Expert Systems With Applications 133 (2019) 21–33

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Max-margin multi-scale convolutional factor analysis model with application to image classification

Yuchen Guo, Lan Du*, Jian Chen

National Lab of Radar Signal Processing, Xidian University, Xi'an, Shaanxi, 710071, China

ARTICLE INFO

Article history:
Received 2 February 2019
Revised 25 March 2019
Accepted 6 April 2019
Available online 26 April 2019

Keywords:
Factor analysis (FA)
Max-margin learning
Multi-scale convolution kernels
Image classification
Synthetic aperture radar (SAR) image

ABSTRACT

This paper presents a max-margin multi-scale convolutional factor analysis (MMCFA) model, which explores the strongly discriminative principle of max-margin learning to improve the classification performance of multi-scale convolutional factor analysis (CFA) model with application to image data classification. Compared with the traditional factor analysis (FA) model, the CFA model can maintain the spatial correlation among the image pixels in two-dimensional space and capture the structural information from images via convolution kernels. Moreover, to extract multi-level features, multi-scale convolution kernels are adopted to capture richer features at different scales of images. Since the unsupervised model may not offer discriminative factors for the classification task, it is expected to introduce the supervised information to the multi-scale CFA model when supervised information is available. To deal with it, a latent variable support vector machine (LVSVM) is linked to the factors learned from multi-scale CFA model, yielding max-margin discrimination, as the classification criterion in the feature space in our proposed model. The multi-scale CFA model and LVSVM learn parameters jointly in a united framework via the Gibbs inference. Experimental results on mixed national institute of standards and technology (MNIST) dataset, Fashion-MNIST dataset, SVHN dataset and measured synthetic aperture radar (SAR) images show that the learned convolution kernels and factors can describe data information well and the proposed model has excellent classification performance.

1. Introduction

Statistical generative models have received intensive attention from data analysis community (Blei, Ng, & Jordan, 2003; Jain, Duin, & Mao, 2000; Zhu, Ahmed, & Xing, 2012; Chen et al., 2010; Du, Liu, & Bao, 2008; Tunç, Dagl, & Gökmen, 2012). By projecting every observation into a low dimensional latent space, statistical generative models could capture some abstract feature from the data, e.g., the topic assignment in the latent Dirichlet allocation (LDA) (Blei et al., 2003; Zhu et al., 2012) model and the factors in the factor analysis (FA) (Chen et al., 2010; Du et al., 2008; Tunç et al., 2012) model. In many practical applications, supervised information can be easily obtained besides the observations and it often provides useful high-level or direct summarization of the data, but it is not directly used in many original statistical generative models to influence parameter inference. It is expected that introducing the supervised information to statistical generative models could guide the models towards discovering more discriminative latent representations that may be more interesting to the user's goal, such as classification tasks.

To explore this potential, using supervised information to train predictive statistical generative models has attracted a lot of attentions, where the inferred discriminant latent factors are considered as input features (Zhu et al., 2012; Chen et al., 2015; Polson, & Scott, 2011; Yu, Yu, Tresp, Kriegel, & Wu, 2006). Yu et al. (2006) propose a linear supervised probabilistic principal component analysis (SPPCA) model, which incorporates the label information into the projection phase. However, it only considers the classification problem as the imputation of missing values without using any classification criterion. Another idea to explore the supervised information is to combine the generator with the discriminator (Zhu et al., 2012; Chen et al., 2015; Polson, & Scott, 2011), the features leaned from generator would tend to be more discriminative by jointly learning with discriminator in a united framework. Max-margin learning has been very successful in building the discriminators (Polson, & Scott, 2011) and the structured output prediction models (Zhu et al., 2012; Chen et al., 2015) in the last decade. Research on learning max-margin models in the presence of latent models has received increasing attention because of the promise of using latent variables to capture the underlying structures for some complex problems. Zhu et al. (2012) propose a maximum entropy discrimination latent Dirichlet allocation (MedLDA) model which seeks a regularized posterior

* Corresponding author.
E-mail address: dulan@mail.xidian.edu.cn (L. Du).

https://doi.org/10.1016/j.eswa.2019.04.012
0957-4174/

distribution of the predictive function in a feasible space. In MedLDA, the predictive function is defined by a set of expected margin constraints generalized from the SVM-style margin constraints. Max-margin constraints are also introduced to improve the discriminative power of the probabilistic topic model in Zhu et al. (2012). However, images should be translated through bag-of-words representation (Gao, Tsang, & Ma, 2014; Lazebnik, Schmid, & Ponce, 2006) before analyzed by the LDA model. And the learning of bag-of-words representation and the training process of LDA model are independent of each other.

FA model is a statistical generative model which is effective on discovering latent structures and revealing hidden explanatory factors for complex data in statistics and machine learning. The latent factors in FA model show the description of data in low-dimensional hidden space. Rather than training the LDA model based on the discrete bag-of-words representation for image data, we can train FA model directly with image data. Some image processing methods based on the improved FA models have been proposed in recent years. In Chen et al. (2010), a nonparametric mixture of FA model is employed to the application of compressive sensing (CS). For the face classification task, a class dependent factor analysis (CDFA) model has been proposed (Tunç et al., 2012), in which each class has its own factor loading matrix instead of a common loading matrix for all classes.

Therefore, in this paper, we consider to combine the FA model and latent variable support vector machine (LVSVM) (Polson, & Scott, 2011) into a united framework. The LVSVM is linked to the factors learned from FA model, yielding max-margin discrimination, as the classification criterion in the feature space. The interplay between the likelihood function of FA model and maximum margin constraint induced by LVSVM can yield latent representations more discriminative and reasonable for supervised prediction tasks.

Nevertheless, since the two-dimensional image data should be stretched into vectors in FA models (Chen et al., 2010; Tunç et al., 2012), only one-dimensional correlation of image pixels can be considered while the two-dimensional correlation is ignored. In image processing researches, to maintain the spatial correlation among the image pixels, there has been significant recent interest in convolutional models, including deconvolutional networks (Zeiler, Krishnan, Taylor, & Fergus, 2010), convolutional networks (LeCun, Bottou, Bengio, & Haffner, 1998), convolutional restricted Boltzmann machines (RBMs) (Lee, Grosse, Ranganath, & Ng, 2009; Lee, Pham, Largman, & Ng, 2009; Norouzi, Ranjbar, & Mori, 2009) and convolutional neural network (CNN) (Alex, Ilya, & Geoff, 2012; Han, Li, & Zhu, 2019; Sellami, & Hwang, 2019). One of the key characteristic of these algorithms is the exploitation of convolution operator, which plays an important role in analyzing the correlation among image pixels in two-dimensional space. Moreover, the size of convolution kernels is much smaller than the size of images, leading to the fact that features extracted by these models focus more on the local details.

Inspired by this, it is desirable to introduce the two-dimensional convolution operator into FA model to represent the observations via the convolution of kernels and factors. In this paper, we refer to the FA model exploiting the two-dimensional convolution operation as the convolutional factor analysis (CFA) model. The learned convolution kernels can notice the correlation between the image pixels in local regions, thus solving the problem that the traditional FA model ignores the spatial correlation among the image pixels. In addition, the convolution kernels in CFA model can capture the structural information from images, such as the sub-stroke attribution feature information of mixed national institute of standards and technology (MNIST) image data or the contour information of measured synthetic aperture radar (SAR) images.

If we fix the scale of convolution kernels, the CFA model can only extract features based on the size of the scale. However, CNN model (Alex et al., 2012; Han et al., 2019; Sellami, & Hwang, 2019), which is a representative deep learning model, could extract more information-rich features in different layers. The output features of different convolution layers reflect the characteristics at different levels of the images, of which low-level features of bottom convolution layer correspond to localized details within small scale while high-level features of top convolution layer represent more global information within large scale (Erhan, Bengio, Courville, & Vincent, 2009). To get multi-level features at different levels in a single layer, Chang, Han, Zhong, Snijders and Mao (2018) propose a multi-scale convolutional sparse coding method which automatically learns convolution kernels at different scales in a single layer.

Multi-scale convolution kernels are also employed in our CFA model to extract features at different levels jointly. The kernels at smaller scales mainly capture lower-level features, while the kernels at larger scales are more responsible for higher-level features. Therefore, the multi-scale CFA model will show more comprehensive information of images compared with the CFA model with convolution kernels at a single scale.

In this paper, we develop a max-margin multi-scale convolutional factor analysis (MMCFA) model, a supervised statistical generative model consisting of multi-scale CFA model and LVSVM. By jointly learning the generative model and max-margin classifier, both of the spatial correlation among the image pixels and supervised information are considered in MMCFA model. The parameters of MMCFA model can be effectively inferred via the simple and efficient Gibbs inference (Geman, & Geman, 1984). In the application of MMCFA model to MNIST dataset, Fashion-MNIST dataset, SVHN dataset and SAR automatic target recognition (ATR), which has been widely studied over the years (Ding, & Wen, 2019; Dong, & Kuang, 2015), good results are obtained. Compared with the existing deep learning approaches which get good performance in image classification, our MMCFA model is a single layer model based on Bayesian theory while has excellent classification performance. And by using multi-scale convolution kernels, our model could extract information-rich features from images at different levels in a single layer. The main contributions of this study are summarized in the following:

- Convolution operator is applied to modify the traditional FA model. The kernels in CFA model can maintain the spatial correlation among the image pixels and capture the structural information from images.
- Multi-scale convolution kernels are employed to extract features at different levels, thus providing more comprehensive information of images.
- Our MMCFA model is a combination of LVSVM and multi-scale CFA model. LVSVM incorporates supervised information into multi-scale CFA model by influencing the learning of factors, and parameters of the two models are learned jointly in a united framework.

The rest of the paper is structured as follows. We will have a brief review of the FA model and LVSVM in Section 2. Further, we introduce our MMCFA model in Section 3. Then the model learning and classification method are discussed in Section 4. The detailed experiments are conducted on synthetic, MNIST and measured SAR datasets to evaluate the effectiveness and efficiency of our model in Section 5. Finally, we summarize this paper in Section 6.

2. Background

We begin with a brief overview of the fundamentals of FA model and LVSVM, which constitute the major building blocks of the proposed MMCFA model.

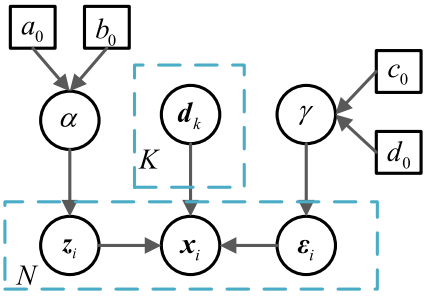


Fig. 1. Graphical representation of the VB-FA model. The detailed generative process is described in Eq. (2).

