Thinking of Images as What They Are: Compound Matrix Regression for Image Classification

Zhigang Ma
University of Trento
Italy
ma@disi.unitn.it

Yi Yang
The University of Queensland, Australia
yee.i.yang@gmail.com

Feiping Nie
University of Texas at Arlington, USA
feipingnie@gmail.com

Nicu Sebe
University of Trento
Italy
sebe@disi.unitn.it

Abstract

In this paper, we propose a new classification framework for image matrices. The approach is realized by learning two groups of classification vectors for each dimension of the image matrices. One novelty is that we utilize compound regression models in the learning process, which endows the algorithm increased degree of freedom. On top of that, we extend the two-dimensional classification method to a semi-supervised classifier which leverages both labeled and unlabeled data. A fast iterative solution is then proposed to solve the objective function. The proposed method is evaluated by several different applications. The experimental results show that our method outperforms several classification approaches. In addition, we observe that our method attains respectable classification performance even when only few labeled training samples are provided. This advantage is especially desirable for real-world problems since precisely annotated images are scarce.

1 Introduction

Several works have indicated that matrix form may be a more natural representation of an image to reflect its 2D structure [Ye *et al.*, 2004][Yang *et al.*, 2004][Kong *et al.*, 2005][Pirsiavash *et al.*, 2009][Cao *et al.*, 2013]. Typical examples include objects, faces and handwritten digits. However, most existing classification algorithms require that an image be represented by a vector, which is normally obtained by concatenating each row (or column) of an image matrix.

The vectorization of an image has some drawbacks. First, the spatial correlation of the image is broken. Second, it causes the dimensionality increase in multiplication effect, and thus heavily increased computational complexity. The curse of dimensionality additionally causes the over-fitting problem if we use a small number of examples to train the classifiers.

Aiming to preserve the correlation within the image matrix while reducing the computation complexity, researchers have proposed two-dimensional based analyzing methods for images that are better represented as matrices [Ye *et al.*, 2004][Yang *et al.*, 2004][Kong *et al.*, 2005]. A well known approach within this paradigm is the two-dimensional subspace learning based classification. This approach is normally achieved by a two-step process. First, it eliminates noise and redundancy from the original data by projecting the data into a lower dimensional subspace. Then it applies classifiers on the low dimensional data for classification. A merit is that both computational efficiency and classification accuracy can be obtained. Classical works include the two-dimensional LDA [Ye *et al.*, 2004] extended from Linear Discriminant Analysis (LDA) [Fukunaga, 1990], two-dimensional PCA [Yang *et al.*, 2004] extended from Principle Component Analysis (PCA) [Jolliffe, 2002], *etc.*

Besides these two-dimensional methods, researchers have also developed tensor subspace analysis algorithms [Vasilescu and Terzopoulos, 2003][Wang and Ahuja, 2005][He *et al.*, 2005][Xu *et al.*, 2005] which can handle even more dimensions. These methods can be readily employed for image matrices by setting the tensor order to two. Typical tensor based subspace learning methods include Tensor LGE [He *et al.*, 2005], Tensor LPP [He *et al.*, 2005] and so on.

The aforementioned methods are able to preserve the spatial correlation of an image and to avoid the curse of dimensionality. Nonetheless, for classification they require a non-convenient two-step process, *i.e.*, subspace learning followed by different classifiers. Although the first step processes image matrices directly, the classifying step still requires the data to be vectorized. Besides, the separation of subspace learning and classification does not guarantee the classifiers benefit the most from the learned subspace. Recently, Pirsiavas *et al.* [Pirsiavash *et al.*, 2009] have proposed a bilinear-SVM classifier which is able to classify image matrices in an integrated framework and Hou *et al.* have proposed a regression model for matrix data classification in [Hou *et al.*, 2013]. These approaches are encouraging, however, they need many labeled training data but labeled data are expensive to acquire. The over-fitting problem is likely to occur when the number of training data remains small. It would be more appealing if a classifier classifies image matrices with good performance by using only limited labeled training samples. This is precisely what we want to do in this paper.

Previous work has shown that if properly designed, semi-supervised learning [Zhu, 2007] is able to attain improved performance for many applications by using a small amount

of labeled training data [Cohen *et al.*, 2004][Liu *et al.*, 2013][Yang *et al.*, 2013][Ma *et al.*, 2011]. A popular approach to realize semi-supervised learning is by building the graph Laplacian. For instance, Yang *et al.* have proposed a semi-supervised ranking scheme by learning a robust Laplacian matrix through Local Regression and Global Alignment [Yang *et al.*, 2012]. Cai *et al.* have proposed a semi-supervised dimensionality reduction algorithm called Semi-supervised Discriminant Analysis (SDA) in [Cai *et al.*, 2007]. In [Fergus *et al.*, 2010], Fergus *et al.* have integrated graph Laplacian into a semi-supervised semantic label sharing method for learning with many categories. Sharma *et al.* [Sharma *et al.*, 2010] present a new constrained spectral clustering method that builds upon graph Laplacian embedding to propagate pairwise constraints and to cluster the data for shape segmentation.

