Online Completion of Ill-conditioned Low-Rank Matrices

Ryan Kennedy and Camillo J. Taylor
Computer and Information Science
University of Pennsylvania
Philadelphia, PA, USA
{kenry, cjtaylor}@cis.upenn.edu

Laura Balzano
Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI, USA
girasole@umich.edu

***Abstract*—We consider the problem of online completion of ill-conditioned low-rank matrices. While many matrix completion algorithms have been proposed recently, they often struggle with ill-conditioned matrices and take a long time to converge. In this paper, we present a new algorithm called Polar Incremental Matrix Completion (PIMC) to address this problem. Our method is based on the GROUSE algorithm, and we show how a polar decomposition can be used to maintain an estimate of the singular value matrix to better deal with ill-conditioned problems. The method is also online, allowing it to be applied to streaming data. We evaluate our algorithm on both synthetic data and a real "structure from motion" dataset from the computer vision community, and show that PIMC outperforms similar methods.**

***Index Terms*—matrix completion, online optimization, condition number**

I. Introduction

Low-rank matrix structure has found applications in a great number of domains, and the applicability of low-rank matrix completion results to real data problems is quite promising since datasets often have missing or unobserved values. Since the seminal results of [6], [7], many algorithms have been developed for low-rank matrix completion [1], [5], [9], [10], [11], [13]. However, the low-dimensional structure found in real data is rarely well-behaved: singular values of large data matrices often drop off in such a way that it is not obvious at what point we are distinguishing signal from noise. In these scenarios, the suite of existing matrix completion algorithms all struggle to find the true low-rank component, both with regards to achieving low error and with regards to the number of algorithm iterations it takes to get a good result. Recently, several algorithms have been proposed which improve performance for matrices with large condition numbers [10], [9], but these algorithms still have difficulty for extremely ill-conditioned problems. Furthermore, these algorithms are batch and cannot easily be used for streaming data.

This paper makes the following contributions. First, we show how the GROUSE algorithm for online matrix completion [1] can be re-interpreted via the Incremental Singular Value Decomposition (ISVD) [4] as finding the solution to a specific least-squares problem. Based on this interpretation, we then present a modification to this algorithm which drastically improves its performance for matrices with large condition number. We also demonstrate experimentally that our algorithm outperforms other batch matrix completion algorithms on extremely ill-conditioned problems.

II. The ISVD Formulation of GROUSE

We begin by briefly describing the GROUSE algorithm [1] and its relation to the incremental singular value decomposition (ISVD) [2]. The ISVD algorithm [4], [3] is a simple method for computing the SVD of a matrix by updating an initial decomposition one column at a time. Given a matrix $A_t \in \mathbb{R}^{n \times m}$ at time t whose SVD is $A_t = U_t \Sigma_t V_t^T$, we wish to compute the SVD of a new matrix with a single column added: $A_{t+1} = \begin{bmatrix} A_t & v_t \end{bmatrix}$. Defining weights $w_t = U_t^T v_t$ and residual $r_t = v_t - U_t w_t$, we have

$$A_{t+1} = \begin{bmatrix} U_t & \frac{r_t}{\|r_t\|} \end{bmatrix} \begin{bmatrix} \Sigma_t & w_t \\ 0 & \|r_t\| \end{bmatrix} \begin{bmatrix} V_t^T & 0 \\ 0 & 1 \end{bmatrix} . \qquad (1)$$

We compute an SVD of the center matrix,

$$\begin{bmatrix} \Sigma_t & w_t \\ 0 & \|r_t\| \end{bmatrix} = \hat{U} \hat{\Sigma} \hat{V}^T , \qquad (2)$$

which yields the new SVD, $A_{t+1} = U_{t+1} \Sigma_{t+1} V_{t+1}^T$ where

$$U_{t+1} = \begin{bmatrix} U_t & \frac{r_t}{\|r_t\|} \end{bmatrix} \hat{U}; \;\; \Sigma_{t+1} = \hat{\Sigma}; \;\; V_{t+1} = \begin{bmatrix} V_t & 0 \\ 0 & 1 \end{bmatrix} \hat{V} . \qquad (3)$$

If only the top k singular vectors are needed, then we can apply the heuristic of dropping the smallest singular value and the associated singular vector after each such update.

