# Robust Low-Rank Matrix Completion using Truncated-Quadratic Loss Function

**Speaker:** WANG Zhiyong

The third-year PhD student

**Email:** z.y.wang@my.cityu.edu.hk

**Supervisor:** Prof. SO Hing Cheung

3 Feb. 2022

# Outline

## What is Matrix Completion?

Matrix completion (MC) refers to restoring the missing entries of an incomplete matrix by making use of low-rank property. For example, find all ? entries in $\boldsymbol{X}_\Omega$

$$\boldsymbol{X}_\Omega = \begin{bmatrix} 1 & ? & ? & 4 & ? \\ ? & 2 & 5 & ? & ? \\ ? & ? & 4 & 5 & ? \\ 5 & ? & ? & ? & 4 \end{bmatrix} \Longrightarrow \Omega = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

where $\Omega$ is used to denote the index set of the observed entries of $\boldsymbol{X}$, and

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

# Introduction

## Why Matrix completion is Important?

It is a core problem in many applications including:

1. Recommender System
2. Image Inpainting
3. Hyperspectral Remote Sensing
4. Video Background Recovery
5. ...

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

This is, the dominant information of high-dimensional data is contained in its low-dimensional subspace.

# Introduction

Netflix Prize, whose goal was to 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.


Figure 1: Recommendation system.

# Introduction

Image is inherently a matrix whose entries are the pixel values. Image inpainting aims to restore a low-rank image with missing pixels possibly in the presence of noise:


Figure 2: Image inpainting.

# Introduction

## How to Recover an Incomplete Matrix?

Matrix completion aims to find $\boldsymbol{M} \in \mathbb{R}^{m \times n}$, given the incomplete observations $\boldsymbol{X}_\Omega$ with the low-rank property of $\boldsymbol{X}$, which can be formulated as:

$$\min_{\boldsymbol{M}} \ \operatorname{rank}(\boldsymbol{M}), \quad \text{s.t. } \|\boldsymbol{M}_\Omega - \boldsymbol{X}_\Omega\|_F \leq \epsilon_F \tag{1}$$

where

$$\|\boldsymbol{X}_\Omega\|_F = \left( \sum_{(i,j)\in\Omega} |\boldsymbol{X}_{ij}|^2 \right)^{1/2}$$

That is, among all matrices $\boldsymbol{M}$ satisfying $\|\boldsymbol{M}_\Omega - \boldsymbol{X}_\Omega\|_F \leq \epsilon_F$, we look for the one with minimum rank.

Note that (1) includes two special cases:

1. When $\epsilon_F = 0$, $M_\Omega = X_\Omega$ is required.
2. When $\Omega$ is the complete set of $m \times n$ entries, the problem is also called low-rank matrix approximation.

# Introduction

However, solving (1) is difficult because the rank minimization problem is computationally infeasible.

A popular and practical approach is to replace the nonconvex rank by convex nuclear norm:

$$\min_{\boldsymbol{M}} \ \|\boldsymbol{M}\|_*, \quad \text{s.t. } \|\boldsymbol{X}_\Omega - \boldsymbol{M}_\Omega\|_F \leq \epsilon_F \tag{2}$$

where $\|\boldsymbol{M}\|_*$ is sum of singular values of $\boldsymbol{M}$.

Representative algorithms include semi-definite programming (SDP) and singular value thresholding (SVT) but the complexity is high as full SVD calculation is required per iteration.

# Introduction

On the other hand, matrix completion algorithms based on factorization are suggested:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \ \|\boldsymbol{X}_\Omega - (\boldsymbol{U}\boldsymbol{V})_\Omega\|_F^2 \tag{3}$$

where $\boldsymbol{U} \in \mathbb{R}^{m\times r}$ and $\boldsymbol{V} \in \mathbb{R}^{r\times n}$ are low-rank matrices. Corresponding algorithms include low-rank matrix fitting (LMaFit) and proximal matrix completion (PAM).

