{\mathrm{T}}[\,X^{\mathrm{T}}\quad -e^{(X)}\,],\\
H^{(Y)}&=[\,Y^{\mathrm{T}}\quad -e^{(Y)}\,]^{T}[Y^{\mathrm{T}}-e^{(Y)}],\\
K^{(X)}&=\begin{bmatrix}U^{(X)}(V^{(X)})^{\mathrm{T}}\\ r^{(X)}\end{bmatrix},\quad K^{(Y)}=\begin{bmatrix}U^{(Y)}(V^{(Y)})^{\mathrm{T}}\\ r^{(Y)}\end{bmatrix}.
\end{aligned} \tag{24}$$

It is noted that if we employ the computational approach of GEPSVM to solve the two proximal tensor planes $(U^{(X)})^{\mathrm{T}}XV^{(X)}-r^{(X)}=0$ and $(U^{(Y)})^{\mathrm{T}}YV^{(Y)}-r^{(Y)}=0$, then the proximal tensor planes can be respectively defined by the eigenvectors corresponding to the leading eigenvalues of the following two trace ratio problems (Guo, Li, Yang, Shu, & Wu, 2003; Jia, Nie, & Zhang, 2009; Wang, Yan, Xu, Tang, & Huang, 2007):

$$\underset{K^{(X)}\neq 0}{Max}\, P(K^{(X)}) = \frac{tr((K^{(X)})^{\mathrm{T}} G^{(X)} K^{(X)})}{tr((K^{(X)})^{\mathrm{T}} H^{(X)} K^{(X)})},$$
$$Max_{K^{(Y)}\neq 0} Q(K^{(Y)}) = \frac{tr((K^{(Y)})^{\mathrm{T}} G^{(Y)} K^{(Y)})}{tr((K^{(Y)})^{\mathrm{T}} H^{(Y)} K^{(Y)})}. \quad (25)$$

It is worthy of noticing that the above maximization problems can be solved by using the eigen-decomposition, which involves the matrix inverse operation. As a result, we firstly need to regularize the problems in Eq. (25) as GEPSVM, otherwise the computation process may be unstable due to the matrix inverse operation on the matrices $G^{(X)}(H^{(X)})^{-1}$ and $G^{(Y)}(H^{(Y)})^{-1}$, i.e., singularity problems. To efficiently solve this problem and obtain a specific solution, we introduce a practical method. We take the left problem of Eq. (25) for example. To solve this trace ratio (TR) problem, Guo et al. (2003) have definitely indicated that the global optimum of TR problem can be equivalently solved by using the trace difference problem, namely $\max tr((K^{(X)})^{\mathrm{T}}(G^{(X)} - \mu H^{(X)})K^{(X)})$, based on the concept of Foley-Sammon transform. In this way, the optimal TR value μ^* and $K^{(X)}$ can be effectively obtained. According to Jia et al. (2009), the objective $tr((K^{(X)})^{\mathrm{T}}(G^{(X)} - \mu H^{(X)})K^{(X)})$ of the trace difference problem can be considered as the generalized maximum margin criterion (MMC) problem with tuning parameter μ involved or the standard MMC (Li et al., 2006) with $\mu = 1$. In this paper, we focus on employing the generalized maximum margin criterion to optimize our problems and solving the eigenvectors $U^{(X)}$, $V^{(X)}$, $U^{(Y)}$ and $V^{(Y)}$ for designing tensor classifiers.

Based on the margin maximization criterion, each of the two nonparallel proximal tensor planes is respectively obtained by the eigenvector corresponding to the largest eigenvalues of two scale-reduced standard eigenvalue problems. Namely, the two tensor planes $(U^{(X)})^{\mathrm{T}} X V^{(X)} - r^{(X)} = 0$ and $(U^{(Y)})^{\mathrm{T}} Y V^{(Y)} - r^{(Y)} = 0$ can be obtained by solving the following two generalized maximum margin criterion based eigen-problems from Eq. (25):

$$\underset{K^{(X)}\neq 0}{Max}\, P(K^{(X)}) = (K^{(X)})^{\mathrm{T}}((1-\ell^{\dagger})G^{(X)} - \ell^{\dagger} H^{(X)})K^{(X)}, \quad (26)$$
$$\underset{K^{(Y)}\neq 0}{Max}\, Q(K^{(Y)}) = (K^{(Y)})^{\mathrm{T}}((1-\ell^{\dagger})G^{(Y)} - \ell^{\dagger} H^{(Y)})K^{(Y)}, \quad (27)$$

where $\ell^{\dagger} \in (0,1)$ is a balancing parameter, which will be used in the later readings without introduction. In other words, we determine the optimal vectors $U^{(X)}$, $V^{(X)}$, $U^{(Y)}$ and $V^{(Y)}$ so that nearby data pairs belonging to the same class are as close together as possible and sample data pairs of different classes are as far apart as possible.

Clearly, the eigenvector problems expressed by Eqs. (26) and (27) with the normalized constraints $(K^{(X)})^{\mathrm{T}}K^{(X)} = 1$ and $(K^{(Y)})^{\mathrm{T}}K^{(Y)} = 1$ are typical eigen-problems, from which one can find that there is no need for computing any matrix inversion in optimizing the above margin criterions. Notice that the normalized constraints $(K^{(X)})^{\mathrm{T}}K^{(X)} = 1$ and $(K^{(Y)})^{\mathrm{T}}K^{(Y)} = 1$ are only defined for aiding the optimizations. One might have noticed that the transforming basis vectors $U^{(X)}$ and $V^{(X)}$ are dependent on each other, that is $K^{(X)}$ and $K^{(Y)}$ are not fixed while depend on $U^{(X)}$ and $V^{(X)}$. As a results, the projection axes $U^{(X)}$ and $V^{(X)}$ cannot be solved independently. All of these discussions are suitable for the computation process of the second proximal tensor plane $(U^{(Y)})^{\mathrm{T}} Y V^{(Y)} - r^{(Y)} = 0$, that is to say, $U^{(Y)}$ and $V^{(Y)}$ involved in the second tensor plane can not be solved independently as well.

In this paper, we aim at computing the basis vectors $U^{(X)}$, $U^{(Y)}$, $V^{(X)}$ and $V^{(Y)}$ as follows. We next describe a simple but effective computational technique to solve the optimization problems as that of He et al. (2005). To simplify such a problem, we first fix $U^{(X)}$ and $U^{(Y)}$. In the simulations, we initially set $U^{(X)}$ and $U^{(Y)}$ to be the vectors with all ones. Thus if we let $\overrightarrow{X}_i = U^{(X)\mathrm{T}} X_i$ and $\overrightarrow{Y}_i = U^{(Y)\mathrm{T}} Y_i$, then the tensor classifiers can be rewritten as

$$f(\overrightarrow{X}) = \overrightarrow{X} V^{(X)} - r_1^{(X)}, \quad f(\overrightarrow{Y}) = \overrightarrow{Y} V^{(Y)} - r_1^{(Y)}, \quad (28)$$

which are identical to the vector space model based linear classifiers. In this way, basis vectors $V^{(X)}$ and $V^{(Y)}$ can be obtained by solving the following two optimization problems from Eqs. (22) and (23):

$$\underset{(V^{(X)}, r_1^{(X)})\neq 0}{Max} \frac{\left\| \overrightarrow{Y} V^{(X)} - e^{(X)} r_1^{(X)} \right\|^2}{\left\| \overrightarrow{X} V^{(X)} - e^{(X)} r_1^{(X)} \right\|^2}, \quad \underset{(V^{(Y)}, r_1^{(Y)})\neq 0}{Max} \frac{\left\| \overrightarrow{X} V^{(Y)} - e^{(Y)} r_1^{(Y)} \right\|^2}{\left\| \overrightarrow{Y} V^{(Y)} - e^{(Y)} r_1^{(Y)} \right\|^2}, \quad (29)$$

where $(V^{(X)}, r_1^{(X)}) \neq 0 \Rightarrow \overrightarrow{X} V^{(X)} - e^{(X)} r_1^{(X)} \neq 0$ and $\left(V^{(Y)}, r_1^{(Y)}\right) \neq 0 \Rightarrow \overrightarrow{Y} V^{(Y)} - e^{(Y)} r_1^{(Y)} \neq 0$. It is obvious that the new-formulated optimization problems in Eq. (29) are equivalent to the GEPSVM optimization problems. It is noted that, with the constraints that vectors $U^{(X)}$ and $U^{(Y)}$ are pre-initialized, then the problems in Eq. (29) can be computed independently only depending on the training data points.

Once the projection axes $V^{(X)}$ and $V^{(Y)}$ are computed, we similarly let $\widehat{X}_i = X_i V^{(X)}$, $\widehat{Y}_i = Y_i V^{(Y)}$. Thus, the tensor classifiers can be transformed to the following linear classifiers in vector space and are rewritten as follows:

$$f(\widehat{X}) = \widehat{X}^{\mathrm{T}} U^{(X)} - r_2^{(X)}, \quad f(\widehat{Y}) = \widehat{Y}^{\mathrm{T}} U^{(X)} - r_2^{(Y)}. \quad (30)$$

By a complete analogous argument, we formulate the similar problems in Eq. (31) to determine $\left(U^{(X)}, r_2^{(X)}\right)$ and $\left(U^{(Y)}, r_2^{(Y)}\right)$ for constructing the two proximal planes in Eq. (30). Therefore, the basis vectors $U^{(X)}$ and $U^{(Y)}$ can be obtained by solving the following two optimization problems Eqs. (22) and (23):

$$\underset{(U^{(X)}, r_2^{(X)})\neq 0}{Max} \frac{\left\| \widehat{Y}^{\mathrm{T}} U^{(X)} - e^{(X)} r_2^{(X)} \right\|^2}{\left\| \widehat{X}^{\mathrm{T}} U^{(X)} - e^{(X)} r_2^{(X)} \right\|^2}, \quad \underset{(U^{(Y)}, r_2^{(Y)})\neq 0}{Max} \frac{\left\| \widehat{X}^{\mathrm{T}} U^{(Y)} - e^{(Y)} r_2^{(Y)} \right\|^2}{\left\| \widehat{Y}^{\mathrm{T}} U^{(Y)} - e^{(Y)} r_2^{(Y)} \right\|^2}, \quad (31)$$

where $(U^{(X)}, r_2^{(X)}) \neq 0 \Rightarrow \widehat{X}^{\mathrm{T}} U^{(X)} - e^{(X)} r_2^{(X)} \neq 0$ and $\left(U^{(Y)}, r_2^{(Y)}\right) \neq 0 \Rightarrow \widehat{Y}^{\mathrm{T}} U^{(Y)} - e^{(Y)} r_2^{(Y)} \neq 0$. We similarly observe that the problems in Eq. (31) are also equivalent to the standard GEPSVM optimization problems. Next we detail the computational step of the first tensor plane $(U^{(X)})^{\mathrm{T}} X V^{(X)} - r^{(X)} = 0$ as an example.

