Neural Networks 98 (2018) 34–41

Contents lists available at ScienceDirect

Neural Networks

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

Matrix completion by deep matrix factorization

Jicong Fan *, Jieyu Cheng

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

ARTICLE INFO

Article history:
Received 28 April 2017
Received in revised form 12 September 2017
Accepted 19 October 2017
Available online 3 November 2017

Keywords:
Matrix completion
Matrix factorization
Deep learning
Image inpainting
Collaborative filtering

ABSTRACT

Conventional methods of matrix completion are linear methods that are not effective in handling data of nonlinear structures. Recently a few researchers attempted to incorporate nonlinear techniques into matrix completion but there still exists considerable limitations. In this paper, a novel method called deep matrix factorization (DMF) is proposed for nonlinear matrix completion. Different from conventional matrix completion methods that are based on linear latent variable models, DMF is on the basis of a nonlinear latent variable model. DMF is formulated as a deep-structure neural network, in which the inputs are the low-dimensional unknown latent variables and the outputs are the partially observed variables. In DMF, the inputs and the parameters of the multilayer neural network are simultaneously optimized to minimize the reconstruction errors for the observed entries. Then the missing entries can be readily recovered by propagating the latent variables to the output layer. DMF is compared with state-of-the-art methods of linear and nonlinear matrix completion in the tasks of toy matrix completion, image inpainting and collaborative filtering. The experimental results verify that DMF is able to provide higher matrix completion accuracy than existing methods do and DMF is applicable to large matrices.

1. Introduction

Matrix completion (Candes & Recht, 2009; Fan & Chow, 2017b; Lu, Gong, Yan, Yuan, & Li, 2012) is a task to recover the missing entries of partially observed matrices of low-rank. Matrix completion has been widely applied to real problems such as collaborative filtering (Bobadilla, Ortega, Hernando, & Gutirrez, 2013; Fan & Chow, 2017a; Su & Khoshgoftaar, 2009; Zhuang et al., 2017), image inpainting (Hu, Zhang, Ye, Li, & He, 2013), and classification (Luo, Liu, Tao, & Xu, 2015). One category of classical matrix completion methods is based on matrix factorization (Koren, Bell, & Volinsky, 2009), in which a low-rank matrix is approximated by the multiplication of a thin matrix and a short matrix (Wen, Yin, & Zhang, 2012; Zhou, Wilkinson, Schreiber, & Pan, 2008). Matrix factorization based methods are non-convex and they are sensitive to the given or estimated rank of the incomplete matrix. In Candes and Recht (2009), Lin, Chen, and Ma (2010), and Shen, Wen, and Zhang (2014), nuclear-norm minimization was utilized for low-rank matrix completion. Nuclear-norm, defined as the sum of the singular values of a matrix, is a convex relaxation for matrix rank. A few extensions of nuclear-norm minimization were also applied to matrix completion (Cao, Chen, Ye, Zhao, & Zhou, 2017; Hu et al., 2013; Lu, Tang, Yan, & Lin, 2014; Nie, Huang, & Ding, 2012). For example, in Hu et al. (2013), truncated nuclear-norm minimization was applied to matrix completion. Truncated nuclear-norm minimization aims at minimizing the sum of the smallest few singular values, where the largest few singular values are preserved because they often contain useful information. In Liu and Li (2016), a method called low-rank factor decomposition was proposed for high-coherence matrix completion where the data were drawn from multiple subspaces. The method is a combination of matrix factorization and nuclear-norm minimization. Compared with factorization based methods, nuclear-norm related methods often provide higher recovery accuracy but have significantly higher computational complexity because of the singular value decomposition (SVD) in each iteration, even when economy or truncated SVD is performed.

The aforementioned matrix completion methods are all linear methods because the low-rank assumption indicates that the data are from linear latent variable models, in which the latent variables are much fewer than the observed variables. Therefore, they are not effective in handling data that are drawn from nonlinear latent variable models. To recover the missing entries for data of nonlinear-structures, in Salakhutdinov, Mnih, and Hinton (2007), restricted Boltzmann machines (RBMs) was proposed for collaborative filtering. Recently, in Sedhain, Menon, Sanner, and Xie (2015), AutoEncoder based collaborative filtering (AECF) was proposed. In AECF, the missing entries were replaced by predefined constants and all the data were used to train AutoEncoders in which the reconstruction errors of the observed entries were minimized. However, the influence of the biases introduced by

* Corresponding author.
E-mail addresses: fanj.c.rick@gmail.com (J. Fan), jieyu.cheng1990@gmail.com (J. Cheng).

https://doi.org/10.1016/j.neunet.2017.10.007
0893-6080/

the pre-defined constants cannot be ignored and the performances are sensitive to the pre-defined constants. More recently, in Si, Chiang, Hsieh, Rao, and Dhillon (2016), a method called goal-directed inductive matrix completion was proposed. The method decomposed the incomplete matrix X as $X = \Phi(A)C\Phi(B)^T$, where $\Phi(A)$ and $\Phi(B)$ are the nonlinear mapped features of the side information (A and B), and C is a low-rank matrix to learn. In Liu et al. (2016), a kernelized matrix factorization was proposed for collaborative filtering. The incomplete matrix X is written as $X = UKV^T$, where U and V are matrices to learn and K is a kernel matrix for a set of randomly chosen dictionary vectors. In Alameda-Pineda, Ricci, Yan, and Sebe (2016), a nonlinear matrix completion method was proposed for classification. The method puts the nonlinear mapped features and the known/unknown labels into a low-rank matrix. As a result, the unknown labels can be computed through rank-minimization. However, these nonlinear extensions for matrix completion have considerable limitations. For example, in Si et al. (2016), the nonlinear operations are only for the side information. If there is no available side information (e.g., in image inpainting and classification) or the side information is incomplete, the nonlinear operations will be inapplicable. The methods in Si et al. (2016) and Liu et al. (2016) are still linear methods because the 3-factor decomposition of X indicates X is given by linear transformation. The method proposed in Alameda-Pineda et al. (2016) is only applicable to classification task because it requires to organize all the missing entries into a sub-matrix. As a result, it is inapplicable to more general matrix completion problems such as image inpainting and collaborative filtering.

