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

W.-J.Zeng and H.C.So, "Outlier-robust matrix completion via l_p-minimization," *IEEE Transactions on Signal Processing*, vol.66, no.5, pp.1125-1140, March 2018

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 $\boldsymbol{X}_\Omega \in \mathbb{R}^{n_1 \times n_2}$ be a matrix with missing entries:

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

where Ω 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 $\boldsymbol{M} \in \mathbb{R}^{n_1 \times n_2}$, given the incomplete observations $\boldsymbol{X}_\Omega$ with the low-rank information of $\boldsymbol{X}$, which can be mathematically formulated as:

$$\min_{\boldsymbol{M}} \ \text{rank}(\boldsymbol{M}), \quad \text{s.t. } \boldsymbol{M}_\Omega = \boldsymbol{X}_\Omega$$

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.

...

Alice	1			4	
Bob		2	5		
Carol			4	5	
Dave	5				4

⋮

How to Recover an Incomplete Matrix?

Directly solving the noise-free version:

$$\min_{\boldsymbol{M}} \ \text{rank}(\boldsymbol{M}), \quad \text{s.t. } \boldsymbol{M}_\Omega = \boldsymbol{X}_\Omega$$

or noisy version:

$$\min_{\boldsymbol{M}} \ \text{rank}(\boldsymbol{M}), \quad \text{s.t. } \|\boldsymbol{M}_\Omega - \boldsymbol{X}_\Omega\|_F \leq \epsilon_F$$

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:

$$\min_{\boldsymbol{M}} \|\boldsymbol{M}\|_*, \quad \text{s.t. } \boldsymbol{M}_\Omega = \boldsymbol{X}_\Omega$$

or

$$\min_{\boldsymbol{M}} \|\boldsymbol{M}\|_*, \quad \text{s.t. } \|\boldsymbol{M}_\Omega - \boldsymbol{X}_\Omega\|_F \leq \epsilon_F$$

where $\|\boldsymbol{M}\|_*$ equals the sum of singular values of $\boldsymbol{M}$. However, complexity of nuclear norm minimization is still high and this approach is not robust if $\boldsymbol{X}_\Omega$ contains outliers.

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

$$\min_{\boldsymbol{U},\boldsymbol{V}} f_2(\boldsymbol{U},\boldsymbol{V}) := \|(\boldsymbol{U}\boldsymbol{V})_\Omega - \boldsymbol{M}_\Omega\|_F^2$$

where $\boldsymbol{U} \in \mathbb{R}^{n_1 \times r}$ and $\boldsymbol{V} \in \mathbb{R}^{r \times n_2}$. 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 ℓ_p-norm where $0 < p \le 2$:

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

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

$$\|\boldsymbol{X}_\Omega\|_p = \left(\sum_{(i,j)\in\Omega} |\boldsymbol{X}_{ij}|^p \right)^{1/p}$$

Iterative l_p-Regression

To ℓ_p-norm minimization, our first idea is to adopt the alternating minimization strategy:

$$\boldsymbol{V}^{k+1} = \arg\min_{\boldsymbol{V}} \ \|(\boldsymbol{U}^k\boldsymbol{V})_\Omega - \boldsymbol{X}_\Omega\|_p^p$$

and

$$\boldsymbol{U}^{k+1} = \arg\min_{\boldsymbol{U}} \ \|(\boldsymbol{U}\boldsymbol{V}^{k+1})_\Omega - \boldsymbol{X}_\Omega\|_p^p$$

where the algorithm is initialized with $\boldsymbol{U}^0$, and $\boldsymbol{U}^k$ represents the estimate of $\boldsymbol{U}$ at the kth iteration.

After determining $\boldsymbol{U}$ and $\boldsymbol{V}$, the target matrix is obtained as $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{V}$.

We now focus on solving:

$$\min_{\boldsymbol{V}} \ f_p(\boldsymbol{V}) := \|(\boldsymbol{U}\boldsymbol{V})_\Omega - \boldsymbol{X}_\Omega\|_p^p$$

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

Denoting the ith row of $\boldsymbol{U}$ and the jth column of $\boldsymbol{V}$ as $\boldsymbol{u}_i^T$ and $\boldsymbol{v}_j$, where $\boldsymbol{u}_i, \boldsymbol{v}_j \in \mathbb{R}^r$, $i = 1, \cdots, n_1$, $j = 1, \cdots, n_2$, the problem can be rewritten as:

$$\min_{\boldsymbol{V}} \ f_p(\boldsymbol{V}) := \sum_{(i,j)\in\Omega} |\boldsymbol{u}_i^T \boldsymbol{v}_j - \boldsymbol{X}_{ij}|^p$$

