Matrix-Variate Restricted Boltzmann Machine Classification Model

Jinghua Li[1](✉), Pengyu Tian[1], Dehui Kong[1], Lichun Wang[1],
Shaofan Wang[1], and Baocai Yin[2]

[1] Beijing Key Laboratory of Multimedia and Intelligent Software Technology,
Faculty of Information Technology, Beijing University of Technology,
Beijing 100124, China
lijinghua@bjut.edu.cn

[2] College of Computer Science and Technology, Faculty of Electronic
Information and Electrical Engineering, Dalian University of Technology,
Dalian 116620, China

Abstract. Recently, Restricted Boltzmann Machine (RBM) has demonstrated excellent capacity of modelling vector variable. A variant of RBM, Matrix-variate Restricted Boltzmann Machine (MVRBM), extends the ability of RBM and is able to model matrix-variate data directly without vectorized process. However, MVRBM is still an unsupervised generative model, and is usually used to feature extraction or initialization of deep neural network. When MVRBM is used to classify, additional classifiers are necessary. This paper proposes a Matrix-variate Restricted Boltzmann Machine Classification Model (ClassMVRBM) to classify 2D data directly. In the novel ClassMVRBM, classification constraint is introduced to MVRBM. On one hand, the features extracted by MVRBM are more discriminative, on the other hand, the proposed model can be directly used to classify. Experiments on some publicly available databases demonstrate that the classification performance of ClassMVRBM has been largely improved, resulting in higher image classification accuracy than conventional unsupervised RBM, its variants and Restricted Boltzmann Machine Classification Model (ClassRBM).

Keywords: ClassMVRBM · MVRBM · RBM

1 Introduction

Currently, more and more multiple array data are widely acquired in modern computer vision research, such as 2D images, 3D videos and 4D light fields etc. [19]. It is well known that vectorizing multiway data is a common used method, however, such vectorization process inevitably leads to possible data structure break and dimension curse. How to model the multiway data more appropriately so as to process and analyze it effectively is the key problem. Many methods have been proposed during the past years. Take 2D images (matrix-style) for example, such as 2D Principle Component Analysis (2DPCA) [1, 2], and 2D Linear Discriminant Analysis (2DLDA) [3].

H. Song et al. (Eds.): SIMUtools 2019, LNICST 295, pp. 486–497, 2019.
https://doi.org/10.1007/978-3-030-32216-8_47

Unfortunately, 2DPCA and 2DLDA are still linear methods, which both aim to find an optimal linear projection matrix to reduce dimension or classify.

RBM is an effective model for nonlinear modeling [4], it is becoming one of the most popular methods, which is widely used in speech/image feature extraction, feature representation [5] and the initialization of deep neural network, typical RBM variants include Gaussian-Bernoulli RBM (GBRBM) [6], Improved Gaussian-Bernoulli RBM (IGBRBM) [7] and Tensor-variate Restricted Boltzmann Machines [8] etc. Especially, Larochelle et al. [9] proposed ClassRBM to implement the classification task, which extended the ability of RBM. After that, Peng et al. [10] integrates infinite RBM and the classification RBM for Radar high resolution range profile recognition. However, when RBM and ClassRBM are used to process image signals, the 2D image matrices must be transformed into 1D image vectors in advance, such process leads to possible high dimensional vector and spatial structural damage of image. Qi et al. [11] proposed MVRBM model, which has been successfully applied to represent 2D signal. Furthermore, Liu et al. [12] proposed improved MVRBM named MVGRBM, which assumes the matrix data entries follow Gaussian distributions. However, MVRBM and MVGRBM are still unsupervised generative models. When the goal is to classify the image data, an additional classifier must be introduced, such as nearest neighbor classifier or neural network. Inspired by Hugo, this paper adds the label constraint to the existing MVRBM model, i.e., we propose a Matrix-variate Restricted Boltzmann Machine Classification Model (ClassMVRBM), which is capable of classifying the images directly.

2 Definition of ClassMVRBM Model

In this section, we propose a ClassMVRBM model for image classification. Firstly, we introduce the definition of the fundamental MVRBM, and then the definition of the proposed model is detailed.