2.1. FA model

FA model is a method for modeling the correlations in multidimensional data by a low-dimensional latent variable. Denote the N observed data set by $\mathbf{X}=\{\mathbf{x}_i\}_{i=1}^N$. The FA model assumes that $\mathbf{x}_i$ is generated via a linear mapping from a low-dimensional latent variable plus an observation noise, and $\mathbf{x}_i$ can be represented as follows

$$\mathbf{x}_i = \mathbf{D}\mathbf{z}_i + \boldsymbol{\mu} + \boldsymbol{\varepsilon}_i \tag{1}$$

where μ is the mean vector of the data; $\mathbf{z}_i$ is a latent factor and assumed Gaussian distributed as $N(\mathbf{z}_i|0,\mathbf{I})$ with $N(\bullet)$ denoting the Gaussian distribution; $\mathbf{D}$ is the factor loading matrix that maps the latent factor $\mathbf{z}_i$ to the observed data $\mathbf{x}_i$; the noise variable $\boldsymbol{\varepsilon}_i$ is assumed as $N(\boldsymbol{\varepsilon}_i|0,\boldsymbol{\Psi})$. The noise covariance $\boldsymbol{\Psi}$ is a diagonal matrix and can be thought of as being sensor noise. Thus, $\mathbf{x}_i \sim N(\mu, \mathbf{D}\mathbf{D}^{\mathrm{T}} + \boldsymbol{\Psi})$.

The unknown parameters μ, $\mathbf{D}$ and $\boldsymbol{\Psi}$ in Eq. (1) can be estimated via the iterative expectation-maximization (EM) algorithm (Rubin, & Thayer, 1982). However, the learning of the parameters is very sensitive to the initial values of the unknown parameters, which easily makes the model converge to the local optimal solution. In Nielsen (2004), a hierarchical FA model based on the VB algorithm, i.e., VB-FA, was developed to mitigate the influence of the initial parameter values and could be expressed as

$$\begin{aligned}
&\mathbf{x}_i = \mathbf{D}\mathbf{z}_i + \boldsymbol{\mu} + \boldsymbol{\varepsilon}_i \\
&\mathbf{z}_i \sim \mathcal{N}(0, \alpha^{-1}\mathbf{I}), \alpha \sim \mathrm{Ga}(a_0, b_0) \\
&\mathbf{d}_k \sim \mathcal{N}(0, \mathbf{I}), \\
&\boldsymbol{\varepsilon}_i \sim \mathcal{N}(0, \gamma^{-1}\mathbf{I}), \gamma \sim \mathrm{Ga}(c_0, d_0)
\end{aligned} \tag{2}$$

where $\mathbf{d}_k$ is the kth column of the factor loading matrix $\mathbf{D}$ with $k \in \{1, \ldots, K\}$; α and γ are the precisions of $\mathbf{z}$ and $\boldsymbol{\varepsilon}$, respectively; a_0,b_0, c_0, d_0 are preset hyperparameters, and $\mathrm{Ga}(\bullet)$ denotes the Gamma distribution parameterized by the arguments in the parentheses. For further illustration, we present the graphical representation of the VB-FA model in Fig. 1.

In this paper, we mainly refer to VB-FA as the traditional FA model.

2.2. LVSVM

Given a labeled training dataset $\{(\mathbf{x}_i, y_i)|y_i \in \{-1, +1\}\}_{i=1}^N$, support vector machines (SVM) describes a binary linear classification with the decision function $\hat{y}_i = \mathrm{sign}(\boldsymbol{\omega}^{\mathrm{T}}\tilde{\mathbf{x}}_i)$, where $\boldsymbol{\omega}$ is the weighted coefficient and $\tilde{\mathbf{x}}_i = [\mathbf{x}_i; 1]$ is augmented feature vector.$\mathbf{x}_i$ is classified as a positive sample $(+1)$ when$\hat{y}_i \geq 0$, while as a negative sample (-1) in case of $\hat{y}_i < 0$. SVM is a max-margin method, in which the margin is defined as the smallest distance between the decision boundary and any of the samples. To maximize the margin, SVM deals with the problem

$$\begin{aligned}
&\min_{\boldsymbol{\eta}, \xi_n} \frac{1}{2}\|\boldsymbol{\omega}\|_2^2 + C_0 \sum_{i=1}^N \xi_i \\
&s.t. \quad y_i\boldsymbol{\omega}^{\mathrm{T}}\tilde{\mathbf{x}}_i \geq 1 - \xi_i \\
&\quad \xi_i \geq 0, i = 1, \ldots, N
\end{aligned} \tag{3}$$

where C_0 is a positive tuning parameter. Problem Eq. (3) can be solved by convex optimization algorithm.

Conventionally, the underlying discriminative objective is a linear hinge loss function, $\max(1 - y_i\boldsymbol{\omega}^{\mathrm{T}}\tilde{\mathbf{x}}_i, 0)$, and is hard to be modeled using traditional Bayesian analysis. Unlike the conventional way, Polson and Scott (2011) present a latent variable representation of SVM (LVSVM). They use the complete likelihood function of the weighted coefficient of SVM to represent the entire optimization function. The optimization criterion is expressed as a conditional Gaussian model, so that SVM can be combined with a probabilistic model. Specifically, the optimization of Eq. (3) mode is equivalent to finding the following coefficient of pseudo-posterior distribution

$$p(\boldsymbol{\omega}|\mathbf{y}) \propto \exp(-d(\boldsymbol{\omega})) \propto C_\alpha \phi(\mathbf{y}|\boldsymbol{\omega}) p(\boldsymbol{\omega}) \tag{4}$$

where $\mathbf{y}=[y_1,y_2,\ldots, y_N]$; C_α is the normalized constant of pseudo-posterior distribution, and does not need to be solved in classical algorithm analysis; $p(\boldsymbol{\omega})$ is prior distribution of coefficient $\boldsymbol{\omega}$; $\varphi(\mathbf{y}|\boldsymbol{\omega})$ is the pseudo-likelihood contribution, and can be expressed as

$$\phi(\mathbf{y}|\boldsymbol{\omega}) = \prod_i \phi(y_i|\boldsymbol{\omega}) = \exp\left(-2C_0 \sum_{i=1}^N \max\left(1 - y_i\boldsymbol{\omega}^{\mathrm{T}}\tilde{\mathbf{x}}_i, 0\right)\right) \tag{5}$$

Further, $\varphi_i(y_i|\boldsymbol{\omega})$ is expressed as a location-scale mixture of normal to deal with hinge loss function

$$\begin{aligned}
\phi_i(y_i|\boldsymbol{\omega}) &= \exp\left\{-2C_0 \max\left(1 - y_i\boldsymbol{\omega}^{\mathrm{T}}\tilde{\mathbf{x}}_i, 0\right)\right\} \\
&= \int_0^\infty \frac{1}{\sqrt{2\pi\lambda_i}} \exp\left(-\frac{\left(\lambda_i + C_0\left(1 - y_i\boldsymbol{\omega}^{\mathrm{T}}\tilde{\mathbf{x}}_i\right)\right)^2}{2\lambda_i}\right) d\lambda_i
\end{aligned} \tag{6}$$

$\varphi_i(y_i|\boldsymbol{\omega})$ can be regarded as the marginal from a joint distribution $\varphi_i(y_i,\lambda_i|\boldsymbol{\omega})$, which is conjugate to multivariate normal prior distribution.

We employ a Student-t prior which implemented via the hierarchical construction of normal-gamma distribution on $\boldsymbol{\omega}$:

$$\boldsymbol{\omega} \sim N\left(0, \sigma^{-1}\mathbf{I}\right), \ \sigma \sim \mathrm{Ga}(a_0, b_0) \tag{7}$$

According to Eqs. (4) and (5), we can express the SVM pseudo-posterior distribution as the marginal distribution of a higher dimensional distribution with the augmented variables $\boldsymbol{\lambda}$. Then complete data pseudo-posterior distribution can be written down as

$$\begin{aligned}
p\left(\boldsymbol{\omega}, \boldsymbol{\lambda}, \sigma|\mathbf{y}\right) &\propto \prod_{i=1}^N \phi_i(y_i, \lambda_i|\boldsymbol{\omega}) p(\boldsymbol{\omega}|\sigma) p(\sigma) \\
&\propto \prod_{i=1}^N \lambda_i^{-\frac{1}{2}} \exp\left(-\frac{\left(\lambda_i + C_0\left(1 - y_i\boldsymbol{\omega}^{\mathrm{T}}\tilde{\mathbf{x}}_i\right)\right)^2}{2\lambda_i}\right) \\
&\quad N\left(\boldsymbol{\omega}; 0, \sigma^{-1}\mathbf{I}\right)\mathrm{Ga}(\sigma; a_0, b_0)
\end{aligned} \tag{8}$$

Consequently, Gibbs sampling algorithm (Chen et al., 2015; Geman, & Geman, 1984) can be implemented to repeatedly sample each random variable from its conditional distribution. The augmented data space allows the SVM optimality criterion to be expressed.

3. Max-margin multi-scale convolutional factor analysis model

3.1. Multi-scale CFA model

As discussed in Section 1, the convolutional networks are widely used for data representation, especially in the image processing field. The convolution kernels can reflect the correlation between the image pixels in local regions. And multi-scale convolution kernels in a single layer can provide more comprehensive information of images (Chang et al., 2018; Nah, Kim, & Lee, 2017). Exploiting the ideas, we could develop a multi-scale CFA model in which the factor loading matrix $\mathbf{D}$ is not a projection matrix, but a set of convolution kernels at different scales.

Consider N images $\{\mathbf{x}_i\}_{i=1}^N$, with $\mathbf{x}_i \in \mathbb{R}^{N_x \times N_y}$, where N_x and N_y represent the number of pixels in images. We consider each image $\mathbf{x}_i$ is expanded in terms of convolution kernels. Let $\mathbf{D} = \{\mathbf{d}_k^l \in \mathbb{R}^{N_x^l \times N_y^l}\}_{k=1,l=1}^{K,L}$ be the multi-scale convolution kernel bank with L different scales, and K kernels per scale, where $N_x^l \ll N_x$ and $N_y^l \ll N_y$. Define $\mathbf{Z} = \{\mathbf{Z}_i\}_{i=1}^N$ as the set of factors, where $\mathbf{Z}_i = \{\mathbf{Z}_{i,k}^l \in \mathbb{R}^{(N_x - N_x^l+1)\times(N_y-N_y^l+1)}\}_{k=1,l=1}^{K,L}$ consists of $K \times L$ factors maps for the representation of image $\mathbf{X}_i$. So the image $\mathbf{X}_i$ can be represented as:

$$\mathbf{x}_i = \sum_{l=1}^{L}\sum_{k=1}^{K} \mathbf{d}_k^l * \mathbf{Z}_{i,k}^l + \boldsymbol{\mu} + \boldsymbol{\varepsilon}_i \tag{9}$$

where $*$ is the convolution operator; μ is the mean matrix of the data; $\boldsymbol{\varepsilon}_i \in \mathbb{R}^{N_x \times N_y}$ is the noise matrix. Since the size of all factor weight $\mathbf{Z}_i$, i.e., $(N_x - N_x^l + 1) \times (N_y - N_y^l + 1) \times K \times L$, is much larger than that of the input $\mathbf{x}_i$, i.e., $N_x \times N_y$, our model is overcomplete and runs the risk of having infinite solutions for the convolution kernels, where some of them can not reflect the inputs' structures. Therefore, in this model, we perform max-pooling on each factor weight $\mathbf{Z}_{i,k}^l$, where all elements of $\mathbf{Z}_{i,k}^l$ are set to zero other than the maximum one in the pooling region, to facilitate the learning of the convolution kernels with certain structures. A similar method was also adopted in Lee, Pham et al. (2009), and Norouzi, Ranjbai, & Mori (2009).