It has recently been shown that the GROUSE algorithm [1] has a close relationship to this ISVD algorithm [2]. Let $\hat{A}_t = U_t R_t^T$ be an estimated rank-k factorization of A_t such that U_t has orthonormal columns. Given a new column v_t with missing data, let $\Omega_t \subseteq \{1, \ldots, N\}$ be the set of observed entries. If w_t and r_t are now the least-squares weight and residual vector, respectively, defined with respect to *only the set of observed indices* Ω_t, then we can write

$$\begin{bmatrix} U_t R_t^T & \tilde{v}_t \end{bmatrix} = \begin{bmatrix} U_t & \frac{r_t}{\|r_t\|} \end{bmatrix} \begin{bmatrix} I & w_t \\ 0 & \|r_t\| \end{bmatrix} \begin{bmatrix} R_t & 0 \\ 0 & 1 \end{bmatrix}^T , \qquad (4)$$

where $\tilde{v}_t$ has imputed values, defined as

$$\tilde{v}_t = \begin{cases} v_{\Omega_t} & \text{on } \Omega_t \\ U_t w_t & \text{otherwise} \end{cases} .$$

Note the similarity of Equations (1) and (4). Taking the SVD of the center matrix to be

$$\begin{bmatrix} I & w_t \\ 0 & \|r_t\| \end{bmatrix} = \hat{U}\hat{\Sigma}\hat{V}^T, \tag{5}$$

it was shown in [2] that updating U_t to

$$U_{t+1} = \begin{bmatrix} U_t & \frac{r_t}{\|r_t\|} \end{bmatrix} \hat{U} \tag{6}$$

and subsequently dropping the last column is equivalent to GROUSE for a specific step size, which performs gradient descent directly on the Grassmann manifold. Combining Equations (4) and (5), updating R_t then becomes

$$R_{t+1} = \begin{bmatrix} R_t & 0 \\ 0 & 1 \end{bmatrix} \hat{V}\hat{\Sigma}, \tag{7}$$

and dropping the last column provides a corresponding update for the matrix R_t. The result is a new rank-k factorization $\hat{A}_{t+1} = U_{t+1}R_{t+1}^T$.

We may get insight into this version of GROUSE by examining this interpretation using what we know about the SVD. By the Eckart-Young theorem [8], the process of Equations (5) and (6) are finding the closest rank-k matrix to $\begin{bmatrix} U_t & \tilde{v}_t \end{bmatrix}$ with respect to the Frobenius norm. In other words, we can interpret this new algorithm as solving the minimization problem

$$\min_{\text{rank}(M)=k} \| \begin{bmatrix} U_t & \tilde{v}_t \end{bmatrix} - M\|_F^2 \ . \tag{8}$$

The updated U_{t+1} is then given by the top k left singular vectors of M (or any orthonormal vectors which span this subspace). Let $M = \hat{U}\hat{Z}^T$, where $\hat{U} \in \mathbb{R}^{n \times k}$, and

$$\hat{Z} = \begin{bmatrix} \hat{z}_1 \\ \vdots \\ \hat{z}_k \\ w \end{bmatrix} = \begin{bmatrix} \hat{Z}_k \\ w \end{bmatrix} \in \mathbb{R}^{(k+1)\times k} \ ; \tag{9}$$

note this enforces the rank-k constraint on M. By plugging into (8), we see that each iteration of this algorithm amounts to minimizing the following cost function:

$$U_{t+1} = \arg\min_{\hat{U}} \left\{ \min_{\hat{Z}_k} \|U_t - \hat{U}\hat{Z}_k\|_F^2 + \min_{w} \|\tilde{v}_t - \hat{U}w\|_2^2 \right\} \tag{10}$$

This has an intuitive interpretation: the first term requires that U_{t+1} have a column span close to that of the current subspace U_t and the second term requires that the new vector $\tilde{v}$ can be well-approximated by a linear combination of the columns of U_{t+1} . The updated matrix is the one that minimizes the combination of these two competing costs.