2.1 Definition of MVRBM

The MVRBM [11] is a bipartite undirected probabilistic graphical model connecting stochastic matrix-style visible units and matrix-style hidden units by tensor-style weights. To formulate the model, we define the follow variables: $X = [x_{ij}] \in \mathbb{R}^{I \times J}$ is a matrix variable of the visual layer, and corresponds to the input observation. $H = [h_{kl}] \in \mathbb{R}^{K \times L}$ is a matrix variable of the hidden layer, and represents the features extracted from the input. $\mathcal{W}=[w_{ijkl}]\in \mathbb{R}^{I\times J\times K\times L}$ denotes the connecting relationship of X and H, which is a fourth-order tensor. $B = [b_{ij}] \in \mathbb{R}^{I \times J}$ and $C = [c_{kl}] \in \mathbb{R}^{K \times L}$ are the matrix-style biases in the visual units and the hidden ones. Therefore, $\Theta'=\{\mathcal{W}, B, C\}$ defines all the model parameters of MVRBM. The MVRBM defines an energy function for joint configuration (X, H) as shown in formula (1):

$$E(X, H; \Theta') = -\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{l=1}^{L} x_{ij}h_{kl}w_{ijkl} - \sum_{i=1}^{I}\sum_{j=1}^{J} x_{ij}b_{ij} - \sum_{k=1}^{K}\sum_{l=1}^{L} h_{kl}c_{kl}. \quad (1)$$

Based on the aforementioned energy function (1), MVRBM defines the joint probability distribution of the visual variates and hidden ones as formula (2):

$$p(\mathrm{X}, H; \Theta') = \frac{1}{Z(\Theta')} \exp\{-E(\mathrm{X}, H; \Theta')\}, \tag{2}$$

where $Z(\Theta')$ is the normalization constant. Maximum likelihood estimation is generally introduced to solve the model parameter Θ', and the log likelihood of X is defined by formula (3).

$$\underset{\Theta'}{\mathrm{Max}}\,\ell = \frac{1}{N}\sum_{n=1}^{N} \log\left(\sum\nolimits_{H \in \mathbb{H}} \exp\left\{-E(\mathbf{X}^{(n)}, H;\ \Theta')\right\}\right) - \log Z(\Theta'), \tag{3}$$

here, N represents the number of the samples and $\mathbf{X}^{(n)}$ means the n^{th} input sample.

2.2 Definition of ClassMVRBM

MVRBM has been successfully used to represent 2D signal, however, MVRBM is still an unsupervised generative model. This paper aims to design an improved MVRBM with the performance of classification, to this end, the classification constraint is added to the existing MVRBM. Specially, as depicted in Fig. 1, we connect an additional label layer to the previous hidden layer. Therefore, in the novel model there are two branches, and the left is the original MVRBM, while the right one is the newly added classification one.

To introduce our model, we define additional variables as follows: $\mathbf{y} = [y_t] \in \mathbb{R}^T$ is a label vector, and indicates the classification of the input data by one-hot coding. $\mathcal{P} = [p_{tkl}] \in \mathbb{R}^{T\times K\times L}$ is the connecting weight of $\mathbf{y}$ and H, indicating the relationship between the label variable and the hidden features. $\mathbf{d} = [d_t] \in \mathbb{R}^T$ is the bias vector of the label layer. Refer to (1), we define the novel joint energy function formulated as below.

$$\begin{aligned} E(X, \mathbf{y}, H; \Theta) = &- \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{l=1}^{L} x_{ij}h_{kl}w_{ijkl} - \sum_{i=1}^{I}\sum_{j=1}^{J} x_{ij}b_{ij} - \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{l=1}^{L} y_t h_{kl} p_{tkl} \\ &- \sum_{t=1}^{T} y_t d_t - \sum_{k=1}^{K}\sum_{l=1}^{L} h_{kl}c_{kl}, \end{aligned} \tag{4}$$

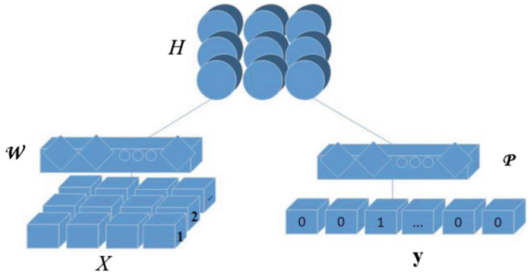

Fig. 1. Graphical illustration of ClassMVRBM

In view of $\mathcal{W}$ is a four-order tensor, which enables the model parameters to increase greatly. To reduce the model parameters, this paper assumes $w_{ijkl} = u_{ki}v_{lj}$ by tensor analysis and decomposition [13], therefore, the revised energy function is in the following:

$$E(X, \mathbf{y}, H; \Theta) = -\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{l=1}^{L} x_{ij}h_{kl}u_{ki}v_{lj} - \sum_{i=1}^{I}\sum_{j=1}^{J} x_{ij}b_{ij} - \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{l=1}^{L} y_t h_{kl} p_{tkl} \\ - \sum_{t=1}^{T} y_t d_t - \sum_{k=1}^{K}\sum_{l=1}^{L} h_{kl}c_{kl}. \tag{5}$$