The construction in Eq. (9) may be viewed as a special class of FA model. Specially, we can rewrite Eq. (9) as:

$$\mathbf{x}_i = \sum_{l=1}^{L}\sum_{k=1}^{K}\sum_{s=1}^{S} z_{i,k,s}^l \mathbf{d}_{k,s}^l + \boldsymbol{\mu} + \boldsymbol{\varepsilon}_i \tag{10}$$

where $\mathbf{d}_{k,s}^l \in \mathbb{R}^{N_x \times N_y}$ represents a shifted version of $\mathbf{d}_k^l$. In detail, $\mathbf{d}_k^l$ is firstly padded with zeros to scale $N_x \times N_y$, and then $\mathbf{d}_{k,s}^l$ is generated via circularly shifting the elements in the zero-padded matrix with all S possible shifts. The model can be implemented via the following hierarchical Bayesian framework:

$$\begin{aligned}
z_{i,k,s}^l &\sim \mathcal{N}(0, 1/\alpha_{i,k,s}^l), \alpha_{i,k,s}^l \sim \mathrm{Ga}(a_0, b_0),\\
d_{k,j}^l &\sim \mathcal{N}(0, 1/\beta_{k,j}^l), \beta_{k,j}^l \sim \mathrm{Ga}(c_0, d_0), j = 1, \ldots J_l,\\
\boldsymbol{\varepsilon}_i &\sim \mathcal{N}(0, \gamma_i^{-1}\mathbf{I}), \gamma \sim \mathrm{Ga}(e_0, f_0),
\end{aligned} \tag{11}$$

where J_l ($J_l = N_x^l \times N_y^l$) denotes the number of elements in the convolution kernels $\mathbf{d}_k^l$, and $d_{k,j}^l$ is the jth element of $\mathbf{d}_k^l$; a_0, b_0, c_0, d_0, e_0, and f_0 are preset hyperparameters.

Compared to the traditional FA model, the main differences in model structure come from three aspects as follows.

Firstly, the convolution operator is introduced into the multi-scale CFA model, where $\mathbf{d}_k^l$ is a convolution kernel rather than a basis vector in the traditional FA model. By expanding the images in terms of convolution kernels., kernels learned by the multi-scale CFA model can reflect the correlation between the image pixels in two-dimensional space, thus solving the problem that the traditional FA model ignores the spatial correlation among the image pixels. Moreover, convolution kernels have extremely small size, i.e., $J_l \ll N_x \times N_y$, leading to the fact that features extracted by these models focus more on the local details and are prone to capture basic structures in images.

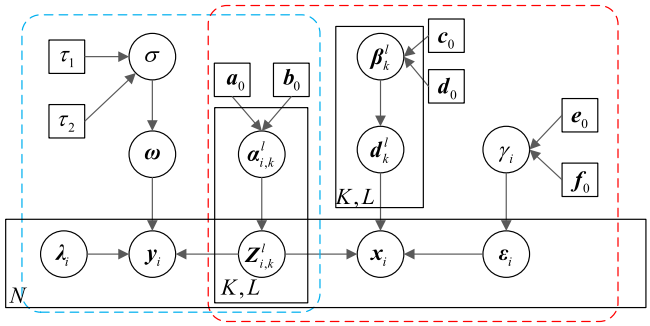


Fig. 2. Graphical representation of the MMCFA model. The multi-scale CFA model structure is in the red dotted line and describes the generation of $\mathbf{x}_i$. And the LVSVM structure is in the blue dotted line and is describes the classification process for the category of $\mathbf{x}_i$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Secondly, multi-scale convolution kernels are employed to extract features at different levels. As discussed in Section 1, the CFA model can only extract features of a single level based on the size of the scale. In multi-scale CFA model, the kernels at smaller scales mainly capture lower-level features, while the kernels at larger scales are more responsible for higher-level features, thus providing more comprehensive information of images.

3.2. MMCFA model

Assume that a label $y_i \in \{1, \ldots, C\}$ is associate with each of the N images, so that the training set may be denoted $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$. We wish to learn a classifier that maps the factors $\mathbf{Z}_i = \{\mathbf{Z}_{i,k}^l\}_{k=1,l=1}^{K,L}$ to an associated label y_i. The $\mathbf{Z}_i$ can be unfolded into the vector $\mathbf{z}_i$. We desire the classifier mapping $\mathbf{z}_i \to y_i$ and our goal is to learn the multi-scale CFA model and classifier jointly.

As mentioned in Section 1, LVSVM is chosen as the classifier, since it is able to model label $\mathbf{y}$ in the hinge loss function under Bayesian framework and has the successful behavior of max-margin models in classification. The MMCFA model structure can be expressed as

$$\begin{aligned}
\mathbf{x}_i &= \sum_{l=1}^{L}\sum_{k=1}^{K}\sum_{s=1}^{S} z_{i,k,s}^l \mathbf{d}_{k,s}^l + \boldsymbol{\mu} + \boldsymbol{\varepsilon}_i\\
z_{i,k,s}^l &\sim \mathcal{N}\left(0, 1/\alpha_{i,k,s}^l\right), \alpha_{i,k,s}^l \sim \mathrm{Ga}(a_0, b_0),\\
d_{k,j}^l &\sim \mathcal{N}\left(0, 1/\beta_{k,j}^l\right), \beta_{k,j}^l \sim \mathrm{Ga}(c_0, d_0), j = 1, \ldots J_l^2,\\
\boldsymbol{\varepsilon}_i &\sim \mathcal{N}(0, \gamma_i^{-1}\mathbf{I}), \gamma \sim \mathrm{Ga}(e_0, f_0),\\
& y_i, \lambda_i|\boldsymbol{\omega}, \tilde{\mathbf{w}}_i \sim \phi(y_i, \lambda_i|\boldsymbol{\omega}, \tilde{\mathbf{w}}_i),\\
\boldsymbol{\omega}|\sigma &\sim \mathcal{N}(0, \sigma^{-1}\mathbf{I}), \sigma \sim \mathrm{Ga}(\tau_1, \tau_2)
\end{aligned} \tag{12}$$

where $\tilde{\mathbf{z}}_i = [\mathbf{z}_i; 1]$ is the augmented feature vector of $\mathbf{w}_i$. In MMCFA, we consider the latent representation $\mathbf{w}_i$ as the input features to learn a discriminative latent space described by $\mathbf{D}$. In case of small sample size, $\mathbf{z}_i$ is expanded by $\mathbf{Z}_i'$ ($\mathbf{Z}_i' = [\sum_{k=1}^K \mathbf{Z}_{i,k}^1; \sum_{k=1}^K \mathbf{Z}_{i,k}^2; \ldots; \sum_{k=1}^K \mathbf{Z}_{i,k}^L]$) rather than $\mathbf{Z}_i$ in order to solve the fitting problem. To further analyze and compare with the multi-scale CFA model, we show the graphical representation of the MMCFA model in Fig. 2.

According to Fig. 2, it is clear that the data $\mathbf{x}_i$ generation and classification process are combined through the factor $\mathbf{Z}_i$ in MMCFA model. Therefore, $\mathbf{Z}_i$ is determined by not only the data $\mathbf{x}_i$ but also its label information, thus guaranteeing the linear separability of

the data in the hidden space. Details for training and inference are provided in the Section 4.

4. Model learning and classification

In this section, we infer model parameters by a Markov chain Monte Carlo (MCMC) (Link & Eaton, 2012) method based on Gibbs sampling (Geman, & Geman, 1984). Samples are constituted by iteratively drawing each random variable (model parameters and latent variables) from its conditional posterior distribution given the most recent values of all the other random variables. In the proposed model, all conditional distributions used to draw samples are analytic and relatively standard in Bayesian analysis.

4.1. Inference with Gibbs sampling

In the training phase, we seek to estimate the posterior distribution of the model parameters. By Bayesian's rule, the posterior probability density function (the full pseudo-posterior function) can be expressed as

$$\begin{aligned} &p(\mathbf{Z},\mathbf{D},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\omega},\boldsymbol{\lambda},\sigma\,|\mathbf{X},\mathbf{y}) \\ &\quad\propto p(\mathbf{X},\mathbf{Z},\mathbf{D},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma})p(\boldsymbol{\omega},\boldsymbol{\lambda},\sigma\,|\mathbf{y}) \\ &\quad=\prod_{i=1}^{N}p(\mathbf{x}_i|\mathbf{Z}_i,\mathbf{D},\gamma_i)\prod_{l=1}^{L}\prod_{k=1}^{K}p\left(\mathbf{Z}_{i,k}^{l}|\boldsymbol{\alpha}_{i,k}^{l}\right)p\left(\boldsymbol{\alpha}_{i,k}^{l}\right)p\left(\mathbf{d}_k^l\middle|\boldsymbol{\beta}_k^l\right)p\left(\boldsymbol{\beta}_k^l\right) \\ &\quad\times\prod_{i=1}^{N}\phi_i(y_i,\lambda_i|\boldsymbol{\omega})p(\boldsymbol{\omega}|\sigma)p(\sigma) \end{aligned} \tag{13}$$

The conditional posteriors used to draw samples can be derived via Eq. (13). The posterior computation with the MCMC method based on Gibbs sampling is realized in the present study (Gelfand, & Smith, 1990; Geman, & Geman, 1984).

For each MCMC iteration, the samples are drawn from the following conditional distributions:

