Outlier-Robust Matrix Completion via $l_p$-Minimization

Hing Cheung So

http://www.ee.cityu.edu.hk/~hcso

Department of Electronic Engineering, City University of Hong Kong

Outline

- Introduction
- Matrix Completion via $l_p$-norm Factorization
  - Iterative $l_p$-Regression
  - Alternating Direction Method of Multipliers (ADMM)
- Numerical Examples
- Concluding Remarks
- List of References
Introduction

What is Matrix Completion?

The aim is to recover a low-rank matrix given only a subset of its possibly noisy entries, e.g.,

$$
\begin{pmatrix}
1 & ? & ? & 4 & ? \\
? & 2 & 5 & ? & ? \\
? & ? & 4 & 5 & ? \\
5 & ? & ? & ? & 4
\end{pmatrix}
$$
Let $X_\Omega \in \mathbb{R}^{n_1 \times n_2}$ be a matrix with missing entries:

$$[X_\Omega]_{ij} = \begin{cases} X_{ij}, & \text{if } (i, j) \in \Omega \\ 0, & \text{otherwise} \end{cases}$$

where $\Omega$ is a subset of the complete set of entries $[n_1] \times [n_2]$, while the unknown entries are assumed zero.

Matrix completion refers to finding $M \in \mathbb{R}^{n_1 \times n_2}$, given the incomplete observations $X_\Omega$ with the low-rank information of $X$, which can be mathematically formulated as:

$$\min_M \text{ rank}(M), \quad \text{s.t. } M_\Omega = X_\Omega$$

That is, among all matrices consistent with the observed entries, we look for the one with minimum rank.
Why Matrix Completion is Important?

It is a core problem in many applications including:

- Collaborative Filtering
- Image Inpainting and Restoration
- System Identification
- Node Localization
- Genotype Imputation

It is because many real-world signals can be approximated by a matrix whose rank is $r \ll \max\{n_1, n_2\}$.

Netflix Prize, whose goal was to accurately predict user preferences with the use of a database of over 100 million movie ratings made by 480,189 users in 17,770 films,
which corresponds to the task of completing a matrix with around 99% missing entries.

<table>
<thead>
<tr>
<th></th>
<th>Alice</th>
<th>Bob</th>
<th>Carol</th>
<th>Dave</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td></td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
How to Recover an Incomplete Matrix?

Directly solving the noise-free version:

$$\min_M \text{ rank}(M), \quad \text{s.t. } M_\Omega = X_\Omega$$

or noisy version:

$$\min_M \text{ rank}(M), \quad \text{s.t. } \|M_\Omega - X_\Omega\|_F \leq \epsilon_F$$

is difficult because the rank minimization problem is NP-hard.

A popular and practical solution is to replace the nonconvex rank by convex nuclear norm:
\[ \min_M \|M\|_*, \quad \text{s.t. } M_\Omega = X_\Omega \]

or

\[ \min_M \|M\|_*, \quad \text{s.t. } \|M_\Omega - X_\Omega\|_F \leq \epsilon_F \]

where \( \|M\|_* \) equals the sum of singular values of \( M \). However, complexity of nuclear norm minimization is still high and this approach is not robust if \( X_\Omega \) contains outliers.

Another popular direction which is computationally simple is to apply low-rank matrix factorization:

\[ \min_{U,V} f_2(U,V) := \|(UV)_\Omega - M_\Omega\|_F^2 \]

where \( U \in \mathbb{R}^{n_1 \times r} \) and \( V \in \mathbb{R}^{r \times n_2} \). Again, the Frobenius norm is not robust against impulsive noise.
Matrix Completion via $l_p$-norm Factorization

To achieve outlier resistance, we robustify the matrix factorization formulation via generalization of the Frobenius norm to $\ell_p$-norm where $0 < p \leq 2$:

$$
\min_{U,V} f_p(U, V) := \|(UV)_\Omega - X_\Omega\|_p^p
$$

where $\| \cdot \|_p$ denotes the element-wise $\ell_p$-norm of a matrix:

$$
\|X_\Omega\|_p = \left( \sum_{(i,j) \in \Omega} |X_{ij}|^p \right)^{1/p}
$$
Iterative $l_p$-Regression

To $\ell_p$-norm minimization, our first idea is to adopt the alternating minimization strategy:

$$V^{k+1} = \arg \min_V \| (U^kV)_\Omega - X_\Omega \|_p$$

and

$$U^{k+1} = \arg \min_U \| (UV^{k+1})_\Omega - X_\Omega \|_p$$

where the algorithm is initialized with $U^0$, and $U^k$ represents the estimate of $U$ at the $k$th iteration.