Defining matrix-style variables $\mathrm{U} = [u_{ki}] \in \mathbb{R}^{K \times I}$ and $\mathrm{V} = [v_{lj}] \in \mathbb{R}^{L \times J}$, therefore, $\Theta = \{U, V, \mathcal{P}, B, C, \mathbf{d}\}$ indicates all model parameters of ClassMVRBM, here, the definition of $\mathcal{W}, B, C$ is same to that of MVRBM. Based on the formula (5), the joint distribution of $X, \mathbf{y}, H$ is defined as follows:

$$p(X, \mathbf{y}, H; \Theta) = \frac{\exp(-E(X, \mathbf{y}, H))}{Z(\Theta)}, \tag{6}$$

$Z(\Theta)$ is the normalized constant and written as:

$$Z(\Theta) = \sum_{X, \mathbf{y}, H} \exp(\{-E(X, \mathbf{y}, H; \Theta)\}). \tag{7}$$

3 Optimization of ClassMVRBM Model

Given the training data pairs $D_{\text{train}} = \{(\mathbf{X}^{(1)}, \mathbf{y}^{(1)}), \cdots, (\mathbf{X}^{(n)}, \mathbf{y}^{(n)}), \cdots (\mathbf{X}^{(N)}, \mathbf{y}^{(N)})\}$, the most popular training objective for RBMs and its variants is generative, that is, maximizing the joint probability is the training objective. Therefore, the equivalent minimized negative log likelihood objective function can be written as:

$$\min L_{gen}(D_{\text{train}}) = -\sum_{n=1}^{N} (\log p(X^{(n)}, \mathbf{y}^{(n)})) = -\sum_{n=1}^{N} (\log p(\mathbf{y}^{(n)} | X^{(n)}) + \log p(X^{(n)})) \\ = -\sum_{n=1}^{N} \log p(\mathbf{y}^{(n)} | X^{(n)}) - \sum_{n=1}^{N} \log p(X^{(n)}). \tag{8}$$

According to (8), our proposed ClassMVRBM includes two parts, one is the conditional probability part, and the other is the marginal distribution of the input samples. Since our training data are labeled and for the test sample, a good prediction of the target classification is the only interesting point, therefore, this paper only focuses on the supervised part in (8), that is, the condition probability is the only objective function as follows.

$$\min_{\Theta} L(\Theta) = -\sum_{n=1}^{N} \log(p(\mathbf{y}^{(n)} \mid X^{(n)}; \Theta)). \tag{9}$$

For a single sample pair $(\mathbf{X}^{(n)}, \mathbf{y}^{(n)})$, we derive:

$$\begin{aligned} \log p(\mathbf{y}^{(n)}|X^{(n)}) &= \log \frac{p(X^{(n)}, \mathbf{y}^{(n)})}{p(X^{(n)})} \\ &= \log \sum_{H} \exp(-E(X^{(n)}, \mathbf{y}^{(n)}, H)) - \log \sum_{\mathbf{y},H} \exp(-E(X^{(n)}, \mathbf{y}, H)). \end{aligned} \tag{10}$$

With respect to any parameter θ of Θ in ClassMVRBM, the gradient of $\log p(\mathbf{y}^{(n)}|X^{(n)})$ is:

$$\begin{aligned} &\frac{\partial \log p(\mathbf{y}^{(n)}|X^{(n)})}{\partial \theta} \\ &= \frac{\sum_{H} \exp(-E(X^{(n)},\mathbf{y}^{(n)},H)) \frac{\partial(-E(X^{(n)},\mathbf{y}^{(n)},H))}{\partial \theta}}{\sum_{H} \exp(-E(X^{(n)},\mathbf{y}^{(n)},H))} - \frac{\sum_{\mathbf{y},H} \exp(-E(X^{(n)},\mathbf{y},H)) \frac{\partial(-E(X^{(n)},\mathbf{y},H))}{\partial \theta}}{\sum_{\mathbf{y},H} \exp(-E(X^{(n)},\mathbf{y},H))} \\ &= \sum_{H} p(H|X^{(n)}, \mathbf{y}^{(n)}) \frac{\partial}{\partial \theta}(-E(X^{(n)}, \mathbf{y}^{(n)}, H)) - \sum_{\mathbf{y},H} p(\mathbf{y}, H|X^{(n)}) \frac{\partial}{\partial \theta}(-E(X^{(n)}, \mathbf{y}, H)). \end{aligned} \tag{11}$$