In this paper, we propose a novel method called deep matrix factorization (DMF) for matrix completion. DMF is based on a nonlinear latent variable model in which the latent variables are much fewer than the observed variables. Therefore, DMF is able to recover incomplete matrices in which the data have nonlinear structures. For an incomplete matrix, DMF aims at learning a multi-layer neural network to approximate the nonlinear latent variable model. In the deep-structure neural network of DMF, the inputs are the low-dimensional unknown latent variables and the outputs are the partially observed variables. DMF factorizes an incomplete matrix into a thin matrix and a short matrix through the deep-structure neural network. In DMF, the inputs and the parameters of the neural network are simultaneously optimized to minimize the reconstruction errors for the observed entries, and the optimization is solved by BFGS or iRprop$^+$. Then the missing entries can be recovered by propagating the latent variables to the output layer. The proposed method is compared with state-of-the-art methods of matrix completion in the tasks of toy matrix completion, image inpainting and collaborative filtering. The experimental results verify that the proposed method is more accurate and efficient than other methods.

The major contributions of this paper are summarized as the following two aspects. First, we thoroughly analyze the limitations of existing matrix completion methods and propose a novel method DMF for matrix completion. Different from linear matrix completion methods (e.g., Hu et al., 2013; Liu and Li, 2016), DMF is exactly a nonlinear matrix completion method and is able to recover incomplete matrices in which the data have nonlinear structures. Compared with the nonlinear methods proposed in Alameda-Pineda et al. (2016), Liu et al. (2016) and Si et al. (2016), DMF is more versatile and applicable to various tasks of matrix completion. Extensive experimental results show that DMF can provide more accurate recovery than other methods do. Second, we analyze the computational complexity of DMF and show that DMF is more efficient than nuclear-norm and truncated nuclear-norm related matrix completion methods; we also analyze the influence of the number of neural network hidden layers on the performance of DMF. It is worth noting that the neural network of DMF is quite different from conventional neural networks. In conventional neural networks, the input and output variables are known and complete. But in DMF, the input variables are absolutely unknown and the output variables are incomplete.

The remaining contents of this paper are organized as follows. Section 2 details our proposed method DMF. Section 3 compares the proposed method with other state-of-the-art methods in the tasks of toy matrix completion, image inpainting and collaborative filtering. Section 4 draws a conclusion for this paper.

2. Matrix completion by Deep Matrix Factorization (DMF)

2.1. Model of DMF

Given an incomplete matrix $X \in \mathbb{R}^{m\times n}$, matrix completion is to recover the missing entries of X through the observed entries. A widely-used assumption for matrix completion is that the observed entries are sampled uniformly at random (Candes & Recht, 2009). Conventional matrix completion methods assume that X is of low-rank, i.e., $rank(X) = r < min(m, n)$. Therefore, the missing entries of X can be recovered through rank-minimization. Because direct minimizing rank is NP-hard, rank is usually approximated with nuclear-norm (Candes & Recht, 2009), truncated nuclear-norm (Hu et al., 2013), or other techniques (Nie et al., 2012). With the property of low-rank, X can be factorized as

$$X = PZ, \tag{1}$$

where $P \in \mathbb{R}^{m\times r}$ and $Z \in \mathbb{R}^{r\times n}$. Consequently, the missing entries of X can also be recovered through matrix factorization (MF) (Wen et al., 2012).

Assuming that the rows and columns of X are variables and samples respectively, the low-rank assumption of conventional matrix completion methods indicates that X is from a linear latent variable model, i.e.,

$$x = Pz \tag{2}$$

where x (observed variables) is a column of X and z (latent variables) is a column of Z. In practice, real datasets are often from nonlinear latent variable models and hence have nonlinear structures. Specifically, X can be modeled by

$$x = f(z), \tag{3}$$

where $z \in \mathbb{R}^r$ are the latent variables and $f(\cdot)$ is a nonlinear map. The model in (3) can also be formulated in matrix form, i.e.,

$$X = f(Z), \tag{4}$$

where $f(\cdot)$ is performed on each column of $Z \in \mathbb{R}^{r\times n}$ individually. Clearly, conventional matrix completion methods are not effective in recovering the missing entries of X given by the nonlinear latent variable model (4).

For an incomplete matrix X given by model (4), it is possible to recover the missing entries because the number of latent variables is much smaller than that of observed variables, i.e., $r < m$ (which means X is redundant). If $f(\cdot)$ and Z can be computed via the observed entries, the missing entries can be readily obtained. Therefore, in this paper, we propose to solve the following problem

$$\begin{aligned} &\min_{Z,f} \pi(f) + \tfrac{\beta}{2n}\|Z\|_F^2, \\ &s.t.\ [f(Z)]_{i,j} = X_{i,j},\ (i,j) \in \Omega, \end{aligned} \tag{5}$$

where $\pi(f)$ denotes a penalization or constraint on $f(\cdot)$, β is a regularization parameter for the penalization on Z, and Ω denotes the positions of observed entries of X. Since there exist infinite solutions of $f(\cdot)$ and Z that can produce the observed entries of X, the regularizations on $f(\cdot)$ and Z are necessary and can result