- Sampling $\mathbf{Z}_{i,k}^{l}$ (i.e., $\{z_{i,k,s}^{l}\}_{s\in S}$): we have $p(z_{i,k,s}^{l}|-)=\mathcal{N}(\mu_{z_{i,k,s}^{l}},\Sigma_{z_{i,k,s}^{l}})$, with

$$\begin{aligned} \Sigma_{z_{i,k,s}^{l}} &= \left(\mathbf{d}_{k,s}^{l\ \mathrm{T}}\mathbf{d}_{k,s}^{l}\gamma_i+\alpha_{i,k,s}^{l}+\lambda_i^{-1}C_0^2\omega_n^2\right)^{-1} \\ \mu_{z_{i,k,s}^{l}} &= \Sigma_{z_{i,k,s}^{l}}\left(\gamma_i\mathbf{x}_{i,k,s}^{\mathrm{T}}\mathbf{d}_{k,s}^{l}+y_i\omega_nC_0\left(1+\lambda_i^{-1}C_0\xi_{i,k,s}^{l}\right)\right) \end{aligned} \tag{14}$$

where $\mathbf{x}_{i,k}=\mathbf{x}_{-i}+\mathbf{Z}_{i,k}^{l}*\mathbf{d}_k^l$, $\mathbf{x}_{-i}=\mathbf{x}_i-\sum_{l=1}^{L}\sum_{k=1}^{K}\mathbf{Z}_{i,k}^{l}*\mathbf{d}_k^l$, $\xi_{i,k,s}^{l}=1-y_i\boldsymbol{\omega}^{\mathrm{T}}\tilde{\mathbf{z}}_i+y_i\omega_n\tilde{z}_{i,n}$, $\omega_n(n=(lk-1)\times(N_x-N_x^l+1)\times(N_y-N_y^l+1)+s)$ denotes the nth element of $\boldsymbol{\omega}$, and $\tilde{z}_{i,n}$ denotes the nth element of $\tilde{\mathbf{z}}_i$ ($n=(l-1)\times(N_x-N_x^l+1)\times(N_y-N_y^l+1)+s$ in the case of small sample size). Taking advantage of the convolution property, we simultaneously update the posterior mean and covariance of the coefficients for the shifted versions of one convolution kernel element. Consequently,

$$\begin{aligned} \boldsymbol{\Sigma}_{\mathbf{Z}_{i,k}^{l}} &= \mathbf{1}\oslash\left(\gamma_i||\mathbf{d}_k^l||_2^2+\boldsymbol{\alpha}_{i,k}^{l}+\lambda_i^{-1}C_0^2\boldsymbol{\omega}_{k,l}^2\right) \\ \boldsymbol{\mu}_{\mathbf{Z}_{i,k}^{l}} &= \gamma_i\boldsymbol{\Sigma}_{\mathbf{Z}_{i,k}^{l}}\odot\left(\mathbf{x}_{-i}*\mathbf{d}_k^l+||\mathbf{d}_k^l||_2^2+y_i\tilde{\omega}_{k,l}C_0\left(1+\lambda_i^{-1}C_0\xi_{i,k}^{l}\right)\right) \end{aligned} \tag{15}$$

where both $\boldsymbol{\Sigma}_{\mathbf{Z}_{i,k}^{l}}$ and $\boldsymbol{\mu}_{\mathbf{Z}_{i,k}^{l}}$ have the same size with $\mathbf{Z}_{i,k}^{l}$; $\xi_{i,k}^{l}=1-y_i\boldsymbol{\omega}^{\mathrm{T}}\tilde{\mathbf{z}}_i+y_i\boldsymbol{\omega}_{k,l}\tilde{\mathbf{z}}_{i,k,l}$, $\tilde{\mathbf{z}}_i$ denotes the elements of $\tilde{\mathbf{z}}$ expanded by $\mathbf{Z}_i^l$, and $\tilde{\mathbf{z}}_{i,k,l}$ denotes the elements of $\tilde{\mathbf{z}}$ expanded by $\mathbf{Z}_{i,k}^{l}$; $\boldsymbol{\omega}_{k,l}$ denotes the elements of $\boldsymbol{\omega}$ that maps $\tilde{\mathbf{z}}_{i,k,l}$ to label; the symbol $\odot$ is the element-wise product operator and $\oslash$ is the element-wise division operator.

After updating $\mathbf{Z}_{i,k}^{l}$: we perform max-pooling on it at each iteration. In detail, within each max-pooling region, we set all coefficients to zero other than the maximum-amplitude coefficient.

- Sampling $\mathbf{d}_k^l$, we have $p(d_{k,j}^{l}|-)=\mathcal{N}(\mu_{d_{k,j}^{l}},\Sigma_{d_{k,j}^{l}})$, and the updated equations, where $\mu_{d_{k,j}^{l}}$ and $\Sigma_{d_{k,j}^{l}}$ are the jth element of the following vectors $\boldsymbol{\mu}_{d_k^l}$ and $\boldsymbol{\Sigma}_{d_k^l}$ respectively

$$\begin{aligned} \boldsymbol{\Sigma}_{d_k^l} &= \mathbf{1}\oslash\left(\sum_{i=1}^{N}\gamma_i||\mathbf{Z}_{i,k}^{l}||_2^2+\beta_k^l\right) \\ \boldsymbol{\mu}_{d_k^l} &= \boldsymbol{\Sigma}_{d_k^l}\odot\left(\sum_{i=1}^{N}\gamma_i\left(\mathbf{x}_{-i}*\mathbf{Z}_{i,k}^{l}+\mathbf{d}_k^l||\mathbf{Z}_{i,k}^{l}||_2^2\right)\right) \end{aligned} \tag{16}$$

- Sampling $\boldsymbol{\omega}$: we have $p(\boldsymbol{\omega}|-)=\mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\omega}},\boldsymbol{\Sigma}_{\boldsymbol{\omega}})$, with

$$\begin{aligned} \boldsymbol{\mu}_{\boldsymbol{\omega}} &= \boldsymbol{\Sigma}_{\boldsymbol{\omega}}\sum_{i=1}^{N}C_0y_i\left(1+\lambda_i^{-1}C_0\right)\tilde{\mathbf{z}}_i \\ \boldsymbol{\Sigma}_{\boldsymbol{\omega}} &= \left(\sigma\mathbf{I}+\sum_{i=1}^{N}\lambda_i^{-1}C_0^2\tilde{\mathbf{z}}_i\tilde{\mathbf{z}}_i^{\mathrm{T}}\right)^{-1} \end{aligned} \tag{17}$$

- Sampling λ_i: we have $p(\lambda_i|-)=\mathcal{GIG}(\lambda_i;\frac{1}{2},1,C_0^2(1-y_i\boldsymbol{\omega}^{\mathrm{T}}\tilde{\mathbf{z}}_i)^2)$, where $\mathcal{GIG}(.)$ denotes the generalized inverse Gaussian distribution (Devroye, 1986). We can infer that λ_i^{-1} obeys the inverse Gaussian distribution

$$p\left(\lambda_i^{-1}|-\right)=\mathcal{IG}\left(\lambda_i^{-1};1/\left|1-y_i\boldsymbol{\omega}^{\mathrm{T}}\tilde{\mathbf{z}}_i\right|,1\right) \tag{18}$$

where $\mathcal{IG}(x;a,b)=\sqrt{\frac{b}{2\pi x^3}}\exp(-(2a^2x)^{-1}b(x-a)^2)$, $a>0,b>0$.

- Sampling γ_i: we have $p(\gamma_i|-)=Ga(\tilde{e},\tilde{f})$, and the updated equations for $p(\gamma_i|-)$ are as follows:

$$\begin{aligned} \tilde{e} &= e_0+\tfrac{1}{2}N_xN_y \\ \tilde{f} &= f_0+\tfrac{1}{2}\left\|\mathbf{x}_i-\sum_{l=1}^{L}\sum_{k=1}^{K}\mathbf{d}_k^l*\mathbf{Z}_{i,k}^{l}\right\|_2^2 \end{aligned} \tag{19}$$

- Sampling $\alpha_{i,k,s}^{l}$: we have $p(\alpha_{i,k,s}^{l}|-)=Ga(\tilde{a},\tilde{b})$, and the updated equations for $p(\alpha_{i,k,s}^{l}|-)$ are as follows:

$$\begin{aligned} \tilde{a} &= a_0+\tfrac{1}{2} \\ \tilde{b} &= b_0+\tfrac{1}{2}z_{i,k,s}^{l\ 2} \end{aligned} \tag{20}$$

- Sampling $\beta_{k,s}^{l}$: we have $p(\beta_{k,s}^{l}|-)=Ga(\tilde{c},\tilde{d})$, and the updated equations for $p(\beta_{k,s}^{l}|-)$ are as follows:

$$\begin{aligned} \tilde{c} &= c_0+\tfrac{1}{2} \\ \tilde{d} &= d_0+\tfrac{1}{2}d_{k,j}^{l\ 2} \end{aligned} \tag{21}$$

- Sampling σ: we have $p(\sigma|-)=Ga(\tilde{\tau}_1,\tilde{\tau}_2)$, and the updated equations for $p(\sigma|-)$ are as follows:

$$\begin{aligned} \tilde{\tau}_1 &= \tau_1+\tfrac{1}{2}(K+1) \\ \tilde{\tau}_2 &= \tau_2+\tfrac{1}{2}\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\omega} \end{aligned} \tag{22}$$

On the basis of conditional distributions given above and Gibbs sampling, we can construct the Markov chain to draw samples iteratively with an initial condition (Algorithm 1).

4.2. Prediction procedure for a new sample

As introduced before, the training stage is supervised, while the prediction stage is unsupervised. So the conditional distribution parameter of the test is different from these deduced before. For a new sample $\mathbf{x}^*$, without any label information, the classification result is given by the following steps:

- Sampling $\mathbf{Z}_{\mathbf{x}^*,k}^{l}$ based on Eq. (23):

$$\begin{aligned} \boldsymbol{\Sigma}_{\mathbf{Z}_{\mathbf{x}^*,k}^{l}} &= \mathbf{1}\oslash\left(\gamma_{\mathbf{x}^*}||\mathbf{d}_k^l||_2^2+\boldsymbol{\alpha}_{\mathbf{x}^*,k}^{l}\right) \\ \boldsymbol{\mu}_{\mathbf{Z}_{\mathbf{x}^*,k}^{l}} &= \gamma_{\mathbf{x}^*}\boldsymbol{\Sigma}_{\mathbf{Z}_{\mathbf{x}^*,k}^{l}}\odot\left(\mathbf{x}_{-\mathbf{x}^*}*\mathbf{d}_k^l+||\mathbf{d}_k^l||_2^2\right) \end{aligned} \tag{23}$$

Algorithm 1 Training step of the MMCFA model.

Input: $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, T, $T_{\text{burn-in}}$, a_0, b_0, c_0, d_0 e_0, f_0, τ_1, τ_2
Output: Parameters $\mathbf{D}$, $\boldsymbol{\omega}$;
1: Initialize the parameter $\mathbf{D}$, $\mathbf{Z}$, $\boldsymbol{\omega}$, $\boldsymbol{\lambda}$, $\boldsymbol{\gamma}$, σ.
2: **for** $t = 1$ to T **do**
  3: **for** $l = 1$ to L **do**
    4: **for** $k = 1$ to K **do**
      5: Sample the parameter $\boldsymbol{\alpha}_k^l$ from Eq. (20);
      6: Sample the parameter $\mathbf{Z}_k^l$ from Eq. (15);
      7: Sample the parameter $\boldsymbol{\beta}_k^l$ from Eq. (21);
      8: Sample the parameter $\mathbf{d}_k^l$ from Eq. (16);
    9: **end for**
  10: **end for**
  11: Sample the parameter $\boldsymbol{\gamma}$ from Eq. (19);
  12: Sample the parameter $\boldsymbol{\lambda}$ from Eq. (18);
  13: Sample the parameter σ from Eq. (22);
  14: Sample the parameter $\boldsymbol{\omega}$ from Eq. (17);
15: **if** $t > T_{\text{burn-in}}$ **then**
  Collect parameters $\mathbf{D}$, $\boldsymbol{\omega}$;
16: **end if**
17: **end for**

where $\mathbf{x}_{-\mathbf{x}^*} = \mathbf{x}^* - \sum_{l=1}^{L}\sum_{k=1}^{K} \mathbf{Z}_{-\mathbf{x}^*,k}^l * \mathbf{d}_k^l$; $\gamma_{\mathbf{x}^*}$ can be sampled based on Eq. (19); $\boldsymbol{\alpha}_{\mathbf{x}^*,k}^l$ can be sampled based on Eq. (20).

In this step, all parameters are sampled T times as in the training phase.