According to (11), the two terms around the minus sign needed to be solved, respectively. Analyze the parameters $\Theta = \{U, V, \mathcal{P}, B, C, \mathbf{d}\}$ to be optimized, since the optimization process does not include the reconstruction of the input X, and the input biases are not involved in the computation of $p(\mathbf{y}|X)$, the gradient with respect to B is 0. The bias vector $\mathbf{d}$ in the label layer is special and only the label position is updated. In this paper, we assume the position of classification label is t, the gradient of the bias component d_t is as follows:

$$\frac{\partial \log p(y_t^{(n)}|X^{(n)})}{\partial d_t} = 1 - p(y_t^{(n)}|X^{(n)}), \quad y_t^{(n)} \in \{1, \cdots, M\}. \tag{12}$$

Here, M is the number of categories. For the other parameters $\theta \in \{U, V, \mathcal{P}, C\}$, the derivative of the log likelihood function with respect to every parameter is computed below. Firstly $\frac{\partial E}{\partial \theta}$ is calculated, and then $\sum_{H} p(H|X^{(n)}, \mathbf{y}^{(n)}) \frac{\partial}{\partial \theta}(-E(X^{(n)}, \mathbf{y}^{(n)}, H))$ and $\sum_{\mathbf{y},H} p(\mathbf{y}, H|X^{(n)}) \frac{\partial}{\partial \theta}(-E(X^{(n)}, \mathbf{y}, H))$ in the objective function (11) are calculated, respectively. To calculate the gradient $\frac{\partial E}{\partial \theta}$, we first take calculating $\frac{\partial E}{\partial U}$ as an example. According to (5), we have

$$\frac{\partial E(X^{(n)}, \mathbf{y}^{(n)}, H; \Theta)}{\partial u_{ki}} = -\sum_{j,l} x_{ij}^{(n)} v_{lj} h_{kl}. \tag{13}$$

The corresponding matrix-style representation is:

$$\frac{\partial E(X^{(n)}, \mathbf{y}^{(n)}, H; \Theta)}{\partial U} = -HVX^{(n)T}. \tag{14}$$

Similarly, the derivatives with respect to other parameters can be calculated, and we discover that the gradients $\frac{\partial E}{\partial \theta}(\theta \in \{U, V, \mathcal{P}, C\})$ all include h_{kl} or H, furthermore, for any binary hidden variable unit h_{kl} in the hidden layer H,

$$\sum_{h_{kl} \in \{0,1\}} p(h_{kl} | X^{(n)}, \mathbf{y}^{(n)}) \times h_{kl} = p(h_{kl} = 1 | X^{(n)}, \mathbf{y}^{(n)}). \tag{15}$$

The activation probability of one single unit in the hidden layer is defined by the following,

$$p(h_{kl} = 1 | X, \mathbf{y}; \Theta) = \sigma(c_{kl} + \sum_{i=1}^{I} \sum_{j=1}^{J} x_{ij} u_{ki} v_{lj} + \sum_{t=1}^{T} y_t p_{klt}), \tag{16}$$

where σ is the sigmoid function, $\sigma(a) = 1/(1 + \exp(-a))$. It is easy to see that the hidden unit is influenced by not only the visual layer but also the labeled one. In terms of matrix representation, the aforementioned conditional probability can be written as:

$$p(H = 1 | X^{(n)}, \mathbf{y}^{(n)}; \Theta) = \sigma(C + UX^{(n)}V^T + \mathcal{P}_{t\bullet\bullet}) \ . \tag{17}$$

Here, σ applies on the entries of the corresponding matrices. $\mathcal{P}_{t\bullet\bullet}$ denotes all the weights between the t^{th} label component in the label vector and the units in the hidden layer.

With regard to the second term in (11),

$$\sum_{\mathbf{y}, H \in \{0,1\}} p(\mathbf{y}, H | X^{(n)}) H = p(\mathbf{y} | X^{(n)}) = \frac{\sum_{H} \exp(-E(X^{(n)}, \mathbf{y}, H))}{\sum_{\mathbf{y}^*, H} \exp(-E(X^{(n)}, \mathbf{y}^*, H))}, \tag{18}$$