Inspired by the progress of two-dimensional analysis and semi-supervised learning, we propose a new classification scheme for image matrices. The proposed method leverages two groups of classification vectors with compound regression models, and the graph Laplacian based semi-supervised learning jointly for classification. Semi-supervised learning enables our method to leverage both labeled and unlabeled data. The proposed method thereby overcomes the deficiency of [Pirsiavash *et al.*, 2009].

Another novelty of our method is that it exploits compound regression models. This is different from traditional single regression model as several regression models are integrated. In this way, these regression models work collaboratively for the learning process, which provides us with a larger space to find the optimal solution. In other words, the degree of freedom is increased. The classification performance can be enhanced subsequently.

We name the new method Semi-supervised Two-dimensional Classification (SSTC). The main contributions of our work are summarized as:

- SSTC can conduct classification directly using the data in matrix form. The image 2D structure is intact in our framework so that the spacial correlation is preserved. Additionally, the efficiency is guaranteed by avoiding the generation of high-dimensional vectors and the flexibility is increased by using compound regression models.
- Our method is implemented in the semi-supervised scenario, which utilizes both labeled and unlabeled data for classification. It is cost-saving and at the same time is able to prevent over-fitting few labeled training data.
- The experiments on different applications show that our method yields good results even when few labeled samples are available. This attribute is attractive for real-world applications since labeled training data are difficult to obtain.

2 Classifying Image Matrices

In this section, we present the formulation of our Semi-supervised Two-dimensional Classification (SSTC) framework followed by a detailed solution for solving the objective function.

2.1 Learning Two Groups of Classification Vectors

Classification aims to learn a predictor f that for an input datum x predicts an output y. For traditional classification algorithms, x is a vector representation of an image. f is decided by minimizing the following regularized empirical error based on a set of training data $\{x_i, y_i\}_{i=1}^n$ where y_i indicates the label of x_i:

$$\min_f \sum_{i=1}^{n} loss\left(f(x_i), y_i\right) + \mu\Omega(f). \tag{1}$$

$loss(\cdot)$ is a loss function and $\Omega(f)$ is the regularization function on f with μ as its parameter.

Among others, ridge regression has shown to be effective for classifying vector data. Denoting $X = [x_1, x_2, \cdots, x_n] \in \mathbb{R}^{d\times n}$ as the training set, the objective function of traditional ridge regression is then defined as

$$\min_{w_j} \sum_{j=1}^{c}\sum_{i=1}^{n} (w_j^T x_i - y_i)^2 + \mu \sum_{j=1}^{c} w_j^T w_j, \tag{2}$$

where $w_j|_{j=1}^c \in \mathbb{R}^{d\times 1}$ are used to classify each datum x_i into c different classes.

Different from traditional way which copes with vector data, we propose to classify image matrices directly. To begin with, we denote $\mathcal{X} = [X_1, X_2, \cdots, X_n] \in \mathbb{R}^{p\times q\times n}$ as the training set where $X_i \in \mathbb{R}^{p\times q}(1 \le i \le n)$ is the i-th datum and n is the total number of the training data. Let $Y = [y_1, y_2, ..., y_n] \in \{0, 1\}^{c\times n}$ be the labels. y_i^r denote the r-th datum of y_i and $y_i^r = 1$ if X_i is in the r-th class, while $y_i^r = 0$ otherwise.

Eq. (2) indicates that the training process for $w_j|_{j=1}^c$ is actually c separate procedures. In this sense, we consider classifying $X_i|_{i=1}^n$ to each class separately. Inspired by [Hou *et al.*, 2013], for the r-th class we use two groups of classification vectors $u_j|_{j=1}^m \in \mathbb{R}^{p\times 1}$ and $v_j|_{j=1}^m \in \mathbb{R}^{q\times 1}$ and rewrite Eq. (2) as:

$$\min_{u_j, v_j} \sum_{i=1}^{n} \left(\sum_{j=1}^{m} u_j^T X_i v_j - y_i^r \right)^2 + \mu \sum_{j=1}^{m} \left\| u_j v_j^T \right\|_F^2. \tag{3}$$

$m(m > 1)$ indicates that we have m compound regression models. Compared with single regression model, *i.e.*, $m = 1$, using compound regression models provides us with larger space to search for the solution.

To step further, we extend Eq. (3) to a semi-supervised scenario using the graph Laplacian. We first define an affinity matrix A whose element A_{ij} reflects the similarity between X_i and X_j as

$$A_{ij} = \begin{cases} 1 & X_i \text{ and } X_j \text{ are } k \text{ nearest neighbors;} \\ 0 & \text{otherwise.} \end{cases}$$

The graph Laplacian is then constructed through $L = D - G$ where L is the graph Laplacian matrix and D is a diagonal matrix with $D_{ii} = \sum_{j=1}^{n} A_{ij}$.