Since $f_p(\boldsymbol{V})$ is decoupled w.r.t. $\boldsymbol{v}_j$, it is equivalent to solving the following n_2 independent subproblems:

$$\min_{\boldsymbol{v}_j} \; f_p(\boldsymbol{v}_j) := \sum_{i \in \mathcal{I}_j} |\boldsymbol{u}_i^T \boldsymbol{v}_j - \boldsymbol{X}_{ij}|^p, \; j = 1, \cdots, n_2$$

where $\mathcal{I}_j = \{j_1, \cdots, j_{|\mathcal{I}_j|}\} \subseteq \{1, \cdots, n_1\}$ denotes the set containing the row indices for the jth column in Ω. Here, $|\mathcal{I}_j|$ stands for the cardinality of $\mathcal{I}_j$ and in general $|\mathcal{I}_j| > r$.

For example, consider $\boldsymbol{X}_\Omega \in \mathbb{R}^{4\times 3}$:

$$\boldsymbol{X}_\Omega = \begin{bmatrix} 0 & \times & 0 \\ \times & 0 & \times \\ 0 & \times & \times \\ \times & 0 & \times \end{bmatrix}$$

For $j = 1$, the $(2,1)$ and $(4,1)$ entries are observed, and thus $\mathcal{I}_1 = \{2, 4\}$. Similarly, $\mathcal{I}_2 = \{1, 3\}$ and $\mathcal{I}_3 = \{2, 3, 4\}$. Combining the results yields $\sum_{j=1}^{n_2} |\mathcal{I}_j| = |\Omega|$.

Define $\boldsymbol{U}_{\mathcal{I}_j} \in \mathbb{R}^{|\mathcal{I}_j| \times r}$ containing the $|\mathcal{I}_j|$ rows indexed by $\mathcal{I}_j$:

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

and $\boldsymbol{b}_{\mathcal{I}_j} = [\boldsymbol{X}_{j_1 j}, \cdots, \boldsymbol{X}_{j_{|\mathcal{I}_j|} j}]^T \in \mathbb{R}^{|\mathcal{I}_j|}$, then we obtain:

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

which is a robust linear regression in ℓ_p-space.

For $p = 2$, it is a least squares (LS) problem with solution being $\boldsymbol{v}_j = \boldsymbol{U}_{\mathcal{I}_j}^{\dagger} \boldsymbol{b}_{\mathcal{I}_j}$, and the corresponding computational complexity is $\mathcal{O}(|\mathcal{I}_j| r^2)$.

For $0 < p < 2$, the ℓ_p-regression can be efficiently solved by the iteratively reweighted least squares (IRLS). At the tth iteration, the IRLS solves the following weighted LS problem:

$$\boldsymbol{v}_j^{t+1} = \arg\min_{\boldsymbol{v}_j} \ \|\boldsymbol{W}^t(\boldsymbol{U}_{\mathcal{I}_j}\boldsymbol{v}_j - \boldsymbol{b}_{\mathcal{I}_j})\|_2^2$$

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

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

The ξ_i^t is the ith element of $\boldsymbol{\xi}^t = \boldsymbol{U}_{\mathcal{I}_j}\boldsymbol{v}_j^t - \boldsymbol{b}_{\mathcal{I}_j}$ and $\epsilon > 0$. As only one LS problem is required to solve in each IRLS iteration, its complexity is $\mathcal{O}(|\mathcal{I}_j|r^2 N_{\mathrm{IRLS}})$. Hence the total complexity for all n_2 ℓ_p-regressions is $\mathcal{O}(|\Omega|r^2 N_{\mathrm{IRLS}})$ due to $\sum_{j=1}^{n_2} |\mathcal{I}_j| = |\Omega|$.

Due to the same structure in $\boldsymbol{U}^{k+1} = \arg\min_{\boldsymbol{U}} \ \|(\boldsymbol{U}\boldsymbol{V}^{k+1})_\Omega - \boldsymbol{X}_\Omega\|_p^p$,

The ith row of $\boldsymbol{U}$ is updated by

$$\min_{\boldsymbol{u}_i^T} \ \|\boldsymbol{u}_i^T \boldsymbol{V}_{\mathcal{J}_i}^{k+1} - \boldsymbol{b}_{\mathcal{J}_i}^T\|_p^p$$

where $\mathcal{J}_i = \{i_1, \cdots, i_{|\mathcal{J}_i|}\} \subseteq \{1, \cdots, n_2\}$ is the set containing the column indices for the ith row in Ω.