The first term of this minimization problem can be scaled by a parameter λ in order to allow for a trade-off between the two terms, and by bringing λ inside the norm and incorporating it into $\hat{Z}_k$, this is equivalent to scaling U_t:

$$\arg\min_{\hat{U}} \left\{ \min_{\hat{Z}_k} \|U_t\sqrt{\boldsymbol{\lambda}} - \hat{U}\hat{Z}_k\|_F^2 + \min_{w} \|\tilde{v}_t - \hat{U}w\|_2^2 \right\} \tag{11}$$

A larger λ will lead to a smaller change; it can be used as a regularization parameter.

In contrast to the ISVD algorithm, GROUSE does not make any use of the singular values of the matrix. By not using an estimate of the singular values, GROUSE can have difficulty converging for ill-conditioned matrices. This is demonstrated in Figure 1, where GROUSE was run on a rank-5 matrix with no missing data and no noise. Even in this ideal setup, the condition number of the matrix has a large effect on the convergence rate of GROUSE.

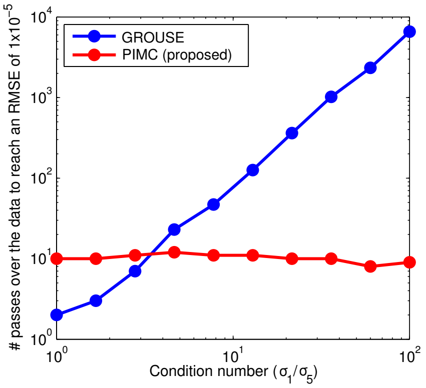


Fig. 1: **Effect of the condition number of a matrix on the convergence.** We consider a rank-5 matrix of size 500×500 with no noise or missing data. We plot the number of passes over the data that were required to reach an RMSE error of 1×10^{-5}. As the condition number increases, GROUSE convergence slows while that of our proposed algorithm PIMC remains constant.

This convergence issue has been previously noticed in batch matrix completion algorithms, and several algorithms have been presented which alter the optimization on the Grassmann manifold in order to take into account the non-isotropic scaling of the space by incorporating the singular values into the optimization [9], [10]. These algorithms have demonstrated improved performance on ill-conditioned matrices, but are limited to the batch setting. Furthermore, as we show in Section IV, even these algorithms have trouble with extremely ill-conditioned matrices. We take a similar approach and incorporate the use of singular values into GROUSE, which allows for accurate matrix completion even for very ill-conditioned matrices in an online manner.

III. PIMC FOR MATRIX COMPLETION

In order to improve the convergence of GROUSE for ill-conditioned matrices, we would like to use U_tS_t as a representative of the current subspace, where S_t is an estimate of the singular values, rather than just U_t. However, we cannot directly use ISVD and just drop the last column at each iteration to maintain a constant rank for two reasons. First, the resulting singular values may not be a good estimate for the real singular values because of the missing data.

Second, the ISVD requires V_t to be orthogonal, so while with GROUSE it is straightforward to re-process a data vector that has previously been processed by removing the column from R_t, with ISVD it is not possible.

We therefore propose a new algorithm, which we call Polar Incremental Matrix Completion (PIMC). Let $\hat{A}_t = U_t R_t^T$ be the current estimate of a matrix completion problem at time t. We represent R_t by its *polar decomposition*

$$R_t = \tilde{V}_t \tilde{S}_t \ , \tag{12}$$

where $\tilde{V}_t \in \mathbb{R}^{m \times k}$ has orthonormal columns and $\tilde{S}_t \in \mathbb{R}^{k \times k}$ is positive semidefinite. This polar decomposition exists for any matrix R_t, and if $\bar{U}\bar{S}\bar{V}^T = R_t$ is an SVD of R_t, then the factors can be written explicitly as

$$\tilde{V}_t = \bar{U}\bar{V}^T \text{ and } \tilde{S}_t = \bar{V}\bar{S}\bar{V}^T \ . \tag{13}$$

The matrix $\tilde{V}_t$ now has orthonormal columns, similar to V_t from ISVD. Likewise $\tilde{S}_t$ is an estimate of the singular values of the space, although it may no longer be diagonal.