Suppose l data are labeled in the training set, *i.e.*, $\forall i > l$, $y_i|_{i=l+1}^n = 0^{c\times 1}$. For semi-supervised learning, l is usually much smaller than n. To avoid being heavily dependent

on the few labels provided with the labeled training data, we bring in a predicted label vector $f^r = [f_1^r, \ldots, f_n^r] \in \mathbb{R}^{1\times n}$ for all the training data where f_i^r is the predicted label value of $X_i \in \mathcal{X}$. f^r should be consistent with the known labels of the training data and be smooth on the graph Laplacian so it can be optimized through the following objective function [Zhu *et al.*, 2003][Zhu, 2007]:

$$\min_{f^r} Tr\left((f^r)^T L f^r\right) + Tr\left((f^r - y^r)^T S (f^r - y^r)\right), \quad (4)$$

where $Tr(\cdot)$ denotes the trace operator; S is a selection matrix whose diagonal element $S_{ii} = \infty$ if X_i is labeled and $S_{ii} = 1$ otherwise; $y^r = [y_1^r, ..., y_l^r, 0, ..., 0] \in R^{1\times n}$ is the ground truth labels of the training data.

By incorporating Eq. (4) into Eq. (3), we propose our objective function for the r-th class as:

$$\min_{u_j, v_j, f^r} f^r L (f^r)^T + (f^r - y^r) S (f^r - y^r)^T$$
$$+\lambda \sum_{i=1}^{n} \left(\sum_{j=1}^{m} u_j^T X_i v_j - f_i^r \right)^2 + \mu \sum_{j=1}^{m} \left\| u_j v_j^T \right\|_F^2. \quad (5)$$

2.2 Optimization

We propose an iterative approach to solve the objective problem of Eq. (5).

(1) Fixing $v_j|_{j=1}^m$ and optimizing $u_j|_{j=1}^m$ and f^r:

Denoting $b_i^j = X_i v_j$ and $D_v = \begin{bmatrix} v_1^T v_1 I_p & & \\ & \ldots & \\ & & v_m^T v_m I_p \end{bmatrix}$ where $I_p \in \mathbb{R}^{p\times p}$ is an identity matrix, Eq. (5) is rewritten as:

$$\min_{u_j, f^r} f^r L (f^r)^T + (f^r - y^r) S (f^r - y^r)^T$$
$$+\lambda \sum_{i=1}^{n} \left(\begin{bmatrix} u_1^T & u_2^T & \ldots & u_m^T \end{bmatrix} \begin{bmatrix} b_i^1 \\ b_i^2 \\ \ldots \\ b_i^m \end{bmatrix} - f_i^r \right)^2$$
$$+\mu \begin{bmatrix} u_1^T & u_2^T & \ldots & u_m^T \end{bmatrix} D_v \begin{bmatrix} u_1 \\ u_2 \\ \ldots \\ u_m \end{bmatrix}. \quad (6)$$

Let $u = \begin{bmatrix} u_1 \\ u_2 \\ \ldots \\ u_m \end{bmatrix} \in \mathbb{R}^{pm\times 1}$ and $B_i = \begin{bmatrix} b_i^1 \\ b_i^2 \\ \ldots \\ b_i^m \end{bmatrix} \in \mathbb{R}^{pm\times 1}$, we have:

$$\min_{u, f^r} f^r L (f^r)^T + (f^r - y^r) S (f^r - y^r)^T$$
$$+\lambda \sum_{i=1}^{n} \left(u^T B_i - f_i^r\right)^2 + \mu u^T D_v u. \quad (7)$$

Denoting $B = [B_1, B_2, ..., B_i] \in \mathbb{R}^{pm\times n}$, Eq. (7) becomes:

$$\min_{u, f^r} f^r L (f^r)^T + (f^r - y^r) S (f^r - y^r)^T + Tr(u^T G_v^{-1} u)$$
$$-2\lambda Tr(u^T B f^r) + \lambda Tr\left((f^r)^T f^r\right) \quad (8)$$

where $Tr(\cdot)$ denotes the trace operator and $G_v = (\lambda B B^T + \mu D_v)^{-1}$. Setting the derivative of Eq. (8) *w.r.t.* u to 0, we have

$$2\lambda B B^T u - 2B(f^r)^T + 2\mu D_v u = 0$$
$$\Rightarrow u = \lambda (\lambda B B^T + \mu D_v)^{-1} B (f^r)^T = \lambda G_v B (f^r)^T \quad (9)$$

Substituting u in Eq. (8) with Eq. (9), it becomes:

$$\min_{f^r} f^r L (f^r)^T + (f^r - y^r) S (f^r - y^r)^T$$
$$-\lambda^2 Tr(f^r B^T G_v B f^r) + \lambda Tr\left((f^r)^T f^r\right). \quad (10)$$

Algorithm 1: Optimizing the classification vectors $u_j|_{j=1}^m$ and $v_j|_{j=1}^m$.