Of which, $\mathbf{y}$ in the numerator represents a special category, while $\mathbf{y}^*$ in the denominator represents all possible categories. Where,

$$\begin{aligned}
&\sum_{H} \exp(-E(X^{(n)}, \mathbf{y}, H)) \\
&= \exp(d_t) \sum_{h_{11}} \exp(h_{11}(c_{11} + \sum_{t} \mathcal{P}\mathbf{y} + \sum_{i,j} UX^{(n)}V^T)) \cdots \sum_{h_{kl}} \exp(h_{kl}(c_{kl} + \sum_{t} \mathcal{P}\mathbf{y} + \sum_{i,j} UX^{(n)}V^T)) \\
&= \exp(d_t)(1 + \exp(c_{11} + \sum_{t} \mathcal{P}\mathbf{y} + \sum_{i,j} UX^{(n)}V^T)) \cdots \exp(1 + (c_{kl} + \sum_{t} \mathcal{P}\mathbf{y} + \sum_{i,j} UX^{(n)}V^T)) \\
&= \exp(d_t + \sum_{k,l} \log(1 + \exp(c_{kl} + \sum_{t} \mathcal{P}\mathbf{y} + \sum_{i,j} UX^{(n)}V^T)))
\end{aligned} \tag{19}$$

In summary, we have:

$$\begin{aligned}
\frac{\partial L(\Theta)}{\partial U} &= \sum_{n=1}^{N}\sigma(C+UX^{(n)}V^{T}+\mathcal{P}_{t\bullet\bullet})VX^{(n)^{T}}-\sum_{n=1}^{N}\sum_{y^{*}}p(y^{*}\mid X^{(n)})\sigma(C+UX^{(n)}V^{T}+\mathcal{P}_{t^{*}\bullet\bullet})VX^{(n)^{T}}\\
\frac{\partial L(\Theta)}{\partial V} &= \sum_{n=1}^{N}\sigma(C+UX^{(n)}V^{T}+\mathcal{P}_{t\bullet\bullet})^{T}UX^{(n)}-\sum_{n=1}^{N}\sum_{y^{*}}p(y^{*}\mid X^{(n)})\sigma(C+UX^{(n)}V^{T}+\mathcal{P}_{t^{*}\bullet\bullet})^{T}UX^{(n)}\\
\frac{\partial L(\Theta)}{\partial \mathcal{P}_{y^{*}\bullet\bullet}} &= \sum_{n=1}^{N}\sigma(C+UX^{(n)}V^{T}+\mathcal{P}_{t\bullet\bullet})\mathbf{y}^{(n)}-\sum_{n=1}^{N}\sum_{\mathbf{y}^{*}}p(\mathbf{y}^{*}\mid X^{(n)})\sigma(C+UX^{(n)}V^{T}+\mathcal{P}_{t^{*}\bullet\bullet})\\
\frac{\partial L(\Theta)}{\partial C} &= \sum_{n=1}^{N}\sigma(C+UX^{(n)}V^{T}+\mathcal{P}_{t\bullet\bullet})-\sum_{n=1}^{N}\sum_{y^{*}}p(y^{*}\mid X^{(n)})\sigma(C+UX^{(n)}V^{T}+\mathcal{P}_{t^{*}\bullet\bullet})
\end{aligned} \tag{20}$$

The gradient values used to update the parameters of the model are as follows:

$$\begin{aligned}
\Delta U^{(t)} &= \alpha\Delta U^{(t-1)}+\lambda\left(\frac{\partial L(\Theta)}{\partial U}-\xi_{1}U^{(t-1)}\right)\\
\Delta V^{(t\ )} &= \alpha\Delta V^{(t-1)}+\lambda\left(\frac{\partial L(\Theta)}{\partial V}-\xi_{1}V^{(t-1)}\right)\\
\Delta \mathcal{P}^{(t)} &= \alpha\Delta \mathcal{P}^{(t-1)}+\lambda\left(\frac{\partial L(\Theta)}{\partial \mathcal{P}}-\xi_{2}\mathcal{P}^{(t-1)}\right)\\
\Delta C^{(t)} &= \alpha\Delta C^{(t-1)}+\lambda\frac{\partial L(\Theta)}{\partial C}\\
\Delta \mathbf{d}^{(t)} &= \alpha\Delta \mathbf{d}^{(t-1)}+\lambda\frac{\partial L(\Theta)}{\partial \mathbf{d}}
\end{aligned} \tag{21}$$