In this paper, we apply the margin maximization criterion which has the advantages of reasonable motivation in principle and simplicity, for optimizing the M^{3}PSTM problems. That is, finally we can formulate the optimization problems based on standard and scale-reduced eigenvalue problems. It is important to notice that the small sample size problem does not exist since the inverse matrix operation is avoided effectively if the margin maximization criterion is employed. So as to obtain the first tensor plane, i.e., $(U^{(X)})^{\mathrm{T}} X V^{(X)} - r^{(X)} = 0$, from solving the problems of Eqs. (26) and (27), it is equivalent to computing the first pair of basis vectors $U^{(X)}$ and $V^{(X)}$ for determining the first tensor plane in the left formula of Eq. (21). We similarly define

$$\begin{aligned} g^{(X)} &= [\,\overrightarrow{Y} \quad -e^{(Y)}\,]^{\mathrm{T}}[\,\overrightarrow{Y} \quad -e^{(Y)}\,], \\ h^{(X)} &= [\,\overrightarrow{X} \quad -e^{(X)}\,]^{\mathrm{T}}[\,\overrightarrow{X} \quad -e^{(X)}\,], \\ g^{(Y)} &= [\,\widehat{Y}^{\mathrm{T}} \quad -e^{(Y)}\,]^{\mathrm{T}}[\,\widehat{Y}^{\mathrm{T}} \quad -e^{(Y)}\,], \\ h^{(Y)} &= [\,\widehat{X}^{\mathrm{T}} \quad -e^{(X)}\,]^{\mathrm{T}}[\,\widehat{X}^{\mathrm{T}} \quad -e^{(X)}\,], \\ k^{(X)} &= \begin{bmatrix} V^{(X)} \\ r_1^{(X)} \end{bmatrix}, \quad k^{(Y)} = \begin{bmatrix} U^{(X)} \\ r_2^{(X)} \end{bmatrix}, \end{aligned} \quad (32)$$

where $g^{(X)}$, $h^{(X)}$, $g^{(Y)}$ and $h^{(Y)}$ are symmetric matrices in $R^{(n_1+1)\times(n_1+1)}$ or $R^{(n_2+1)\times(n_2+1)}$. Thus, the vectors $U^{(X)}$ and $V^{(X)}$ can be obtained by the eigenvectors corresponding to the largest eigenvalues of two generalized maximum margin criterion based standard and

scale-reduced eigenvalue problems as maximizing $M(k^{(X)})$ and $N(k^{(Y)})$ with respect to the orthogonal constraints $(k^{(X)})^{\mathrm{T}}k^{(X)}=1$ and $(k^{(Y)})^{\mathrm{T}}k^{(Y)}=1$, where

$$\underset{k^{(X)}\neq 0}{Max}\, M(k^{(X)}) = (k^{(X)})^{\mathrm{T}}((1-\ell^{\dagger})g^{(X)} - \ell^{\dagger}h^{(X)})k^{(X)}, \tag{33}$$

$$\underset{k^{(Y)}\neq 0}{Max}\, N(k^{(Y)}) = (k^{(Y)})^{\mathrm{T}}((1-\ell^{\dagger})g^{(Y)} - \ell^{\dagger}h^{(Y)})k^{(Y)}. \tag{34}$$

In other words, M^{3}PSTM seeks $k^{(X)}$ and $k^{(Y)}$ such that points of dataset X should be close to the two planes $\overrightarrow{X}^{\mathrm{T}}V^{(X)} - r^{(X)} = 0$ and $\widehat{X}^{\mathrm{T}}U^{(X)} - r^{(X)} = 0$ (i.e., $(k^{(X)})^{\mathrm{T}}h^{(X)}k^{(X)}$ and $(k^{(Y)})^{\mathrm{T}}h^{(Y)}k^{(Y)}$ should be minimized) and the data samples of dataset Y should be far away from the planes (i.e., the terms $(k^{(X)})^{\mathrm{T}}g^{(X)}k^{(X)}$ and $(k^{(Y)})^{\mathrm{T}}g^{(Y)}k^{(Y)}$ will be maximized). The maximum of the problem in Eq. (33) or Eq. (34) is obtained by an eigenvector of the standard eigenvalue problem in Eq. (33) or Eq. (34) corresponding to the largest eigenvalue $\lambda_1^{(X)}$ or $\lambda_2^{(X)}$. If we represent the eigenvector by $k_1^{(X)}$ or $k_1^{(Y)}$, then $k_1^{(X)} = \begin{bmatrix}(U^{(X)})^{\mathrm{T}} & r_1^{(X)}\end{bmatrix}^{\mathrm{T}}$ and $k_1^{(Y)} = \begin{bmatrix}(V^{(X)})^{\mathrm{T}} & r_2^{(X)}\end{bmatrix}^{\mathrm{T}}$ will determine the tensor plane $(U^{(X)})^{\mathrm{T}}XV^{(X)} - r^{(X)} = 0$, which is closest to the sample data points in the dataset X and furthest away from the data points in the dataset Y. Thus the first proximal tensor plane in the left of Eq. (21) can be solved by the MATLAB (MATLAB, 1994–2001) commands: $eig((1-\ell^{\dagger})\ g^{(X)} - \ell^{\dagger}h^{(X)})$ and $eig((1-\ell^{\dagger})g^{(Y)} - \ell^{\dagger}h^{(Y)})$, each of which can produce n_1+1 or n_2+1 eigenvalues and eigenvectors of the margin maximization optimization problems. Note that term $r^{(X)}$ in the first tensor plane is given by $\left(r_1^{(X)} + r_2^{(X)}\right)/2$. By an complete similar argument, we can employ the similar method to achieve $U^{(Y)}$, $V^{(Y)}$ and $r^{(Y)}$ from the maximum margin problems in Eq. (23) and then determine the second proximal tensor plane in the right formula of Eq. (21) which is closest to the points of dataset Y and furthest from the sample points in dataset X. Based on the advantages of MMC, i.e., reasonable motivation in principle and simplicity, the optimization problems of M^{3}PSTM can be effectively and steadily solved.

After vectors $U^{(X)}$ and $V^{(X)}$, $U^{(Y)}$ and $V^{(Y)}$ are obtained, the training stage of M^{3}PSTM is completed. For the testing phase, the class label of a new coming image z will be determined by the following decision rules:

z has the same class label as

$$\times \begin{cases} X, & \text{if } \|U^{(X)\mathrm{T}}zV^{(X)} - r^{(X)}\| < \|U^{(Y)\mathrm{T}}zV^{(Y)} - r^{(Y)}\|, \\ Y, & \text{if } \|U^{(X)\mathrm{T}}zV^{(X)} - r^{(X)}\| > \|U^{(Y)\mathrm{T}}zV^{(Y)} - r^{(Y)}\|, \\ X \text{ or } Y, & \text{if } \|U^{(X)\mathrm{T}}zV^{(X)} - r^{(X)}\| = \|U^{(Y)\mathrm{T}}zV^{(Y)} - r^{(Y)}\|, \end{cases} \tag{35}$$

Table 1
Maximum Margin Multisurface Proximal Support Tensor Machine (M^{3}PSTM) algorithm.

Input:	Image data or 2^{nd} order tensor$X \in R^{n_1\times n_2}$ and $Y \in R^{n_1\times n_2}$
Output:	Two $(d_1\times d_2)$-dimensional tensor spaces $U^{(X)}\otimes V^{(X)}$ and $U^{(Y)}\otimes V^{(Y)}$
Step 1:	Compute matrices $G^{(X)}$, $H^{(X)}$, $G^{(Y)}$, $H^{(Y)}$, $K^{(X)}$, $K^{(Y)}$ defined in Eq. (24)
Step 2:	Initial $U^{(X)}$ and $U^{(Y)}$ to the vectors with all ones in $R^{(n_1+1)\times(n_1+1)}$ or $R^{(n_2+1)\times(n_2+1)}$
Step 3:	Linear planes $f(\overrightarrow{X}) = \overrightarrow{X}^{\mathrm{T}}V^{(X)} - r_1^{(X)}$ and $f(\overrightarrow{Y}) = \overrightarrow{Y}^{\mathrm{T}}V^{(Y)} - r_1^{(Y)}$ are obtained, where $V^{(X)}, r_1^{(X)}, V^{(Y)}$ and $r_1^{(Y)}$ are obtained by using eigen-decomposition of $((1-\ell^{\dagger})G^{(X)} - \ell^{\dagger}H^{(X)})$
Step 4:	Compute matrices $g^{(X)}$, $h^{(X)}$, $g^{(Y)}$, $h^{(Y)}$, $k^{(X)}$, $k^{(Y)}$ defined in Eq. (32)
Step 5:	Define $\widehat{X_i} = X_i^{\mathrm{T}}V^{(X)}$, $\widehat{Y_i} = Y_i^{\mathrm{T}}V^{(Y)}$ and obtain the linear planes $f(\widehat{X}) = \widehat{X}^{\mathrm{T}}U^{(X)} - r_2^{(X)}$ and $f(\widehat{Y}) = \widehat{Y}^{\mathrm{T}}U^{(X)} - r_2^{(Y)}$, where $U^{(X)}, r_2^{(X)}, U^{(Y)}$ and $r_2^{(Y)}$ are obtained by conducting eigen- decomposition of $((1-\ell^{\dagger})g^{(X)} - \ell^{\dagger}h^{(X)})$
Step 6:	Design tensor classifiers $f(X) = \langle X, U^{(X)}(V^{(X)})^{\mathrm{T}}\rangle - r^{(X)}$ and $f(Y) = \langle Y, U^{(Y)}(V^{(Y)})^{\mathrm{T}}\rangle - r^{(Y)}$

where $\|U^{(X)\mathrm{T}}zV^{(X)} - r^{(X)}\|$ is the tensor represented distance between the sample z and the first tensor plane in Eq. (21). Similarly, $\|U^{(Y)\mathrm{T}}zV^{(Y)} - r^{(Y)}\|$ denotes the tensor represented distance between sample z and the second tensor plane in Eq. (21). The implementation procedures of the M^{3}PSTM algorithm are detailed in Table 1.