Data: Training data $\mathcal{X} \in \mathbb{R}^{p\times q\times n}$
Training data labels $Y \in \mathbb{R}^{c\times n}$
Parameters λ and μ.
Result: Converged $u_j|_{j=1}^m$ and $v_j|_{j=1}^m$ for c classes.
begin
  Compute the graph Laplacian matrix $L \in \mathbb{R}^{n\times n}$
  Compute the decision matrix $S \in \mathbb{R}^{n\times n}$
  for $r \leftarrow 1$ **to** c **do**
    Set $t = 0$ and initialize $(v_j|_{j=1}^m)_t$ as $\begin{bmatrix} 0 \cdots 0 & \underset{j^{th}}{1} & 0 \cdots 0 \end{bmatrix}$
    Reformulate $(v_j|_{j=1}^m)_t$ to v_t
    $y^r = Y(r, :)$
    repeat
      Update f_{t+1}^r using (11)
      Update u_{t+1} using (12)
      Update f_{t+1}^r using (15)
      Update v_{t+1} using (16)
      $t = t + 1$.
    until *Convergence*
  Return $u_j|_{j=1}^m$ and $v_j|_{j=1}^m$ for c classes.

Setting the derivative of Eq. (10) *w.r.t.* f^r to 0, we obtain:

$$2f^r L + 2(f^r - y^r) S - 2\lambda^2 f^r B^T G_v B + 2\lambda f^r = 0$$
$$\Rightarrow f^r (L + S - \lambda^2 B^T G_v B + \lambda I_n) = y^r S$$
$$\Rightarrow f^r = y^r S E_v, \quad (11)$$

where I_n denotes an $n \times n$ identity matrix and $E_v = (L + S - \lambda^2 B^T G_v B + \lambda I_n)^{-1}$. Substituting Eq. (9) with Eq. (11), we have:

$$u = \lambda G_v B E_v S (y^r)^T \quad (12)$$

which can be easily converted to $u_j|_{j=1}^m$.

(2) Fixing $u_j|_{j=1}^m$ and optimizing $v_j|_{j=1}^m$ and f^r:

Similarly, we denote $v = \begin{bmatrix} v_1 \\ v_2 \\ \ldots \\ v_m \end{bmatrix}$, $c_i^j = X_i^T u_j$, $C_i = \begin{bmatrix} c_i^1 \\ c_i^2 \\ \ldots \\ c_i^m \end{bmatrix} \in \mathbb{R}^{qm\times 1}$ and $C = [C_1, C_2, ...C_i] \in \mathbb{R}^{qm\times n}$, $D_u =$

$\begin{bmatrix} u_1^T u_1 I_q & & \\ & \ldots & \\ & & u_m^T u_m I_q \end{bmatrix}$ where $I_q \in \mathbb{R}^{q\times q}$ is an identity matrix. Then Eq. (5) becomes:

$$\min_{v,f^r} f^r L(f^r)^T + (f^r - y^r)S(f^r - y^r)^T$$
$$+\lambda Tr\left((v^T C - f^r)^T (v^T C - f^r)\right) + \mu v^T D_u v. \quad (13)$$

Setting the derivative of Eq. (13) *w.r.t.* v to 0, it becomes

$$v = \lambda G_u C (f^r)^T, \quad (14)$$

where $G_u = (\lambda C C^T + \mu D_u)^{-1}$. By substituting v into Eq. (13) and setting its derivative *w.r.t.* f^r to 0, we get:

$$f^r = y^r S E_u, \quad (15)$$

where $E_u = (L + S - \lambda^2 C^T G_u C + \lambda I_n)^{-1}$. Consequently, we obtain

$$v = \lambda G_u C E_u S (y^r)^T \quad (16)$$

and then $v_j|_{j=1}^m$.

The optimization of $u_j|_{j=1}^m$, $v_j|_{j=1}^m$ and f^r is iterated until convergence. The detailed iteration process is given in Algorithm 1.

Once the u and v for each class are obtained, we can easily get the two groups of classification vectors $u_j|_{j=1}^m$ and $v_j|_{j=1}^m$. Then we propose Algorithm 2 to predict the labels of the testing data.

Algorithm 2: The classification process.

Data: Testing data $\mathcal{X}_{te} \in \mathbb{R}^{p\times q\times n_{te}}$
Classification vectors $u_j|_{j=1}^m$ and $v_j|_{j=1}^m$ for c classes
Result: Predicted labels $Ypre$ of the testing data.
begin
  for $i \leftarrow 1$ **to** n_{te} **do**
    for $r \leftarrow 1$ **to** c **do**
      Compute the c regression values of X_{te}^i with $\sum_{j=1}^m (u_j^r)^T X_{te}^i (v_j^r)^T \Big|_{r=1}^c$
    Predict the label of X_{te}^i as
    $Ypre_{ir} = \arg\max_r \sum_{j=1}^m (u_j^r)^T X_{te}^i (v_j^r)^T \Big|_{r=1}^c$

By the following theorem, we can verify that the proposed iterative approach in Algorithm 1 converges and the global solutions of $u_j|_{j=1}^m$ and $v_j|_{j=1}^m$ are obtained.