After computing $\boldsymbol{U}$ and $\boldsymbol{V}$, the recovered matrix is:

$$\boldsymbol{M} = \boldsymbol{U}\boldsymbol{V} \tag{4}$$

Despite its computational attractiveness, the $\ell_2$-norm is not robust to gross errors or outliers.

# Algorithm Development

## Robustification by Truncated-Quadratic Function

To achieve outlier robustness, the Frobenius norm is replaced by a robust M-estimation function $\phi(\cdot)$:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \ \|\boldsymbol{X}_\Omega - (\boldsymbol{U}\boldsymbol{V})_\Omega\|_F^2 \Rightarrow$$

$$\min_{\boldsymbol{U},\boldsymbol{V}} \ \phi\left(\boldsymbol{X}_\Omega - (\boldsymbol{U}\boldsymbol{V})_\Omega\right), \quad \phi\left(\boldsymbol{X}_\Omega\right) = \sum_{(i,j)\in\Omega} \phi\left(\boldsymbol{X}_{ij}\right)$$

Truncated-quadratic function is used since it is bounded from above:

$$\phi(x) = \begin{cases} \dfrac{x^2}{2}, & |x| < e \\ \dfrac{e^2}{2}, & |x| \geq e \end{cases}$$

# Algorithm Development

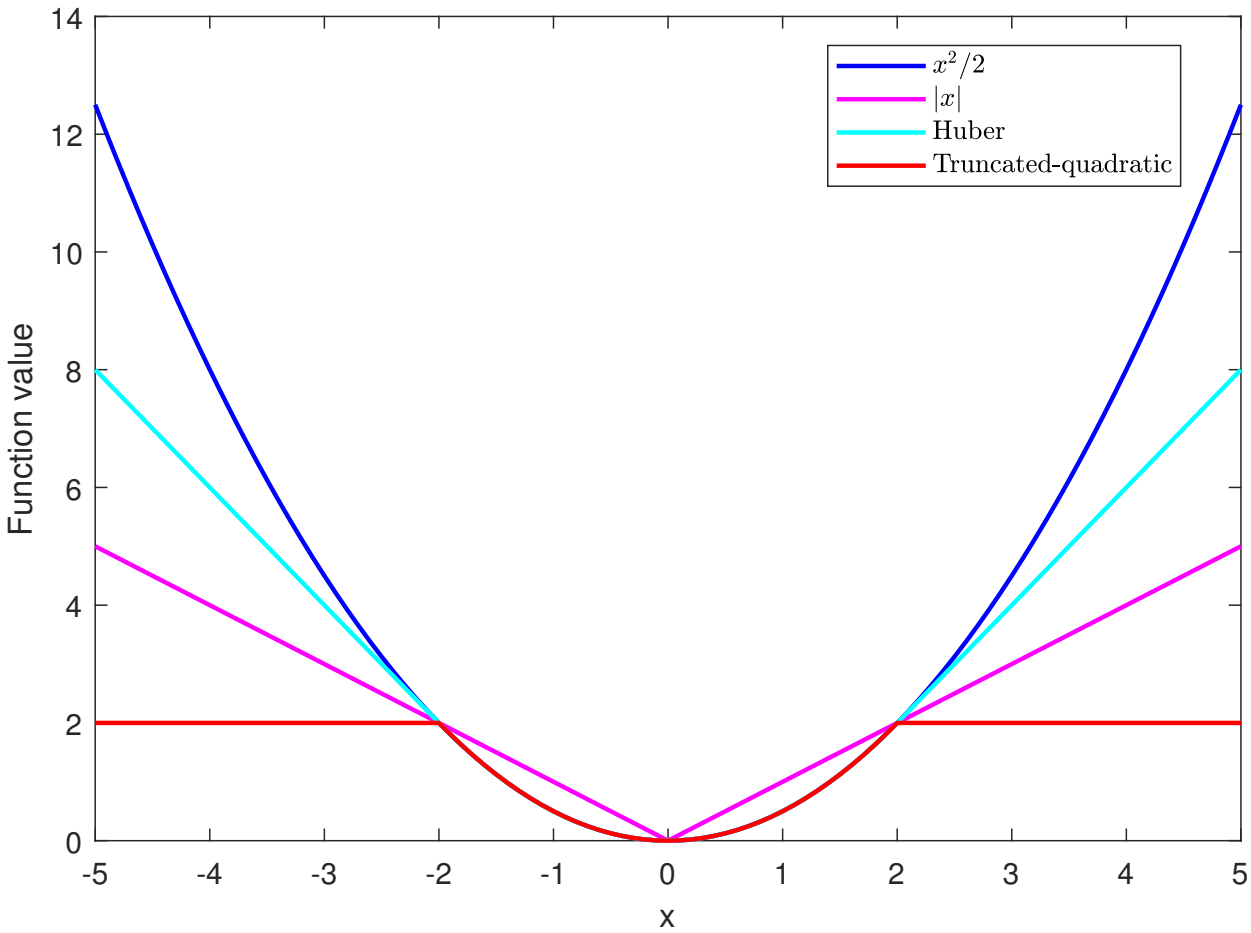


Figure 3: Different loss functions.

When $e = 2$ is chosen, all residual errors are upper bounded by 2.

# Algorithm Development

## Reformulations via Half-Quadratic Optimization

Applying half-quadratic optimization [1], $\phi(x)$ can be written as:

$$\phi(x) = \min_{y} \; Q(x,y) + \varphi(y) \tag{5}$$

where $Q(x,y)$ is a quadratic function while $\varphi(\cdot)$ is the dual function of $\phi(\cdot)$ with $y$ being the auxiliary variable.

The original problem can then be expressed as:

$$\min_{x} \phi(x) = \min_{x,y} \; Q(x,y) + \varphi(y) \tag{6}$$