We additionally choose to scale S_t to account for the fact that $U_t S_t$ is still only an approximation to past data due to the missing entries. When data are missing, the weights w_t are defined with respect to only the data that are observed, but we use the *interpolated vector* $\tilde{v}_t = U_t w_t + r_t$ in our update. Recalling that the sum of squares of the singular values is equal to the sum of column 2-norms, the singular values will therefore be increasing with respect to this interpolated vector rather than with respect to only the observed data as we would like.

Instead, we will re-scale the singular value matrix S_t to account for only the observed entries. To do so, we keep a running sum of the norm of the actual observed data,

$$s_t^2 = s_{t-1}^2 + \|v_{\Omega_t}\|_2^2 \ , \tag{14}$$

and at each iteration scale S_t by $\gamma \frac{s_t}{\|S_t\|_F}$, where γ is a fixed constant. The resulting factorization is given by

$$A_{t+1} = \begin{bmatrix} U_t & \frac{r_t}{\|r_t\|} \end{bmatrix} \begin{bmatrix} \frac{\gamma s_t}{\|S_t\|_F} S_t & w_t \\ 0 & \|r_t\| \end{bmatrix} \begin{bmatrix} \frac{\|S_t\|_F}{\gamma s_t} R_t^T & 0 \\ 0 & 1 \end{bmatrix} \ . \tag{15}$$

Our method, PIMC, then finds the SVD of the center matrix and subsequently drops the last singular value and the corresponding singular vectors at each iteration.

Note that the use of S_t effectively scales U_t at each iteration, in a similar way to adding a regularization parameter λ in Equation 11, and so we do not explicitly set λ in our experiments. The full algorithm is shown in Algorithm 1.

IV. Experiments

We compare our proposed algorithm PIMC to the ISVD formulation of GROUSE, LMaFit [13], APGL [11], ScGrad [10], and qGeom [9]; the latter two are batch algorithms designed to perform well on ill-conditioned matrices by modifying the metric on the Grassmann manifold. We used MATLAB code from the respective authors with default parameters. For PIMC, γ was set to 0.01 for all experiments.

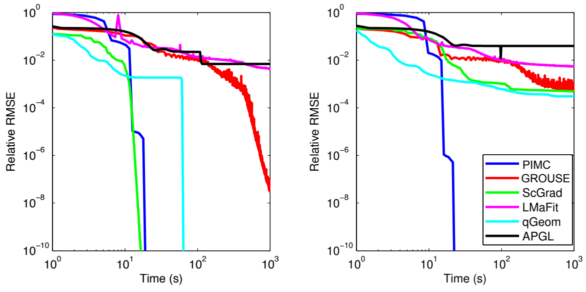
Algorithm 1 PIMC for matrix completion

```
procedure PIMC ($A, \gamma, t_{\max}$)
    Initialize $U_1, S_1, R_1, s_0$
    for $t \leftarrow 1, \dots, k_{\max}$ do
        Select a column $i$ of $A$: $v_t = A(:, i)$
        Estimate weights: $w_t = \arg\min_a \|U_{\Omega_t} a - v_{\Omega_t}\|_2^2$
        Update the scaling weight: $s_t^2 = s_{t-1}^2 + \|v_{\Omega_t}\|_2^2$
        Compute residual: $r_{\Omega_t} = v_{\Omega_t} - U_{\Omega_t} w_t; \quad r_{\Omega_t^C} = 0$
        Zero-out row of $R_t$: $R_t(i, :) = 0$
        if re-orthogonalizing $R_t$ then
            Compute polar decomposition: $R_t = \tilde{V}_t \tilde{S}_t$
            Update matrices: $R_t = \tilde{V}_t; \quad S_t = S_t \tilde{S}_t^T$
        end if
        Compute SVD of center matrix:
            $\hat{U}\hat{S}\hat{V}^T = SVD\left(\begin{bmatrix} \frac{\gamma s_t}{\|S_t\|_F} S_t & w_t \\ 0 & \|r_t\| \end{bmatrix}\right)$
        Update $U_t$: $U_{t+1} = \begin{bmatrix} U_t & \frac{r_t}{\|r_t\|} \end{bmatrix} \hat{U}$
        Update $S_t$: $S_{t+1} = \hat{S}$
        Set up last column for $R_t$ update:
            $z = \begin{bmatrix} 0 & \dots & 0 \end{bmatrix}^T; \quad z(i) = 1$
        Update $R_t$: $R_{t+1} = \begin{bmatrix} \frac{\|S_t\|_F}{\gamma s_t} R_t & z \end{bmatrix} \hat{V}$
        Drop last singular value and corresponding singular vectors
    end for
    return $U_{t_{\max}}, S_{t_{\max}}, R_{t_{\max}}$
end procedure
```