Theorem 1 *The objective function value shown in (5) monotonically decreases in each iteration until convergence using the iterative approach in Algorithm 1.*

To prove *Theorem 1*, we first give the following lemma:

Lemma 1 *By fixing $v_j|_{j=1}^m$, we obtain the global solutions for $u_j|_{j=1}^m$ and f^r. In the same manner, by fixing $u_j|_{j=1}^m$, we obtain the global solutions for $v_j|_{j=1}^m$ and f^r.*

Proof. By fixing $v_j|_{j=1}^m$, the objective function in Eq. (5) is converted to the problem in Eq. (8). It can be seen that Eq. (8) is a convex optimization problem *w.r.t.* $u_j|_{j=1}^m$ and f^r. Therefore, we can obtain the global solutions for $u_j|_{j=1}^m$ and f^r by setting the derivative of Eq. (8) *w.r.t.* them to zero respectively. Based on the similar theory, we also prove that by fixing $u_j|_{j=1}^m$, we obtain the global solutions for $v_j|_{j=1}^m$ and f^r □.

Next, we prove *Theorem 1* as follows:

*Proof.*Suppose after the t-th iteration, we obtain $u_j^t|_{j=1}^m$, $v_j^t|_{j=1}^m$ and f_t^r. In the next iteration, we fix $v_j|_{j=1}^m$ as $v_j^t|_{j=1}^m$ and solve for $u_j^{t+1}|_{j=1}^m$ and f_{t+1}^r. According to *Lemma 1*, we obtain:

$$\begin{aligned} &f_{t+1}^r L(f_{t+1}^r)^T + (f_{t+1}^r - y^r)S(f_{t+1}^r - y^r)^T \\ &+\lambda\sum_{i=1}^n \left(\sum_{j=1}^m (u_j^{t+1})^T X_i v_j^t - (f_i^r)_{t+1}\right)^2 + \mu\sum_{j=1}^m \left\|u_j^{t+1}(v_j^t)^T\right\|_F^2 \\ &\le f_t^r L(f_t^r)^T + (f_t^r - y^r)S(f_t^r - y^r)^T \\ &+\lambda\sum_{i=1}^n \left(\sum_{j=1}^m (u_j^t)^T X_i v_j^t - (f_i^r)_t\right)^2 + \mu\sum_{j=1}^m \left\|u_j^t(v_j^t)^T\right\|_F^2 \end{aligned} \quad (17)$$

Similarly, when we fix $u_j|_{j=1}^m$ as $u_j^t|_{j=1}^m$ the following inequality holds:

$$\begin{aligned} &f_{t+1}^r L(f_{t+1}^r)^T + (f_{t+1}^r - y^r)S(f_{t+1}^r - y^r)^T \\ &+\lambda\sum_{i=1}^n \left(\sum_{j=1}^m (u_j^t)^T X_i v_j^{t+1} - (f_i^r)_{t+1}\right)^2 + \mu\sum_{j=1}^m \left\|u_j^t(v_j^{t+1})^T\right\|_F^2 \\ &\le f_t^r L(f_t^r)^T + (f_t^r - y^r)S(f_t^r - y^r)^T \\ &+\lambda\sum_{i=1}^n \left(\sum_{j=1}^m (u_j^t)^T X_i v_j^t - (f_i^r)_t\right)^2 + \mu\sum_{j=1}^m \left\|u_j^t(v_j^t)^T\right\|_F^2 \end{aligned} \quad (18)$$

By integrating Eq. (17) and Eq. (18), we have:

$$\begin{aligned} &f_{t+1}^r L(f_{t+1}^r)^T + (f_{t+1}^r - y^r)S(f_{t+1}^r - y^r)^T \\ &+\lambda\sum_{i=1}^n \left(\sum_{j=1}^m (u_j^{t+1})^T X_i v_j^{t+1} - (f_i^r)_{t+1}\right)^2 \\ &+\mu\sum_{j=1}^m \left\|u_j^{t+1}(v_j^{t+1})^T\right\|_F^2 \\ &\le f_t^r L(f_t^r)^T + (f_t^r - y^r)S(f_t^r - y^r)^T \\ &+\lambda\sum_{i=1}^n \left(\sum_{j=1}^m (u_j^t)^T X_i v_j^t - (f_i^r)_t\right)^2 + \mu\sum_{j=1}^m \left\|u_j^t(v_j^t)^T\right\|_F^2 \end{aligned} \quad (19)$$

Eq. (19) demonstrates that the objective function value decreases after each iteration. Thus, we have proved *Theorem 1* □.

3 Experiments

In this section, we apply SSTC to facial expression recognition, face recognition, head pose estimation, object recognition and handwritten digit recognition. We compare

SSTC to 7 algorithms which are 2DLDA [Ye *et al.*, 2004], Tensor LPP (T-LPP) [He *et al.*, 2005], BilinearSVM [Pirsiavash *et al.*, 2009], TaylorBoost [Saberian *et al.*, 2011], CRC-RLS [Zhang *et al.*, 2011], Semi-supervised Discriminant Analysis (SDA) [Cai *et al.*, 2007] and Manifold Regularization (MR) [Belkin *et al.*, 2006]. The performance is evaluated by accuracy.