It is worth noting that $x$ is easy to solve since $Q(x,y)$ is a quadratic function, and if the solution of $y$ is available, the problem can be solved by alternating updates of $x$ and $y$.

# Algorithm Development

There are two forms of $Q(x,y)$, and the corresponding solutions of $y$ have been derived as follows (See our paper):

Additive form:

$$\phi(x) = \min_y \ \frac{(x-y)^2}{2} + \varphi_A(y), \quad Q_A(x,y) = \frac{(x-y)^2}{2}$$

$$y^* := \arg\min_y \ Q_A(x,y) + \varphi_A(y) = \begin{cases} 0, & |x| < e \\ x, & |x| \geq e \end{cases} \tag{7}$$

Multiplicative form:

$$\phi(x) = \min_y \ \frac{y \cdot x^2}{2} + \varphi_M(y), \quad Q_M(x,y) = \frac{y \cdot x^2}{2}$$

$$y^* := \arg\min_y \ Q_M(x,y) + \varphi_M(y) = \begin{cases} 1, & |x| < e \\ 0, & |x| \geq e \end{cases} \tag{8}$$

# Algorithm Development

## Solvers using Block Coordinate Descent

Recall:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \ \phi\left(\boldsymbol{X}_{\Omega}-(\boldsymbol{U}\boldsymbol{V})_{\Omega}\right) \Leftrightarrow \min_{\boldsymbol{U},\boldsymbol{V}} \ \sum_{(i,j)\in\Omega} \phi\left(\boldsymbol{X}_{i,j}-(\boldsymbol{U}\boldsymbol{V})_{i,j}\right)$$

The additive form is:

$$\begin{aligned}
&\min_{x}\phi(x)=\min_{x,y} \ \frac{(x-y)^2}{2}+\varphi_A(y) \Rightarrow \\
&\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{N}} \ \sum_{(i,j)\in\Omega}\frac{1}{2}\left(\boldsymbol{X}_{i,j}-(\boldsymbol{U}\boldsymbol{V})_{i,j}-\boldsymbol{N}_{i,j}\right)^2+\varphi_A(\boldsymbol{N}_{i,j}) \\
&=\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{N}} \ \frac{1}{2}\left\|\boldsymbol{X}_{\Omega}-(\boldsymbol{U}\boldsymbol{V})_{\Omega}-\boldsymbol{N}_{\Omega}\right\|_F^2+\varphi_A(\boldsymbol{N}_{\Omega})
\end{aligned} \tag{9}$$