4. Maximum margin criterion based Multi-weight Vector Projection Support Tensor Machine (M^{3}VSTM)

4.1. The objective function

In this section, we present the problem of M3VSTM, which relies in the tensorized representation for MVSVM and aims at computing two pair of transforming axes, i.e., $(U^{(X)},V^{(X)})$ and $(U^{(Y)},V^{(Y)})$, for representing datasets X and Y, one for each dataset. Based on tensor representations, each pattern X_i, Y_i of X, Y is represented as tensors $\widetilde{X}_i = U^{(X)\mathrm{T}}X_iV^{(X)}$, $\widetilde{Y}_i = U^{(X)\mathrm{T}}Y_iV^{(X)}$ by using $U^{(X)}$ and $V^{(X)}$. Similarly, each pattern X_i, Y_i of the datasets X, Y can be represented as tensors $\widehat{X}_i = U^{(Y)\ \mathrm{T}}X_iV^{(Y)}$, $\widehat{Y}_i = U^{(Y)\mathrm{T}}Y_iV^{(Y)}$ by using by using $U^{(Y)}$ and $V^{(Y)}$. According to Cai, He, and Han (2009), we can have $\|uv^{\mathrm{T}}\|^2 = tr(uv^{\mathrm{T}}vu^{\mathrm{T}}) = (v^{\mathrm{T}}v)tr(uu^{\mathrm{T}}) = (v^{\mathrm{T}}v)(u^{\mathrm{T}}u)$, thus the optimization problems in Eqs. (5) and (6) can be transformed into the following generalized maximum margin criterion based variant problems:

$$\underset{U^{(X)},V^{(X)}}{Max}\,(1-\ell^{\dagger})\left\|\frac{1}{m_2}\sum_{i=1}^{m_2}\widetilde{Y}_i - \frac{1}{m_1}\sum_{j=1}^{m_1}\widetilde{X}_j\right\|^2 - \ell^{\dagger}\sum_{i=1}^{m_1}\left\|\widetilde{X}_i - \frac{1}{m_1}\sum_{j=1}^{m_1}\widetilde{X}_j\right\|^2,$$
$$s.t.\ \|U^{(X)}V^{(X)\mathrm{T}}\|^2 = (V^{(X)\mathrm{T}}V^{(X)})(U^{(X)\mathrm{T}}U^{(X)}) = 1, \tag{36}$$

$$\underset{U^{(Y)},V^{(Y)}}{Max}\,(1-\ell^{\dagger})\left\|\frac{1}{m_1}\sum_{i=1}^{m_1} - \frac{1}{m_2}\sum_{j=1}^{m_2}\right\|^2 - \ell^{\dagger}\sum_{i=1}^{m_2}\left\| - \frac{1}{m_2}\sum_{j=1}^{m_2}\right\|,$$
$$s.t.\ \|U^{(Y)}V^{(Y)\mathrm{T}}\|^2 = (V^{(Y)\mathrm{T}}V^{(Y)})(U^{(Y)\mathrm{T}}U^{(Y)}) = 1. \tag{37}$$

Similarly, in M^{3}VSTM, the intra-class scatter difference is measured by minimizing the tensoried distances between tensorized representations of intra-class points in the $(U^{(X)},V^{(X)})$-space and their sample means. Also, M^{3}VSTM aims at optimizing the inter-class scatters by maximizing the scatter difference between the tensorized representations of the class means of different classes. By substituting $\wp^{(1)}$, $\wp^{(2)}$ and $\wp^{(3)}$ in Eq. (7) into Eqs. (36) and (37), we can have the following problems:

$$\underset{U^{(X)},V^{(X)}}{Max}\,(1-\ell^{\dagger})U^{(X)\mathrm{T}}\wp^{(1)}V^{(X)} - \ell^{\dagger}U^{(X)\mathrm{T}}\wp^{(2)}V^{(X)},$$
$$s.t.\ (V^{(X)\mathrm{T}}V^{(X)})(U^{(X)\mathrm{T}}U^{(X)}) - 1 = 0, \tag{38}$$

$$\underset{U^{(Y)},V^{(Y)}}{Max}\,(1-\ell^{\dagger})U^{(Y)\mathrm{T}}\wp^{(1)}V^{(Y)} - \ell^{\dagger}U^{(Y)\mathrm{T}}\wp^{(3)}V^{(Y)},$$
$$s.t.\ (V^{(Y)\mathrm{T}}V^{(Y)})(U^{(X)\mathrm{T}}U^{(Y)}) - 1 = 0. \tag{39}$$

It is noted that, when solving vectors $U^{(X)}$, $V^{(X)}$, $U^{(Y)}$, $V^{(Y)}$, the above two problems are dependent with each other as well, thus they can not be solved independently. Next we show the computational method.

4.2. Computational analysis

We firstly show how to compute the vectors $U^{(X)}$ and $V^{(X)}$ for representing the data points belonging to the dataset X from the problem in Eq. (36). Since $\|A\|^2 = tr(A^{\mathrm{T}}A)$, we can obtain

$$
\begin{aligned}
&U^{(X)\mathrm{T}}\wp^{(1)}V^{(X)}\\
&= tr\left[\left(\frac{1}{m_2}\sum_{i=1}^{m_2}\widetilde{Y}_i-\frac{1}{m_1}\sum_{j=1}^{m_1}\widetilde{X}_j\right)^{\mathrm{T}}\left(\frac{1}{m_2}\sum_{i=1}^{m_2}\widetilde{Y}_i-\frac{1}{m_1}\sum_{j=1}^{m_1}\widetilde{X}_j\right)\right]\\
&= tr\left[\frac{1}{m_2^2}\sum_{i=1}^{m_2}\sum_{j=1}^{m_2}\widetilde{Y}_i^{\mathrm{T}}\widetilde{Y}_j+\frac{1}{m_1^2}\sum_{i=1}^{m_1}\sum_{j=1}^{m_1}\widetilde{X}_i^{\mathrm{T}}\widetilde{X}_j\right.\\
&\quad\left.-\frac{1}{m_1m_2}\sum_{i=1}^{m_2}\sum_{j=1}^{m_1}\widetilde{Y}_i^{\mathrm{T}}\widetilde{X}_j-\frac{1}{m_1m_2}\sum_{j=1}^{m_1}\sum_{i=1}^{m_2}\widetilde{X}_j^{\mathrm{T}}\widetilde{Y}_i\right]\\
&= tr\left[\frac{1}{m_2^2}\sum_{i=1}^{m_2}\sum_{j=1}^{m_2}V^{(X)\mathrm{T}}Y_i^{\mathrm{T}}U^{(X)}U^{(X)\mathrm{T}}Y_jV^{(X)}\right.\\
&\quad+\frac{1}{m_1^2}\sum_{i=1}^{m_1}\sum_{j=1}^{m_1}V^{(X)\mathrm{T}}X_i^{\mathrm{T}}U^{(X)}U^{(X)\mathrm{T}}X_jV^{(X)}\Big]\\
&\quad-tr\left[\frac{1}{m_1m_2}\sum_{j=1}^{m_1}\sum_{i=1}^{m_2}V^{(X)\mathrm{T}}Y_i^{\mathrm{T}}U^{(X)}U^{(X)\mathrm{T}}X_jV^{(X)}\right.\\
&\quad\left.+\frac{1}{m_1m_2}\sum_{i=1}^{m_2}\sum_{j=1}^{m_1}V^{(X)\mathrm{T}}X_j^{\mathrm{T}}U^{(X)}U^{(X)\mathrm{T}}Y_iV^{(X)}\right]\\
&= tr\left[V^{(X)\mathrm{T}}\left(\partial_1^{(U)}+\partial_2^{(U)}-\partial_3^{(U)}-\partial_4^{(U)}\right)V^{(X)}\right],
\end{aligned}
\tag{40}
$$

where $\partial_1^{(U)}=\frac{1}{m_2^2}\sum_{i=1}^{m_2}\sum_{j=1}^{m_2}Y_i^{\mathrm{T}}U^{(X)}U^{(X)\mathrm{T}}Y_j$, $\partial_2^{(U)}=\frac{1}{m_1^2}\sum_{i=1}^{m_1}\sum_{j=1}^{m_1}X_i^{\mathrm{T}}U^{(X)}U^{(X)\mathrm{T}}X_j$, $\partial_3^{(U)}=\frac{1}{m_1m_2}\sum_{j=1}^{m_1}\sum_{i=1}^{m_2}Y_i^{\mathrm{T}}U^{(X)}U^{(X)\mathrm{T}}X_j$ and $\partial_4^{(U)}=\frac{1}{m_1m_2}\sum_{i=1}^{m_2}\sum_{j=1}^{m_1}X_j^{\mathrm{T}}U^{(X)}\ \ U^{(X)\mathrm{T}}Y_i$. Similarly, we have