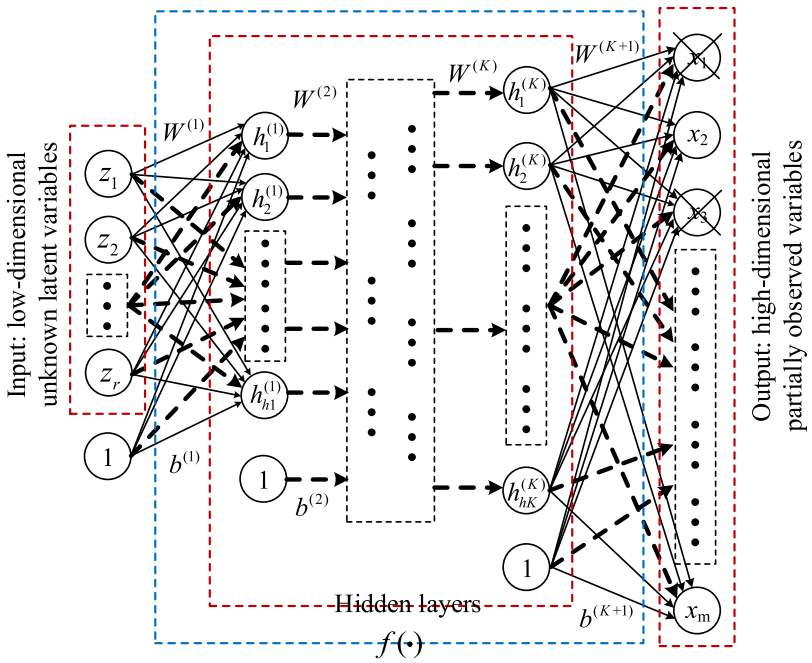

Fig. 1. Neural network structure of DMF based matrix completion.

in simpler models. On the other hand, the regularizations are able to avoid overfitting and improve the generalization ability of the trained model for predicting the unknown entries of X. In practice, the observed entries of X are often corrupted by small noises and there is no need to make $[f(Z)]_{i,j}$ equal to $X_{i,j}$ exactly. One just needs to minimize the difference between $[f(Z)]_{i,j}$ and $X_{i,j}$ as much as possible. Therefore, problem (5) can be modified as

$$\min_{Z,f} \tfrac{1}{2n}\|\Psi \odot (X - f(Z))\|_F^2 + \tfrac{\beta}{2n}\|Z\|_F^2 + \lambda\pi(f). \tag{6}$$

In (6), λ is a regularization parameter, $\odot$ denotes the Hadamard product, and Ψ is a binary matrix in which the entry Ψ_{ij} is 1 if $(i,j) \in \Omega$ and 0 otherwise. The missing entries of X can be temporally set as arbitrary values (e.g., zeros) because the Hadamard product with Ψ makes the missing entries have no contribution to the objective function. By solving (6), the missing entries can be recovered as

$$X_{i,j} = [f(Z)]_{i,j},\ (i,j) \in \bar{\Omega}, \tag{7}$$

where $\bar{\Omega}$ denotes the positions of missing entries of X.

It is difficult to solve (6) directly because $f(\cdot)$ is nonlinear and no more information about $f(\cdot)$ is available. Therefore, we first propose to use a single-layer neural network to approximate $f(\cdot)$. As a result, problem (6) can be approximated as

$$\begin{aligned}\min_{Z,W,b} \tfrac{1}{2n}\sum_{i=1}^{n}\|\Psi_i \odot (x_i - g(Wz_i + b))\|^2 \\ + \tfrac{\beta}{2n}\|Z\|_F^2 + \tfrac{\lambda}{2}\|W\|_F^2,\end{aligned} \tag{8}$$

where $W \in \mathbb{R}^{m\times r}$ denotes the weight matrix, $b \in \mathbb{R}^m$ denotes the bias vector, Ψ_i is the ith column of Ψ, λ is the weight-decay parameter, and $g(\cdot)$ denotes the activation function such as sigmoid function or hyperbolic tangent function. By solving (8), $f(\cdot)$ and Z can be obtained and we get

$$f(Z) = g(WZ + B), \tag{9}$$

where B consists of n columns of b. By substituting (9) into (7), the missing entries of X can be obtained. It can be seen that if the activation function $g(\cdot)$ is linear, problem (8) will degrade to conventional factorization based matrix completion.

It is well known that multi-layer neural networks are often more effective than single-layer neural networks in approximating nonlinear functions (Hinton & Salakhutdinov, 2006; Schmidhuber, 2015). In other words, deep structures often outperform shallow structures (Bengio, 2009; Bengio, Courville, & Vincent, 2013). Therefore, problem (6) can be further approximated as

$$\begin{aligned}\min_{Z,\Theta} \tfrac{1}{2n}\sum_{i=1}^{n} \| \Psi_i \odot (x_i - g^{(K+1)}(g^{(K)}(\cdots g^{(1)}(z_i, \Theta^{(1)}) \\ \cdots, \Theta^{(K)}), \Theta^{(K+1)}))\|^2 + \tfrac{\beta}{2n}\|Z\|_F^2 + \tfrac{\lambda}{2}\sum_{j=1}^{K+1}\|W^{(j)}\|_F^2,\end{aligned} \tag{10}$$

where $\Theta^{(j)} = \{W^{(j)}, b^{(j)}\}$, $g^{(j)}(t, \Theta^{(j)}) = g^{(j)}(W^{(j)}t + b^{(j)})$, $j = 1, 2\cdots, K+1$, and K is the number of hidden layers. The model in (10) is called deep matrix factorization (DMF) based matrix completion. In DMF, high-dimensional X is factorized into low-dimensional Z and $W^{(1)}$ through multi-layer nonlinear mappings. The neural network structure of DMF is shown in Fig. 1, in which the number of nodes in each layer is in an order of $r < h_1 < h_2 < \cdots < h_K < m$. Clearly, the neural network of DMF is quite different from conventional neural networks including regression machines and AutoEncoders. In conventional neural networks, both input and output are known and complete. In DMF, the input is unknown and the output is incomplete. As can be seen, in Fig. 1, $f(\cdot)$ is approximated with a multi-layer neural network that has a deep structure, i.e.,

$$\begin{aligned}f(z) =& g^{(K+1)}(W^{(K+1)}g^{(K)}(W^{(K+1)}(\cdots \\ & g^{(1)}(W^{(1)}z + b^{(1)})\cdots) + b^{(K)}) + b^{(K+1)}).\end{aligned} \tag{11}$$

By solving (10), $f(\cdot)$ and Z can be obtained and then the missing entries of X can be recovered with (7).