After determining $U$ and $V$, the target matrix is obtained as $M = UV$. 
We now focus on solving:

$$\min_V f_p(V) := \| (UV)_\Omega - X_\Omega \|_p^p$$

for a fixed $U$. Note that $(\cdot)^k$ is dropped for notational simplicity.

Denoting the $i$th row of $U$ and the $j$th column of $V$ as $u_i^T$ and $v_j$, where $u_i, v_j \in \mathbb{R}^r$, $i = 1, \ldots, n_1, j = 1, \ldots, n_2$, the problem can be rewritten as:

$$\min_V f_p(V) := \sum_{(i,j) \in \Omega} |u_i^T v_j - X_{ij}|^p$$

Since $f_p(V)$ is decoupled w.r.t. $v_j$, it is equivalent to solving the following $n_2$ independent subproblems:
\[
\min_{v_j} f_p(v_j) := \sum_{i \in I_j} |u_i^T v_j - X_{ij}|^p, \quad j = 1, \ldots, n_2
\]

where \( I_j = \{j_1, \ldots, j_{|I_j|}\} \subseteq \{1, \ldots, n_1\} \) denotes the set containing the row indices for the \( j \)th column in \( \Omega \). Here, \(|I_j|\) stands for the cardinality of \( I_j \) and in general \(|I_j| > r\).

For example, consider \( X_\Omega \in \mathbb{R}^{4 \times 3} \):

\[
X_\Omega = \begin{bmatrix}
0 \times 0 \\
\times 0 \times \\
0 \times \times \\
\times 0 \times 
\end{bmatrix}
\]

For \( j = 1 \), the \( (2, 1) \) and \( (4, 1) \) entries are observed, and thus \( I_1 = \{2, 4\} \). Similarly, \( I_2 = \{1, 3\} \) and \( I_3 = \{2, 3, 4\} \). Combining the results yields \( \sum_{j=1}^{n_2} |I_j| = |\Omega| \).
Define $\mathbf{U}_{\mathcal{I}_j} \in \mathbb{R}^{\left|\mathcal{I}_j\right| \times r}$ containing the $\left|\mathcal{I}_j\right|$ rows indexed by $\mathcal{I}_j$:

$$
\mathbf{U}_{\mathcal{I}_j} = 
\begin{bmatrix}
\mathbf{u}_{j_1}^T \\
\vdots \\
\mathbf{u}_{j_{|\mathcal{I}_j|}}^T
\end{bmatrix}
$$

and $\mathbf{b}_{\mathcal{I}_j} = [\mathbf{X}_{j_{1j}}, \ldots, \mathbf{X}_{j_{|\mathcal{I}_j|j}}]^T \in \mathbb{R}^{\left|\mathcal{I}_j\right|}$, then we obtain:

$$
\min_{\mathbf{v}_j} f_p(\mathbf{v}_j) := \|\mathbf{U}_{\mathcal{I}_j} \mathbf{v}_j - \mathbf{b}_{\mathcal{I}_j}\|_p^p
$$

which is a robust linear regression in $\ell_p$-space.

For $p = 2$, it is a least squares (LS) problem with solution being $\mathbf{v}_j = \mathbf{U}_{\mathcal{I}_j}^\dagger \mathbf{b}_{\mathcal{I}_j}$, and the corresponding computational complexity is $\mathcal{O}(\left|\mathcal{I}_j\right| r^2)$. 
For $0 < p < 2$, the $\ell_p$-regression can be efficiently solved by the iteratively reweighted least squares (IRLS). At the $t$th iteration, the IRLS solves the following weighted LS problem:

$$v_j^{t+1} = \arg\min_{v_i} \left\| W^t(U_{I_j}v_j - b_{I_j}) \right\|_2^2$$

where $W^t = \text{diag}\{w_1^t, \cdots, w_{n_1}^t\}$ with

$$w_i^t = \frac{1}{(|\xi_i^t|^2 + \epsilon)^{1-p/2}}$$

The $\xi_i^t$ is the $i$th element of $\xi^t = U_{I_j}v_j^t - b_{I_j}$ and $\epsilon > 0$. As only one LS problem is required to solve in each IRLS iteration, its complexity is $O(|I_j|^2 N_{IRLS})$. Hence the total complexity for all $n_2 \ell_p$-regressions is $O(|\Omega|^2 N_{IRLS})$ due to $\sum_{j=1}^{n_2} |I_j| = |\Omega|$. 
Due to the same structure in $U^{k+1} = \arg \min_U \left\| (UV^{k+1})_{\Omega} - X_{\Omega} \right\|_p^p$,

The $i$th row of $U$ is updated by

$$\min_{u_i^T} \left\| u_i^T V^{k+1}_{J_i} - b_{J_i}^T \right\|_p^p$$

where $J_i = \{i_1, \ldots, i_{|J_i|}\} \subseteq \{1, \ldots, n_2\}$ is the set containing the column indices for the $i$th row in $\Omega$.