- Computing the label of $\mathbf{x}^*$ based on Eq. (24):

$$\hat{y} = \text{sign}\left(\frac{1}{T_0}\sum_{t=1}^{T_0} \boldsymbol{\omega}_t^{\mathrm{T}} \tilde{\mathbf{z}}_t\right) \tag{24}$$

where $\mathbf{z}_t$ denotes the tth sample of the latent variable, and $\boldsymbol{\omega}_t$ denotes the tth collected SVM coefficients, and $T_0 = T - T_{\text{burn-in}}$. For MMCFA we average over all the collected samples from Gibbs sampler to predict the label of $\mathbf{x}^*$.

5. Experiments

In this section, we evaluate the performance of our proposed model in the synthesized data, MNIST, Fashion-MNIST, SVHN and MSTAR datasets. For all the datasets, the model is learned via Gibbs sampling.

5.1. Synthesized data

In this subsection, we design some simulation examples to demonstrate that our multi-scale CFA model is capable of extracting the basic structures of observed data and convolution kernels at different scales can capture structures at different levels.

Four building blocks shown in Fig. 3(a) are used to constitute four hundred synthetic samples, one of which is shown in Fig. 3(b). Each sample is generated by situating the building blocks arbitrarily. The building blocks are of size 8×8, and the synthetic samples are of size 64×64.

We model the above synthetic data via our multi-scale CFA model on scales of two. In this experiment, we assume the number of convolution kernels at different scales are the same and determined by minimizing of the average relative reconstruction error, i.e., root mean square error (RMSE). Here, the RMES is defined as $\frac{1}{N}\sum_{n=1}^{N} \|\hat{\mathbf{x}}_n - \mathbf{x}_n\|_2 / \|\mathbf{x}_n\|_2$, where $\mathbf{x}_n$ denotes the nth original sample, $\hat{\mathbf{x}}_n$ represents the corresponding recovery sample, and N denotes the number of samples. The large scale convolution kernels are of size 8×8 while the small scale convolution kernels are of size 4×4.

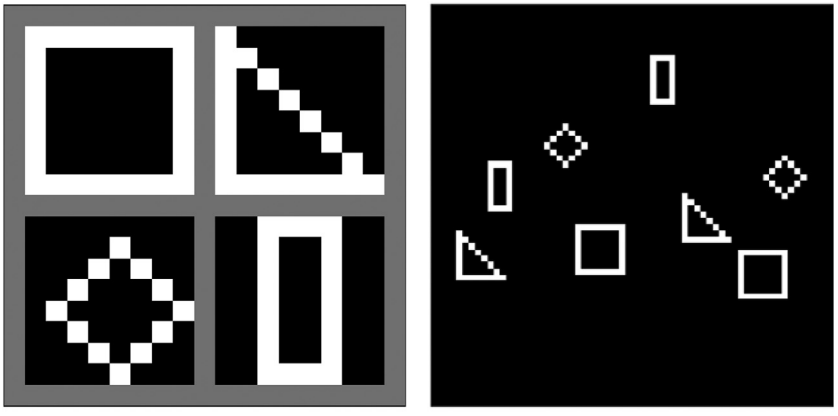
Fig. 4(a) and (b) show the learned convolution kernels via multi-scale CFA model with the number of kernels being set to be four. Fig. 4(c) depicts one of the reconstructed sample via multi-scale CFA model corresponding to the sample shown in Fig. 3(b). As we can see from Fig. 4(a) and(b), the learned large scale convolution kernels of our model are similar to the building blocks in this experiment, i.e., the four graphics elements: diamond, rectangle, box and triangle, while the learned small convolution kernels show some detail structures, i.e., two diagonal lines of different sizes.

Fig. 5 shows the learned convolution kernels in different iterations. 20 results are selected at equal interval from the first 60 iterations. During iterations, the small scale convolution kernels always extract detail structural information such as line segments in different angles and corner of the rectangles. And by comparing with the building blocks, we can see from Fig. 5(b) that the large scale convolution kernels always extract the global structural features from the synthesized data in the iterations.

5.2. MNIST dataset

5.2.1. Data description and performance of feature extraction

The MNIST dataset has 70k (60k train, 10k test) labeled images of hand-written digits. Digits in the MNIST dataset are stored in images of 28 × 28 pixels. Pixel intensities are in the range of 0 and 255. Table 1 shows the distribution of train and test set of the dataset. Fig. 6 illustrates the samples randomly selected from each class. In this experiment, we use all the 60k training samples for model training. And normalization preprocessing is adopted.



(a) (b)

Fig. 3. (a) Four building blocks are used for synthetic data generation. (b) One of four hundred samples are shown here, which are generated by using eight building blocks depicted in (a) and situating them arbitrarily. Each sample is of size 64×64.

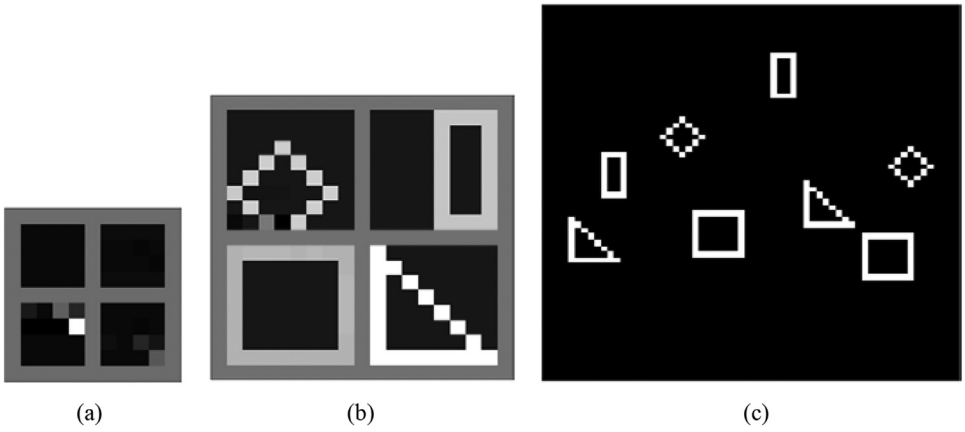


Fig. 4. (a) Convolution kernels at small scale, which are learned via multi-scale CFA model. (b) Convolution kernels at large scale, which are learned via multi-scale CFA model. (c) A reconstructed sample via multi-scale CFA model, corresponding to the sample shown in Fig. 3(b).

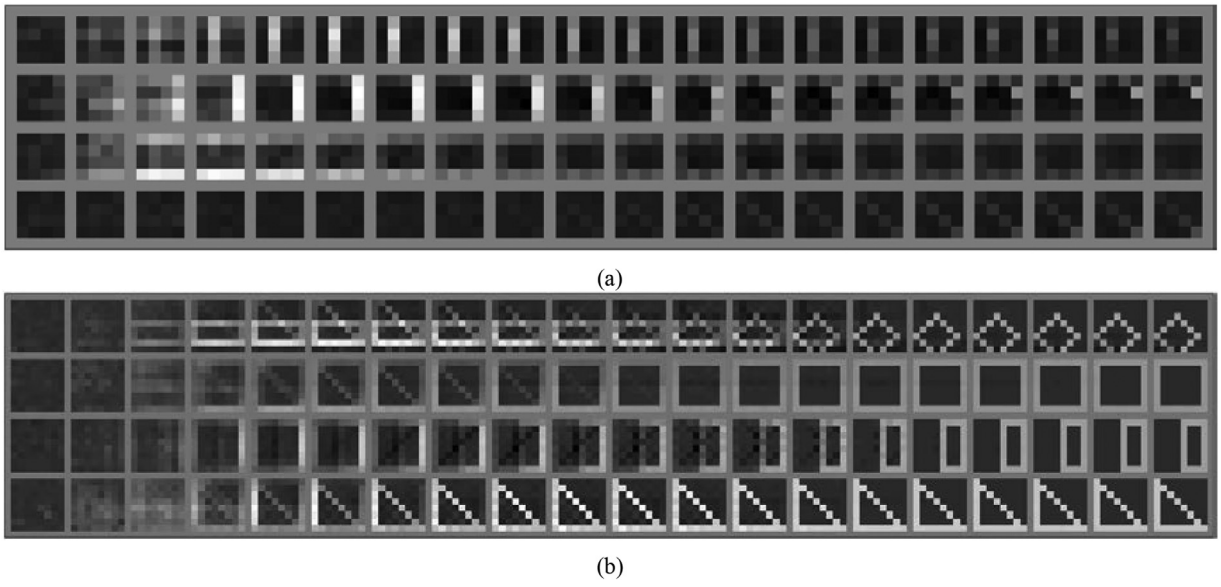


Fig. 5. 20 results of our multi-scale CFA model in the iterations. Rows represent different convolution kernels, and columns represent different iterations. (a) The learned small scale convolution kernels. (b) The learned large scale convolution kernels.

Table 1
Distribution of digits in train and test set of MNIST dataset.

	0	1	2	3	4	5	6	7	8	9	Total
Train	5923	6742	5958	6131	5842	5421	5918	6265	5851	5949	60,000
Test	980	1135	1032	1010	982	892	958	1028	974	1009	10,000

Fig. 6. MNIST samples of each class.

As stated in Section 3, the convolution kernels learned by our MMCFA model can reflect the basic structural information of images. To verify this, we train a MMCFA model with three scales (7×7 (64), 11×11 (81), 19×19 (81)) using MNIST dataset. As shown in Fig. 7(a) and (b), kernels at the small and medium scales can capture some details of the images, e.g., dots and short line segments. Meanwhile, in comparison with Fig. 6, we note convolution kernels at large scale, shown in Fig. 7(c), take on forms characteristic of digits and part of digits. By comparing Fig. 7(a), (b) and (c), kernels at different scales show different levels of structural features of the MNIST dataset, and offering the potential to improve the separability of the factors in classification task.

5.2.2. Classification results

For the classification task, firstly we compare the classification performance of our MMCFA model with different parameter settings, including different sizes of scale, different number of convolution kernels of each scale and different number of scales. In Table 2, "1-MMCFA" represents our model with single scale; "2-MMCFA" represents our model with two scales; "3-MMCFA" represents our model with three scales; "4-MMCFA" represents our model with four scales. As shown in Table 2, the 3-MMCFA (7×7 (64), 11×11 (81), 19×19 (81)) obtains the highest classification accuracy on MNIST dataset.