Using previous example, only $(1,2)$ entry is observed for $i=1$, and thus $\mathcal{J}_1 = \{2\}$. Similarly, $\mathcal{J}_2 = \{1,3\}$, $\mathcal{J}_3 = \{2,3\}$ and $\mathcal{J}_4 = \{1,3\}$. Here, $\boldsymbol{V}_{\mathcal{J}_i}^{k+1} \in \mathbb{R}^{r \times |\mathcal{J}_i|}$ contains $|\mathcal{J}_i|$ columns indexed by $\mathcal{J}_i$ and $\boldsymbol{b}_{\mathcal{J}_i}^T = [\boldsymbol{X}_{ii_1}, \cdots, \boldsymbol{X}_{ii_{|\mathcal{J}_i|}}]^T \in \mathbb{R}^{|\mathcal{J}_i|}$. The involved complexity is $\mathcal{O}(|\mathcal{J}_i| r^2 N_{\text{IRLS}})$ and hence the total complexity for solving all n_1 ℓ_p-regressions is $\mathcal{O}(|\Omega| r^2 N_{\text{IRLS}})$ due to $\sum_{i=1}^{n_1} |\mathcal{J}_i| = |\Omega|$.

Algorithm 1 Iterative ℓ_p-Regression for Robust Matrix Completion

Input: $\boldsymbol{X}_\Omega$, Ω, and rank r

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

Determine $\{\mathcal{I}_j\}_{j=1}^{n_2}$ and $\{\mathcal{J}_i\}_{i=1}^{n_1}$ according to Ω.

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

// Fix $\boldsymbol{U}^k$, optimize $\boldsymbol{V}$

for $j = 1, 2, \cdots, n_2$ **do**

$\boldsymbol{v}_j^{k+1} \leftarrow \arg\min_{\boldsymbol{v}_j} \|\boldsymbol{U}_{\mathcal{I}_j}^k \boldsymbol{v}_j - \boldsymbol{b}_{\mathcal{I}_j}\|_p^p$

end for

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

for $i = 1, 2, \cdots, n_1$ **do**

$(\boldsymbol{u}_i^T)^{k+1} \leftarrow \arg\min_{\boldsymbol{u}_i^T} \|\boldsymbol{u}_i^T \boldsymbol{V}_{\mathcal{J}_i}^{k+1} - \boldsymbol{b}_{\mathcal{J}_i}^T\|_p^p$

end for

Stop if a termination condition is satisfied.

end for

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

ADMM

Assign:

$$\boldsymbol{E}_\Omega = (\boldsymbol{U}\boldsymbol{V})_\Omega - \boldsymbol{X}_\Omega$$

The proposed robust formulation is then equivalent to:

$$\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{E}_\Omega} \|\boldsymbol{E}_\Omega\|_p^p, \quad \text{s.t. } \boldsymbol{E}_\Omega = (\boldsymbol{U}\boldsymbol{V})_\Omega - \boldsymbol{X}_\Omega$$

Its augmented Lagrangian is:

$$\begin{aligned}\mathcal{L}_\mu(\boldsymbol{U},\boldsymbol{V},\boldsymbol{E}_\Omega,\boldsymbol{\Lambda}_\Omega) = & \|\boldsymbol{E}_\Omega\|_p^p + \langle \boldsymbol{\Lambda}_\Omega, (\boldsymbol{U}\boldsymbol{V})_\Omega - \boldsymbol{E}_\Omega - \boldsymbol{X}_\Omega \rangle \\ & + \frac{\mu}{2}\|(\boldsymbol{U}\boldsymbol{V})_\Omega - \boldsymbol{E}_\Omega - \boldsymbol{X}_\Omega\|_F^2\end{aligned}$$

where $\boldsymbol{\Lambda}_\Omega \in \mathbb{R}^{n_1 \times n_2}$ with $[\boldsymbol{\Lambda}_\Omega]_{ij} = 0$ for $(i,j) \notin \Omega$ contains $|\Omega|$ Lagrange multipliers.