Using previous example, only $(1,2)$ entry is observed for $i = 1$, and thus $J_1 = \{2\}$. Similarly, $J_2 = \{1,3\}$, $J_3 = \{2,3\}$ and $J_4 = \{1,3\}$. Here, $V_{J_i}^{k+1} \in \mathbb{R}^{r \times |J_i|}$ contains $|J_i|$ columns indexed by $J_i$ and $b_{J_i}^T = [X_{i1}, \ldots, X_{i|J_i|}]^T \in \mathbb{R}^{|J_i|}$. The involved complexity is $O(|J_i|r^2N_{IRLS})$ and hence the total complexity for solving all $n_1 \ell_p$-regressions is $O(|\Omega|r^2N_{IRLS})$ due to $\sum_{i=1}^{n_1} |J_i| = |\Omega|$.
Algorithm 1 Iterative $\ell_p$-Regression for Robust Matrix Completion

\textbf{Input:} $X_\Omega$, $\Omega$, and rank $r$

\textbf{Initialize:} Randomly initialize $U^0 \in \mathbb{R}^{n_1 \times r}$

Determine $\{I_j\}_{j=1}^{n_2}$ and $\{J_i\}_{i=1}^{n_1}$ according to $\Omega$.

\textbf{for} $k = 0, 1, \cdots$ \textbf{do}

\hspace{1em} // Fix $U^k$, optimize $V$

\hspace{2em} \textbf{for} $j = 1, 2, \cdots, n_2$ \textbf{do}

\hspace{3em} $v_j^{k+1} \leftarrow \arg \min_{v_j} \| U^k_{I_j} v_j - b_{I_j} \|^p$

\hspace{2em} \textbf{end for}

\hspace{1em} // Fix $V^{k+1}$, optimize $U$

\hspace{2em} \textbf{for} $i = 1, 2, \cdots, n_1$ \textbf{do}

\hspace{3em} $(u_i^T)^{k+1} \leftarrow \arg \min_{u_i^T} \| u_i^T V^{k+1}_{J_i} - b_{J_i}^T \|^p$

\hspace{2em} \textbf{end for}

\hspace{1em} \textbf{Stop} if a termination condition is satisfied.

\textbf{end for}

\textbf{Output:} $M = U^{k+1} V^{k+1}$
ADMM

Assign:

\[ E_\Omega = (UV)_\Omega - X_\Omega \]

The proposed robust formulation is then equivalent to:

\[
\min_{U,V,E_\Omega} \|E_\Omega\|_p^p, \quad \text{s.t.} \quad E_\Omega = (UV)_\Omega - X_\Omega
\]

Its augmented Lagrangian is:

\[
\mathcal{L}_\mu(U, V, E_\Omega, \Lambda_\Omega) = \|E_\Omega\|_p^p + \langle \Lambda_\Omega, (UV)_\Omega - E_\Omega - X_\Omega \rangle + \frac{\mu}{2} \| (UV)_\Omega - E_\Omega - X_\Omega \|_F^2
\]

where \( \Lambda_\Omega \in \mathbb{R}^{n_1 \times n_2} \) with \( [\Lambda_\Omega]_{ij} = 0 \) for \( (i, j) \not\in \Omega \) contains \( |\Omega| \) Lagrange multipliers.
The Lagrange multiplier method aims to find a saddle point of:

$$\max_{\Lambda_\Omega} \min_{U, V, E_\Omega} \mathcal{L}_\mu(U, V, E_\Omega, \Lambda_\Omega)$$