Of which, λ is the learning rate, α is the momentum, and ξ_1 and ξ_2 is the weight regularizer. The training algorithm of ClassMVRBM is presented as follows.

```
Algorithm 1: Training of ClassMVRBM
Input: Training pairs set $D=\{(\mathbf{X}^{(1)},\mathbf{y}^{(1)}),\cdots,(\mathbf{X}^{(n)},\mathbf{y}^{(n)}),\cdots,(\mathbf{X}^{(N)},\mathbf{y}^{(N)})\}$ , learning rate $\lambda$ (0.01), momentum $\alpha$ (0.5) , weight regularizer $\xi_1$ (0.001) , weight regularizer $\xi_2$ (0.00001), the maximum iteration number Z (100 ) , batch size $s$ ( $s$ =2M, that is, there two sample pairs for each category in each batch) .
Output: Model parameters $\Theta=\{U,V,\mathcal{P},B,C,\mathbf{d}\}$
Initialization: Initialize $U,V$ and $\mathcal{P}$ randomly, set the bias $C=B=0,\mathbf{d}=0$ and the gradient increments $\Delta U=\Delta V=\Delta\mathcal{P}=\Delta C=\Delta B=\Delta\mathbf{d}=0$.
for iteration step $z=1\rightarrow Z$ do
    Divide $D$ into $N/2M$ batches $D_1,\cdots,D_{m,}\cdots,D_{N/2M}$ of size $s$
    for batch $m=1\rightarrow N/2M$ do
        for the sample pairs $(\mathbf{X}^{(n)},\mathbf{y}^{(n)})\in D_m$ do
            $y^0\leftarrow\mathbf{y}^{(n)}$, $X^0\leftarrow\mathbf{X}^{(n)}$, $\overline{H^0}\leftarrow sigmoid(C+U\mathbf{X}^0V^T+\mathcal{P}_{y^0\bullet\bullet})$
            #Notation: $a\leftarrow b$ means $a$ is set to $b$ , $y^0$ corresponds to the label component.
            for $\theta\in\Theta$ do      # Update $\theta\in\Theta$:
                Compute the derivatives of the objective function with respect to all the parameters refer to Eq. (20), here, $N$ in the Eq. (20) is equal to batchsize $s$ .
                Update the gradient with Eq. (21)
                Update $\Theta$ with a gradient-based solver $\theta=\theta-\Delta\theta$
            end for
        end for
    end for
end for
```

4 Experimental Results

To evaluate the performance of our method, we conduct two types of experiments on four publicly available image databases. The first type of experiment aims to evaluate the classification performance of ClassMVRBM relative to RBM, MVRBM and other unsupervised methods, and the second type of experiment aims to compare the classification performance of ClassMVRBM with ClassRBM, respectively for 2D signal and the general vectorized 1D signal. In addition, we also conduct the sensitivity test for some parameters.

4.1 The Experimental Datasets

This paper conducts experiments on image databases MNIST, Ballet, ETH80 and Coil_20. All programs are coded by MATLAB and implemented on an Intel Core i7, 3.60 GHz CPU machine with 12 GB RAM. The datasets are listed as follows:

MNIST Database [14]**:** MNIST is a dataset of handwritten digits images database including 60,000 training samples and 10,000 testing samples. Each image is one digit among 0–9, and each one is a gray image with the size of 28 × 28.

Ballet Database [15]**:** This dataset includes 8 kinds of complex ballet actions, totally 44 videos clips are cut from the Ballet DVD video, and each clip has 107–506 frames. The paper randomly selects 200 frames from each kind of action for training, while the remaining images are used for testing. Similarly, all images are down-sampled to 32 × 32 and transformed to gray scale.

ETH80 Database [16]**:** ETH80 dataset includes 8 categories (apples, cars, cows, cups, dogs, horses, pears and tomatoes). Each category consists of 10 different objects, and each object is collected from 41 different views. Therefore, there are totally 8 × 10 × 41 = 3280 images. We randomly select the images of 21 views (8 × 10 × 21 = 1680) for training while the others from the additional 20 views (8 × 10 × 20 = 1600) for testing. All images are down-sampled to 32 × 32 and transformed to gray scale.

Coil_20 Database [18]**:** There are 20 kinds of different objects in this database, and each object includes 72 images taken under different views, and all images are down-sampled to 32 × 32, and transformed to gray scale. We randomly select 36 images for training while the rest 36 images for testing.

4.2 Experimental Results

Experiment 1: The Classification Performance Evaluation of ClassMVRBM and Other Unsupervised RBMs and Variants

This section aims to compare the classification accuracy of our proposed ClassMVRBM with other unsupervised methods such as RBM, IGBRBM, MVRBM and MVIGRBM. Note that RBM, IGBRBM, MVRBM and MVIGRBM are unsupervised and mainly used to extract features of the input, we use the nearest neighbor classifier for classification. The comparative experiments are conducted on image datasets MNIST, Ballet and ETH80. Table 1 shows the classification accuracy of five algorithms: RBM, IGBRBM, MVRBM, MVGRBM and ClassMVRBM on three datasets. Of which, the classification accuracy of unsupervised RBM, IGBRBM, MVRBM and MVGRBM are reported in [11] when the iteration times and all parameters are adjusted to the best. In the same way, the classification accuracy of our proposed ClassMVRBM is obtained when the iteration times are 100 and all the other parameters are adjusted to the most optimal by grid search. In the Table 1, the bold figures are the best results in the comparison.