A. Synthetic data without noise

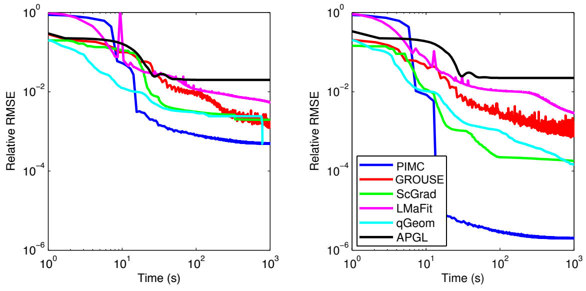
We generated a 5000×5000 matrix W of rank 5 as the product $W = XSY^T$ where X and Y are random 5000×5 matrices with orthonormal columns and S is a 5×5 diagonal matrix containing the singular values. The smallest singular value was set to be $\sigma_5 = 1 \times 10^3$ and they varied logarithmically up to σ_1. 95% of the entries were removed uniformly at random.

Results are shown in Figure 2 for two values of σ_1. In all cases, the algorithms that took account of an estimate of the singular values of the space – PIMC, ScGrad and qGeom – outperformed the other matrix completion algorithms. However, with an increase of one order of magnitude, the performance of ScGrad and qGeom suffers (Figure 2b). We note that the authors of both of these algorithms only performed experiments for condition numbers up to around 10, while here we have gone up to 1000. Our proposed algorithm PIMC converges in roughly the same amount of time regardless of the condition number. We hypothesize that this may be due to the fact that ScGrad and qGeom both perform an alternating optimization, having to retract back onto the manifold at each iteration, while PIMC has no alternation and remains orthogonal the entire time.



(a) $\frac{\sigma_1}{\sigma_5} = 1 \times 10^2$ (b) $\frac{\sigma_1}{\sigma_5} = 1 \times 10^3$

Fig. 2: **Comparison without noise.** Random 5000×5000, rank-5 matrices with no noise and 95% of their entires missing were generated with singular values that varied logarithmically from $\sigma_1 = 1 \times 10^3$ up to σ_5. In all cases, PIMC converges in roughly the same amount of time.



(a) $\tau = 0.3$ (b) $\tau = 0.1$

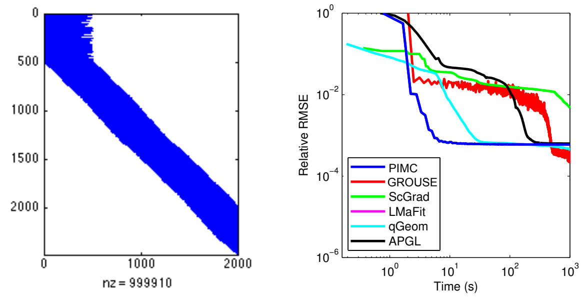
Fig. 3: **Comparison with noise.** Singular values were set to decay exponentially from $\sigma_1 = 1 \times 10^7$ as $\sigma_i = \sigma_{i-1} * \tau$ and 95% of their entires missing were generated. The rank to estimate was set to 5 and we measure the error with respect to the best rank-5 matrix taken from the full data. τ was set to 0.3 and 0.1, resulting in matrices with σ_1 being 123 and 10000 times larger than σ_5.