Applying block coordinate descent (BCD), $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{N}$ are updated in an alternating and iterative manner:

# Algorithm Development

$$
\begin{aligned}
\boldsymbol{U}^{k+1} &= \arg\min_{\boldsymbol{U}} \ \frac{1}{2}\left\|\boldsymbol{X}_{\Omega} - \left(\boldsymbol{U}\boldsymbol{V}^{k}\right)_{\Omega} - \boldsymbol{N}_{\Omega}^{k}\right\|_{F}^{2} \\
\boldsymbol{V}^{k+1} &= \arg\min_{\boldsymbol{V}} \ \frac{1}{2}\left\|\boldsymbol{X}_{\Omega} - \left(\boldsymbol{U}^{k+1}\boldsymbol{V}\right)_{\Omega} - \boldsymbol{N}_{\Omega}^{k}\right\|_{F}^{2} \\
\boldsymbol{N}^{k+1} &= \arg\min_{\boldsymbol{N}} \ \frac{1}{2}\left\|\boldsymbol{X}_{\Omega} - \left(\boldsymbol{U}^{k+1}\boldsymbol{V}^{k+1}\right)_{\Omega} - \boldsymbol{N}_{\Omega}\right\|_{F}^{2} + \varphi_{A}(\boldsymbol{N}_{\Omega})
\end{aligned}
$$

For $\boldsymbol{U}^{k+1}$, it can be estimated row-by-row using linear least squares (LLS), resulting in closed-form solution:

$$
\boldsymbol{u}_{i}^{k+1} = \left(\left(\boldsymbol{V}_{\mathcal{J}_i}^{k}\right)^{T}\right)^{\dagger} (\boldsymbol{D}_{i,\mathcal{J}_i}^{k})^{T}, \quad i = 1, \cdots, m
$$

where $\boldsymbol{V}_{\mathcal{J}_i}^{k}$ and $\boldsymbol{D}_{i,\mathcal{J}_i}^{k}$ are constructed from $\boldsymbol{V}^{k}$ and $\boldsymbol{D}^{k} = \boldsymbol{X}_{\Omega} - \boldsymbol{N}_{\Omega}^{k}$, respectively.

# Algorithm Development

Similarly, $\boldsymbol{V}^{k+1}$ is estimated column-by-column using LLS:

$$\boldsymbol{v}_j^{k+1} = \left(\boldsymbol{U}_{\mathcal{I}_j}^{k+1}\right)^{\dagger} \boldsymbol{D}_{\mathcal{I}_j,j}^{k}, \quad j = 1, \cdots, n$$

where $\boldsymbol{U}_{\mathcal{I}_j}^{k+1}$ and $\boldsymbol{D}_{\mathcal{I}_j,j}^{k}$ are constructed from $\boldsymbol{U}^k$ and $\boldsymbol{D}^k = \boldsymbol{X}_\Omega - \boldsymbol{N}_\Omega^k$, respectively.

Defining $\boldsymbol{R}_\Omega = \boldsymbol{X}_\Omega - (\boldsymbol{UV})_\Omega$, we have:

$$\boldsymbol{N}_{i,j} = \begin{cases} 0, & |\boldsymbol{R}_{i,j}| < e \\ \boldsymbol{R}_{i,j}, & |\boldsymbol{R}_{i,j}| \geq e \end{cases}, \quad (i,j) \in \Omega$$