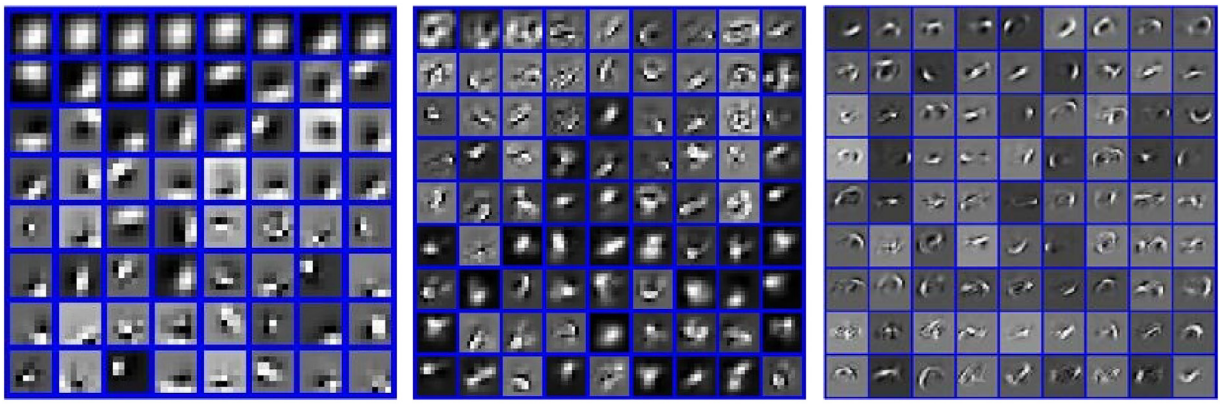


Fig. 7. The learned convolution kernels for MNIST data. (a) 64 small scale kernels of size 7×7. (b) 81 medium scale kernels of size 11×11 (c) 81 large scale kernels of size 19×19.

Table 2
Classification performance of MMCFA model for MNIST dataset.

	Parameter set	Accuracy
1-MMCFA	7×7 (64)	91.72% ± 1.36
	10×10 (64)	92.46% ± 1.54
	13×13 (81)	94.01% ± 1.67
	15×15 (81)	94.13% ± 1.14
	17×17 (81)	93.37% ± 1.09
2-MMCFA	7×7 (49), 10×10 (64)	95.73% ± 0.96
	7×7 (49), 13×13 (81)	97.94% ± 0.34
	7×7 (64), 15×15 (81)	98.43% ± 0.33
	7×7 (64), 17×17 (64)	96.45% ± 0.71
	9×9 (49), 17×17 (81)	96.35% ± 0.49
3-MMCFA	7×7 (49), 15×15 (64), 19×19 (81)	95.19% ± 0.97
	7×7 (64), 15×15 (64), 23×23 (64)	97.64% ± 0.43
	7×7 (49), 11×11 (49), 17×17 (81)	98.33% ± 0.51
	7×7 (64), 11×11 (81), 19×19 (81)	**98.74%** ± 0.22
	7×7 (64), 11×11 (64), 23×23 (64)	98.17% ± 0.46
4-MMCFA	7×7 (49), 11×11 (49), 15×15 (64), 19×19 (64)	96.33% ± 0.63
	7×7 (64), 11×11 (64), 17×17 (64), 19×19 (81)	96.58% ± 0.72
	7×7 (64), 11×11 (64), 15×15 (64), 23×23 (64)	97.19% ± 0.40
	7×7 (49), 11×11 (64), 17×17 (81), 23×23 (81)	97.94% ± 0.58
	7×7 (49), 11×11 (64), 19×19 (64), 23×23 (64)	97.74% ± 0.67

Table 3
Classification accuracies obtained by different classification methods for MNIST dataset.

Algorithm	Accuracy
FA	96.02%
SVM	93.95%
PCA	93.51%
C2 features (Borji et al., 2008)	96.40%
Freeman chain code template (Zaqout, 2011)	92.90%
KNN (LeCun et al., 1998)	97.17%
3-layer NN (LeCun et al., 1998)	97.55%
3-MMCFA	**98.74%**

Secondly, we compare the proposed model with some other MNIST classification methods, including traditional FA model (Zhu et al., 2012), linear SVM, Principal Component Analysis (PCA) with linear SVM (Wang, Han, Lu, Wu, & Huang, 2007), some feature based classification methods (Borji, Hamidi, & Mahmoudi, 2008; Zaqout, 2011), neural networks (NN) (LeCun et al., 1998) and K-nearest-neighbors (Cover, &Hart, 1967). In Table 3, "3-MMCFA" represents our proposed model with three scales; "FA" represents the traditional FA model using statistical classification framework (Zhu et al., 2012); "SVM" represents directly using original images as input features in linear SVM; "PCA" represents PCA with linear SVM (Wang et al., 2007); "C2 features" represents using the C2 feature in linear SVM (Borji et al., 2008); "Freeman chain code template" represents a statistical approach using some features concept of the freeman chin code template (Zaqout, 2011); "KNN" represents the K-nearest-neighbors (Cover, &Hart, 1967) method; "3-layer NN" represents neural networks with 3 layers (LeCun et al., 1998). Due to the feature extraction capability of multi-scale convolution kernels and the joint learning with LVSVM, we can see the classification accuracy of our proposed model is higher than the traditional FA model, which uses statistical classification framework (Jain et al., 2000) to utilize the label information. Compared to linear SVM, the features extracted by our model are obviously more discriminative than the original images. Compared to some traditional unsupervised methods (PCA Wang et al., 2007) C2 features Borji et al., 2008 and Freeman chain code template Zaqout, 2011), our proposed model yields highest accuracy due to the constraint of supervised information. Some common classification methods are also considered as comparison methods (NN LeCun et al., 1998 and KNN Cover, &Hart, 1967), which also utilize the supervised information during the training procedure. And our MMCFA model obtains the highest accuracy.

5.2.3. Parameters used in Gibbs sampling

As discussed in Geman and Geman (1984), and Gelfand and Smith (1990), when the Gibbs sampling is used to estimate a posterior distribution, a key issue is to determine when the procedure has essentially converged. In order to investigate the convergence of the Gibbs sampling in our model, we give the classification accuracies with different values of burn-in parameter $I_{burn-in}$, collection parameter I_{num} and sampling interval I_{space} respectively. Fig. 8 shows the classification accuracies of our model with different values of $I_{burn-in}$ for MNIST dataset, where 400 samples are collected with the interval of ten. We can see from Fig. 9 that the performance has been quite stable when the number of burn-in samples is larger than 3000. The standard deviation of classification accuracies for the final 1000 iterations in the burn-in step for MNIST is 0.0081. Therefore, the training procedure has converged through 3000 iterations in the burn-in step, and we set the value of $I_{burn-in}$ to be 4000 in our experiments.

In Fig. 9, we present the classification accuracies of our model with different values of I_{num} for MNIST dataset, where the value of $I_{burn-in}$ is set to be 4000 and the sampling interval is set to be 10. It is apparent that when the number of collection samples is larger than 600, the classification accuracies have converged to stable values. The standard deviation of classification accuracies for

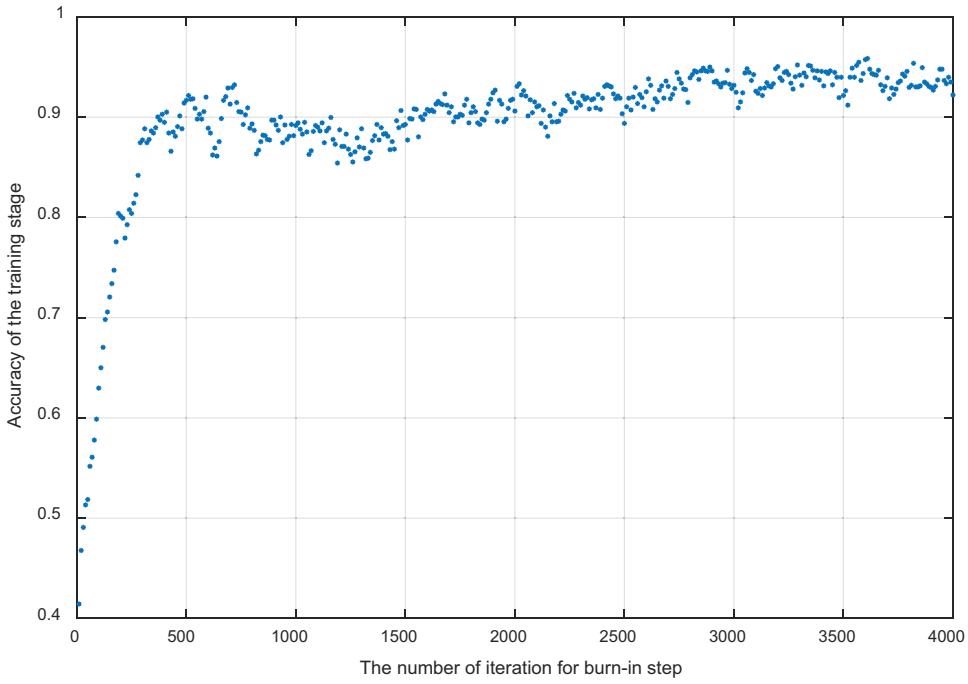


Fig. 8. The classification performance with different numbers of burn-in samples for MNIST dataset.

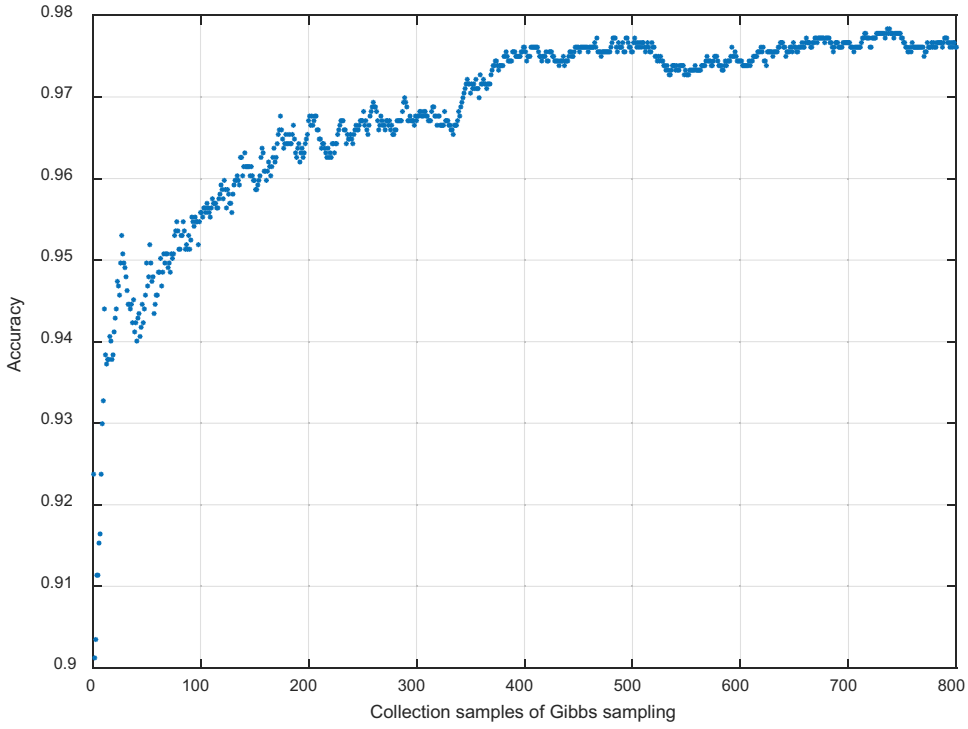


Fig. 9. The classification performance with different numbers of collection samples for MNIST dataset.

the final 200 iterations in the collection procedure for the MNIST dataset is 7.1280e−04. Therefore, the MMCFA model has converged when the value of I_{num} is larger than 600, and we set the value of I_{num} to be 800 in our experiments.