The Lagrange multiplier method aims to find a saddle point of:

$$\max_{\boldsymbol{\Lambda}_\Omega} \min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{E}_\Omega} \mathcal{L}_\mu(\boldsymbol{U},\boldsymbol{V},\boldsymbol{E}_\Omega,\boldsymbol{\Lambda}_\Omega)$$

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

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

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

$$\boldsymbol{\Lambda}_\Omega^{k+1} = \boldsymbol{\Lambda}_\Omega^k + \mu\left((\boldsymbol{U}^{k+1}\boldsymbol{V}^{k+1})_\Omega - \boldsymbol{E}_\Omega^{k+1} - \boldsymbol{X}_\Omega\right)$$

Ignoring the constant term independent of $(\boldsymbol{U}, \boldsymbol{V})$, it is shown that

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

is equivalent to:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \left\| (\boldsymbol{U}\boldsymbol{V})_\Omega - \left(\boldsymbol{E}_\Omega^k - \frac{\boldsymbol{\Lambda}_\Omega^k}{\mu} + \boldsymbol{X}_\Omega \right) \right\|_F^2$$

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_{ℓ_2} is the required iteration number.

For the problem of

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

It can be simplified as:

$$\min_{\boldsymbol{E}_\Omega} \frac{1}{2} \left\| \boldsymbol{E}_\Omega - \boldsymbol{Y}_\Omega^k \right\|_F^2 + \frac{1}{\mu} \|\boldsymbol{E}_\Omega\|_p^p$$

where

$$\boldsymbol{Y}_\Omega^k = (\boldsymbol{U}^{k+1}\boldsymbol{V}^{k+1})_\Omega + \frac{\boldsymbol{\Lambda}_\Omega^k}{\mu} - \boldsymbol{X}_\Omega$$

We only need to consider the entries indexed by Ω because other entries of $\boldsymbol{E}_\Omega$ and $\boldsymbol{Y}_\Omega^k$ which are not in Ω are zero.

Defining $\boldsymbol{e}_\Omega$, $\boldsymbol{y}_\Omega^k$, $\boldsymbol{\lambda}_\Omega^k$, and $\boldsymbol{t}_\Omega^k \in \mathbb{R}^{|\Omega|}$ as the vectors that contain the observed entries in $\boldsymbol{E}_\Omega$, $\boldsymbol{Y}_\Omega^k$, $\boldsymbol{\Lambda}_\Omega^k$, and $(\boldsymbol{U}^k\boldsymbol{V}^k)_\Omega$, we have the equivalent vector optimization problem:

$$\min_{\boldsymbol{e}_\Omega} \frac{1}{2}\left\|\boldsymbol{e}_\Omega - \boldsymbol{y}_\Omega^k\right\|_2^2 + \frac{1}{\mu}\|\boldsymbol{e}_\Omega\|_p^p$$

whose solution can be written in proximity operator:

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

Denoting e_i and y_i, $i = 1, \cdots, |\Omega|$, as the ith entry of $\boldsymbol{e}$ and $\boldsymbol{y}$, 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:

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

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

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

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

where $t_i = \mathrm{sgn}(y_i) r_i$ 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

$$\boldsymbol{\Lambda}_{\Omega}^{k+1} = \boldsymbol{\Lambda}_{\Omega}^{k} + \mu \left((\boldsymbol{U}^{k+1}\boldsymbol{V}^{k+1})_{\Omega} - \boldsymbol{E}_{\Omega}^{k+1} - \boldsymbol{X}_{\Omega} \right)$$

It is converted in vector form:

$$\boldsymbol{\lambda}_{\Omega}^{k+1} = \boldsymbol{\lambda}_{\Omega}^{k} + \mu \left(\boldsymbol{t}_{\Omega}^{k+1} - \boldsymbol{e}_{\Omega}^{k+1} - \boldsymbol{x}_{\Omega} \right)$$

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

Note that at each iteration, $(\boldsymbol{U}\boldsymbol{V})_{\Omega}$ instead of $\boldsymbol{U}\boldsymbol{V}$ is needed to compute, whose complexity is $\mathcal{O}(|\Omega|r)$ because only $|\Omega|$ inner products $\{\boldsymbol{u}_i^T \boldsymbol{v}_j\}_{(i,j)\in\Omega}$ are calculated.

The algorithm is terminated when

$$\|\boldsymbol{t}_{\Omega}^{k} - \boldsymbol{e}_{\Omega}^{k} - \boldsymbol{x}_{\Omega}\|_2 < \delta, \quad \delta > 0$$

Algorithm 2 ADMM for Robust Matrix Completion

Input: $\boldsymbol{X}_\Omega$, Ω, and rank r

Initialize: $\boldsymbol{e}^0 = \boldsymbol{0}$ and $\boldsymbol{\lambda}^0 = \boldsymbol{0}$

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

1) Solve LS matrix factorization

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

using Algorithm 1 with $p = 2$.