Assuming that the sampling rate of the observed entries of X is ρ, then the number of observed entries is $|\Omega| = \rho mn$ and the number of observed entries of $x_i (i = 1, \ldots, n)$ is about ρm. It is worth noting that for the model (4), if $\rho m < r$, Z cannot be determined, which indicates that the missing entries cannot be determined. Therefore, a necessary condition for successfully recovering the missing entries of X by (8) or (10) is $\rho \geq r/m$. In DMF, the first

step is to design the structure of the neural network, including the input size (r), the number of hidden layers (K), the number of nodes in each hidden layer (h_k), and the activation functions. The optimal r is the dimensionality of the latent variable z. But in practice, r is unknown and also difficult to estimate. However, as mentioned previously, we have $r < (1-\delta)m$, where $\delta = 1-\rho$ is the missing rate. For the activation functions, sigmoid function and hyperbolic tangent function are widely-used choices (LeCun, Bottou, Orr, & Müller, 1998). In DMF, type of the activation functions for the output layer should be determined according to the range of the data in X. Alternatively, the data should be transformed to match the output ranges of the activation functions. For example, in image inpainting task, the activation functions for the output layers should be sigmoid functions because the pixel values are in (or can be easily transformed into) the interval [0, 1].

2.2. Optimization of DMF

The optimization problem of DMF in (10) is nonconvex and hence we propose to find the local minima by nonlinear optimization techniques such as BFGS (Liu & Nocedal, 1989) and improved resilient backpropagation (iRprop$^+$) algorithm (Igel & Hüsken, 2000). In addition, because the input variables Z are unknown, the optimization of DMF should be solved by batch-wise approach but not mini-batch and stochastic approaches. In Igel and Hüsken (2003), it was shown that iRprop$^+$ often outperformed other methods in optimizing neural networks and iRprop$^+$ is more efficient than BFGS for large-scale problems. In this paper, we use BFGS to solve DMF when the size of X is relatively small (e.g., $m, n < 1000$) and use iRprop$^+$ otherwise. In general, BFGS and iRprop$^+$ require only the gradient of the objective function. The objective function of DMF in (10) can be written as

$$J(Z, \Theta) = L(Z, \Theta) + \Gamma(Z, \Theta), \tag{12}$$

where $L(Z, \Theta)$ denotes the reconstruction error for the observed entries of X and $\Gamma(Z, \Theta)$ denotes the penalizations of Z and the network parameters. Both $\partial L(Z, \Theta)/\partial Z$ and $\partial L(Z, \Theta)/\partial \Theta$ can be computed by back-propagation algorithm. Hence the gradient of $J(Z, \Theta)$ can be obtained.

It is worth noting that the initializations of Θ and Z are also important to DMF. Θ can be initialized according to LeCun et al. (1998) and the entries of Z can be initialized as zeros. The well-known training strategy in deep learning, greedy layer-wise training (Bengio et al., 2013), can also be applied to DMF. Specifically, we can perform the following steps: first solve (8) with a relatively large r but much smaller than m; then reduce the dimensionality of Z as low as possible through training stacked AutoEncoders; perform fine-tuning on the network of (10) that is formed by the trained decoders and trained network of (8). However, the computational cost will increase significantly. Empirically, we find that direct solving (10) is effective enough to recover incomplete matrices.

3. Experiments

In this section, the proposed method DMF will be compared with five state-of-the-art methods of matrix completion in the tasks of toy matrix completion, image inpainting and collaborative filtering. The methods include matrix factorization (MF) based method solved by LMaFit (Wen et al., 2012), nuclear-norm minimization (NNM) based method solved by IALM (Lin et al., 2010), truncated nuclear-norm minimization (TNNM) based method solved by ADMM (Hu et al., 2013), AECF method (Sedhain et al., 2015) solved by nonlinear conjugate gradient method, and the low-rank factor decomposition (LRFD) method (Liu & Li, 2016). The parameters of all compared methods are carefully adjusted to present their best performances. Particularly, in image inpainting, the missing entries for the input of AECF are predefined as the result of MF. Otherwise, the performance of AECF could be unsatisfactory significantly. In DMF, the parameters β and λ are chosen from $\{0.001, 0.005, 0.01, 0.1\}$. The performances for image inpainting and collaborative filtering are evaluated by the normalized mean absolute error (NMAE) (Wen et al., 2012) defined as

$$NMAE = \frac{1}{(X_{max} - X_{min})|\bar{\Omega}|} \sum_{(i,j)\in\bar{\Omega}} |\hat{X}_{ij} - X_{ij}|, \tag{13}$$

where $\hat{X}$ is the recovered matrix, X is the true matrix, $\bar{\Omega}$ denotes the positions of the missing entries, X_{max}, X_{min} are the maximum and minimum of X respectively.

Table 1
Relative errors (%) for the toy example of nonlinear data with different missing rates (δ).