According to Table 1, the classification accuracy of ClassMVRBM is much higher than other four unsupervised methods on the MNIST, Ballet and ETH80 datasets. It can be concluded when adding the classification constraint to MVRBM, on one side, the extracted feature representations are discriminative, on the other side, when the conditional probability is directly used to classify, our proposed discriminative model pays more attention to the difference between categories, which enables the proposed method be obviously more robust for modeling relative less and more complicated input data such as Ballet and ETH80 datasets. Therefore, our proposed model demonstrates the significant superiority.

Table 1. Classification accuracy of ClassMVRBM and other unsupervised methods

	RBM	IGBRBM	MVRBM	MVGRBM	ClassMVRBM
MNIST	0.9515	0.9398	0.9670	0.9700	**0.9725**
Ballet	0.3779	0.9216	0.3505	0.9357	**0.9509**
ETH80	0.5281	0.8750	0.3969	0.8894	**0.9053**

Experiment 2: The Classification Accuracy Comparison of ClassMVRBM with ClassRBM

In this experiment, we will compare the classification accuracy of ClassRBM and ClassMVRBM on three databases: Ballet, ETH80 and Coil_20. ClassMVRBM and ClassRBM are both classification models, the difference lies in when ClassRBM is used to classify 2D images, the images need to be vectorized firstly. To make the comparison fair, the number of neurons in the hidden layer of ClassRBM and ClassMVRBM is set consistent. That is, when the hidden dimension of ClassMVRBM is 20×20, then the hidden dimension of ClassRBM is 400. Table 2 demonstrates the classification accuracy of ClassRBM and ClassMVRBM when all parameters are adjusted to the most optimal by grid search. According to Table 2, it is easy to see that the classification performance of ClassMVRBM is better than that of ClassRBM. It's not difficult to conclude that to model 2D signal, ClassMVRBM performs better than ClassRBM, which is due to that ClassMVRBM does not vectorize the images and keeps the spatial structure better.

Table 2. Classification accuracy comparison of ClassMVRBM and ClassRBM

Methods	ClassRBM	ClassMVRBM
Ballet	0.9114	**0.9509**
ETH80	0.5078	**0.9053**
Coil_20	0.9779	**0.9896**

ClassMVRBM is sensitive to the hidden size and iteration times. In the following experiments, we discuss the classification accuracy of ClassMVRBM under different hidden sizes and iteration times. As regards to the hidden size, the grid search method [17] is introduced to find the optimal hidden size so as to attain the highest classification accuracy. According to the preceding description of the datasets, the input size of Ballet, Coil_20 and ETH80 are all 32 × 32, for the sake of dimensionality reduction, we conduct experiments successively assuming the hidden size is 15 × 15, 18 × 18, 20 × 20, 25 × 25, 28 × 28 and 32 × 32. As shown in Table 3, the larger the hidden size, the higher the classification accuracy, however, when the hidden size increases to 32 × 32, the classification accuracy decreases instead. Especially, 28 × 28 is the optimal hidden size and with the highest classification performance. It is not difficult to conclude when the hidden size is small, the extracted feature dimension is limited, and the less model parameters generally leads to the under fitting, thus the smaller hidden size brings the lower classification accuracy. But when the hidden size is more than 28 × 28, the classification accuracy decreases, which probably results from the overfitting caused by the excessive model parameters.

The influence of the iteration times for classification performance is reported in Table 4. Note that as the iteration times increased, the classification accuracy increased. However, when the iteration times are more than 200, the accuracy decreased. According to Table 4, the best optimal iteration times are about 100 and we can conclude that the increased iteration times over 100 probably lead to the over fitting. This implies that our proposed classification model converges rapidly.

Table 3. Classification accuracy comparison under different hidden layer sizes on various datasets

Hidden size	15 × 15	18 × 18	20 × 20	25 × 25	28 × 28	32 × 32
Ballet	0.8165	0.8432	0.8875	0.9165	**0.9509**	0.9053
Coil_20	0.3999	0.3999	0.5139	0.9229	**0.9896**	0.8653
ETH80	0.7888	0.7975	0.8388	0.8546	**0.9053**	0.8632

Table 4. Classification accuracy comparison under different iteration times on various datasets

Iteration times	10	30	50	100	200	500
Ballet	0.3838	0.6657	0.8547	**0.9365**	0.9309	0.9073
Coil_20	0.6753	0.8289	0.9264	**0.9719**	0.9597	0.9253
ETH80	0.6516	0.7713	0.8782	**0.8946**	0.8830	0.8632

5 Conclusions

In this paper, we introduce a novel classification model called Matrix-variate Restricted Boltzmann Machine Classification Model (ClassMVRBM). Inspired by ClassRBM, ClassMVRBM integrates classification constraints to MVRBM and presents the optimized objective function of conditional probability to solve the model parameters.

