Paper:

Separable Algorithms for Matrix Factorization with Presence of Missing Data

Shi-Xin Wang, Min Gan, Guang-Yong Chen

College of Mathmatics and Computer Science, Fuzhou University, Fuzhou, Fujian, China
E-mail: sx.wang96@outlook.com;
aganmin@aliyun.com;
gychen_fzu@163.com

Abstract: Low-rank matrix factorization (LRMF) frequently appears in various tasks in computer vision, e.g., bundle adjustment. Singular value decomposition (SVD) is a well-known approach to solve LRMF. However, it fails when the matrix is relatively large or the elements of target are lost. In this paper, we formulate the LRMF as a separable nonlinear least squares problem. An iterative algorithm, a combination of variable projection (VP) algorithm and BFGS method (named VP-BFGS), is proposed to solve this problem. The algorithm first utilizes the VP strategy to eliminate part of the parameters (i.e., a matrix), and then the BFGS method is used to estimate the other matrix. In numerical experiments, compared with the joint method, Gauss-Newton method and LM method, the VP-BFGS method achieves competitive performance, especially when the ratio of deficiency to existence is high.

Keywords: low-rank matrix factorization; quasi-Newton method; variable projection; separable nonlinear least-squares

1. Introduction

Many tasks in computer vision can attribute to decomposing a large matrix $M \in \mathbb{R}^{m\times n}$ into a product of two low-rank matrices, e.g., bundle adjustment in visual SLAM [3][4][6], 3D reconstruction from images [6], blind source separation and principal component analysis. Mathematically, the decomposition task can be represented as

$$M \to UV^T \tag{1}$$

where $U \in \mathbb{R}^{m\times r}$, $V \in \mathbb{R}^{n\times r}$, r is the rank that is usually much smaller than m,n, i.e., $r \ll min(m,n)$. The well-known singular value decomposition (SVD) is an effective method for solving these problems [1]. Unfortunately, the SVD method fails when the data set is severely incomplete. In structure from motion, observation matrix is collected by moving sensors, which signifies that the feature points can be barely integrated. To solve the case of missing data and stronge noise in low-rank matrix factorization, we formulate (1) as a separable nonlinear least squares(SNLS) problem

$$\min_{U,V} F(U,V) = \min_{U,V} ||W \odot (UV^T - M)||_F^2 \tag{2}$$

where $W \in \mathbb{R}^{m\times n}$ is a mask matrix that marks the existence of value in M with ones and zeros, $\odot$ is the Hadamard or element-wise product, and the $||\cdot||_F$ is the Frobenius norm.

In the past decades, researchers have proposed various algorithms for solving problem (2), including joint method, alternating least squares algorithm (ALS), embedded point iteration (EPI), and variable projection (VP). Joint optimization regards LRMF as an ordinary nonlinear optimization, as it ignores the separation structure of the model, optimizing all parameters at once. The ALS method considers the separable structure presented in (2) and optimizes the parameters U and V in an alternating way. Whereas the LRMF is of obvious separation structure [9], of which ALS takes the advantage to some extent to optimize the two variables alternately. However, in [2], Buchanan and Fitzgibbon pointed out that the ALS algorithm converges very slowly for ill-condition datasets (e.g., incomplete or strong noise ones). The EPI strategy [4] is often used to accelerate the classical bundle adjustment, which comprehensively utilizes the separable structure.

The VP algorithm was first proposed by Golub and Pereyra [10], which makes full use of the separable structure. In this paper, we apply the VP strategy to the LRMF task and propose an VP-BFGS method. The proposed method takes advantage of the separable structure of (2), and utilizes the VP strategy to eliminate the parameters V, resulting in a reduced function that only contains U. Then, the BFGS method that just requires the gradient information of the reduced function is used to update the remaining parameter U.

The rest of this paper is organized as follows.

• In Section 3, some classical methods for solving separable nonlinear least-squares are revisited explicitly.

• In Section 4, the VP-BFGS is first proposed to solve LRMF. Unlike VP in the previous literature, VP-BFGS optimizes the reduced objective function by BFGS method instead of LM algorithm. Therefore, the VP-BFGS benefits from both the simple derivation of BFGS and the high efficiency of VP.

• In Section 5, the objective function of the regularized VP is represented.