$$
\begin{aligned}
&U^{(X)\mathrm{T}}\wp^{(2)}V^{(X)}\\
&=\sum_{i=1}^{m_1}tr\left[\left(\widetilde{X}_i-\frac{1}{m_1}\sum_{j=1}^{m_1}\widetilde{X}_j\right)^{\mathrm{T}}\left(\widetilde{X}_i-\frac{1}{m_1}\sum_{j=1}^{m_1}\widetilde{X}_j\right)\right]\\
&= tr\left[\sum_{i=1}^{m_1}\widetilde{X}_i^{\mathrm{T}}\widetilde{X}_i-\frac{1}{m_1}\sum_{i,j=1}^{m_1}\widetilde{X}_j^{\mathrm{T}}\widetilde{X}_i\right]\\
&= tr\left[\sum_{i=1}^{m_1}V^{(X)\mathrm{T}}X_i^{\mathrm{T}}U^{(X)}U^{(X)\mathrm{T}}X_iV^{(X)}-\frac{1}{m_1}\sum_{i,j=1}^{m_1}V^{(X)\mathrm{T}}X_j^{\mathrm{T}}U^{(X)}U^{(X)\mathrm{T}}X_iV^{(X)}\right]\\
&= tr\left[V^{(X)\mathrm{T}}\left(\vartheta_1^{(U)}-\vartheta_2^{(U)}\right)V^{(X)}\right],
\end{aligned}
\tag{41}
$$

where $\vartheta_1^{(U)}=\sum_{i=1}^{m_1}X_i^{\mathrm{T}}U^{(X)}U^{(X)\mathrm{T}}X_i$, $\vartheta_2^{(U)}=\frac{1}{m_1}\sum_{i,j=1}^{m_1}X_j^{\mathrm{T}}U^{(X)}U^{(X)\mathrm{T}}X_i$. If $U^{(X)}$ is set to be vector with all ones, we similarly let $\overrightarrow{X_i}=X_i^{\mathrm{T}}U^{(X)}$ and $\overrightarrow{Y_i}=Y_i^{\mathrm{T}}U^{(X)}$, then $\partial_1^{(U)}$, $\partial_3^{(U)}$, $\partial_3^{(U)}$, $\partial_4^{(U)}$, $\vartheta_1^{(U)}$ and $\vartheta_2^{(U)}$ can be simplified independently only depending on the training data points and are respectively given as

$$
\partial_1^{(U)}=\frac{1}{m_2^2}\sum_{i=1}^{m_2}\sum_{j=1}^{m_2}\overrightarrow{Y_i}\overrightarrow{Y_j}^{\mathrm{T}},\quad \partial_2^{(U)}=\frac{1}{m_1^2}\sum_{i=1}^{m_1}\sum_{j=1}^{m_1}\overrightarrow{X_i}\overrightarrow{X_j}^{\mathrm{T}},
$$

$$
\partial_3^{(U)}=\frac{1}{m_1m_2}\sum_{j=1}^{m_1}\sum_{i=1}^{m_2}\overrightarrow{Y_i}^{\mathrm{T}}\overrightarrow{X_j}^{\mathrm{T}},
$$

$$
\partial_4^{(U)}=\frac{1}{m_1m_2}\sum_{i=1}^{m_2}\sum_{j=1}^{m_1}\overrightarrow{X_j}\overrightarrow{Y_i}^{\mathrm{T}},\quad \vartheta_1^{(U)}=\sum_{i=1}^{m_1}\overrightarrow{X_i}\overrightarrow{X_i}^{\mathrm{T}},\quad \vartheta_2^{(U)}=\frac{1}{m_1}\sum_{i,j=1}^{m_1}\overrightarrow{X_j}\overrightarrow{X_i}^{\mathrm{T}}.
$$

Thus the transforming basis vector $V^{(X)}$ can be computed by solving the following typical eigen-problem:

$$
\left((1-\ell^{\dagger})\left(\partial_1^{(U)}+\partial_2^{(U)}-\partial_3^{(U)}-\partial_4^{(U)}\right)-\ell^{\dagger}\left(\vartheta_1^{(U)}-\vartheta_2^{(U)}\right)\right)v=\lambda^{(v)}v,
\tag{42}
$$

from which the optimal $V^{(X)}$ is selected as the eigenvector corresponding to the biggest eigenvalue of the above eigen-problem. Due to the property of the matrix trace, $\|A\|^2=tr(AA^{\mathrm{T}})$, thus we can similarly obtain

$$
\begin{aligned}
&U^{(X)\mathrm{T}}\wp^{(1)}V^{(X)}\\
&= tr\left[\left(\frac{1}{m_2}\sum_{i=1}^{m_2}\widetilde{Y}_i-\frac{1}{m_1}\sum_{j=1}^{m_1}\widetilde{X}_j\right)\left(\frac{1}{m_2}\sum_{i=1}^{m_2}\widetilde{Y}_i-\frac{1}{m_1}\sum_{j=1}^{m_1}\widetilde{X}_j\right)^{\mathrm{T}}\right]\\
&= tr\left[\frac{1}{m_2^2}\sum_{i=1}^{m_2}\sum_{j=1}^{m_2}\widetilde{Y}_i\widetilde{Y}_j^{\mathrm{T}}+\frac{1}{m_1^2}\sum_{i=1}^{m_1}\sum_{j=1}^{m_1}\widetilde{X}_i\widetilde{X}_j^{\mathrm{T}}-\frac{1}{m_1m_2}\sum_{i=1}^{m_2}\sum_{j=1}^{m_1}\widetilde{Y}_i\widetilde{X}_j^{\mathrm{T}}\right.\\
&\quad\left.-\frac{1}{m_1m_2}\sum_{j=1}^{m_1}\sum_{i=1}^{m_2}\widetilde{X}_j\widetilde{Y}_i^{\mathrm{T}}\right]\\
&= tr\left[U^{(X)\mathrm{T}}\left(\partial_1^{(V)}+\partial_2^{(V)}-\partial_3^{(V)}-\partial_4^{(V)}\right)U^{(X)}\right],
\end{aligned}
\tag{43}
$$

where $\partial_1^{(V)}=\frac{1}{m_2^2}\sum_{i=1}^{m_2}\sum_{j=1}^{m_2}Y_iV^{(X)}V^{(X)\mathrm{T}}Y_j^{\mathrm{T}}$, $\partial_2^{(V)}=\frac{1}{m_1^2}\sum_{i=1}^{m_1}\sum_{j=1}^{m_1}X_iV^{(X)}V^{(X)\mathrm{T}}X_j^{\mathrm{T}}$, $\partial_3^{(V)}=\frac{1}{m_1m_2}\sum_{j=1}^{m_1}\sum_{i=1}^{m_2}Y_iV^{(X)}V^{(X)\mathrm{T}}X_j^{\mathrm{T}}$ and $\partial_4^{(U)}=\frac{1}{m_1m_2}\sum_{i=1}^{m_2}\sum_{j=1}^{m_1}X_jV^{(X)}\ \ V^{(X)\mathrm{T}}Y_i^{\mathrm{T}}$. Similarly, we can have

$$
\begin{aligned}
U^{(X)\mathrm{T}}\wp^{(2)}V^{(X)}&=\sum_{i=1}^{m_1}tr\left[\left(\widetilde{X}_i-\frac{1}{m_1}\sum_{j=1}^{m_1}\widetilde{X}_j\right)\left(\widetilde{X}_i-\frac{1}{m_1}\sum_{j=1}^{m_1}\widetilde{X}_j\right)^{\mathrm{T}}\right]\\
&= tr\left[\sum_{i=1}^{m_1}\widetilde{X}_i\widetilde{X}_i^{\mathrm{T}}-\frac{1}{m_1}\sum_{i,j=1}^{m_1}\widetilde{X}_j\widetilde{X}_i^{\mathrm{T}}\right]\\
&= tr\left[U^{(X)\mathrm{T}}\left(\vartheta_1^{(V)}-\vartheta_2^{(V)}\right)U^{(X)}\right],
\end{aligned}
\tag{44}
$$

where $\vartheta_1^{(U)}=\sum_{i=1}^{m_1}X_iV^{(X)}V^{(X)\ \mathrm{T}}X_i^{\mathrm{T}}$, $\vartheta_2^{(U)}=\frac{1}{m_1}\sum_{i,j=1}^{m_1}X_jV^{(X)}V^{(X)\mathrm{T}}X_i^{\mathrm{T}}$. After the transforming vector $V^{(X)}$ is computed by solving Eq. (42), if we similarly set $\widehat{X}_i=X_iV^{(X)}$ and $\widehat{Y}_i=Y_iV^{(X)}$, then terms $\partial_1^{(V)}$, $\partial_3^{(V)}$, $\partial_3^{(V)}$, $\partial_4^{(V)}$, $\vartheta_1^{(V)}$ and $\vartheta_2^{(V)}$ can be simplified independently only depending on the training data points as well. Thus the projective basis vector $U^{(X)}$ can be computed by solving the following typical eigen-problem:

$$
\left((1-\ell^{\dagger})\left(\partial_1^{(V)}+\partial_2^{(V)}-\partial_3^{(V)}-\partial_4^{(V)}\right)-\ell^{\dagger}\left(\vartheta_1^{(V)}-\vartheta_2^{(V)}\right)\right)u=\lambda^{(u)}u,
\tag{45}
$$

from which the optimal $U^{(X)}$ is selected as the eigenvector corresponding to the biggest eigenvalue of the above eigen-problem. After $U^{(X)}$ and $V^{(X)}$ are obtained, we can then use them to construct the tensor space for representing the data points from the dataset X. With an entirely similar argument, the solutions $U^{(Y)}$ and $V^{(Y)}$ can be solved and then the $(U^{(Y)}, V^{(Y)})$-space can be constructed. The detailed computational issues of computing $U^{(Y)}$ and $V^{(Y)}$ are not provided due to page limitation. It is noted that the optimization problems of M^{3}VSTM are also based on the maximum margin criterion (Li et al., 2006), that is no matrix inverse operation is involved. Therefore we can steadily and effectively obtain the solutions of the M^{3}VSTM problems based on the useful properties of MMC.