In the Gibbs sampling, the sampling interval I_{space} is a constant larger than 1 to eliminate the correlation of adjacent collection samples. The classification accuracies of our model with different values of I_{space} for MNIST dataset are depicted in Fig. 10, where the iterations of burn-in step is 4000 and 800 samples are collected. It is apparent that the value ofI_{space} has little influence on the classification accuracy. Therefore, the sampling interval I_{space} is set to be 10 in our experiments.

5.3. Fashion-MNIST dataset

Fashion-MNIST dataset is a dataset of Zalando's article images and consists of a training set of 60k examples and a test set of 10k examples (Xiao, Rasul, & Vollgraf, 2017). Each samples in the

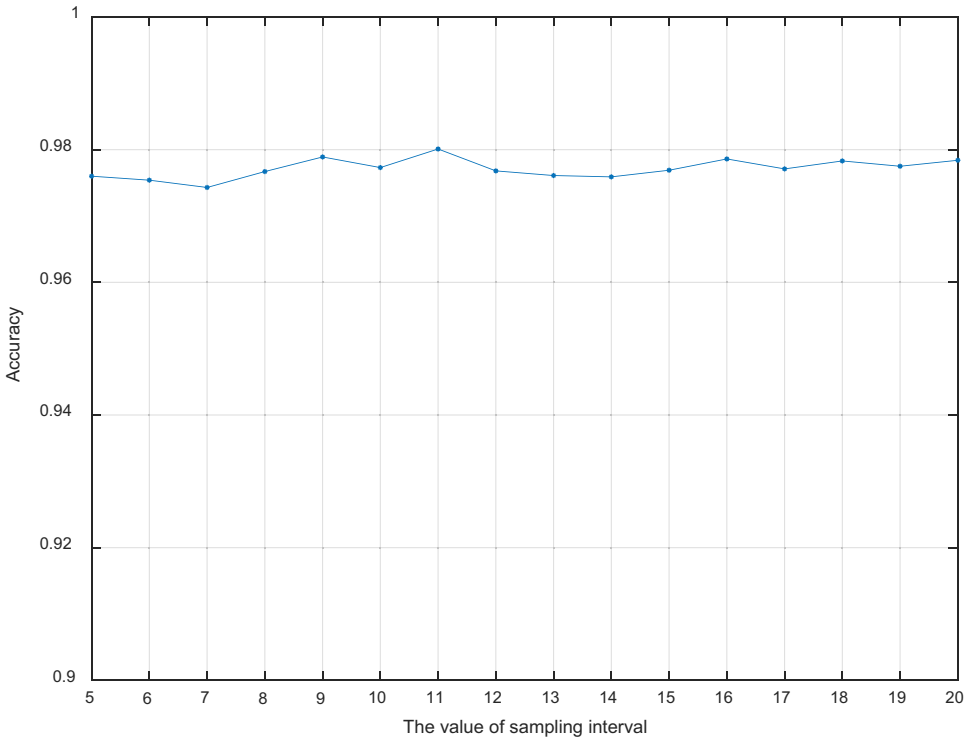


Fig. 10. The classification performance with the value of sampling interval for MNIST dataset.

Table 4
Classification accuracies obtained by different classification methods for Fashion-MNIST dataset.

Algorithm	Accuracy
SVC (Seo, & Shin, 2019)	89.70%
PCA	85.93%
EDEN (Seo, & Shin, 2019)	90.60%
CNN2 (Seo, & Shin, 2019)	91.17%
CNN3 (Xiao et al., 2017)	92.10%
2-MMCFA	91.27%
3-MMCFA	**92.36%**
4-MMCFA	91.54%

Table 5
Classification accuracies obtained by different classification methods for SVHN dataset.

Algorithm	Accuracy
K-means (Netzer, & Wang, 2011)	90.60%
HOG (Netzer, & Wang, 2011)	85.00%
VGG8-ELU (Bawa, & Kumar, 2019)	92.51%
AE (Sankaran et al., 2017)	89.90%
GSAE (Sankaran et al., 2017)	92.40%
2-MMCFA	92.11%
3-MMCFA	92.74%
4-MMCFA	**93.13%**

Fashion-MNIST dataset are stored in images of 28×28 pixels and associated with a label from 10 classes. It intends to serve as a direct drop-in replacement for the original MNIST dataset for benchmarking machine learning algorithms.

On Fashion-MNIST, Table 4 shows the comparative performance of the proposed algorithm along with different existing classification methods. In Table 4, "SVC" represents directly using original images as input in Support Vector Classifier (SVC) (Seo, & Shin, 2019);"PCA" represents PCA with kernel SVM (Wang et al., 2007); "EDEN" represents Evolutionary Deep Learning (EDEN) (Seo, & Shin, 2019); "CNN2" represents CNN consists of two convolutional and max-pooling layers (Seo, & Shin, 2019); "CNN3" represents CNN consists of three convolutional, BN and max-pooling layers (Xiao et al., 2017). From the test accuracy suggested in Table 4, our MMCFA models achieve higher accuracy than the single layer classifier (SVC, and PCA) and even better than the deep learning methods (EDEN, CNN2 and CNN3).

5.4. SVHN dataset

SVHN (Netzer, & Wang, 2011) is a real-world image dataset for developing machine learning and object recognition algorithms with minimal requirement on data preprocessing and formatting. This dataset contains 32×32 color images of digits with ten classes. It is divided into 73,257 training examples and 26,032 test examples. There are 531,131 additional images. We only use training and test examples in our experiments. It can be seen as similar in flavor to MNIST, but incorporates an order of magnitude more labeled data and comes from a significantly harder, unsolved, real world problem.

On SVHN, Table 5 shows the comparative performance of the proposed algorithm along with different existing classification methods. In Table 5, "K-means" represents the K-means clustering method (Netzer, & Wang, 2011); "HOG" represents the widely-used Histograms-of-Oriented-Gradients (HOG) features (Netzer, & Wang, 2011); "VGG8-ELU" represents eight layer VGGNet (Simonyan & Zisserman, 2014) with six convolutional layer and two dense layers (Bawa, & Kumar, 2019); "AE" represents AutoEncoder method (Sankaran, Vatsa, Singh, & Majumdar, 2017); "GSAE" represents Group Sparse AutoEncoder (GSAE) which use an effective regularization method to improve the learning capacity (Sankaran et al., 2017). From the test accuracy suggested in Table 5, our MMCFA models achieve the highest classification accuracy.

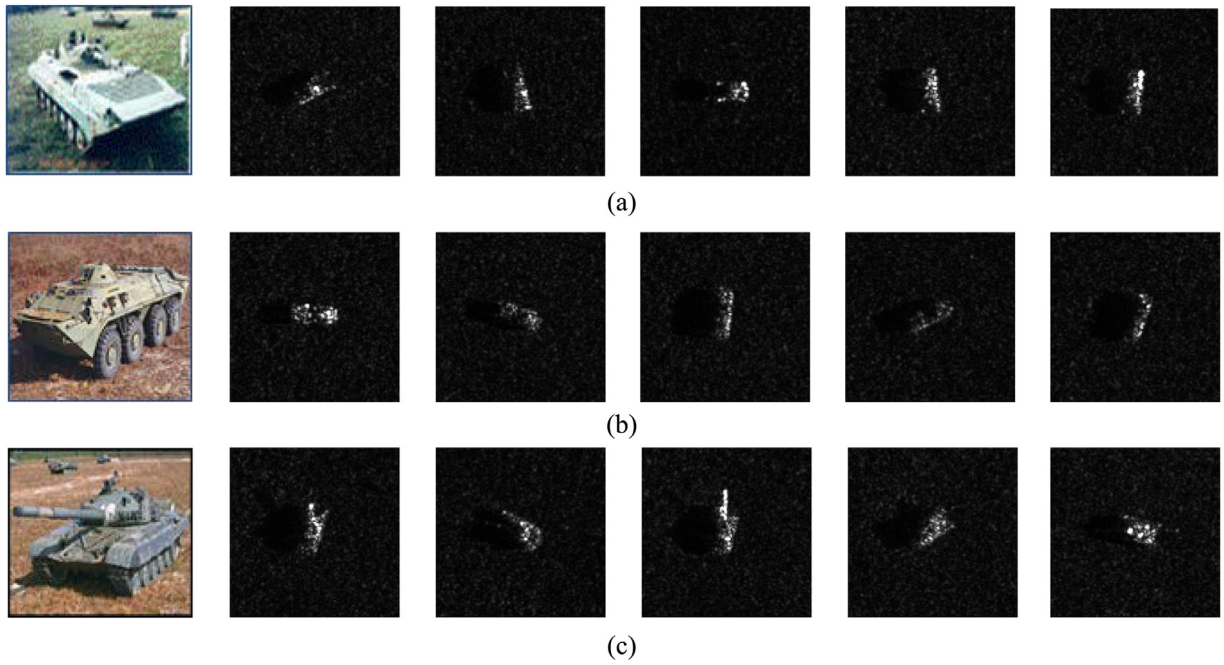


Fig. 11. SAR and optical images of the three-target data. The first column shows the optical images. The remaining columns gives some SAR imagery examples of the three targets. (a) BMP2 target. (b) BTR70 target. (c) T72 target.

Table 6
Type and number of training and test samples for three-target dataset.

Dataset	BMP2			BTR70	T72		
	C21	9566	9563	C71	132	S7	812
Training samples (17°)	233	0	0	233	232	0	0
Test samples(15°)	196	196	195	196	196	191	195

5.5. MSTAR dataset

MSTAR is a measured dataset with three-target data which contains two types of tanks, T72 and BMP2, and a vehicle BTR70. Some SAR and optical imagery examples of these three types of targets are shown in Fig. 11. Each of the target has full aspect coverage from 0° to 360° and different views at 15° and 17° depression angles. The data at depression 17° are used for training and those at depression 15° for test. The number of aspect views available for these targets is listed in Table 6. As shown in Table 6, we only use the images of BMP2-9563, BTR70-C71 and T72-132 at depression angle 17° as the training data; while all images of BMP2-C21, BMP2-9566, BMP2-9563, BTR70-C71, T72-132, T72-S7 and T72-812 at depression angle 15° are used as the test data.

MSTAR images available are of around 128 × 128 pixels, and cropped to 63×63 pixels region of interest in our experiments. The amplitudes of all images differ a lot in the MSTAR dataset. We adopt the normalization preprocessing method because huge difference of the amplitudes between training samples may conceal their own differences between the targets using our model to extract features. Except for normalization, we do not use any other preprocessing methods, such as data augmentation, filtering and image segmentation (Ni & Xu, 2013), in our experiments.

We compare the proposed model with some other SAR ATR methods, including linear SVM, joint Sparse Representation (Zhang, Nasrabadi, Zhang, & Huang, 2012), Principal Component Analysis with kernel SVM (Wang et al., 2007), Particle Swarm Optimization with Hausdorff Distance (Dungan, 2010), as well as three deep methods (Ni & Xu, 2013; Chen and Wang, 2014; Deng, Du, Li, Ding, & Liu, 2017). The correct classification rates of these methods

Table 7
3-target accuracies obtained by different SAR ATR methods for MSTAR dataset.