The value of $e$ is updated via the robust normalized median absolute deviation (MADN) [2] as:

$$e^k = \min\left(\xi\sigma^k, e^{k-1}\right), \quad \sigma^k = 1.4815 \times \mathrm{Med}(|\mathrm{vec}(\boldsymbol{R}_\Omega^k) - \mathrm{Med}(\mathrm{vec}(\boldsymbol{R}_\Omega^k))|)$$

# Algorithm Development

Analogously, the multiplicative form is:

$$\min_{x} \phi(x) = \min_{x,y} \frac{y \cdot x^2}{2} + \varphi_M(y) \Rightarrow$$

$$\begin{aligned} &\min_{U,V,W} \sum_{(i,j)\in\Omega} \frac{1}{2} W_{i,j} \left( X_{i,j} - (UV)_{i,j} \right)^2 + \varphi_M(W_{i,j}) \\ &= \min_{U,V,W} \frac{1}{2} \left\| \sqrt{W_\Omega} \circ (X_\Omega - (UV)_\Omega) \right\|_F^2 + \varphi_M(W_\Omega) \end{aligned} \tag{10}$$

Using BCD, we have

$$U^{k+1} = \arg\min_{U} \frac{1}{2} \left\| \sqrt{W_\Omega^k} \circ \left( X_\Omega - \left( UV^k \right)_\Omega \right) \right\|_F^2$$

$$V^{k+1} = \arg\min_{V} \frac{1}{2} \left\| \sqrt{W_\Omega^k} \circ \left( X_\Omega - \left( U^{k+1}V \right)_\Omega \right) \right\|_F^2$$

$$W^{k+1} = \arg\min_{W} \frac{1}{2} \left\| \sqrt{W_\Omega} \circ \left( X_\Omega - \left( U^{k+1}V^{k+1} \right)_\Omega \right) \right\|_F^2 + \varphi_M(W_\Omega)$$

# Algorithm Development

Recall:

$$\arg\min_{y} \; Q_M(x,y) + \varphi_M(y) = \begin{cases} 1, & |x| < e \\ 0, & |x| \geq e \end{cases}$$

$\boldsymbol{W}^k$ has values of $0$ and $1$ only and we may set $\widetilde{\Omega}^k = \boldsymbol{W}^k \circ \Omega$, that is, we consider outlier-contaminated elements as missing entries:

$$\boldsymbol{U}^{k+1} = \arg\min_{\boldsymbol{U}} \; \frac{1}{2} \left\| \boldsymbol{X}_{\widetilde{\Omega}^k} - \left( \boldsymbol{U}\boldsymbol{V}^k \right)_{\widetilde{\Omega}^k} \right\|_F^2$$

$$\boldsymbol{V}^{k+1} = \arg\min_{\boldsymbol{V}} \; \frac{1}{2} \left\| \boldsymbol{X}_{\widetilde{\Omega}^k} - \left( \boldsymbol{U}^{k+1}\boldsymbol{V} \right)_{\widetilde{\Omega}^k} \right\|_F^2$$

$$\boldsymbol{W}_{i,j} = \begin{cases} 1, & |\boldsymbol{R}_{i,j}| < e \\ 0, & |\boldsymbol{R}_{i,j}| \geq e \end{cases}, \quad (i,j) \in \Omega$$

# Numerical Examples

Synthetic $\mathbf{X} \in \mathbb{R}^{m \times n}$ is generated with standard Gaussian distribution where $m = n$ and $r = m/50$ (Default $50\%$ observed entries, $m = 400$ and $r = 8$).
Gaussian mixture model (GMM) noise with 2 components: variances $\sigma^2$ and $100\sigma^2$ with probabilities $0.9$ and $0.1$ (Default SNR $= 10$dB).
Performance measure includes:

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

We compare our method with existing approaches, including $\mathrm{VBMFL}_1$ [3], $\mathrm{RMC-Huber}$ [4], $\ell_1-\mathrm{ADMM}$ [5], $\mathrm{RMF-MM}$ [6] and $\mathrm{HQ-PF}$ [7], in terms of RMSE and runtime for different matrix dimensions $m \times m$.