The settings of the experiments are as follows: 1) The grey pixel values of the images are used as features. SSTC, 2DLDA, T-LPP and BilinearSVM deal with image matrices while other methods use vectors. 2) Some parameters need to be set. One of them is the parameter k that specifies the k nearest neighbors used to compute the graph Laplacian matrix L. We fix it at 10 empirically. The number of regression models m is fixed at 3 empirically for all the experiments. The regularization parameters, denoted as λ and μ in Eq. (5), are tuned from $\{10^{-4}, 10^{-2}, ..., 10^{2}, 10^{4}\}$. We similarly tune all the regularization parameters for other methods (if any). The best results from the optimal parameters are reported for all the methods. 3) Each dataset is randomly split into two subsets, one as the training set whereas the other as the testing set. 10% training data are labeled. The split is conducted 5 times independently and we report the average results.

3.1 Datasets

We first introduce the datasets used in our experiments.

Facial Expression Recognition: The BU-FE (Binghamton University Facial Expression) database [Yin *et al.*, 2006] is used. It consists of 2500 images of seven facial expressions. The images were cropped to 32×32. We randomly select 1000 images as the training data and the remaining images as the testing data.

Face Recognition: The UMIST face database[1] is used. The database consists of 575 face images from 20 different subjects. Each image was resized to 28×23. We randomly select 400 images as the training set and the remaining images as the testing set.

Head Pose Estimation: We use the Pointing'04 database [Gourier *et al.*, 2004]. It comprises 2790 images from 15 different people. Each image was resized to 40×30. Every head pose is determined by both pan and tilt angles. For pan, the angle varies between -90 and +90 degrees with a step of 15 degrees, resulting in 13 poses. Whereas for tilt, the angle varies as -90, -60, -30, -15, 0, +15, +30, +60, +90 degrees to generate 9 poses. We evaluate the performance on both pan and tilt estimation. For both experiments, 1000 images are randomly chosen as the training data and the rest are the testing data.

Object Recognition: We use the Coil20 dataset [Nene *et al.*, 1996]. Coil20 includes 1440 grey scale images with 20 different objects. The images were resized to 32×32. We use 500 images as the training data and the rest as the testing data.

Handwritten Digit Recognition: We use the USPS database to evaluate the performance on handwritten digit recognition. The database contains 9298 gray-scale handwritten digit images. The images were resized to 16×16. 1000 images are randomly selected as the training set and the remaining images are the testing data.

[1] http://images.ee.umist.ac.uk/danny/database.html

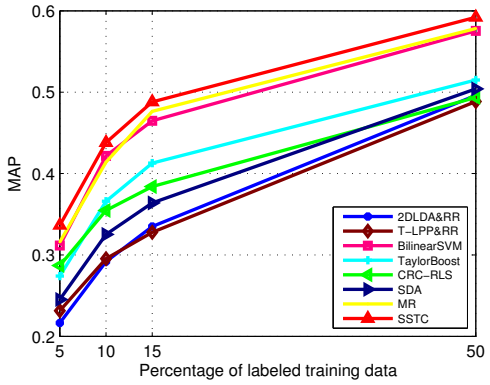

Figure 1: Performance comparison on BU-FE when 5%, 10%, 15% and 50% training data are labeled.

3.2 Experimental Results

The performance comparison of different algorithms are presented in Table 1. It can be seen that our method SSTC is consistently the best algorithm for different applications. We also conduct an experiment to study the performance variance *w.r.t.* 5%, 10%, 15% and 50% labeled training data taking facial expression recognition as an example. The result is displayed in Figure 1.

From the experimental results, we observe that 1) SSTC is consistently the best classification algorithm. 2) As the number of labeled training data increases, the performance of all methods is improved. 3) SSTC, BilinearSVM and MR gain the top performance for all settings, which indicates that preserving image 2D structure and semi-supervised learning both contribute to the performance. 4) When 50% training data are labeled, the advantage of SSTC and MR over other supervised algorithms (except BilinearSVM) decreases noticeably. MR is worse than BilinearSVM. The phenomenon demonstrates that supervised algorithms are preferable when we have a large amount of labeled training data. 5) When less than 50% training data are labeled, SSTC and MR generally have more advantage over other supervised algorithms (except BilinearSVM). SSTC is visibly better than Bilinear SVM when 5% training data are labeled and MR is competitive with BilinearSVM. It indicates that through proper design, semi-supervised approaches do benefit much from the usage of unlabeled data.

3.3 Compound Regression Models vs Single Regression Model

Our algorithm uses compound regression models for classification. In the previous experiments, the number of regression models m was fixed at 3. To show its advantage over single regression model, we reduce m to 1 in this experiment.

Table 2 shows the classification results by using compound regression models ($m = 3$) and using single regression model ($m = 1$). We can see that using compound regression models results noticeable performance gain (from 5.4% to 17.6%

Table 1: Classification results of different applications. The best results are highlighted in bold.