B. Synthetic data with noise

We next test how the algorithms perform with respect to noise using a random 5000×5000 matrix with singular values that decay exponentially as $\sigma_i = \tau\sigma_{i-1}$ with $\sigma_1 = 1 \times 10^7$, for some constant τ. 95% of the data were randomly removed and the estimated rank was set to 5. Results are shown in Figure 3 for $\tau = 0.3$, and 0.1. The error was measured with respect to the best rank-5 matrix as calculated using the SVD of the data matrix before any data were removed. This situation is much more difficult and no algorithm is able to find the optimal solution in any situation due to the lack of separation between the signal and noise subspaces. However, it is again the case that PIMC outperforms other algorithms when the spread of singular values is larger.

C. Structure from motion data

Structure-from-motion involves recovering the full 3D locations of points given their 2D locations tracked over the frames of a video. These tracks can be arranged in a *measurement matrix* where every pair of columns gives the x and y locations of points over all frames and each row contains the 2D locations of all points in a given frame. If the camera is assumed to be affine, then it can be shown that this matrix has rank at most 4 [12]. Missing data occur when points tracks are lost or become occluded.



(a) Banded structure of the data (b) Comparison of algorithms

Fig. 4: **Comparison of algorithms on structure-from-motion dataset**. All algorithms have trouble reaching the optimum due to the banded structure of the data matrix. PIMC converges the fastest and GROUSE has the least error after 1×10^3 seconds. See text for details on the dataset.

We generated a synthetic cylinder of radius 10 and height 5000 with 500 points tracked over 1000 frames. After removing points tracked for fewer than five frames, the resulting measurement matrix has size 2484×2000. The cylinder rotated once every 500 frames, resulting in 80.13% missing data. This matrix has an exact rank-4 solution with a condition number $\sigma_1/\sigma_4 \approx 290$. An interesting aspect of this dataset is that the data are not randomly observed, but appear within a band down the diagonal of the matrix (Figure 4a). This stands in contrast to the theoretical guarantees of convergence for matrix completion which assume that data are observed uniformly at random [6], [7].

Figure 4 shows results on the structure-from-motion dataset. All algorithms perform relatively similarly with PIMC converging fastest and GROUSE achieving the lowest error, but all are unable to find the optimal solution. We have found that the banded structure of the data matrix here makes optimization more difficult than if the data were sampled uniformly at random, and when combined with a large condition number this optimization problem is quite challenging for all algorithms.

V. Conclusion

In this paper we have presented a novel algorithm for matrix completion based on the incremental singular value decomposition. Our method is online and takes into account an estimate of the singular values during optimization to improve convergence for matrices which are ill-conditioned. We have demonstrated that it outperforms other batch algorithms for extremely ill-conditioned matrices.

References

[1] L. Balzano, R. Nowak, and B. Recht. Online identification and tracking of subspaces from highly incomplete information. In *Communication, Control, and Computing (Allerton)*, pages 704–711. IEEE, 2010.

[2] L. Balzano and S. J. Wright. On GROUSE and incremental SVD. In *IEEE Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2013.

[3] M. Brand. Incremental singular value decomposition of uncertain data with missing values. *European Conference on Computer Vision (ECCV)*, pages 707–720, 2002.

[4] J. R. Bunch and C. P. Nielsen. Updating the singular value decomposition. *Numerische Mathematik*, 31:111–129, 1978. 10.1007/BF01397471.

[5] J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2008.

[6] E. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, December 2009.

[7] E. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053 –2080, May 2010.

[8] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[9] B. Mishra, A. A. Karavadi, and R. Sepulchre. A riemannian geometry for low-rank matrix completion. *Tech Report*, 2012.

[10] T. Ngo and Y. Saad. Scaled gradients on grassmann manifolds for matrix completion. In *Advances in Neural Information Processing Systems*, 2012.

[11] K.-C. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of Optimization*, 6(615-640):15, 2010.

[12] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[13] Z. Wen, W. Yin, and Y. Zhang. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation*, 2012.