# Numerical Examples

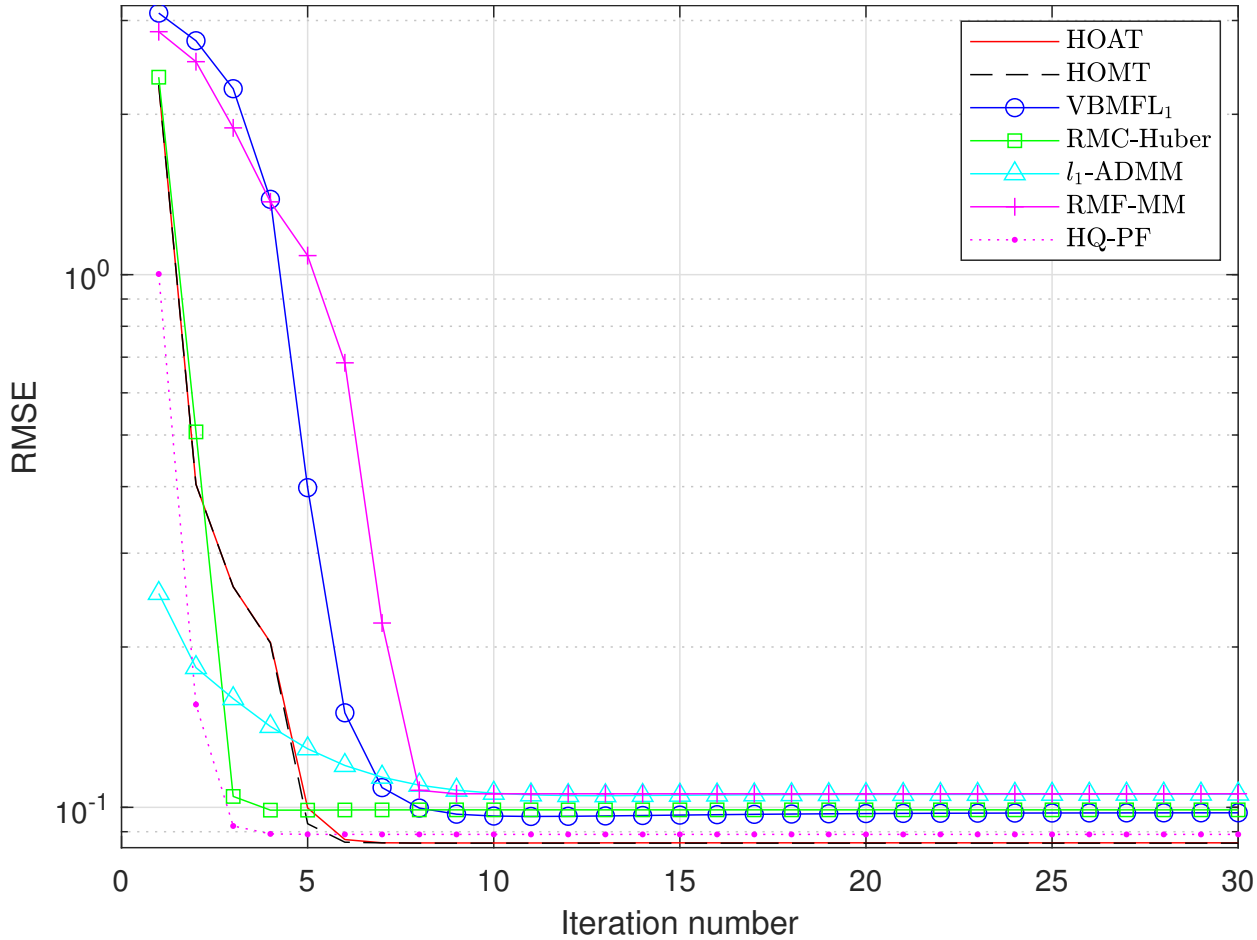


Figure 4: RMSE versus iteration number.