δ	MF	NNM	TNNM	AECF	LRFD	DMF
20%	32.12	33.57	20.88	43.34	31.72	**4.02**
40%	33.35	35.41	24.56	44.89	34.38	**7.14**
60%	35.94	61.08	36.85	47.68	42.58	**10.82**
80%	100	100	100	66.90	100	**38.42**

3.1. Matrix completion on a toy example of nonlinear data

Since the main topic of this paper is matrix completion on nonlinear data, the following nonlinear toy example is considered: $x_1 = z, x_2 = z^2, x_3 = z^3, x_4 = e^z$, and $x_5 = sin(z)$. Fifty samples of z are drawn from a uniform distribution on the open interval $(0, 1)$. As a result, a matrix X with size 5×50 can be obtained. By randomly removing 1, 2, 3, and 4 entries of every column of X, four incomplete matrices with missing rates 20%, 40%, 60%, and 80% can be obtained. The parameters of all matrix completion methods are carefully determined to provide their best performances. Particularly, in DMF, three hidden layers are used; the size of input layer and hidden layers are $\{1, 5, 5, 5\}$ respectively; the activation functions of the hidden layers are hyperbolic tangent function; the activation function of the output layer is linear function. The matrix completion results are evaluated by the relative error defined as $RE = \sqrt{\sum_{(i,j)\in\bar{\Omega}}(\hat{X}_{ij} - X_{ij})^2}/\sqrt{\sum_{(i,j)\in\bar{\Omega}} X_{ij}^2}$. The average results of 500 repeated trials are reported in Table 1. We can see that the proposed method always outperforms other methods significantly. The results verified that DMF is very effective in recovering nonlinear data, which is a considerable challenge to existing methods.

3.2. Single-image inpainting

We use five RGB images (Hu et al., 2013), shown in Fig. 2, to test the performances of matrix completion based single-image inpainting. Each image is resized to $300 \times 300 \times 3$ and unfolded to a pixel matrix of 300×900. Three masks are considered in this study. The first one is random pixel mask for which 50% pixels are randomly removed. The other two masks are text masks with small and large font sizes. The number of hidden layers of DMF is set as 1. The numbers of nodes in the input layer, hidden layers, and output layer are set as [200, 400, 900]. The optimization of DMF is solved by BFGS because the data size is relatively small. Figs. 3, 4, and 5 are a few examples of the original images, masked images, and recovered images given by the six methods. Through careful observation, we can find that, among all methods, DMF always provides the best recoveries for the original images. The NMAEs for all inpainting results are reported in Table 2. In most cases, the recovery errors of DMF are lower than that of other methods.

Fig. 2. Five RGB images for inpainting.

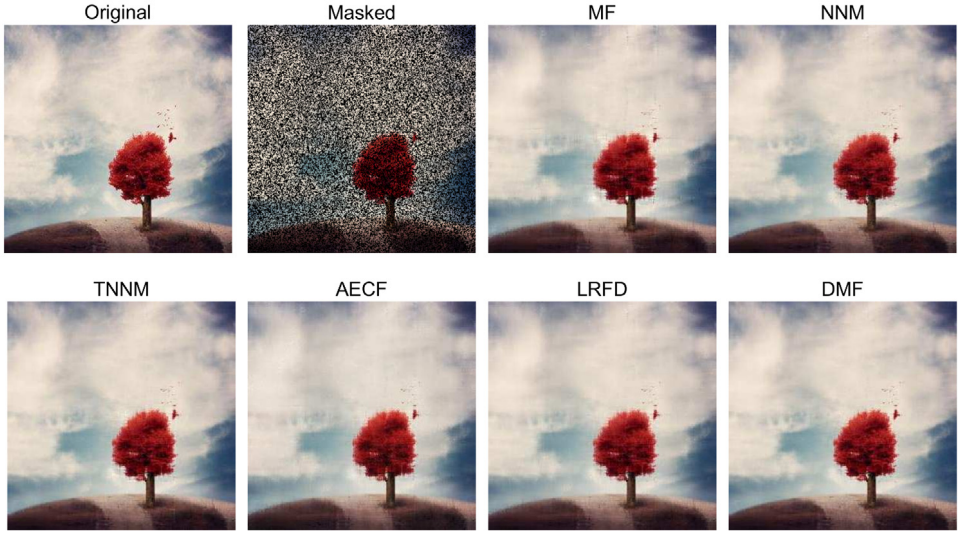

Fig. 3. Inpainting results for image 2 with random mask.

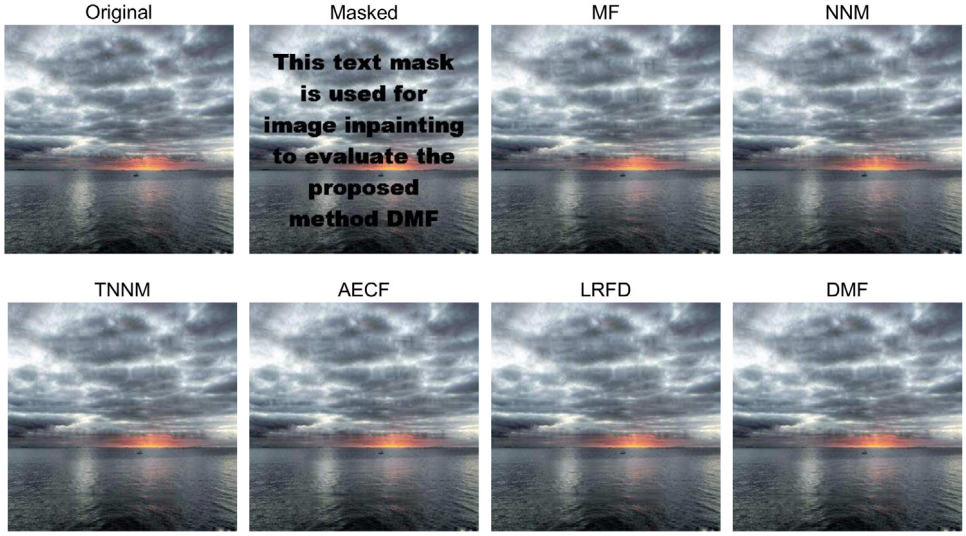

Fig. 4. Inpainting results for image 3 with small text mask.