After $U^{(X)}$ and $V^{(X)}$, $U^{(Y)}$ and $V^{(Y)}$ are obtained, the training stage of M^{3}VSTM is completed. For testing, the class label of a new coming image z will be determined by the following decision rules:

$$
z \text{ has the same class label as } \begin{cases} X, & \text{if } \|U^{(X)\mathrm{T}}zV^{(X)}-U^{(X)\mathrm{T}}M^{(X)}V^{(X)}\|<\|U^{(Y)\mathrm{T}}zV^{(Y)}-U^{(Y)\mathrm{T}}M^{(Y)}V^{(Y)}\|,\\ Y, & \text{if } \|U^{(X)\mathrm{T}}zV^{(X)}-U^{(X)\mathrm{T}}M^{(X)}V^{(X)}\|>\|U^{(Y)\mathrm{T}}zV^{(Y)}-U^{(Y)\mathrm{T}}M^{(Y)}V^{(Y)}\|,\\ X \text{ or } Y, & \text{if } \|U^{(X)\mathrm{T}}zV^{(X)}-U^{(X)\mathrm{T}}M^{(X)}V^{(X)}\|=\|U^{(Y)\mathrm{T}}zV^{(Y)}-U^{(Y)\mathrm{T}}M^{(Y)}V^{(Y)}\|, \end{cases}
\tag{46}
$$

where $M^{(X)}$ and $M^{(Y)}$ are mean images. $\|U^{(X)\mathrm{T}}zV^{(X)} - U^{(X)\mathrm{T}}M^{(X)}V^{(X)}\|$ denotes the tensor represented distance between sample z and the sample mean of the data points in dataset X. Similarly, $\|U^{(X)\mathrm{T}}zV^{(X)} - U^{(X)\mathrm{T}}M^{(X)}V^{(X)}\|$ denotes the tensor represented distance between sample z and the sample mean of the data points in dataset Y.

4.3. Comparison and discussion

In this section, we mainly discuss some issues related to our algorithms. We first compare the regular MVSVM and GEPSVM classification algorithms with our M^{3}PSTM and M^{3}VSTM algorithms from the following two aspects:

(1) The conventional MVSVM and GEPSVM algorithms perform based on the vector space model, taking the vectors in R^n as inputs. Different from MVSVM and GEPSVM, M^{3}PSTM and M^{3}VSTM take the second order tensors (or matrices) in $R^{n_1} \otimes R^{n_2}$ as inputs, where$n_1 \times n_2 \approx n$. For example, a vector $x \in R^n$ can be transformed by some means to a second order tensor data (or matrix) $X \in R^{n_1} \otimes R^{n_2}$.

(2) For given data matrices X and Y in different classes and n-dimensional input space R^n, then the linear classifiers of the regular GEPSVM can be represented as $f(X) = X\varpi^{(X)} - r^{(X)}$ and $f(Y) = Y\varpi^{(Y)} - r^{(Y)}$, in which there are $2(n+1)(\approx 2 \times n_1 \times n_2 + 2)$ parameters $\left(r^{(X)}, r^{(Y)}, \varpi_i^{(X)}, \varpi_i^{(Y)},\ i = 1, 2, \ldots, n\right)$. With similar argument, there are $2n$ $(\approx 2 \times n_1 \times n_2)$ parameters to estimate in MVSVM. Note that the linear classifiers of M^{3}PSTM in $R^{n_1} \otimes R^{n_2}$ can be represented as $g(X) = U^{(X)\mathrm{T}}XV^{(X)} - r^{(X)}$ and $g(Y) = U^{(Y)\mathrm{T}}YV^{(Y)} - r^{(Y)}$, where $U^{(X)}, U^{(Y)} \in R^{n_1}$ and $V^{(X)}, V^{(Y)} \in R^{n_2}$. Thus, there are total $2 \times (n_1 + n_2 + 1)$ parameters $\left(r^{(X)}, r^{(Y)}, U_i^{(X)}, U_i^{(Y)}, V_j^{(X)}, V_j^{(Y)},\ i = 1, 2, \ldots, n_1,\ j = 1, 2, \ldots, n_2\right)$ involved in M^{3}PSTM to be estimated. Similarly, the linear classifiers of M^{3}VSTM in $R^{n_1} \otimes R^{n_2}$ can be represented as $g(X) = U^{(X)\mathrm{T}}XV^{(X)} - U^{(X)\mathrm{T}}M^{(X)}V^{(X)}$ and $g(Y) = U^{(Y)\mathrm{T}}YV^{(Y)} - U^{(Y)\mathrm{T}}M^{(Y)}V^{(Y)}$, where bases $U^{(X)}, U^{(Y)} \in R^{n_1}$ and $V^{(X)}, V^{(Y)} \in R^{n_2}$. Thus, M^{3}VSTM needs to estimate $2 \times (n_1 + n_2)$ parameters $\left(U_i^{(X)}, U_i^{(Y)}, V_j^{(X)}, V_j^{(Y)},\ i = 1, 2, \ldots, n_1,\ j = 1, 2, \ldots, n_2\right)$. This property can make our proposed M^{3}PSTM and M^{3}VSTM classification algorithms computational efficiency and especially applicable for small sample size problems and the classification tasks on datasets involving high dimensionality, because tensoried algorithms, including M^{3}PSTM and M^{3}VSTM, need to estimate small number of parameters than those vectorized MVSVM and GEPSVM.

The proposed M^{3}PSTM and M^{3}VSTM approaches are based on the tensor representation, thus it is important to investigate the choice of subspace, i.e., the size of the tensor. In tensor space model, an image is represented as a tensor and each input pixel in the tensor corresponds to a feature. For an image $x \in R^n$, one can convert it to the second order tensor data $X \in R^{n_1 \times n_2}$, in which $n_1 \times n_2 \approx n$. Suppose that $n_1 \geqslant n_2$, in order to have at least n entries in the tensor while minimizing the size of the tensor, thus we have $(n_1 - 1) \times n_2 < n < n_1 \times (n_2 - 1)$ (He et al., 2005). However, there are still many different choices to determine the values of n_1 and n_2, especially when n is larger. Note that all these (n_1, n_2) combinations can be used in the tensor subspace learning methods. It should be noted that each vector itself in R^n can be considered as a second order tensor in $R^n \otimes R^1$. The validity of the proposed tensoried classifier models will be verified by extensive simulations with the benchmark UCI and real datasets.

5. Simulation results and analysis

In this section, several classification and segmentation simulations will be carried out to show the effectiveness of the proposed M^{3}PSTM and M^{3}VSTM algorithms. The system performance of M^{3}PSTM and M^{3}VSTM is compared with that of GEPSVM and MVSVM. For M^{3}PSTM and M^{3}VSTM, the sample images are represented as matrices or second order tensors. In short, the classification processes of M^{3}PSTM and M^{3}VSTM have three steps. First, we calculate the image subspace from the training set of images. Then the new images to be identified are transformed into $(d_1 \times d_2)$-dimensional tensor space. Finally, new coming images will be identified by the tensor based classifier learned from the training dataset. For the GEPSVM, MVSVM, M^{3}PSTM and M^{3}VSTM problems, each algorithm has a single parameter. The parameters will be estimated and selected by applying 10% of each training fold as a tuning set. The best parameter will be then determined by observing the performance of the three classifiers on datasets. Then the trained classifiers with the best parameter are used for measuring the testing accuracy (the ratio of the number of correctly classified test samples to that of total test samples). All the algorithms are implemented in Matlab 7.1. The eigen-problems involved in these algorithms are all solved by using the *eig* function in Matlab 7.1.We carry out the simulations on a PC with Intel (R) Core (TM) i5 CPU 650 @3.20 GHz 3.19 GHz 4G.

In this present study, seven publicly available UCI datasets from ML UCI Repository (Blake & Merz, 1998) and two real databases are evaluated. These UCI datasets include *Votes*, *Tic-Tac-Toe*, *Balance Scale*, *Iris*, *Contraceptive Method Choice* (*CMC*), *Blood Transfusion* and *Letter Image Recognition*. The real databases include the *USPS Handwritten Digits database* (Hull, 1994) and the Berkeley image segmentation database (Martin, Fowlkes, Tal, & Malik, 2001).

5.1. Classification on UCI dataset

We first employ the standard UCI datasets, that is, Votes ($D = 16$, Num = 435, $C = 2$), Tic-Tac-Toe ($D = 9$, Num = 958, $C = 2$), Balance Scale ($D = 4$, Num = 625, $C = 3$), Iris ($D = 4$, Num = 150, $C = 3$), CMC ($D = 9$, Num = 1473, $C = 3$), Blood Transfusion ($D = 4$, Num = 748, $C = 3$) and Letter Recognition ($D = 16$, Num = 2000, $C = 26$), to objectively evaluate the effectiveness of the proposed M^{3}PSTM and M^{3}VSTM algorithms by comparing their classification performance with GEPSVM and MVSVM. Where D is the number of dimensionality of the original space, Num is the number of data points, and C is the number of classes. It is noted that GEPSVM, MVSVM, M^{3}PSTM and M^{3}VSTM are originally proposed for handling two-class classification problems. In this simulation, for the Balance Scale, CMC and Blood Transfusion datasets, we merge the latter two classes into a single class to create the two-class case. For the Letter Image Recognition, we choose letters 'B' and 'C' for classification. For the Iris dataset, we create three two-class problems, i.e., (Iris Setosa vs. Iris Versicolour), (Iris Setosa vs. Iris Virginica) and (Iris Versicolour vs. Iris Virginica). For classification, the samples of the datasets are represented as vectors in GEPSVM, MVSVM. For M^{3}PSTM and M^{3}VSTM, each sample of the tested datasets will be embedded into the $(d_1 \times d_2)$-dimensional tensor space, satisfying $d_1 = d_2$. Because different training samples from each class for a fixed dataset usually lead to different levels of classification accuracies, it is difficult to compare the performance of the classifiers in a meaningful way. In the simulations, the sample points of each dataset are randomly split into training and testing sets. The classification accuracy rates and standard deviations of the four algorithms will be measured by using 10-fold cross-validation methodology and are treated as our test accuracy metric.