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

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

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

Stop if a termination condition is satisfied.

end for

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

Numerical Examples

$\boldsymbol{X} \in \mathbb{R}^{n_1 \times n_2}$ is generated by multiplying $\boldsymbol{X}_1 \in \mathbb{R}^{n_1 \times r}$ and $\boldsymbol{X}_2 \in \mathbb{R}^{r \times n_2}$ whose entries are standard Gaussian distribution.

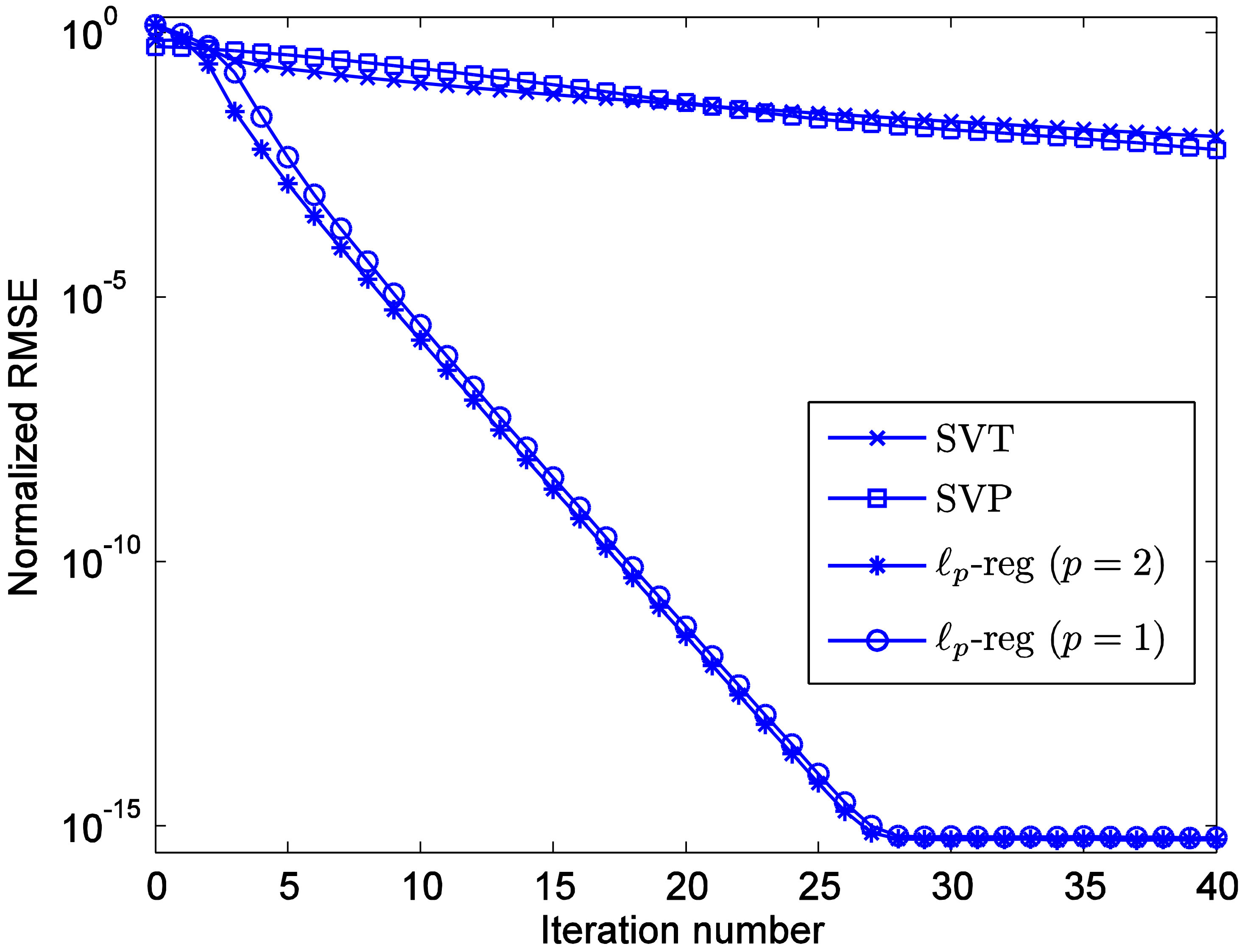
45% entries of $\boldsymbol{X}$ are randomly selected as observations.

$n_1 = 150$, $n_2 = 300$ and $r = 10$.