Dataset	2DLDA	T-LPP	BilinearSVM	TaylorBoost	CRC-RLS	SDA	MR	SSTC
BU-FE	29.2%	29.5%	41.2%	36.6%	36.4%	32.5%	41.4%	**43.5%**
UMIST	59.8%	66.3%	61.7%	54.3%	59.2%	64.6%	65.6%	**68.1%**
Pointing'04	31.4%(Pan) 10.7%(Tilt)	33.3%(Pan) 12.1%(Tilt)	30.6%(Pan) 15.7%(Tilt)	34.8%(Pan) 15.8%(Tilt)	32.1%(Pan) 16.6%(Tilt)	31.1%(Pan) 15.7%(Tilt)	35.3%(Pan) 16.9%(Tilt)	**36.0%**(Pan) **17.4%**(Tilt)
Coil20	59.1%	70.8%	71.7%	66.0%	72.1%	75.0%	74.8%	**76.8%**
USPS	73.0%	71.8%	78.3%	78.9%	75.7%	76.6%	78.3%	**81.8%**

Table 2: Comparison between compound regression models and single regression model.

	BU-FE	UMIST	Pointing'04	Coil20	USPS
Single	40.6%	64.6%	33.4%(Pan) 14.8%(Tilt)	70.2%	75.7%
Compound	43.5%	68.1%	36.0%(Pan) 17.4%(Tilt)	76.8%	81.8%
Relative Improvement	7.1%	5.4%	7.8%(Pan) 17.6%(Tilt)	9.4%	8.1%

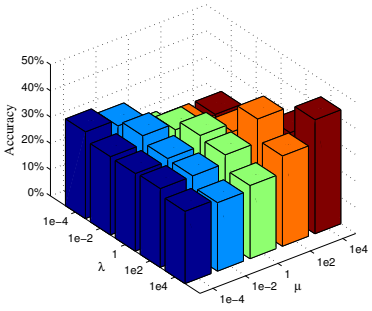

Figure 2: Parameter sensitivity.

on different datasets). This validates that using compound regression model is advantageous as we have larger space to search for the optimal solution. Therefore, the performance can be improved.

3.4 Parameter Sensitivity & Convergence

In this part, we study the performance variance *w.r.t.* the regularization parameters λ and μ. Then an experiment on convergence is presented.

We use the BU-FE dataset for the experiments. Figure 2 shows the parameter sensitivity and we learn that better results are normally obtained when λ and μ are comparable. Figure 3 shows how fast Algorithm 1 converges for one class on this dataset by fixing the parameters at the optimal values when the best classification result is obtained. We observe that for the specified setting Algorithm 1 converges with 4 iterations, which is very fast.

4 Conclusion

Representing images with matrices is better in preserving the spatial correlation and avoiding the curse of dimensionality. We have proposed a method to classify image matrices, which is realized by learning two groups of classification vectors for each dimension of the images. Our method is a

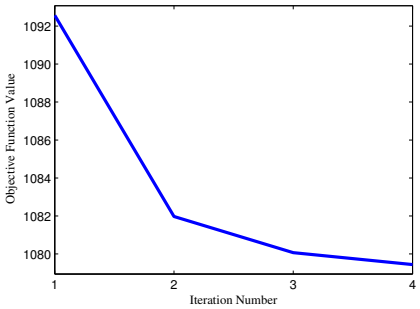

Figure 3: Convergence.

semi-supervised algorithm which uses both labeled and unlabeled data. An efficient iterative algorithm has also been proposed to solve our objective function. Our method processes the image matrices directly to capture the spatial correlation and achieves good results when using few labeled training samples, which is cost-saving. The major novelty of our method is that it uses compound regression models. By using compound regression models, we have a larger space to search for the optimal solution, resulting in improved performance. Experiments on different applications were further conducted to evaluate the efficacy of our method. The results are encouraging and have demonstrated that our method is especially competitive when only few labeled training data are available.

Acknowledgments

This paper was partially supported by the National Program on Key Basic Research Project of China (973 Program) under grant 2010CB327903.

References

[Belkin *et al.*, 2006] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 12:2399–2434, 2006.

[Cai *et al.*, 2007] Deng Cai, Xiaofei He, and Jiawei Han. Semi-supervised discriminant analysis. In *ICCV*, pages 1–7, 2007.

[Cao *et al.*, 2013] Xiaochun Cao, XingxingWei, Yahong Han, Yi Yang, and Dongdai Lin. Robust tensor clustering with non-greedy maximization. In *IJCAI*, 2013.

[Cohen *et al.*, 2004] Ira Cohen, Fabio Gagliardi Cozman, Nicu Sebe, Marcelo Cesar Cirelo, and Thomas S.

Huang. Semisupervised learning of classifiers: Theory, algorithms, and their application to human-computer interaction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(12):1553–1567, 2004.

[Fergus *et al.*, 2010] Robert Fergus, Hector Bernal, Yair Weiss, and Antonio Torralba. Semantic label sharing for learning with many categories. In *ECCV (1)*, pages 762–775, 2010.