• In Section 6, experiments on several datasets are carried out, which indicates the faster convergence and better reconstruction of VP-BFGS than joint and ALS optimization strategies, particularly in datasets critically incomplete.

2. Notations

Some notations and operation rules used in the rest part of this paper are introduced as follows. Given an arbitrary real matrix X, define

$$X^{+} := (X^{T}X)^{-1}X^{T} \tag{3}$$

$$X^{-\lambda} := (X^{T}X+\lambda I)^{-1}X^{T} \tag{4}$$

$vec(X) : \mathbb{R}^{m\times n} \to \mathbb{R}^{mn}$ which vectorize X into $\mathbf{x}$ columnwisely; $\otimes$ is the Kronecker product.

3. Methods for solving SNLS problems

Noting that $||X||_F^2 = trace(X^TX) = ||vec(X)||_2^2$, the optimization problem (2) can be rewritten as follows

$$\begin{aligned} F(U,V) &= ||f(U,V)||_2^2 \\ &= ||\psi vec(W\odot(UV^T-M))||_2^2 \end{aligned} \tag{5}$$

where ψ is a projection matrix of $l\times mn$, where l is the amount of visible elements in M. Define $\boldsymbol{u} := vec(U) \in \mathbb{R}^{mr}$, $\boldsymbol{v} := vec(V) \in \mathbb{R}^{nr}$, $\boldsymbol{m} := vec(M) \in \mathbb{R}^{mn}$; $\hat{W} := \psi diag(vec(W))$. Applying the linear algebra to (5) yields

$$\begin{aligned} f(U,V) &= \psi diag(vec(W))vec(UV^T-M) \\ &= \hat{W}vec(UV^T) - \hat{W}\boldsymbol{m} \\ &= \hat{U}\boldsymbol{v} - \hat{\boldsymbol{m}} = \hat{V}\boldsymbol{u} - \hat{\boldsymbol{m}} \end{aligned} \tag{6}$$

where $\hat{\boldsymbol{m}} := \hat{W}\boldsymbol{m}$, $\hat{U} := \hat{W}(I_n\otimes U)$, $\hat{V} := \hat{W}(V\otimes I_m)$ which can specially expressed as

$$\hat{U} = \begin{bmatrix} \hat{U}_1 & & & \\ & \hat{U}_2 & & \\ & & \ddots & \\ & & & \hat{U}_m \end{bmatrix}, \hat{V} = \begin{bmatrix} \hat{V}_1 \\ \hat{V}_2 \\ \vdots \\ \hat{V}_m \end{bmatrix} \tag{7}$$

The definition of $\hat{U}_i$,$\hat{V}_i$ in (7): let $S_i = \{j|w_{ij}\}$ which contains $\alpha_{i1},...,\alpha_{i|S_i|}(1 \le \alpha_{i1} < ... < \alpha_{i|S_i|})$. The row k of $\hat{U}_i \in \mathbb{R}^{S_i\times r}$ is $v^T_{\alpha_{ik}}$. With regard to $\hat{V}_i$, u_i^T lies in $[(\alpha_{ik}-1)r+1 : \alpha_{ik}r]$ of column k. In the following part, several methods for separable nonlinear least-squares problems are summarized briefly.

3.1. Joint optimization

Joint optimization stacks $\boldsymbol{u}$ and $\boldsymbol{v}$ to form a vector $\mathbf{x} := [\boldsymbol{u};\boldsymbol{v}] \in \mathbb{R}^{mr+nr}$, and updates all parameters at once using second-order optimization(e.g. LM algorithm and BFGS algorithm). Define $J_u := \frac{\partial F}{\partial \boldsymbol{u}}$, $J_v := \frac{\partial F}{\partial \boldsymbol{v}}$. In general, joint optimization obtains the increment of $\Delta\boldsymbol{u}$ by solving

$$\begin{bmatrix} J_u & J_v \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = -f \tag{8}$$

Using the least-squares method, the solution of (8) can be obtained from the following normal equation:

$$\begin{bmatrix} J_u^TJ_u & J_u^TJ_v \\ J_v^TJ_u & J_v^TJ_v \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{u} \\ \Delta\boldsymbol{v} \end{bmatrix} = -\begin{bmatrix} J_u^Tf \\ J_v^Tf \end{bmatrix} \tag{9}$$