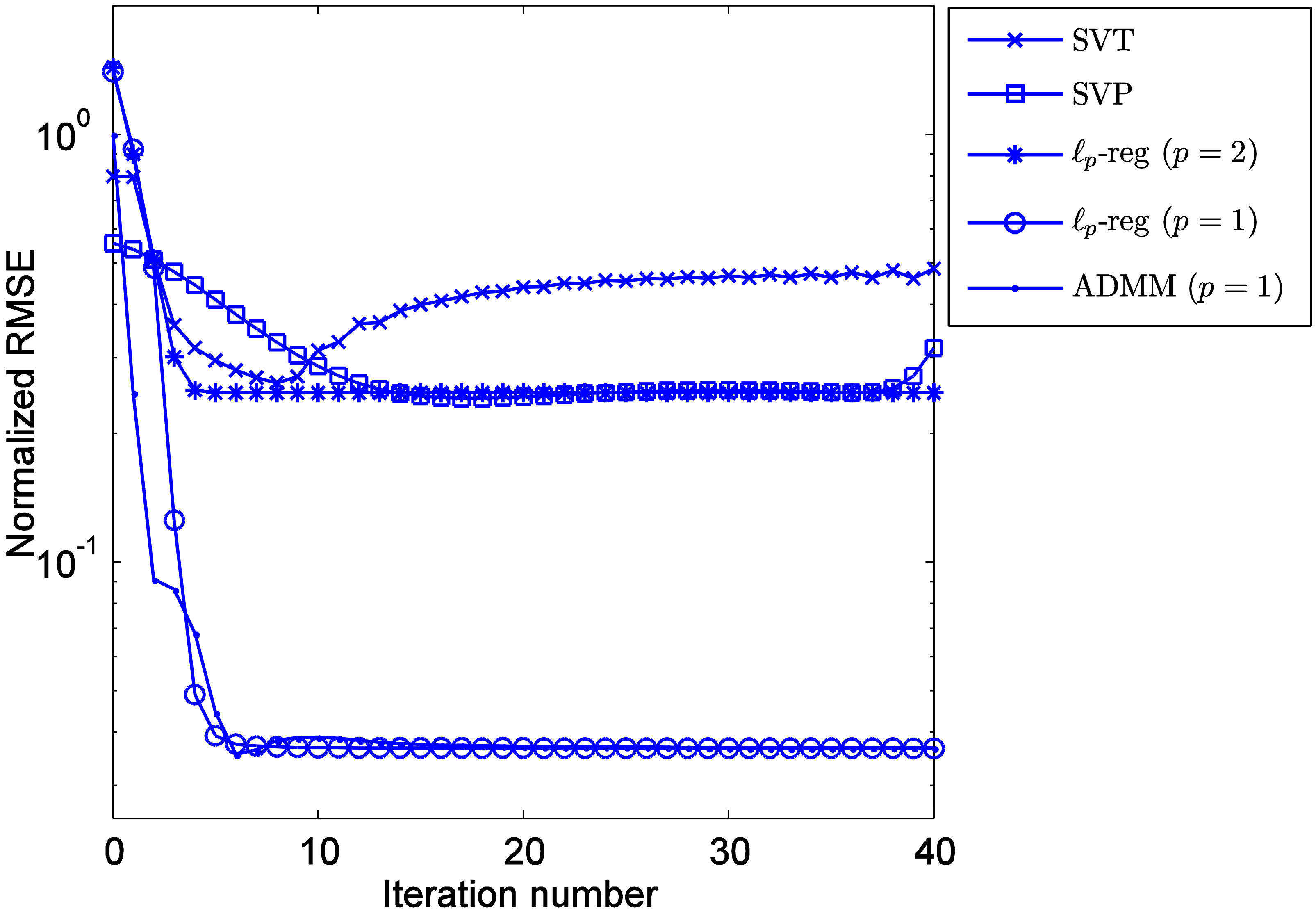
Performance measure is:

$$\mathrm{RMSE}(\widehat{\boldsymbol{M}}) = \sqrt{\mathrm{E}\left\{\frac{\|\widehat{\boldsymbol{M}} - \boldsymbol{X}\|_F^2}{\|\boldsymbol{X}\|_F^2}\right\}}$$