The solution is obtained by applying the ADMM via the following iterative steps:

$$(U^{k+1}, V^{k+1}) = \arg\min_{U, V} \mathcal{L}_\mu(U, V, E_\Omega^k, \Lambda_\Omega^k)$$

$$E_\Omega^{k+1} = \arg\min_{E_\Omega} \mathcal{L}_\mu(U^{k+1}, V^{k+1}, E_\Omega, \Lambda_\Omega^k)$$

$$\Lambda_\Omega^{k+1} = \Lambda_\Omega^k + \mu ((U^{k+1}V^{k+1})_\Omega - E_\Omega^{k+1} - X_\Omega)$$
Ignoring the constant term independent of \((U, V)\), it is shown that

\[
(U^{k+1}, V^{k+1}) = \arg \min_{U, V} \mathcal{L}_\mu(U, V, E^k_\Omega, \Lambda^k_\Omega)
\]

is equivalent to:

\[
\min_{U, V} \left\| (UV)_\Omega - \left( E^k_\Omega - \frac{\Lambda^k_\Omega}{\mu} + X_\Omega \right) \right\|^2_F
\]

which can be solved by Algorithm 1 with \(p = 2\), with a complexity bound of \(\mathcal{O}(K_{\ell_2}|\Omega|r^2)\), where \(K_{\ell_2}\) is the required iteration number.
For the problem of

\[ E^{k+1}_\Omega = \arg \min_{E_\Omega} L_\mu(U^{k+1}, V^{k+1}, E_\Omega, \Lambda^k_\Omega) \]

It can be simplified as:

\[ \min_{E_\Omega} \frac{1}{2} \| E_\Omega - Y^k_\Omega \|_F^2 + \frac{1}{\mu} \| E_\Omega \|_p^p \]

where

\[ Y^k_\Omega = (U^{k+1} V^{k+1})_\Omega + \frac{\Lambda^k_\Omega}{\mu} - X_\Omega \]

We only need to consider the entries indexed by \( \Omega \) because other entries of \( E_\Omega \) and \( Y^k_\Omega \) which are not in \( \Omega \) are zero.
Defining $e_\Omega$, $y^k_\Omega$, $\lambda^k_\Omega$, and $t^k_\Omega \in \mathbb{R}^{|\Omega|}$ as the vectors that contain the observed entries in $E_\Omega$, $Y^k_\Omega$, $\Lambda^k_\Omega$, and $(U^k V^k)_\Omega$, we have the equivalent vector optimization problem:

$$
\min_{e_\Omega} \frac{1}{2} \|e_\Omega - y^k_\Omega\|_2^2 + \frac{1}{\mu} \|e_\Omega\|_p^p
$$

whose solution can be written in proximity operator:

$$
e^{k+1}_\Omega = \text{prox}_{1/\mu}(y^k_\Omega)
$$

Denoting $e_i$ and $y_i$, $i = 1, \cdots, |\Omega|$, as the $i$th entry of $e$ and $y$, and noting the separability of the problem, we solve $|\Omega|$ independent scalar problems instead:

$$
\min_{e_i \in \mathbb{R}} g(e_i) := \frac{1}{2}(e_i - y_i)^2 + \frac{1}{\mu}|e_i|^p, \ i = 1, \cdots, |\Omega|
$$
For $p = 1$, closed-form solution exists:

$$e_i^* = \text{sgn}(y_i) \max(|y_i| - 1/\mu, 0)$$

with a marginal complexity of $\mathcal{O}(|\Omega|)$.

For $p < 1$, the solution of the scalar minimization problem is:

$$e_i^* = \begin{cases} 0, & \text{if } |y_i| \leq \tau \\ \arg\min_{e_i \in \{0, t_i\}} g(e_i), & \text{if } |y_i| > \tau \end{cases}, \tau = \left(\frac{p(1-p)}{\mu}\right)^{\frac{1}{2-p}} + \frac{p}{\mu} \left(\frac{p(1-p)}{\mu}\right)^{\frac{p-1}{2-p}}$$

where $t_i = \text{sgn}(y_i)r_i$ with $r_i$ being the unique root of:

$$h(\theta) := \theta + \frac{p}{\mu} \theta^{p-1} - |y_i| = 0$$

in $\left[(p(1-p)/\mu)^{\frac{1}{2-p}}, |y_i|\right]$ and the bisection method can be used.
Although computing the proximity operator for $p < 1$ still has a complexity of $\mathcal{O}(|\Omega|)$, it is more complicated than $p = 1$ because there is no closed-form solution.

On the other hand, the solution for the case of $p \in (1, 2)$ can be obtained in a similar manner. Again, there is no closed-form solution and calculating the proximity operator for $1 < p < 2$ has a complexity of $\mathcal{O}(|\Omega|)$ although an iterative procedure for root finding is required.