Schur complement [7] is an efficient method to derive the solution

$$\begin{bmatrix} \Delta\boldsymbol{u} \\ \Delta\boldsymbol{v} \end{bmatrix} = \begin{bmatrix} -J_u^{+}(f - J_v(J_v^TP_{J_u}^{\perp}J_v)^{-1}J_v^TP_{J_u}^{\perp}f) \\ -(J_v^TP_{J_u}^{\perp}J_v)^{-1}J_v^TP_{J_u}^{\perp}f \end{bmatrix} \tag{10}$$

where $P_{J_u}^{\perp} := I - J_uJ_u^{+}$ is a orthogonal projection operation that projects a vector to the complement space spanned by the column of J_u.

3.2. Alternating least squares (ALS)

ALS, a instance of block coordinate descent, optimizes the parameters with initial value $\boldsymbol{u}_0$ and $\boldsymbol{v}_0$ alternately. It takes advantages of the separable structure of $f(U,V)$ and eliminates $\boldsymbol{u}$ and $\boldsymbol{v}$ respectively by minimizing (6). For example, for a fixed $\boldsymbol{u}_k$

$$\boldsymbol{v}_{k+1}^{*}(\boldsymbol{u}) = \arg\min_{\boldsymbol{v}} \left\|\hat{U}_k\boldsymbol{v} - \hat{\boldsymbol{m}}\right\|_2^2 \tag{11}$$

$$= \hat{U}_k^{+}\hat{\boldsymbol{m}} \tag{12}$$

Alternately, $\boldsymbol{u}_{k+1}^{*}(\boldsymbol{v}) = \hat{V}_{k+1}^{+}\hat{\boldsymbol{m}}$. However, as mentioned in the introduction, the ALS shows poor performance when it is facing ill-conditioned datasets.

3.3. Embedded point iteration (EPI)

EPI updates the parameter $\boldsymbol{u}$ by employing of the normal equation (9) while $\boldsymbol{v}$ is determined by minimizing (6). Setting $\boldsymbol{v}_k^{*}(\boldsymbol{u})$ be the optimal solution of $F^{*}(U_k,V)$ at iteration k, which implies

$$\frac{\partial F^{*}}{\partial \boldsymbol{u}} = J_v^Tf = 0 \tag{13}$$

Inserting (13) into the normal equation (9)

$$\begin{bmatrix} J_u^TJ_u & J_u^TJ_v \\ J_v^TJ_u & J_v^TJ_v \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{u} \\ \Delta\boldsymbol{v} \end{bmatrix} = -\begin{bmatrix} J_u^Tf \\ 0 \end{bmatrix} \tag{14}$$

we can obtain

$$\Delta\boldsymbol{u} = -(J_u^T(I - J_vJ_v^{+})J_u)^{-1}J_u^Tf \tag{15}$$

Note that the solution of $\Delta\boldsymbol{v}$ in (14) is bypassed. The iteration k is as follows:

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \Delta\boldsymbol{u}$$

$$\boldsymbol{v}_{k+1} = \hat{U}_k^{+}\hat{\boldsymbol{m}}$$

4. Variable projection (VP)

4.1. Derivatives of VP

For the separable nonlinear least-squares problem, the VP algorithm takes into account the relationship between the parameters U and V. It eliminates one part of the parameters using least-squares method, resulting in a reduced function that only contains U. In LRMF, it is always appropriate to get rid of the variables of large dimension(w.l.o.g. $\boldsymbol{v}$). Substituting $\boldsymbol{v}^* = \hat{U}^+\hat{\boldsymbol{m}}$ into (6) yields a reduced objective function

$$F(U,V^*) = ||f(U,V^*)||_2^2 = ||(\hat{U}\hat{U}^+ - I)\hat{\boldsymbol{m}}||_2^2 \quad (16)$$

The form (16) is more complicated than the original optimization problem (6). The key technique of using second-order method to optimize the reduced function is to calculate the derivative of $\boldsymbol{v}^*$:

$$\frac{\mathrm{d}V^*(U)}{\mathrm{d}U} \quad (17)$$

To alleviate computational burden, Kaufman [15] proposed a reduction:

$$\left[\frac{\partial \boldsymbol{v}^*}{\partial \boldsymbol{u}}\right]_{Kauf.} = -\hat{U}^+\hat{V} \quad (18)$$