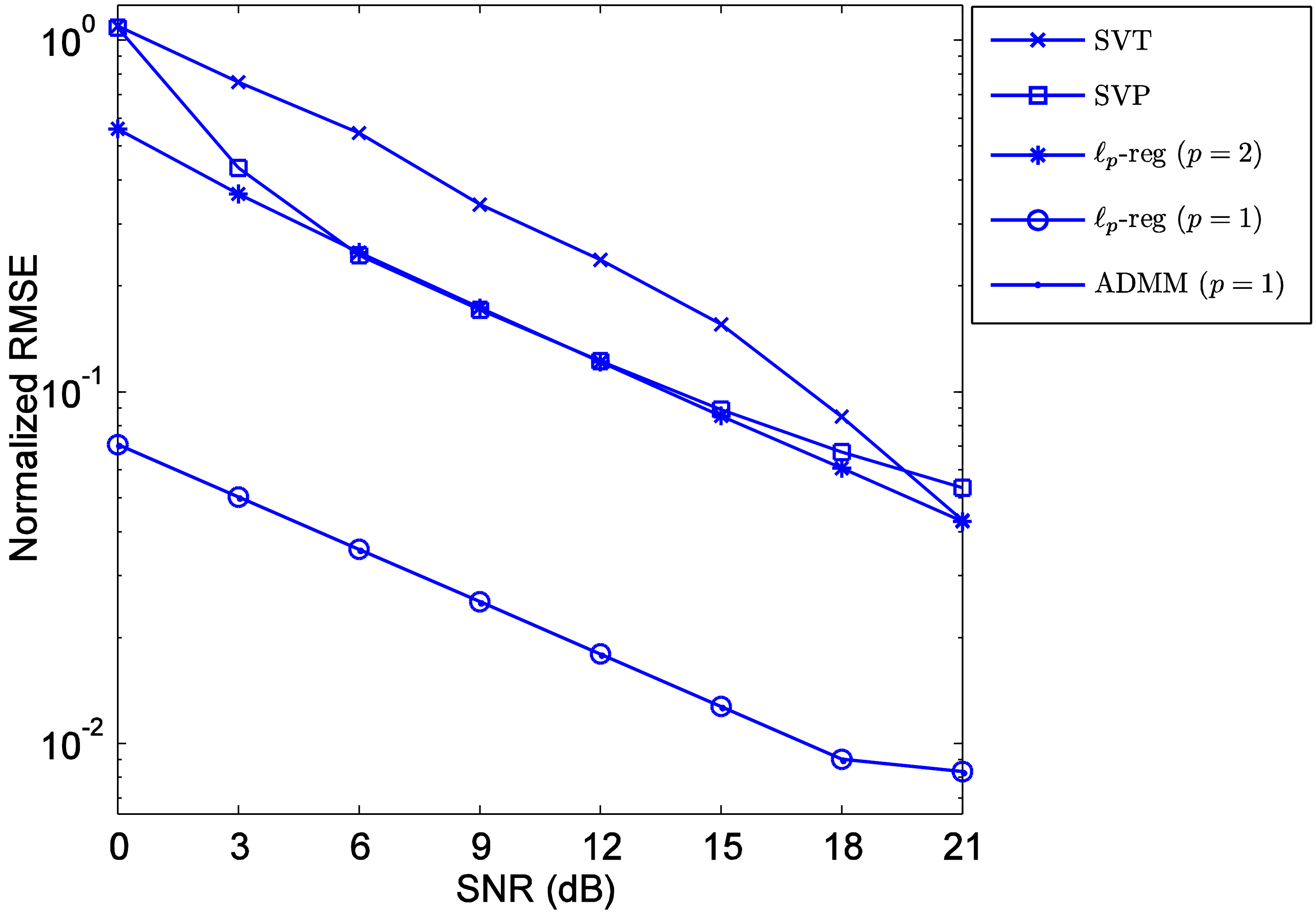
CPU times for attaining $\mathrm{RMSE} \leq 10^{-5}$ of SVT, SVP, ℓ_p-regression with $p = 2$ and $p = 1$ and ADMM with $p = 1$ are 10.7s, 8.0s, 0.28s, 4.5s, and 0.28s, respectively.