Note that the choice of $p = 1$ is more robust than employing $p \in (1, 2)$ and is computationally simpler.
For

$$\Lambda_{\Omega}^{k+1} = \Lambda_{\Omega}^k + \mu \left( (U_{\Omega}^{k+1}V_{\Omega}^{k+1}) - E_{\Omega}^{k+1} - X_{\Omega} \right)$$

It is converted in vector form:

$$\lambda_{\Omega}^{k+1} = \lambda_{\Omega}^k + \mu \left( t_{\Omega}^{k+1} - e_{\Omega}^{k+1} - x_{\Omega} \right)$$

whose complexity is $\mathcal{O}(|\Omega|)$.

Note that at each iteration, $(UV)_{\Omega}$ instead of $UV$ is needed to compute, whose complexity is $\mathcal{O}(|\Omega|r)$ because only $|\Omega|$ inner products $\{u_i^T v_j\}_{(i,j)\in\Omega}$ are calculated.

The algorithm is terminated when

$$\|t_{\Omega}^k - e_{\Omega}^k - x_{\Omega}\|_2 < \delta, \quad \delta > 0$$
**Algorithm 2** ADMM for Robust Matrix Completion

**Input:** $X_\Omega$, $\Omega$, and rank $r$

**Initialize:** $e^0 = 0$ and $\lambda^0 = 0$

**for** $k = 0, 1, \cdots$ **do**

1) Solve LS matrix factorization

$$(U^{k+1}, V^{k+1}) = \arg \min_{U, V} \left\| (UV)_{\Omega} - \left( E^k_{\Omega} - \Lambda^k_{\Omega}/\mu + X_\Omega \right) \right\|^2_F$$

using Algorithm 1 with $p = 2$.

2) Compute $Y^k_{\Omega} = (U^{k+1}V^{k+1})_{\Omega} + \Lambda^k_{\Omega}/\mu - X_\Omega$ and form $y^k_{\Omega}$ and $t^{k+1}_{\Omega} \leftarrow (U^{k+1}V^{k+1})_{\Omega}$.

3) $e^{k+1}_{\Omega} \leftarrow \text{prox}_{1/\mu}(y^k_{\Omega})$

4) $\lambda^{k+1}_{\Omega} \leftarrow \lambda^k_{\Omega} + \mu \left( t^{k+1}_{\Omega} - e^{k+1}_{\Omega} - x_\Omega \right)$

**Stop** if a termination condition is satisfied.

**end for**

**Output:** $M = U^{k+1}V^{k+1}$
Numerical Examples

\( X \in \mathbb{R}^{n_1 \times n_2} \) is generated by multiplying \( X_1 \in \mathbb{R}^{n_1 \times r} \) and \( X_2 \in \mathbb{R}^{r \times n_2} \) whose entries are standard Gaussian distribution. 45% entries of \( X \) are randomly selected as observations.

\( n_1 = 150, \ n_2 = 300 \) and \( r = 10 \).

Performance measure is:

\[
\text{RMSE}(\hat{M}) = \sqrt{\mathbb{E} \left\{ \frac{\|\hat{M} - X\|^2_F}{\|X\|^2_F} \right\}}
\]

CPU times for attaining \( \text{RMSE} < 10^{-5} \) of SVT, SVP, \( \ell_p \) - regression with \( p = 2 \) and \( p = 1 \) and ADMM with \( p = 1 \) are 10.7s, 8.0s, 0.28s, 4.5s, and 0.28s, respectively.
RMSE versus iteration number in noise-free case
RMSE versus iteration number in GMM noise at SNR=6dB
RMSE versus SNR

- SVT
- SVP
- $\ell_p$-reg ($p = 2$)
- $\ell_p$-reg ($p = 1$)
- ADMM ($p = 1$)
Results of image inpainting in salt-and-pepper noise
Concluding Remarks

- Two algorithms for robust matrix completion using low-rank factorization via $\ell_p$-norm minimization with $0 < p \leq 2$ are devised.

- The first tackles the nonconvex factorization with missing data by iteratively solving multiple independent linear $\ell_p$-regressions.

- The second applies ADMM in $\ell_p$-space: At each iteration, it requires solving a LS matrix factorization problem and calculating proximity operator of the $p$th power of $\ell_p$-norm. The LS factorization can be efficiently solved using linear LS regression while the proximity operator has
closed-form solution for $p = 1$ or can be obtained by root finding of a scalar nonlinear equation for $p \neq 1$.

- Both are based on alternating optimization, and have comparable recovery performance and computational complexity of $\mathcal{O}(K|\Omega|r^2)$ where $K$ is a fixed constant of several hundreds to thousands.

- Their superiority over the SVT and SVP in terms of implementation complexity, recovery capability and outlier-robustness is demonstrated.
List of References