[Fukunaga, 1990] Keinosuke Fukunaga. *Introduction to statistical pattern recognition (2nd ed.)*. Academic Press Professional, San Diego, USA, 1990.

[Gourier *et al.*, 2004] Nicolas Gourier, Daniela Hall, and James L. Crowley. Estimating face orientation from robust detection of salient facial features. In *ICPR, International Workshop on Visual Observation of Deictic Gestures*, 2004.

[He *et al.*, 2005] Xiaofei He, Deng Cai, and Partha Niyogi. Tensor subspace analysis. In *NIPS*, 2005.

[Hou *et al.*, 2013] Chenping Hou, Feiping Nie, Dongyun Yi, and Yi Wu. Efficient image classification via multiple rank regression. *IEEE Transactions on Image Processing*, 22(1):340–352, 2013.

[Jolliffe, 2002] Ian T. Jolliffe. *Principal Component Analysis (2nd ed.)*. Springer-Verlag, New York, USA, 2002.

[Kong *et al.*, 2005] Hui Kong, Lei Wang, Eam Khwang Teoh, Jian-Gang Wang, and Ronda Venkateswarlu. A framework of 2D Fisher discriminant analysis: Application to face recognition with small number of training samples. In *CVPR (2)*, pages 1083–1088, 2005.

[Liu *et al.*, 2013] Xiao Liu, Mingli Song, Dacheng Tao, Luming Zhang, Jiajun Bu, and Chun Chen. Semi-supervised node splitting for random forest construction. In *CVPR*, 2013.

[Ma *et al.*, 2011] Zhigang Ma, Yi Yang, Feiping Nie, Jasper R. R. Uijlings, and Nicu Sebe. Exploiting the entire feature space with sparsity for automatic image annotation. In *ACM Multimedia*, pages 283–292, 2011.

[Nene *et al.*, 1996] Sameer A. Nene, Shree K. Nayar, and Hiroshi Murase. Columbia object image library (coil-20). In *Technical Report CUCS-005-96, Columbia University*, 1996.

[Pirsiavash *et al.*, 2009] Hamed Pirsiavash, Deva Ramanan, and Charless Fowlkes. Bilinear classifiers for visual recognition. In *NIPS*, pages 1482–1490, 2009.

[Saberian *et al.*, 2011] Mohammad J. Saberian, Hamed Masnadi-Shirazi, and Nuno Vasconcelos. Taylorboost: First and second-order boosting algorithms with explicit margin control. In *CVPR*, pages 2929–2934, 2011.

[Sharma *et al.*, 2010] Avinash Sharma, Etienne von Lavante, and Radu Horaud. Learning shape segmentation using constrained spectral clustering and probabilistic label transfer. In *ECCV (5)*, pages 743–756, 2010.

[Vasilescu and Terzopoulos, 2003] M. Alex O. Vasilescu and Demetri Terzopoulos. Multilinear subspace analysis of image ensembles. In *CVPR (2)*, pages 93–99, 2003.

[Wang and Ahuja, 2005] Hongcheng Wang and Narendra Ahuja. Rank-r approximation of tensors: Using image-as-matrix representation. In *CVPR (2)*, pages 346–353, 2005.

[Xu *et al.*, 2005] Dong Xu, Shuicheng Yan, Lei Zhang, HongJiang Zhang, Zhengkai Liu, and Heung-Yeung Shum. Concurrent subspaces analysis. In *CVPR (2)*, pages 203–208, 2005.

[Yang *et al.*, 2004] Jian Yang, David Zhang, Alejandro F. Frangi, and Jing-Yu Yang. Two-dimensional PCA: A new approach to appearance-based face representation and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(1):131–137, 2004.

[Yang *et al.*, 2012] Yi Yang, Feiping Nie, Dong Xu, Jiebo Luo, Yueting Zhuang, and Yunhe Pan. A multimedia retrieval framework based on semi-supervised ranking and relevance feedback. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(4):723–742, 2012.

[Yang *et al.*, 2013] Yi Yang, Jiangkuan Song, Zi Huang, Zhigang Ma, Nicu Sebe, and Alexander G. Hauptmann. Multi-feature fusion via hierarchical regression for multimedia analysis. *IEEE Transactions on Multimedia*, 15(3):572–581, 2013.

[Ye *et al.*, 2004] Jieping Ye, Ravi Janardan, and Qi Li. Two-dimensional linear discriminant analysis. In *NIPS*, 2004.

[Yin *et al.*, 2006] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and Matthew J. Rosato. A 3D facial expression database for facial behavior research. In *FG*, pages 211–216, 2006.

[Zhang *et al.*, 2011] Lei Zhang, Meng Yang, and Xiangchu Feng. Sparse representation or collaborative representation: Which helps face recognition? In *ICCV*, pages 471–478, 2011.

[Zhu *et al.*, 2003] Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003.

[Zhu, 2007] Xiaojin Zhu. Semi-supervised learning literature survey. In *Technical Report 1530, University of Wisconsin, Madison*, 2007.