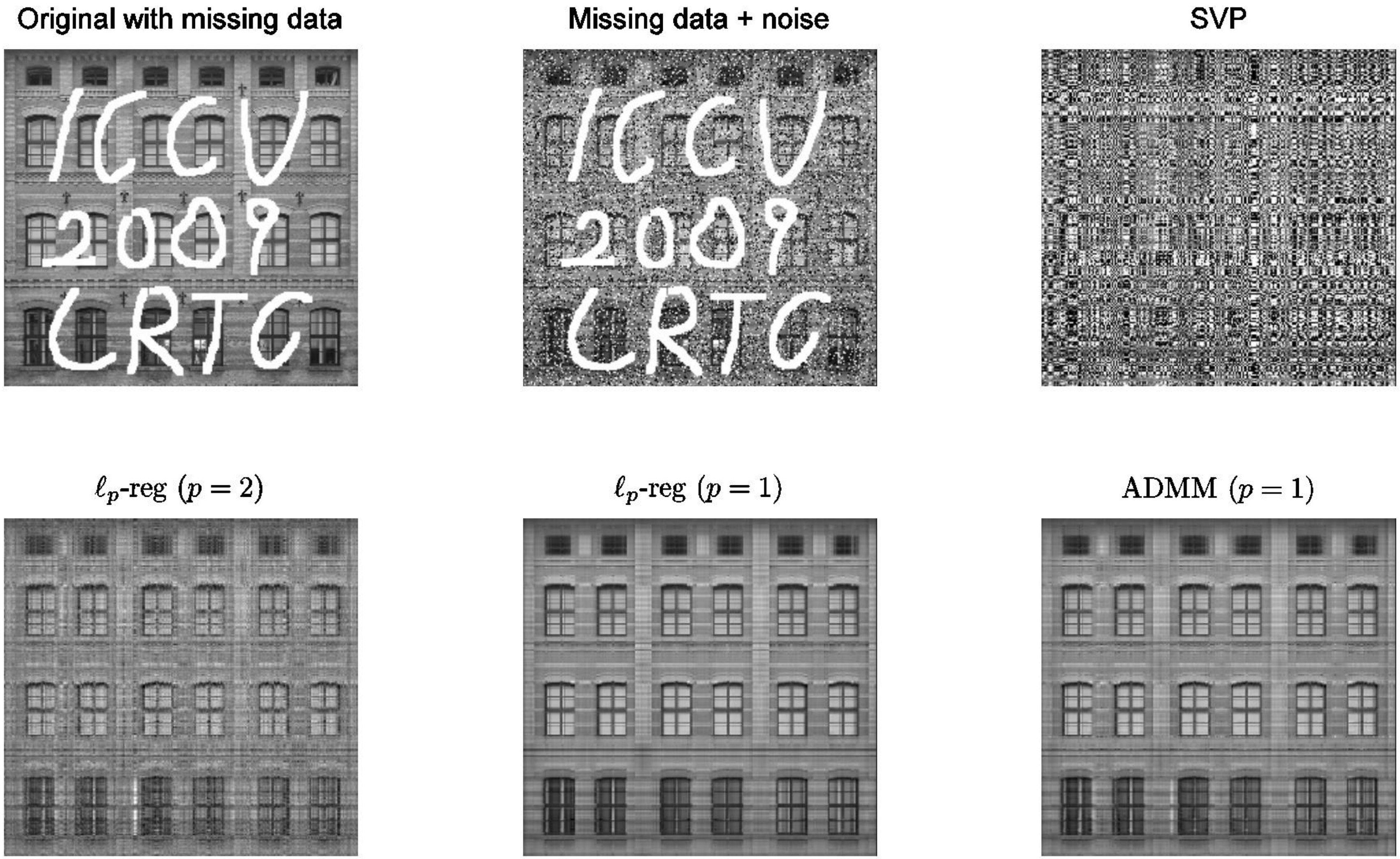
RMSE versus iteration number in noise-free case



RMSE versus iteration number in GMM noise at SNR=6dB



RMSE versus SNR



Results of image inpainting in salt-and-pepper noise

Concluding Remarks

- Two algorithms for robust matrix completion using low-rank factorization via ℓ_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 ℓ_p-regressions.

- The second applies ADMM in ℓ_p-space: At each iteration, it requires solving a LS matrix factorization problem and calculating proximity operator of the pth power of ℓ_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

[1] E. J. Candès and Y. Plan, "Matrix completion with noise," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 9255–936, Jun. 2010.

[2] J.-F. Cai, E. J. Candès and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[3] P. Jain, R. Meka and I. S. Dhillon, "Guaranteed rank minimization via singular value projection," *Advances in Neural Information Processing Systems* , pp. 937–945, 2010.

[4] Y. Koren, R. Bell and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[5] R. H. Keshavan, A. Montanari and S. Oh, "Matrix completion from a few entries," *IEEE Transactions on Information Theory*, vol. 56, no. 6, pp. 2980–2998, Jun. 2010.

[6] R. Sun and Z.-Q. Luo, "Guaranteed matrix completion via nonconvex factorization," *IEEE Transactions on Information Theory*, vol. 62, no. 11, pp. 6535–6579, Nov. 2016.

[7] G. Marjanovic and V. Solo, "On l_q optimization and matrix completion," *IEEE Transactions on Signal Processing*, vol. 60, no. 11, pp. 5714–5724, Nov. 2012.

[8] J. Liu, P. Musialski, P. Wonka and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 208–220, Jan. 2013.