Table 2 summarizes the 10-fold mean classification accuracies and standard deviations over the UCI datasets. The best test record and averaged running time (we compute in seconds) are also reported in Table 2. We have the following observations from Table 2: (1) On the whole, there is an improvement in the generalization performance of M^{3}PSTM and M^{3}VSTM over the corresponding non-tensorized learning methods. In most cases, M^{3}PSTM and M^{3}VSTM deliver comparable to or better classification accuracy rate and best record than the original algorithms. Specifically, M^{3}PSTM and M^{3}VSTM significantly outperform GEPSVM and MVSVM on the Iris (Versicolour vs. Virginica), CMC and Blood datasets. (2) Considering the running time performance, M^{3}PSTM and M^{3}VSTM exhibit the comparative results to GEPSVM and MVSVM in each case.

5.2. Image classification on USPS handwritten digits

In this section, we address a classification task using the real-world *USPS Handwritten Digits database* (Hull, 1994). In this study, the publicly available dataset from URL: http://www.cs.nyu.edu/~roweis/data.html/USPSHandwrittenDigits is used for the simulations. The dataset has 11,000 examples of handwritten digits. There are 8-bit grayscale digit images of '0' through '9' and 1100 images of each digit. In the database, the size of each digit image is 16×16 pixels, with 256 grey levels per pixel. As result, each image is then represented by a 256-dimensional vector in For GEPSVM and MVSVM, and each image of the database will be represented by a (16×16)-dimensional matrix in M^{3}PSTM and M^{3}VSTM for the simulations. The optimal parameters are estimated by tuning on each training fold. Fig. 1 shows some typical sample images of digits '1', '2' and '3' from the USPS database. In the simulations, we randomly choose 200 samples from each digit for training and testing. We create nine two-class problems by using the digits, including (0 vs. 1), (1 vs. 2), (2 vs. 3), (3 vs. 4), (4 vs. 5), (5 vs. 6), (6 vs. 7), (7 vs. 8) and (8 vs. 9). We test GEPSVM, MVSVM, M^{3}PSTM and M^{3}VSTM algorithms. The classification results over different numbers of training data from each digit are reported in Fig. 2. For each fixed training sample size, the results averaged over 10 random splits of the training samples are reported as the test accuracy metric.

Observing from the results in Fig. 2, we conclude that: (1) The experimental results show again that M^{3}PSTM and M^{3}VSTM are comparable or even better than other algorithms in terms of classification accuracy in most cases. (2) The performance of all the methods varies with the increasing number of training data samples. (3) Different pairwise digits produce different accuracy trends and each method tends to perform especially well on some kind of combinations. For instance, M^{3}PSTM achieve the highest accuracy rates on the cases: (0 vs. 1), (1 vs. 2) and (3 vs. 4). M^{3}VSTM exhibits the highest accuracy rates on the cases: (2 vs. 3), (6 vs. 7). GEPSVM is capable of exhibiting the highest accuracy for the case of (4 vs. 5). For the cases of (1 vs. 2) and (6 vs. 7), M^{3}PSTM is able to deliver similar trend to GEPSVM and their accuracies are comparable. Similarly, M^{3}VSTM can obtain the comparative results to MVSVM for cases of (7 vs. 8) and (8 vs. 9). Overall, the performance of both GEPSVM and M^{3}PSTM are superior to MVSVM and M^{3}VSTM for the cases of (3 vs. 4), (4 vs. 5), (7 vs. 8) and (8 vs. 9). In contrast, the overall performance of both MVSVM and M^{3}VSTM are superior to GEPSVM and M^{3}PSTM for the cases of (6 vs. 7). Relatively, MVSVM works poorly on the cases of (1 vs. 2), (2 vs. 3) and (4 vs. 5).

Table 3 presents an overview of the means and standard deviations of the classification accuracy rate achieved by each method. The highest accuracy and average running time computed in seconds are also reported. We can obtain the following observations from Table 3. (1) The superiority of the four algorithms keeps consistent with the plotted results in Fig. 2. For example, the mean accuracy of M^{3}VSTM is 95.72 for the case of (5 vs. 6), followed by

Table 2
Performance comparisons on the seven benchmark UCI datasets.

Method	Result Simulation setting								
	Votes (D = 16, Num = 435)			Letter (D = 16, Num = 2000, B vs. C)			Tic-Tac-Toe (D = 9, Num = 958)		
	Mean ± std	Best	Time	Mean ± std	Best	Time	Mean ± std	Best	Time
GEPSVM	95.62 ± 0.0710	95.69	0.5568	63.92 ± 0.0202	67.34	0.9664	67.89 ± 0.046	73.56	0.1129
M^{3}PSTM	95.63 ± 0.0449	95.68	0.6786	64.65 ± 0.0127	69.84	1.0967	67.90 ± 0.075	74.21	0.1410
MVSVM	95.63 ± 0.0613	96.03	0.5989	84.46 ± 0.0137	85.86	1.0743	68.79 ± 0.031	75.65	0.1019
M^{3}VSTM	95.61 ± 0.0393	96.10	0.7325	85.12 ± 0.0125	86.29	1.0404	67.02 ± 0.065	76.23	0.1389
	Iris (Setosa vs. Versicolour)			Iris (Setosa vs. Virginica)			Iris (Versicolour vs. Virginica)		
	Mean ± std	Best	Time	Mean ± std	Best	Time	Mean ± std	Best	Time
GEPSVM	94.99 ± 0.0086	97.19	0.0160	99.78 ± 0.0074	99.98	0.0158	82.26 ± 0.0056	88.87	0.0129
M^{3}PSTM	97.38 ± 0.0107	98.35	0.0592	99.87 ± 0.0075	99.99	0.0564	83.82 ± 0.0075	89.00	0.0510
MVSVM	98.89 ± 0.0147	99.89	0.0147	99.98 ± 0.0032	99.99	0.0143	83.60 ± 0.0032	88.26	0.0119
M^{3}VSTM	9847 ± 0.0037	98.77	0.0366	99.96 ± 0.0022	99.99	0.0368	84.04 ± 0.0069	89.97	0.0289
	Balance scale (D = 4, Num = 625)			CMC (D = 9, Num = 1473)			Blood (D = 4, Num = 748)		
	Mean ± std	Best	Time	Mean ± std	Best	Time	Mean ± std	Best	Time
GEPSVM	89.15 ± 0.0134	94.58	0.1530	58.65 ± 0.0470	64.88	0.1630	70.48 ± 0.0709	75.26	0.1171
M^{3}PSTM	88.64 ± 0.0099	94.61	0.2479	63.76 ± 0.0219	70.65	0.3979	78.61 ± 0.0159	83.87	0.1911
MVSVM	90.13 ± 0.0078	94.86	0.0921	56.39 ± 0.0142	63.64	0.1597	66.15 ± 0.0165	73.49	0.1054
M^{3}VSTM	92.13 ± 0.0048	95.12	0.1300	62.81 ± 0.0109	71.70	0.2987	73.88 ± 0.0274	78.68	0.1744

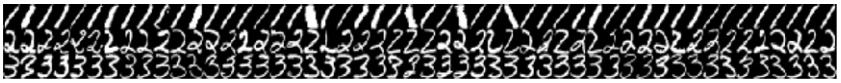

Fig. 1. Sample images of digits '1', '2' and '3' from the USPS handwritten digits database.