A large amount of literature shows that Kaufman-form VP converges analogously to fully-derived VP while benefiting from reduced computational complexity. Kaufman's VP algorithm yields the Jacobian matrix and the gradient of F as follows

$$J_u^* = \frac{\partial f(\boldsymbol{u},\boldsymbol{v}^*(\boldsymbol{u}))}{\partial \boldsymbol{u}} = \frac{\partial f^*}{\partial \boldsymbol{u}} + \frac{\partial f^*}{\partial \boldsymbol{v}}\frac{\mathrm{d}\boldsymbol{v}^*(\boldsymbol{u})}{\mathrm{d}\boldsymbol{u}} = (I - \hat{U}\hat{U}^+)\hat{V} \quad (19)$$

Naturally

$$g_u^* = \frac{\partial F}{\partial \boldsymbol{u}} = J_u^{*T} f = \hat{V}^T(I - \hat{U}\hat{U}^+)f \quad (20)$$

With the Jacobian matrix at hand, the GN, LM, and BFGS method can be used to optimize the reduced function.

When using the GN method, the update of $\boldsymbol{u}$ can be obtained by solving:

$$J_k^T J_k \Delta\boldsymbol{u} = -f_k \quad (21)$$

then,

$$\Delta\boldsymbol{u}_k = -(J_k^T J_k)^{-1} f_k \quad (22)$$

where $J_k := J(\boldsymbol{u}_k)$. Observe (21) that, $J_k^T J_k$ should be fully rank or GN method abort.

The GN algorithm is numerically unstable, to overcome this shortcoming, the LM algorithm introduces a penalty to achieve:

$$\arg\min_{\Delta\boldsymbol{u}} ||f_k + J_k\Delta\boldsymbol{u}||_2^2 + \lambda_k||\Delta\boldsymbol{u}||_2^2 \quad (23)$$

where $\lambda_k > 0$ is a damping parameter. The solution of (23) is determined according to the following procedure:

$$H_k\Delta\boldsymbol{u}_k := (J_k^T J_k + \lambda_k I)\Delta\boldsymbol{u}_k = -J_k^T f_k \quad (24)$$

by solving which, we have

$$\Delta\boldsymbol{u}_k := -H_k^{-1}J_k^T f_k = -J_k^{-\lambda_k} f_k \quad (25)$$

4.2. Broyden-Fletcher-Goldfarb-Shanno method

BFGS method is an efficient quasi-Newton method for solving nonlinear optimization problems. Compared with the GN and LM methods, the BFGS method only requires the gradient information of the objective. which makes it more convenient for some complicate problems. Therefore, in this paper, we combine the VP strategy and BFGS method to propose a VP-BFGS method, which utilizes the VP strategy to eliminate one part of the parameter v and then optimizes the reduced function using the BFGS method.

Define $s_k := \boldsymbol{u}_{k+1} - \boldsymbol{u}_k, y_k := g_{u(k+1)} - g_{uk}$, and $y_k = \Delta H_k s_k$, then the update of approximation Hessian matrix [13] can be expressed as:

$$\Delta H_k = \frac{y_k y_k^T}{y_k^T s_k} - \frac{H_k s_k s_k^T H_k}{s_k^T H_k s_k} \quad (26)$$

To avoid computing the inverse of Hessian matrix, we apply Sherman-Morrison formula to (26). Define $D_{k+1} := H_{k+1}^{-1}$, the update formula can be written as follows

$$D_{k+1} = (I - \frac{s_k y_k^T}{y_k^T s_k})D_k(I - \frac{y_k s_k^T}{y_K^T s_k}) + \frac{s_k s_k^T}{y_k^T s_k} \quad (27)$$

For given $\boldsymbol{u}_k$ and a descent direction $p_k := -D_k g_{uk}^*$ of the reduced function (16), we employ the Armijo inexact line search to obtain an appropriate step length, which satisfies

$$F(\boldsymbol{u}_k + \lambda_k p_k) \leq F(x_k) + \gamma\lambda_k g_k^T p_k, \gamma \in (0,1)$$

In practical terms, quasi-Newton method contains DFP and BFGS. Compared with DFP, BFGS has a self-correcting property, i.e. BFGS will correct the approximate Hessian matrix with deviation during the continuous updating of quasi-Newton matrix [12]. Hence, in this paper, we concentrate on BFGS rather than DFP.