Since the proposed ClassMVRBM directly models the images without the vectorized process, which keeps the spatial structure of the images better. Furthermore, the classification constraint and the conditional probability objective function ensure the discriminability of the learnt features. The experiments are carried out on four benchmark datasets, MNIST, Ballet, Coil_20 and ETH80. The corresponding results demonstrate the superiority of ClassMVRBM. However, the hidden features extracted based on our proposed model still lack the discriminative analysis like the within-class and between-class scatter constraints, we shall extent our work for tackling the task in future.

Acknowledgments. This research is supported by NSFC (No.61772049, 61602486), BJNSF (No.4162009), Beijing Educational Committee (No. KM201710005022) and Beijing Key Laboratory of Computational Intelligence and Intelligent System.

References

1. Wang, H., Wang, J.: 2DPCA with L1-norm for simultaneously robust and sparse modelling. Neural Netw. **46**(10), 190–198 (2013)
2. Ju, F., Sun, Y., Gao, J., Hu, Y., Yin, B.: Image outlier detection and feature extraction via L1-norm-based 2D probabilistic PCA. IEEE Trans. Image Process. **24**(12), 4834–4846 (2015)
3. Li, M., Yuan, B.: 2D-LDA: a statistical linear discriminant analysis for image matrix. Pattern Recogn. Lett. **26**(5), 527–532 (2005)
4. Wang, J., Wang, W., Wang, R., Gao, W.: Image classification using RBM to encode local descriptors with group sparse learning. In: Proceedings of International Conference on Image Processing, pp. 912–916. IEEE, Canada (2015)
5. Dahl, G.E., Dong, Y., Li, D., Acero, A.: Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Trans. Audio Speech Lang. Process. **20**(1), 30–42 (2011)
6. Nair, V., Hinton, G.E.: Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning, Israel, pp. 807–814 (2010)
7. Cho, K., Ilin, A., Raiko, T.: Improved learning of Gaussian-Bernoulli restricted Boltzmann machines. In: Honkela, T., Duch, W., Girolami, M., Kaski, S. (eds.) ICANN 2011. LNCS, vol. 6791, pp. 10–17. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21735-7_2
8. Nguyen, T., Tran, T., Phung, D., Venkatesh, S.: Tensor-variate restricted Boltzmann machines. In: Proceedings of the Twenty-Ninth National Conference on Artificial Intelligence, pp. 2887–2893. AAAI, USA (2015)
9. Larochelle, H., Mandel, M., Pascanu, R., et al.: Learning algorithms for the classification restricted Boltzmann machine. J. Mach. Learn. Res. **13**(1), 643–669 (2012)
10. Peng, X., Gao, X., Li, X.: An infinite classification RBM model for radar HRRP recognition. In: International Joint Conference on Neural Networks, pp. 1442–1448, IEEE, USA (2017)
11. Qi, G., Sun, Y., Gao, J., Hu, Y., Li, J.: Matrix variate restricted Boltzmann machine. In: The proceeding of 2016 International Joint Conference on Neural Networks, pp. 389–395. IEEE, Canada (2016)

12. Liu, S., Sun, Y., Hu, Y., Gao, J., Ju, F., Yin, B.: Matrix variate RBM model with Gaussian distributions. In: The proceeding of 2017 International Joint Conference on Neural Networks, pp. 808–815. IEEE, USA (2017)
13. Gao, J., Guo, Y., Wang, Z.: Matrix neural networks. In: Cong, F., Leung, A., Wei, Q. (eds.) ISNN 2017. LNCS, vol. 10261, pp. 313–320. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59072-1_37
14. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
15. Wang, Y., Mori, G.: Human action recognition by semilatent topic models. IEEE Trans. Pattern Anal. Mach. Intell. **31**(10), 1762 (2009)
16. Leibe, B., Schiele, B.: Analyzing appearance and contour based methods for object categorization. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1–7. IEEE, USA (2003)
17. Tenenbaum, J., Silva, V., Langford, J.: A global geometric framework for nonlinear dimensionality reduction. Science **290**(5500), 2319–2323 (2000)
18. Nene, S., Nayar, S., Murase, H.: Columbia object image library (COIL-20). Technical report CUCS-005-96, USA (1996)
19. Qi, N., Shi, Y., Sun, X., Wang, J., Yin, B., Gao, J.: Multi-dimensional sparse models. IEEE Trans. Pattern Anal. Mach. Intell. **40**(1), 163–178 (2018)