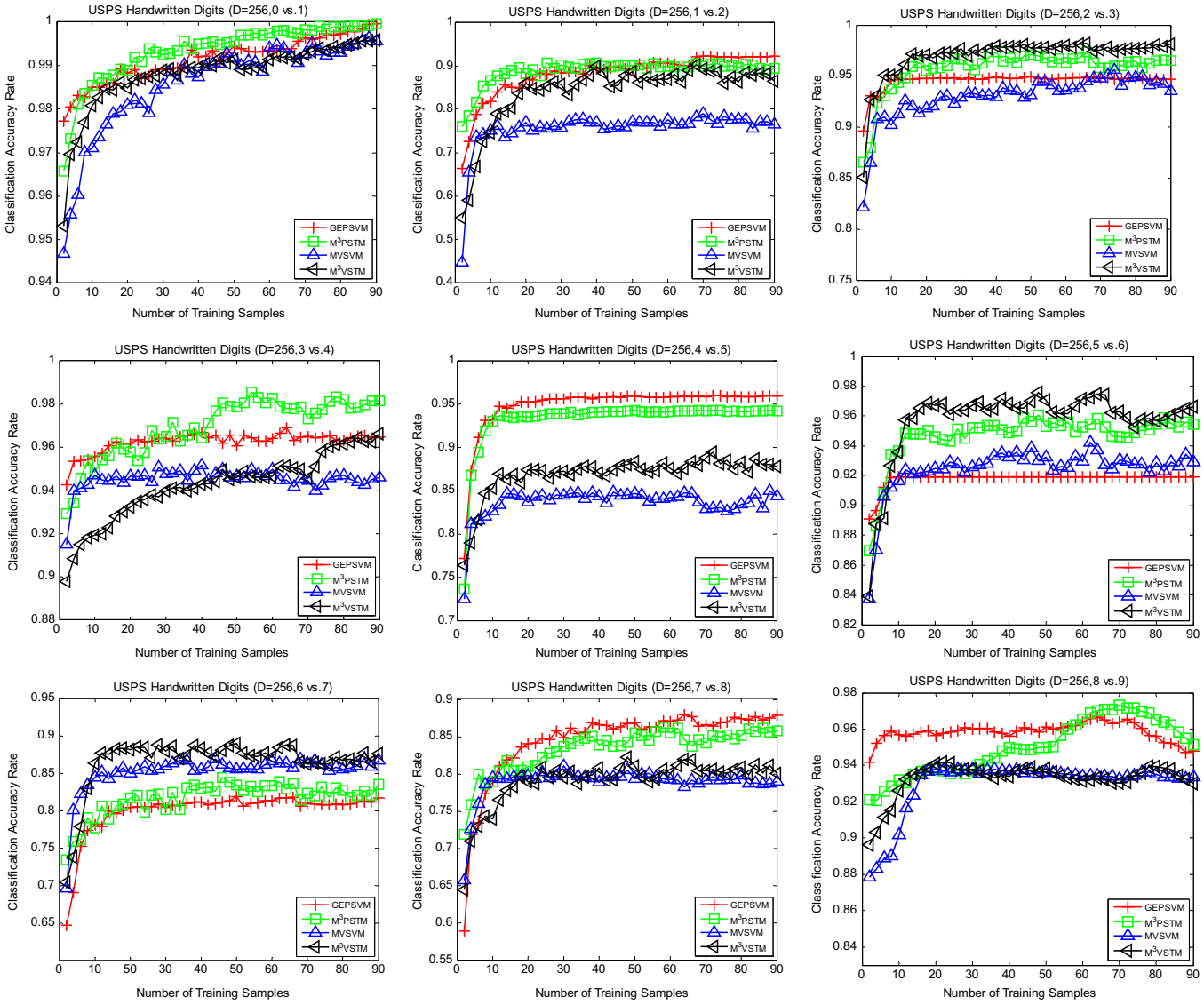

Fig. 2. The classification accuracy rates vs. different numbers of training samples on the USPS database.

Table 3
Performance comparisons on the USPS handwritten digits database.

Method	Result Simulation setting								
	USPS Digit Database (0 vs. 1)			*USPS Digit Database (1 vs. 2)*			*USPS Digit Database (2 vs. 3)*		
	Mean ± std	Best	Time	Mean ± std	Best	Time	Mean ± std	Best	Time
GEPSVM	99.16 ± 0.0011	99.96	5.1168	88.20 ± 0.0087	92.31	5.1100	94.56 ± 0.0032	94.96	5.1122
M^{3}PSTM	99.34 ± 0.0010	99.96	0.5598	88.83 ± 0.0092	90.74	0.5604	95.68 ± 0.0062	97.34	0.5926
MVSVM	98.58 ± 0.0016	99.59	5.0396	75.51 ± 0.0097	78.98	4.9156	92.96 ± 0.0057	95.58	5.0790
M^{3}VSTM	98.82 ± 0.0029	99.64	0.5236	84.14 ± 0.0106	90.30	0.5172	96.95 ± 0.0054	98.19	0.5357
	USPS Digit Database (3 vs. 4)			*USPS Digit Database (4 vs. 5)*			*USPS Digit Database (5 vs. 6)*		
	Mean ± std	Best	Time	Mean ± std	Best	Time	Mean ± std	Best	Time
GEPSVM	96.26 ± 0.0057	96.90	5.12085	94.85 ± 0.0035	95.99	5.5438	91.77 ± 0.0015	91.90	5.7811
M^{3}PSTM	96.95 ± 0.0048	98.57	0.49258	93.21 ± 0.0098	94.29	0.6517	94.68 ± 0.0065	96.06	0.6778
MVSVM	94.51 ± 0.0043	95.18	5.05227	83.59 ± 0.0067	84.98	5.2615	92.41 ± 0.0051	94.21	5.5878
M^{3}VSTM	94.24 ± 0.0063	96.62	0.54190	86.87 ± 0.0088	89.45	0.6050	95.72 ± 0.0075	97.58	0.6255
	USPS Digit Database (6 vs. 7)			*USPS Digit Database (7 vs. 8)*			*USPS Digit Database (8 vs. 9)*		
	Mean ± std	Best	Time	Mean ± std	Best	Time	Mean ± std	Best	Time
GEPSVM	79.95 ± 0.0055	81.82	5.5374	84.49 ± 0.0072	88.00	4.9508	95.84 ± 0.0050	96.67	5.1705
M^{3}PSTM	81.64 ± 0.0097	84.39	0.7049	83.35 ± 0.0102	86.19	0.6270	94.99 ± 0.0064	97.33	0.5679
MVSVM	85.22 ± 0.0087	86.94	5.2914	78.87 ± 0.0095	81.07	4.9909	92.96 ± 0.0026	93.83	5.1589
M^{3}VSTM	86.58 ± 0.0098	88.96	0.6404	79.06 ± 0.0125	82.24	0.5818	93.27 ± 0.0052	94.19	0.5241

94.68 for M^{3}PSTM, followed by 92.41 for MVSTM, and MVSTM delivers the lowest accuracy, i.e., 91.77. Correspondingly, their best records are 97.58, 96.06, 94.21 and 91.90, respectively. (2) Considering the running time performance, we find that the computational time of the M^{3}PSTM and M^{3}VSTM approaches are significantly reduced by comparing with that of GEPSVM and MVSTM in each case. This is because that by utilizing the tensor representations, the number of parameters estimated by M^{3}PSTM and M^{3}VSTM is greatly reduced, which makes the tensorized algorithms faster than those vectorized algorithms in training the learner for testing.

5.3. Application to image segmentation

In this simulation, we prepare an interactive image segmentation task using the benchmark Berkeley segmentation database (Martin et al., 2001). This task focuses on extracting the foreground objects from the natural images. Though many efforts have been made, e.g., Protiere and Sapiro (2007), Xiang, Nie, and Zhang (2008), Wang, Wang, Zhang, Shen, and Quan (2009), Ning, Zhang, Zhang, and Wu (2010), image segmentation is still a challenging problem. In handing interactive image segmentation, the most important problem is how to collect the user specified pixels about foreground and background. In this simulation, eight natural images from the Berkeley database are tested. Each extracted pixel from the images is represented by a 5-dimensional vector φ in R^5, namely $\varphi = [R,G,B,\chi,\eta]^{\mathrm{T}}$, where (R,G,B) denotes the normalized color of the pixel and (χ,η) denotes the spatial coordinate with image width and height. After the pixels are extracted, GEPSVM, MVSTM, M^{3}PSTM and M^{3}VSTM classifiers are applied to determine the class labels of the pixels. The visual measurement results of M^{3}PSTM and M^{3}VSTM are compared with those of GEPSVM, MVSTM. For the GEPSVM and MVSVM algorithms, each pixel is represented by a 5-dimensional vector. In M^{3}PSTM and M^{3}VSTM, each vector φ is considered as a second order tensor in $R^5 \otimes R^1$. Based on the obtained class labels of the pixels, the image regions are classified into foreground and background.

Fig. 3 exhibits the comparison results. The row (a) illustrates the original tested images. The row (b) shows the source images with user specified pixels, where blue and red colors indicate different segments. The rows (c) and (d) illustrate the segmentation results of GEPSVM and our M^{3}PSTM. The rows (e) and (f) exhibit the segmentation results of MVSTM and our M^{3}VSTM, respectively. Observing from the results, we see that our proposed M^{3}PSTM and M^{3}VSTM algorithms can deliver visually comparable and even better classification performance than that of the original GEPSVM and MVSTM, especially on the boundaries. Here we take the woman image as an example. We show the two segmented regions

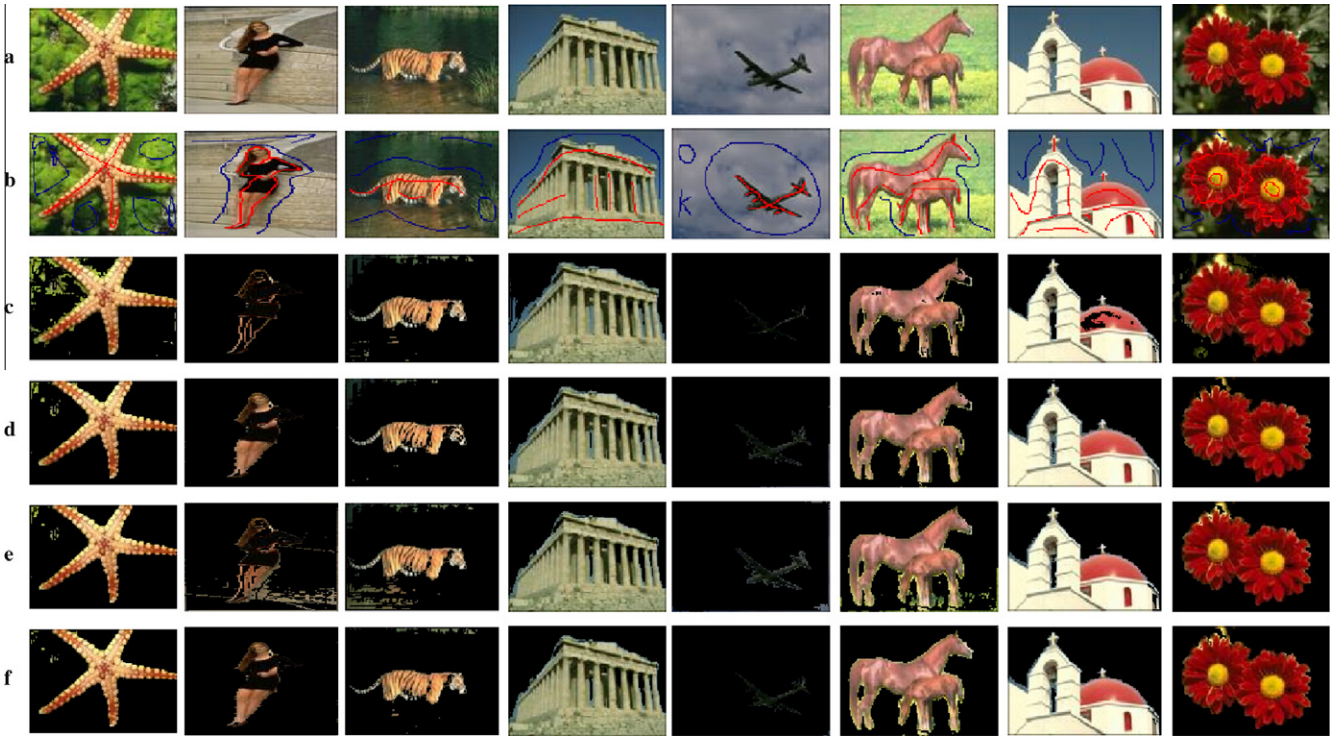

Fig. 3. Natural image segmentation results, where (a) original natural images, (b) the partially labeled images with user specified pixels denoted by different colors, (c) results of GEPSVM, (d) results of M^{3}PSTM, (e) results of MVSTM, (f) results of M^{3}VSTM.