5. Variable projection with Regularization

For the case of ill-condition, the classical VP method usually leads to overfitting. Regularization is an effective method to overcome such shortcoming. By adding the damping term of the estimated parameters to the objective function, we have

$$\min_{U,V} ||W \odot (UV^T - M)||_F^2 + \mu(||U||_F^2 + ||V||_F^2)$$

The RTRMC algorithm proposed by Boumal and Absil [11] introduces regularization into LRMF. The rest of the researchers tend to ignore it in the past decade. Nevertheless, a large number of existing works hold unanimously that VP unregularized converges surpassingly as VP regularized does [6].

6. Experiments

In this section, we carry out numerical experiments on several datasets to verify the performance of the proposed VP-BFGS method. Four algorithms listed in Table 1 are employed in the experiment.

All the experiments are carried out on ASUS notebook GU502LV with Intel i7-10875h CPU and 16GB RAM us-

Table 1. A list of algorithms

ID	Strategy	Second-Order Solver
JO-LM	Joint	LM
JO-BFGS	Joint	BFGS
VP-LM	VP	LM
VP-BFGS	VP	BFGS

Table 2. Datasets used for the experiments

Dataset	Dimension	Filled	Filling Rate%
S0	2×2	4	100
S1	10×10	49	49
S2	30×60	919	51.06
S3	40×80	1620	50.62
S4	80×80	3258	50.91
S5	80×100	3901	48.76
S6	8×2000	7862	49.14
D.M.	8×319	484	18.97
L.M.	6×667	2832	70.76
M.M.	6×475	2234	78.39

ing matlab R2017b.

6.1. Datasets

The four algorithms aforesaid applied to the datasets listed in Table 2. To testify the algorithm performance from different dimension datasets, S0-S6 are randomly generated by Matlab, and the scale of them gradually increases, from 2×2 to 8×2000. We suppose that the presence of the elements in M obeys the normal distribution $\mathcal{N}(1,0)$ and maintains the filling rate at 50% or so. In addition to synthetic datasets, several SfM ones are also within the consideration. We processed the Dino-Trimmed, Library and Merton-2 from [6] by dividing the observation matrix by a hundred, and renamed them Dino-Modified, Library-Modified and Merton-2-Modified respectively. On account of the incompleteness of the observation matrix, the cost values used in the following part are

$$cost := \sqrt{\frac{||W\odot(UV^T-M)||_F^2}{Number\ of\ Visible\ Points}}$$

Datasets those listed in Table 2 are run several times with different initial points. The global solutions and their number of convergence steps are recorded in Table 3.

6.2. Experimental Conditions

To keep a fair comparison, for each algorithm, we generated each element of $\boldsymbol{u}_0$ and $\boldsymbol{v}_0$ that obeys $\mathcal{N}(1,0)$. Each iteration was going on until either when the upper limit of iterations that we set it to be 300 was reached or the norm of the residual dropped blow tolerance 10e-6. Note that, $\hat{U}$ and $\hat{V}$ are of large size which drag the iterative process to varying degrees, we introduce some speed-up tricks mentioned by [6]:

Table 3. Optimas of each datasets obtained by algorithms.

Dataset	Rank	Optima	From	Iteration
S0	4	2.62e-16	VP-BFGS	2
S1	4	5.64e-7	VP-BFGS	49
S2	4	2.044	VP-LM	47
S3	4	0.4502	VP-BFGS	109
S4	4	0.4998	VP-BFGS	237
S5	4	0.5048	VP-BFGS	300
S6	4	0.1963	VP-LM	146
D.M.	4	1.47e-13	VP-BFGS	2
L.M.	4	0.198	VP-LM	10
M.M.	4	0.1862	VP-LM	9

• Under the assistance of QR-decomposition [14], given that $\hat{U}=Q_{\hat{U}}R_{\hat{U}}$, the equation (12) can be written as

$$\boldsymbol{v}^*=(\hat{U}^T\hat{U})^{-1}\hat{\boldsymbol{m}}=R_{\hat{U}}^{-1}Q_{\hat{U}}^T\hat{\boldsymbol{m}} \tag{28}$$