3.3. Group-image inpainting

The MNIST dataset of handwritten digits (28×28) (LeCun, Bottou, Bengio, & Haffner, 1998) is utilized to show the performances of all related matrix completion methods in group-image inpainting task, in which each image forms a vector (column) of the matrix. We randomly choose 100 images for each digit and then form a 784×1000 matrix. Two types of masks are considered: random pixel (70%) mask and random square block mask (14×14). The number of hidden layers of DMF is also set as 2. The numbers of nodes in the input layer, hidden layers, and output layer are set as [25, 100, 200, 784]. The optimization of DMF is solved by iRprop$^+$ because the data size is relatively large. Fig. 6 shows the optimization curves of BFGS and iRprop$^+$ for DMF on MNIST digits with random-block mask. As can been seen, BFGS and iRprop$^+$ converged in 1500 iterations and the value of objective function given by iRprop$^+$ is nearly the same as that given by BFGS after 1000 iterations. Additionally, the computational cost of BFGS is at least two times of that of iRprop$^+$. Therefore, iRprop$^+$ is preferable to BFGS in this case.

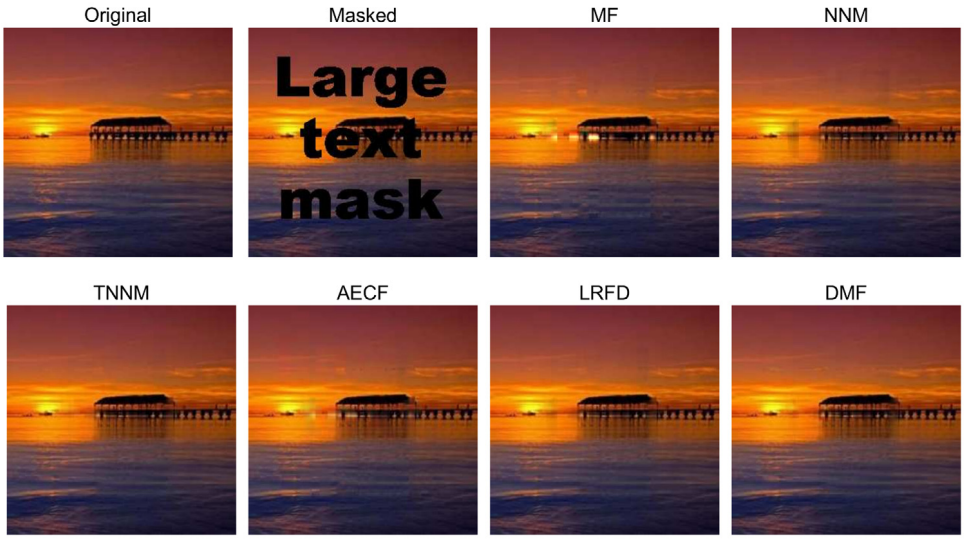

Fig. 5. Inpainting results for image 4 with large text mask.

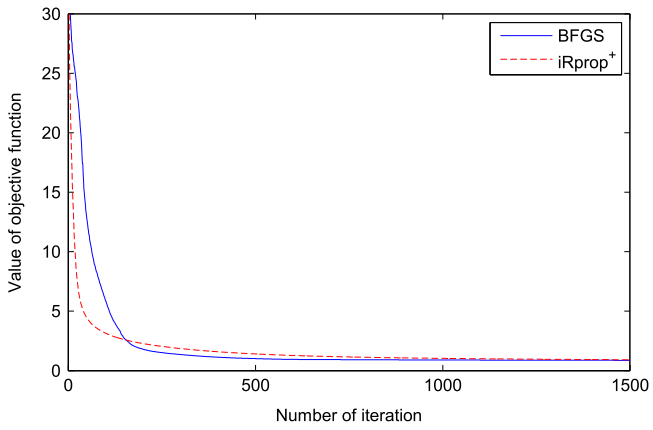

Fig. 6. Optimization curves of BFGS and iRprop$^+$ for DMF on MNIST digits with random-block mask.

Fig. 7 shows examples of recovery results for random-pixel masked digits. For every digit, the recovery performance of DMF is significantly better than that of other methods. Fig. 8 shows examples of recovery results for random-block masked digits. We can see that, the digits recovered by DMF are nearly the same as the original digits. On the contrary, the recovery results given by other methods are unsatisfactory. The average NMAEs of 20 repeated trials are reported in Table 3. Clearly, the recovery errors of DMF are the lowest in the two cases. Table 4 shows the NMAEs of DMF with different number of hidden layers. DMF with three hidden layers performs the best but the superiority to DMF with two hidden layers is not significant. By comparing Table 4 with Table 3, we can find that the recovery errors of DMF with 0 or 1 hidden layers are still significantly lower than that of other methods.

Table 2
NMAE (%) of single-image inpainting.