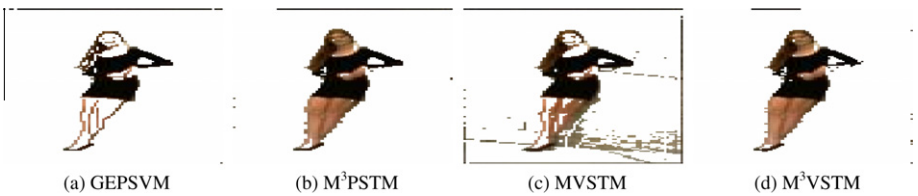

Fig. 4. Details of the two segmented regions in the woman image.

with original image resolution for performance comparison in Fig. 4. Observing from the details, we see clearly that more pixels from the foreground and background are misclassified by GEPSVM and MVSTM. On the contrary, the M^{3}PSTM and M^{3}VSTM algorithms perform better in classifying the pixels and are capable of organizing more satisfactory segmentation results.

6. Concluding remarks

This paper aims at representing the image data as the second order tensors (or, matrices) and designing new tensor classifier models for efficiently classifying and segmenting the images. In particular, we have proposed two novel classification algorithms called *Maximum Margin Multisurface Proximal Support Tensor Machine* (M^{3}PSTM) and *Maximum Margin Multi-weight Vector Projection Support Tensor Machine* (M^{3}VSTM) for learning the linear classifiers in tensor space. M^{3}PSTM and M^{3}VSTM aim at finding the maximum margin presentations of images in tensor space and aims to compute two pairs of optimal projection directions to construct two tensor hyperplanes for image classification and image segmentation. To effectively and steadily obtain the transforming basis vectors, the maximum margin criterion is employed for formulating the optimization problems. As a result, the computational process is always steady, since the matrix inverse operation or matrix singularity has been avoided. We have also discussed that the parameters in our proposed tensorized methods are greatly reduced due to the introduction of the tensorized representations. With tensor representation, M^{3}PSTM and M^{3}VSTM can successfully take into account the spatial locality of the pixels in the images. To verify the effectiveness of our proposed algorithms, thorough comparative simulations on several benchmark UCI and two real databases have been conducted. The classification results demonstrated that the proposed M^{3}PSTM and M^{3}VSTM algorithms are highly competitive with the GEPSVM and MVSVM techniques. The image segmentation results indicate that our algorithms perform better in capturing the details from the image pixels and correctly classifying the segmented regions.

Though the proposed algorithms are proved to be effective for image classification and segmentation, there are several problems sill remains to be investigated. M^{3}PSTM and M^{3}VSTM are naturally linear, so they are incapable of discovering the intrinsic nonlinear structure embedded in the image space. Thus it is interesting to consider developing kernelized M^{3}PSTM and M^{3}VSTM for classification. Recently, semi-supervised classification problem has attracted a lot of attention in machine learning area (Astorino & Fuduli, 2007; Chapelle, Sindhwani, & Keerthi, 2008). Therefore, considering extending M^{3}PSTM and M^{3}VSTM to the semi-supervised case is also a very interesting future work.

References

Astorino, A., & Fuduli, A. (2007). Nonsmooth optimization techniques for semi-supervised classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29*(12), 2135–2142.

Blake, C. L., & Merz, C. J. (1998). UCI repository of machine learning databases. <http://www.ics.uci.edu/~mlearn/MLRepository.html>.

Cai, D., He, X. F., & Han, J. W. (2009). Learning with tensor representation. Technical Report UIUCDCS- R-2006-2716.

Chapelle, O., Sindhwani, V., & Keerthi, S. S. (2008). Optimization techniques for semi-supervised support vector machines. *Journal of Machine Learning Research, 9*, 203–233.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines*. Cambridge, Mass.: Cambridge Univ. Press.

Dempster, A. P. (1971). An overview of multivariate data analysis. *Journal of Multivariate Analysis, 1*(3), 316–346.

Fu, Y., & Huang, T. S. (2008). Image classification using correlation tensor analysis. *IEEE Transactions on Image Processing, 17*(2), 226–234.

Guo, Y., Li, S., Yang, J., Shu, T., & Wu, L. (2003). A generalized Foley–Sammon transform based on generalized fisher discriminant criterion and its application to face recognition. *Pattern Recognition Letter, 24*(1–3), 147–158.

He, X., Cai, D., & Niyogi, P. (2005). Tensor subspace analysis. In *Advances in neural information processing systems, Vancouver, Canada*.

Hull, J. J. (1994). A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 16*(5), 550–554.

Jia, Y., Nie, F., & Zhang, C. (2009). Trace ratio problem revisited. *IEEE Transactions on Neural Network, 20*(4), 729–735.

Li, H., Jiang, T., & Zhang, K. (2006). Efficient and robust feature extraction by maximum margin criterion. *IEEE Transactions on Neural Networks, 17*(1), 157–165.

Liu, J., Chen, S. C., Tan, X. Y., & Zhang, D. Q. (2007). Comments on: Efficient and robust feature extraction by maximum margin criterion. *IEEE Transactions on Neural Networks, 18*(6), 1862–1864.

Mangasarian, O. L. (1999). Arbitrary-norm separating plane. *Operations Research Letters, 24*, 15–23.

Mangasarian, O. L., & Wild, E. W. (2006). Multisurface proximal support vector machine classification via generalized eigenvalues. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28*(1), 69–74.

Martin, D., Fowlkes, C., Tal, D., & Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the 8th international conference on computer vision* (pp. 416–423).

Martinez, A. M., & Kak, A. C. (2001). PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(2), 228–233.

MATLAB, User's Guide (1994–2001). The MathWorks, Inc. <http://www.mathworks.com>.

Ning, J. F., Zhang, L., Zhang, D., & Wu, C. K. (2010). Interactive image segmentation by maximal similarity based region merging. *Pattern Recognition, 43*(2), 445–456.

Parlett, B. N. (1998). *The symmetric eigenvalue problem*. Philadelphia: SIAM.

Protiere, A., & Sapiro, G. (2007). Interactive image segmentation via adaptive weighted distances. *IEEE Transactions on Image Processing, 16*(4), 1046–1057.

Vapnik, V. (1995). *The nature of statistical learning theory*. New York: Springer-Verlag.

Vasilescu, M. A. O., & Terzopoulos, D. (2003). Multilinear subspace analysis for image ensembles. In *Proceedings of the IEEE Conference on computer vision and pattern recognition* (pp. 93–99).

Wang, Z., Chen, S. C., Liu, J., & Zhang, D. Q. (2008). Pattern representation in feature extraction and classifier design: Matrix versus vector. *IEEE Transactions on Neural Networks, 19*(5), 758–769.

Wang, H., Zheng, W., Hu, Z., & Chen, S. (2007). Local and weighted maximum margin discriminant analysis. In *Proceedings of the IEEE computer vision and pattern recognition, Minneapolis, MN* (pp. 1–8).

Wang, H., Yan, S., Xu, D., Tang, X., & Huang, T. (2007). Trace ratio vs. ratio trace for dimensionality reduction. In *Proceedings of the IEEE conference on computer vision and pattern recognition, Minneapolis, MN* (pp. 1–8).

Wang, J. D., Wang, F., Zhang, C. S., Shen, H. C., & Quan, L. (2009). Linear neighborhood propagation and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 31*(3), 1600–1615.

Xiang, S., Nie, F. P., & Zhang, C. S. (2008). Learning a Mahalanobis distance metric for data clustering and classification. *Pattern Recognition, 41*(12), 3600–3612.

Xu, D., & Yan, S. (2009). Semi-supervised bilinear subspace learning. *IEEE Transactions on Image Processing, 18*(7), 1671–1676.

Yang, J., Zhang, D., Frangi, A. F., & Yang, J. Y. (2004). Two-dimensional PCA: A new approach to face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 26*(1), 131–137.

Yan, S., Xu, D., Yang, Q., Zhang, L., Tang, X., & Zhang, H. (2007). Multilinear discriminant analysis for face recognition. *IEEE Transactions on Image Processing, 16*(1), 212–220.

Ye, Q. L., Zhao, C. X., Ye, N., & Chen, Y. N. (2010). Multi-weight vector projection support vector machines. *Pattern Recognition Letters, 31*(13), 2006–2011.

Zhang, Z., & Ye, N. (2011). Learning a tensor subspace for semi-supervised dimensionality reduction. *Soft Computing, 15*(2), 383–395.

Zheng, W., Zou, C., & Zhao, L. (2005). Weighted maximum margin discriminant analysis with kernels. *Neurocomputing, 67*, 357–362.