• The unregularized objective (5) can be regarded as the sum of multiple subproblem [8]:

$$F(U,V^*)=\sum_{j=1}^{n}||W_j\odot(U\boldsymbol{v}_j^*-\boldsymbol{m}_j)||_2^2 \tag{29}$$

by minimizing (29), we can obtain $\boldsymbol{v}_j^*=\hat{U}^+\hat{\boldsymbol{m}}_{\boldsymbol{j}}$, where W_j is the column j of W, $\boldsymbol{v}_j^*$ is the column j of $\boldsymbol{v}^*$ and $\boldsymbol{m}_j$ is the column j of $\boldsymbol{m}$.

6.3. Results

Figure 1 indicates that BFGS based on VP converges in 49 iterations while the other three algorithms run until the upper threshold is nearly reached. For dataset S3, it is a classical "horizontal-line" case, the reason is that the joint optimization falls into the local optimum. And within our expectation, the VP based algorithms converge efficiently with global optima obtained.

In all experiments, the VP based algorithms perform better than the joint optimization algorithm. Take Dino-Modified as the most striking example (see Figure 3). As shown in Figure 3, while joint-based algorithms are descending slowly, VP-LM and VP-BFGS converge in two steps unexpectedly. The "vertical descending" of the cost indicates the efficiency of variable projection. Note that the column dimension is far less than row one in Dino-Modified. By eliminating $\boldsymbol{v}\in\mathbb{R}^{1276}$, the parameters reduces to $\boldsymbol{u}\in\mathbb{R}^{32}$ from $x\in\mathbb{R}^{1308}$, which greatly reduces the space dimension. Similar circumstances occurred on dataset S6 with dimension of 8×2000 (see Figure 4). Compared with reduced parameters $\boldsymbol{u}\in\mathbb{R}^{32}$, $\mathbf{x}\in\mathbb{R}^{8032}$ from joint optimization is inapplicable. As shown in Figure 4, VP-BFGS steps faster than VP-LM, which further confirms the efficiency of the proposed algorithm.

7. Conclusions

Low-rank matrix factorization is an important research topic in computer vision. The classical SVD method is

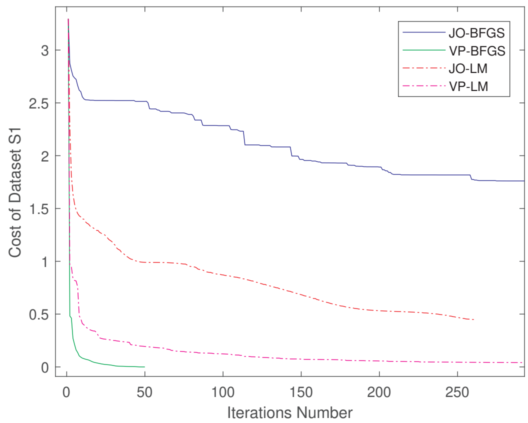

Fig. 1. Results of four algorithms on S1 (10×10)

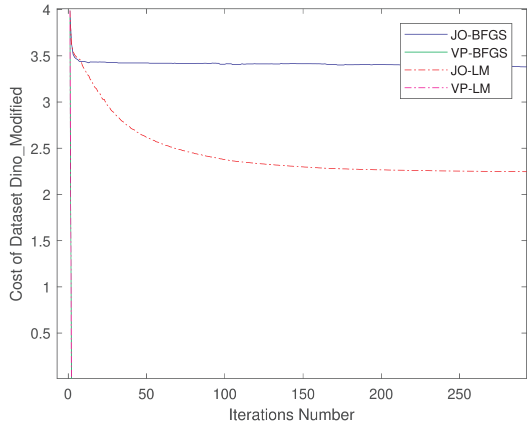

Fig. 3. Results of four algorithms on Dino_Modified (8×319)