Image	Mask	MF	NNM	TNNM	AECF	LRFD	DMF
1	Random	3.02	2.76	**2.64**	2.95	2.71	2.65
	Text-small	3.62	3.40	**3.16**	3.35	3.19	3.18
	Text-large	4.76	4.22	3.92	3.99	3.71	**3.68**
2	Random	2.37	1.60	1.48	2.53	2.02	**1.46**
	Text-small	4.19	3.42	3.03	3.61	3.16	**2.76**
	Text-large	7.80	6.69	5.74	6.06	5.74	**5.17**
3	Random	2.36	1.84	1.66	2.18	1.78	**1.62**
	Text-small	3.84	3.05	2.92	3.65	3.11	**2.84**
	Text-large	7.08	5.45	5.01	5.80	4.83	**4.73**
4	Random	1.10	0.78	**0.71**	1.06	0.91	0.79
	Text-small	2.23	1.67	1.52	2.22	1.70	**1.51**
	Text-large	3.90	2.40	2.14	3.09	2.23	**2.06**
5	Random	6.03	4.69	4.53	5.38	4.66	**4.51**
	Text-small	7.95	6.56	6.28	7.04	6.29	**6.07**
	Text-large	8.70	8.28	7.27	8.01	7.28	**7.14**

Table 3
NMAE (%) for MNIST data.

Mask	MF	NNM	TNNM	AECF	LRFD	DMF
Random-pixel	10.94	9.62	9.38	9.06	9.58	**6.73**
Random-block	18.26	15.62	15.29	15.13	16.19	**12.94**

Table 4
NMAE(%) of DMF with different number of hidden layers for masked digits.

Number of hidden layers	0	1	2	3
Random-pixel mask	7.91	6.88	6.73	**6.68**
Random-block mask	14.53	13.17	12.94	**12.71**

Table 5
NMAE(%) for Jester-joke dataset 1 and MovieLens 100k.

	δ	MF	NNM	TNNM	AECF	LRFD	DMF
Jester-joke	30%	15.94	16.21	15.78	16.13	15.72	**15.68**
	50%	16.32	16.65	16.13	16.41	16.11	**16.03**
	70%	17.24	17.36	16.69	16.97	16.92	**16.51**
MovieLens	30%	18.97	20.61	20.76	19.02	18.64	**18.25**
	50%	20.17	20.98	20.81	19.29	18.96	**18.73**
	70%	20.68	21.53	21.49	19.75	19.54	**19.32**

Table 6
Computational time (second) for MNIST dataset, Jester-joke dataset 1 and MovieLens 100k.

Dataset	Matrix size	MF	NNM	TNNM	AECF	LRFD	DMF
MNIST	784×1000	17	631	594	147	436	165
Jester-joke	100×10^4	5	218	210	64	728	83
MovieLens	1682×943	7	717	713	119	242	128

Fig. 7. Recovery results for MNIST digits with random-pixel mask.

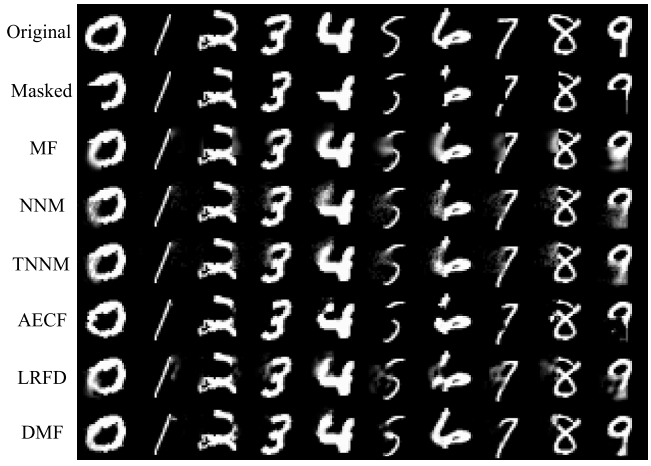

Fig. 8. Recovery results for MNIST digits with random-block mask.

3.4. Collaborative filtering

In this study, two datasets are used for collaborative filtering. The first one is the Jester-joke dataset-1, in which the data are from 24,983 users who have rated 36 or more of 100 jokes. The rating values are within $[-10,10]$. We use the ratings of 10000 users to form a 100×10000 matrix. Another dataset is the MovieLens 100k, which consists of 100000 ratings (1–5) from 943 users on 1682 movies and forms a 1682×943 matrix. Each user has rated at least 20 movies. It is worth noting that the two matrices are originally incomplete. In the Jester-joke data, about 70% of the entries are known. In the MovieLens data, only 6.3% of the entries are known. Therefore, the missing rates in this study are to the known entries, not all entries.

We randomly remove 30%, 50%, and 70% of the known entries of the two datasets to test the performances of the proposed method. For both datasets, the neural networks of DMF have 1 hidden layer; the numbers of nodes in input layer, hidden layer, and output layer are set as [5 50 100] for Jester-joke data and [10 100 1682] for MovieLens data. The optimization of DMF is solved by iRprop$^+$. It is worth noting that, in this collaborative filtering study, DMF with 1 hidden layer is able to give better results than DMF with 0, 2, or more hidden layers do. The average results of 20 repeated trials are reported in Table 5. As can be seen, the proposed DMF outperforms other methods in all cases.

3.5. Computational complexity analysis

The major computational cost in each iteration of DMF is from the computation of the gradient especially when iRprop$^+$ is utilized for optimization. The computation of the gradient based on back-propagation is mainly related to matrix multiplication. As known, the flop count of an $a \times b$ matrix multiplying a $b \times c$ matrix is about $2abc$. Therefore, the flop count of computing gradient in DMF is about $4n(rh_1 + h_1h_2 + \cdots + h_Km)$. In NNM/TNNM/LRFD, the major computational cost in each iteration is from SVD that has a flop count of $14m^2n + 8m^3$ on an $m \times n$ $(m < n)$ matrix. For simplicity, we assume $m < n$. Then the flop count in each of NNM/TNNM/LRFD is much larger than $14m^2n$. In DMF, because $r < h_1 < \cdots < h_K < m$, the flop count in each iteration is much smaller than $4Kmh_Kn$. Apparently, the computational cost of DMF is significantly smaller than that of NNM/TNNM/LRFD. The computational time of the related methods on MNIST dataset, Jester-joke dataset 1 and MovieLens 100k dataset are shown in Table 6. DMF is significantly faster than NNM, TNNM, and LRFD. Although DMF is slower than MF, it can provide significantly higher recovery accuracy than any other methods including MF do, in all experimental cases of this paper. Moreover, although DMF needs to optimize latent variables, its computational time is comparable with that of AECF. The reason is that AECF depends on greedy layer-wise learning plus fine-tuning and the network consists of both encoders and decoders. On the contrary, DMF consists of only decoders and its optimization is implemented all at once.