Algorithm	Accuracy
SVM	88.64%
JSR (Zhang et al., 2012)	93.20%
PCA (Wang et al., 2007)	88.46%
HD (Dungan, 2010)	87.56%
AE-CNN (Chen and Wang, 2014)	90.10%
Euclidean distance AE (Deng et al., 2017)	94.14%
Sparse AE (Ni & Xu, 2013)	92.10%
2-MMCFA	94.37%
3-MMCFA	**94.96%**
4-MMCFA	92.54%

are shown in Table 7. In Table 7, "2-MMCFA" represents our proposed model; "SVM" represents directly using original images as input features in linear SVM; "JSR" represents joint Sparse Representation (Zhang et al., 2012); "PCA" represents PCA with kernel SVM (Wang et al., 2007); "HD" represents Particle Swarm Optimization with Hausdorff Distance (Dungan, 2010); "AE-CNN" represents CNN with sparse AE pretraining (Chen and Wang, 2014); "Euclidean distance AE" represents AE combined with a supervised constraint by a Euclidean distance restriction on AE (Deng et al., 2017); "sparse AE" presents AE model with sparse constraint and other preprocessing methods, such as filtering and image segmentation (Ni & Xu, 2013). As shown in Table 7, compared to some traditional methods (JSR Zhang et al., 2012, PCA Wang et al., 2007 and HD Dungan, 2010), our proposed MMCFA model has achieved the highest accuracy due to the exploitation of multi-scale convolution operator and supervised restriction. Compared to these deep learning methods (sparse AE Ni & Xu, 2013, AE-CNN Chen et al., 2015 and Euclidean Distance AE Deng et al., 2017), the accuracy of our proposed model is also better.

6. Conclusions

In this paper, we propose an improved FA model, namely, MMCFA model, for image classification. The MMCFA model is a combination of the multi-scale CFA model and LVSVM. The multi-scale

CFA model can maintain the spatial correlation among the image pixels in two-dimensional space and capture multi-level structural information from images via multi-scale convolution kernels. Then the factors learned by multi-scale CFA ls linked to the LVSVM as input features. By joint learning with the LVSVM, the MMCFA model can extract discriminative factors for the classification task. To validate the proposed model, it is compared with some traditional classification methods on MNIST, Fashion-MNIST, SVHN and MSTAR datasets. Based on the experimental analysis on the classification accuracy, the proposed model obtains competitive classification performance.

Our model is the modified version of the traditional FA model inspired by the convolutional deep networks. In the future, we will mainly focus on the research of multi-layered models for image classification. By using a deep model, the deep structural features reflecting the physical characteristics of image are expected to be extracted and the ability of classification may be further improved.

Conflicts of interest

The authors declared that they have no conflicts of interest to this work.

Credit authorship contribution statement

Yuchen Guo: Conceptualization, Data curation, Formal analysis, Investigation, Software, Validation, Visualization, Writing - original draft. **Lan Du:** Funding acquisition, Methodology, Project administration, Resources, Supervision, Writing - review & editing. **Jian Chen:** Formal analysis, Supervision, Validation, Writing - original draft.

Acknowledgments

This work was partially supported by the National Science Foundation of China (No. 61771362, No. U1833203, No. 61671354), and 111 Project (B18039).

References

Alex, K., Ilya, S., & Geoff, H. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Proceeding Systems (NIPS), 25*, 1106–1114. https://doi.org/http://dx.doi.org/10.1016/j.protcy.2014.09.007.

Bawa, V. S., & Kumar, V. (2019). Linearized sigmoidal activation: A novel activation function with tractable non-linear characteristics to boost representation capability. *Expert Systems with Applications*. https://doi.org/10.1016/j.eswa.2018.11.042.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Drichlet allocation. *Journal of Machine Learning Reasearch, 3*, 993–1022.

Borji, A., Hamidi, M., & Mahmoudi, F. (2008). Robust handwritten character recognition with features inspired by visual ventral stream. *Neural Processing Letters, 28*, 97–111. https://doi.org/10.1007/s11063-008-9084-y.

Chang, H., Han, J., Zhong, C., Snijders, A. M., & Mao, J. H. (2018). Unsupervised transfer learning via multi-scale convolutional sparse coding for biomedical applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/TPAMI.2017.2656884.

Chen, B., Zhang, H., Zhang, X., Wen, W., Liu, H., & Liu, J. (2015). Max-margin discriminant projection via data augmentation. *IEEE Transactions on Knowledge and Data Engineering*. https://doi.org/10.1109/TKDE.2015.2397444.

Chen, M., Silva, J., Paisley, J., Wang, C., Dunson, D., & Carin, L. (2010). Compressive sensing on manifolds using a nonparametric mixture of factor analyzers: Algorithm and performance bounds. *IEEE Transactions on Signal Processing*. https://doi.org/10.1109/TSP.2010.2070796.

Chen, S., & Wang, H. (2014). SAR target recognition based on deep learning. In *In DSAA 2014 - Proceedings of the 2014 IEEE international conference on data science and advanced analytics* https://doi.org/10.1109/DSAA.2014.7058124.

Cover, T. M., & Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*. https://doi.org/10.1109/TIT.1967.1053964.

Deng, S., Du, L., Li, C., Ding, J., & Liu, H. (2017). SAR automatic target recognition based on euclidean distance restricted autoencoder. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. https://doi.org/10.1109/JSTARS.2017.2670083.

Devroye, L. (1986). General principles in random variate generation. *Non-uniform random variate generation* https://doi.org/10.1007/978-1-4613-8643-8_2.

Ding, B. Y., & Wen, G. J. (2019). Combination of global and local filters for robust SAR target recognition under various extended operating conditions. *Information Sciences, 476*, 48–63.

Dong, G., & Kuang, G. (2015). Classification on the monogenic scale space: Application to target recognition in SAR image. *IEEE Transactions on Image Processing*. https://doi.org/10.1109/TIP.2015.2421440.

Du, L., Liu, H., & Bao, Z. (2008). Radar HRRP statistical recognition: Parametric model and model selection. *IEEE Transactions on Signal Processing*. https://doi.org/10.1109/TSP.2007.912283.

Dungan, K. E. (2010). *Feature-based vehicle classification in wide-angle synthetic aperture radar*. The Ohio State Universit.

Erhan, D., Bengio, Y., Courville, A., & Vincent, P. (2009). *Visualizing higher-layer features of a deep network*. D´Epartement d'Informatique et Recherche Op´Erationnelle Technical Report 1341 https://doi.org/10.2464/jilm.23.425 .

Gao, S., Tsang, I. W. H., & Ma, Y. (2014). Learning category-specific dictionary and shared dictionary for fine-grained image categorization. *IEEE Transactions on Image Processing*. https://doi.org/10.1109/TIP.2013.2290593.

Gelfand, A. E., & Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*. https://doi.org/10.1080/01621459.1990.10476213.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/TPAMI.1984.4767596.

Han, H. M., Li, Y., & Zhu, X. Q. (2019). Convolutional neural network learning for generic data classification. *Information Sciences, 477*, 448–465.

Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/34.824819.

Lazebnik, S., Schmid, C., & Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition* https://doi.org/10.1109/CVPR.2006.68.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*. https://doi.org/10.1109/5.726791.

Lee, H., Grosse, R., Ranganath, R., & Ng, A. Y. (2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th annual international conference on machine learning - ICML '09* https://doi.org/10.1145/1553374.1553453.

Lee, H., Pham, P., Largman, Y., & Ng, A. (2009). Unsupervised feature learning for audio classification using convolutional deep belief networks. *Advances in Nerual Infromation Processing Systems*. https://doi.org/10.1145/1553374.1553453.

Link, W. A., & Eaton, M. J. (2012). On thinning of chains in MCMC. *Methods in Ecology and Evolution*. https://doi.org/10.1111/j.2041-210X.2011.00131.x.

Nah, S., Kim, T. H., & Lee, K. M. (2017). Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings - 30th IEEE conference on computer vision and pattern recognition, CVPR 2017* https://doi.org/10.1109/CVPR.2017.35.

Nielsen, F.B., (2004). Variational Approach to Factor Analysis and Related Models.

Netzer, Y., & Wang, T. (2011). Reading digits in natural images with unsupervised feature learning. *NIPS workshop on deep learning and unsupervised feature learning* https://doi.org/10.2118/18761-MS.

Ni, J. C., & Xu, Y. L. (2013). SAR automatic target recognition based on a visual cortical system. In *Proceedings of the 2013 6th international congress on image and signal processing, CISP 2013* https://doi.org/10.1109/CISP.2013.6745270.

Norouzi, M., Ranjbar, M., & Mori, G. (2009). Stacks of convolutional restricted Boltzmann machines for shift-invariant feature learning. *2009 IEEE computer society conference on computer vision and pattern recognition workshops, CVPR workshops 2009* https://doi.org/10.1109/CVPRW.2009.5206577.

Polson, N. G., & Scott, S. L. (2011). Data augmentation for support vector machines. *Bayesian Analysis*. https://doi.org/10.1214/11-BA601.

Rubin, D. B., & Thayer, D. T. (1982). EM algorithms for ML factor analysis. *Psychometrika*. https://doi.org/10.1007/BF02293851.

Sankaran, A., Vatsa, M., Singh, R., & Majumdar, A. (2017). Group sparse autoencoder. *Image and Vision Computing*. https://doi.org/10.1016/j.imavis.2017.01.005.

Sellami, A., & Hwang, H. (2019). A robust deep convolutional neural network with batch-weighted loss for heartbeat classification. *Expert Systems With Applications, 122*, 75–84.

Seo, Y., & Shin, K. shik (2019). Hierarchical convolutional neural networks for fashion image classification. *Expert Systems with Applications*. https://doi.org/10.1016/j.eswa.2018.09.022.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556v1.

Tunç, B., Dagl, V., & Gökmen, M. (2012). Class dependent factor analysis and its application to face recognition. *Pattern Recognition*. https://doi.org/10.1016/j.patcog.2012.05.017.

Wang, Y., Han, P., Lu, X., Wu, R., & Huang, J. (2007). The performance comparison of Adaboost and SVM applied to SAR ATR. *CIE international conference of radar proceedings* https://doi.org/10.1109/ICR.2006.343515.

Xiao, H., Rasul, K., & Vollgraf, R. (2017). A novel image dataset for benchmarking machine learning algorithms.

Yu, S., Yu, K., Tresp, V., Kriegel, H.-P., & Wu, M. (2006). Supervised probabilistic principal component analysis. In *Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining - KDD '06* https://doi.org/10.1145/1150402.1150454.

Zaqout, I. (2011). A statistical approach for Latin handwritten digit recognition. *International Journal of Advanced Computer Science and Applications, 2*(10), 37–40.

Zeiler, M. D., Krishnan, D., Taylor, G. W., & Fergus, R. (2010). Deconvolutional networks. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition* https://doi.org/10.1109/CVPR.2010.5539957.

Zhang, H., Nasrabadi, N. M., Zhang, Y., & Huang, T. S. (2012). Multi-view automatic target recognition using joint sparse representation. *IEEE Transactions on Aerospace and Electronic Systems*. https://doi.org/10.1109/TAES.2012.6237604.

Zhu, J., Ahmed, A., & Xing, E. (2012). MedLDA: maximum margin supervised topic models. *Journal of Machine Learning Research*. https://doi.org/10.1145/1553374.1553535.