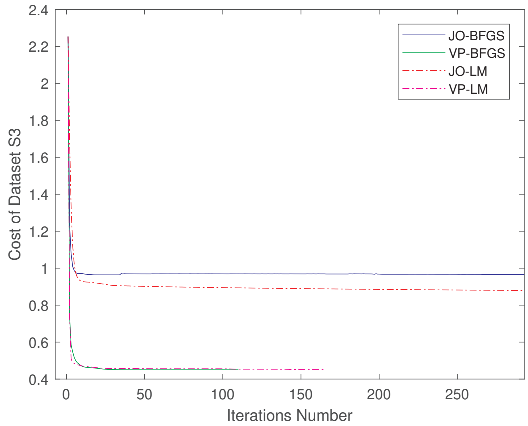

Fig. 2. Results of four algorithms on S3 (40×80)

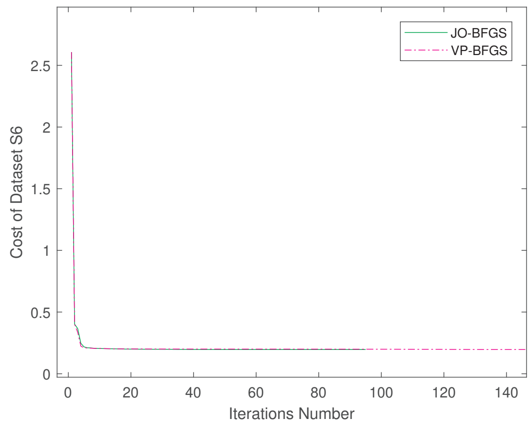

Fig. 4. Results of four algorithms on S6 (8×2000)

an important method to deal with such task, however, it has many limitations in real-world applications. In this paper, we consider the case that the observation matrix is of huge size and incomplete. We convert the task of factorizing a large matrix into solving a separable nonlinear least-squares problem and propose a VP-BFGS method. The method first utilizes the VP strategy to eliminate part of the parameter and employ the BFGS algorithm to optimize the reduced function. The proposed method fully takes advantage of the separable structure, and is proven to be quite valuable on several numerical experiments, achieving faster convergence and better reconstruction performance than the joint and ALS optimization strategies.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 62073082.

References:

[1] Pei C. Optimization Algorithms on Subspaces: Revisiting Missing Data Problem in Low-Rank Matrix[J]. International Journal of Computer Vision, 2008, 80(1):125-142.

[2] Buchanan A M, Fitzgibbon A W. Damped Newton Algorithms for Matrix Factorization with Missing Data[J]. Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.

[3] Hong J H, Zach C, Fitzgibbon A. Revisiting the Variable Projection Method for Separable Nonlinear Least Squares Problems[C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2017.

[4] Jeong Y, Nister D, D Steedly, et al. Pushing the envelope of modern methods for bundle adjustment[C]// 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, 2010.

[5] Minka T P. Old and New Matrix Algebra Useful for Statistics. 1999.

[6] Hyeong J, Fitzgibbon A. Secrets of Matrix Factorization: Approximations, Numerics, Manifold Optimization and Random Restarts[J]. Proceedings, 2015.

[7] Gallier J. The Schur Complement and Symmetric Positive Semidefinite (and Definite) Matrices. 2010.

[8] Hyeong J, Fitzgibbon A. Secrets of Matrix Factorization: Supplementary material to ICCV 2015 submission. Technical report.

[9] Golub G, Pereyra V. Separable nonlinear least squares: the variable projection method and its applications[J]. Speech Communication, 2003, 45(1):63-87.

[10] Golub G H, Pereyra V. The Differentiation of Pseudo-Inverses and Nonlinear Least Squares Problems Whose Variables Separate[J]. Siam Journal on Numerical Analysis, 1973, 10(2):413-432.

[11] Boumal N, Absil P A. RTRMC: A Riemannian trust-region method for low-rank matrix completion. Curran Associates Inc. 2011.

[12] Byrd R H, Nocedal J. A Tool for the Analysis of Quasi-Newton Methods with Application to Unconstrained Minimization[J]. SIAM Journal on Numerical Analysis, 1989, 26(3):727-739.

[13] Dennis J E, Mor& E A. Quasi-Newton Methods, Motivation and Theory[J]. Siam Review, 1977, 19(1):46-89.

[14] Okatani T, Yoshida T , Deguchi K . Efficient algorithm for low-rank matrix factorization with missing components and performance comparison of latest algorithms[J]. IEEE, 2011.

[15] Kaufman L. A variable projection method for solving separable nonlinear least squares problems[J]. Bit, 1975, 15(1):49-57.