4. Conclusion

In this paper, we proposed a novel method called deep matrix factorization (DMF) for matrix completion. DMF is able to recover the missing entries of nonlinear data drawn from nonlinear latent variable models. The experimental results in the tasks of toy matrix completion, image inpainting and collaborative filtering verified that DMF can outperform state-of-the-art methods of linear and nonlinear matrix completion. In addition, the computational complexity of DMF is significantly lower than that of nuclear-norm and truncated nuclear-norm related methods.

References

Alameda-Pineda, X., Ricci, E., Yan, Y., & Sebe, N. (2016). Recognizing emotions from abstract paintings using non-linear matrix completion. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 5240–5248).

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends® in Machine Learning*, *2*(1), 1–127.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1798–1828.

Bobadilla, J., Ortega, F., Hernando, A., & Gutirrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, *46*, 109–132.
Candes, E. J., & Recht, B. (2009). Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, *9*(6), 717–772.
Cao, F., Chen, J., Ye, H., Zhao, J., & Zhou, Z. (2017). Recovering low-rank and sparse matrix based on the truncated nuclear norm. *Neural Networks*, *85*, 10–20.
Fan, J., & Chow, T. W. (2017a). Matrix completion by least-square, low-rank, and sparse self-representations. *Pattern Recognition*, *71*, 290–305.
Fan, J., & Chow, T. W. (2017b). Sparse subspace clustering for data with missing entries and high-rank matrix completion. *Neural Networks*, *93*, 36–44.
Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504–507.
Hu, Y., Zhang, D., Ye, J., Li, X., & He, X. (2013). Fast and accurate matrix completion via truncated nuclear norm regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(9), 2117–2130.
Igel, C., & Hüsken, M. (2000). Improving the Rprop learning algorithm. In *Proceedings of the second international ICSC symposium on neural computation, Vol. 2000*. (NC 2000), (pp. 115–121). Citeseer.
Igel, C., & Hüsken, M. (2003). Empirical evaluation of the improved Rprop learning algorithms. *Neurocomputing*, *50*, 105–123.
Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8).
LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.
LeCun, Y., Bottou, L., Orr, G. B., & Müller, K. R. (1998). Efficient backprop. In G. B. Orr, & K.-R. Müller (Eds.), *Neural networks: Tricks of the trade* (pp. 9–50). Berlin, Heidelberg: Springer Berlin Heidelberg.
Lin, Z., Chen, M., & Ma, Y. (2010). The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices, arXiv:1009.5055v3 [Math.OC].
Liu, X., Aggarwal, C., Li, Y.-F., Kong, X., Sun, X., & Sathe, S. (2016). Kernelized matrix factorization for collaborative filtering. In *SIAM conference on data mining*, (pp. 399–416).
Liu, G., & Li, P. (2016). Low-Rank matrix completion in the presence of high coherence. *IEEE Transactions on Signal Processing*, *64*(21), 5623–5633.
Liu, D. C., & Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, *45*(1), 503–528.
Lu, X., Gong, T., Yan, P., Yuan, Y., & Li, X. (2012). Robust alternative minimization for matrix completion. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, *42*(3), 939–949.
Lu, C., Tang, J., Yan, S., & Lin, Z. (2014). Generalized nonconvex nonsmooth low-rank minimization. In *2014 IEEE conference on computer vision and pattern recognition*, June, (pp. 4130–4137), ISSN: 1063-6919.
Luo, Y., Liu, T., Tao, D., & Xu, C. (2015). Multiview matrix completion for multilabel image classification. *IEEE Transactions on Image Processing*, *24*(8), 2355–2368.
Nie, F., Huang, H., & Ding, C. (2012). Low-rank matrix recovery via efficient schatten P-norm minimization. In *Proceedings of the twenty-sixth AAAI conference on artificial intelligence*. AAAI'12, (pp. 655–661). AAAI Press.
Salakhutdinov, R., Mnih, A., & Hinton, G. (2007). Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th International conference on machine learning* (pp. 791–798). ACM.
Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117.
Sedhain, S., Menon, A. K., Sanner, S., & Xie, L. (2015). Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th International conference on world wide web* (pp. 111–112). ACM.
Shen, Y., Wen, Z., & Zhang, Y. (2014). Augmented Lagrangian alternating direction method for matrix separation based on low-rank factorization. *Optimization Methods & Software*, *29*(2), 239–263.
Si, S., Chiang, K.-Y., Hsieh, C.-J., Rao, N., & Dhillon, I. S. (2016). Goal-Directed inductive matrix completion. In *Proceedings of the 22Nd ACM SIGKDD International conference on knowledge discovery and data mining*. KDD '16, (pp. 1165–1174). New York, NY, USA: ACM.
Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, *2009*.
Wen, Z., Yin, W., & Zhang, Y. (2012). Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation*, *4*(4), 333–361.
Zhou, Y., Wilkinson, D., Schreiber, R., & Pan, R. (2008). Large-scale parallel collaborative filtering for the netflix prize. In *International conference on algorithmic applications in management* (pp. 337–348). Springer.
Zhuang, F., Zhang, Z., Qian, M., Shi, C., Xie, X., & He, Q. (2017). Representation learning via Dual-Autoencoder for recommendation. *Neural Networks*, *90*, 83–89.