Note: HOAT and HOMT are the proposed algorithms

# Numerical Examples

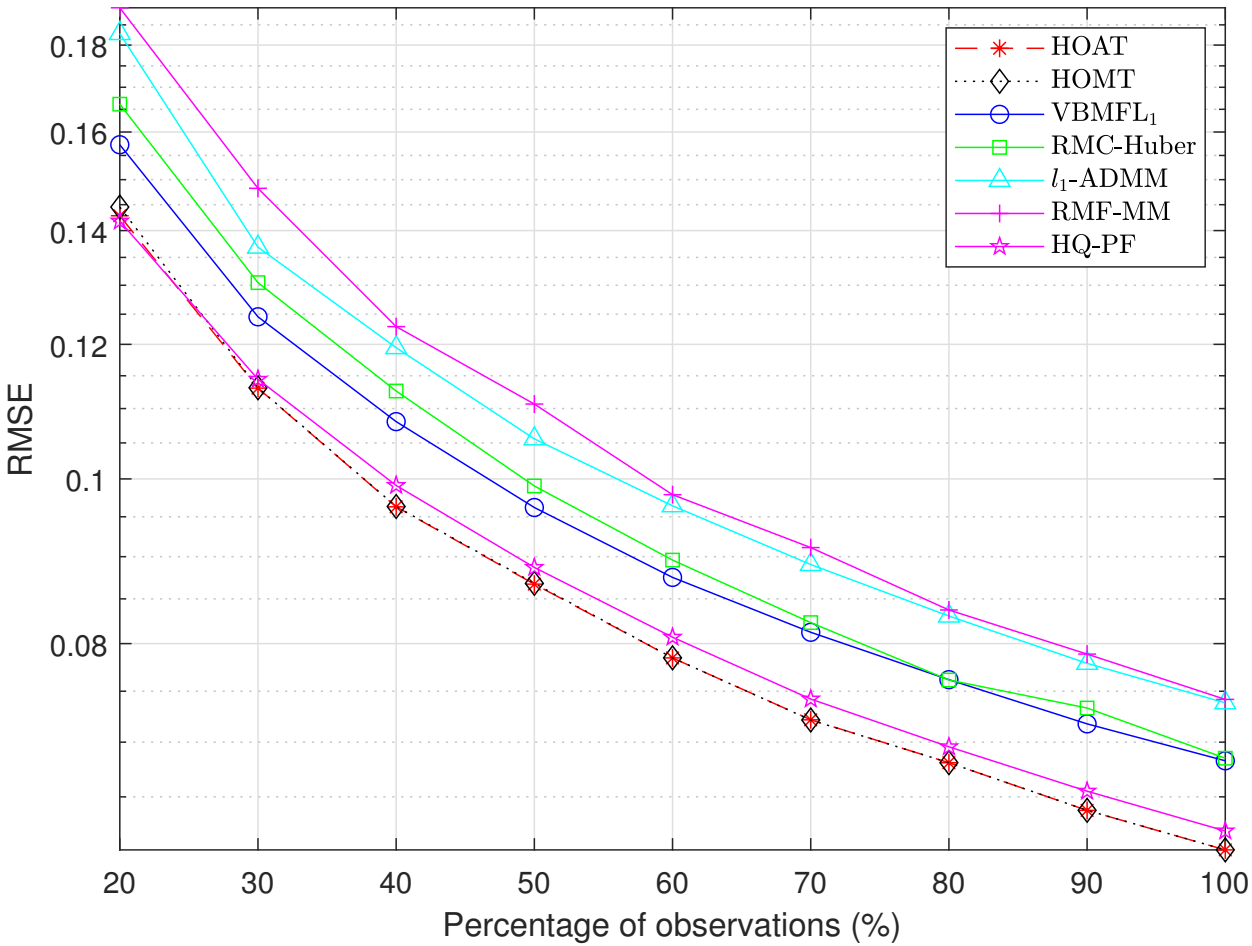


Figure 5: RMSE versus percentage of observations

# Numerical Examples

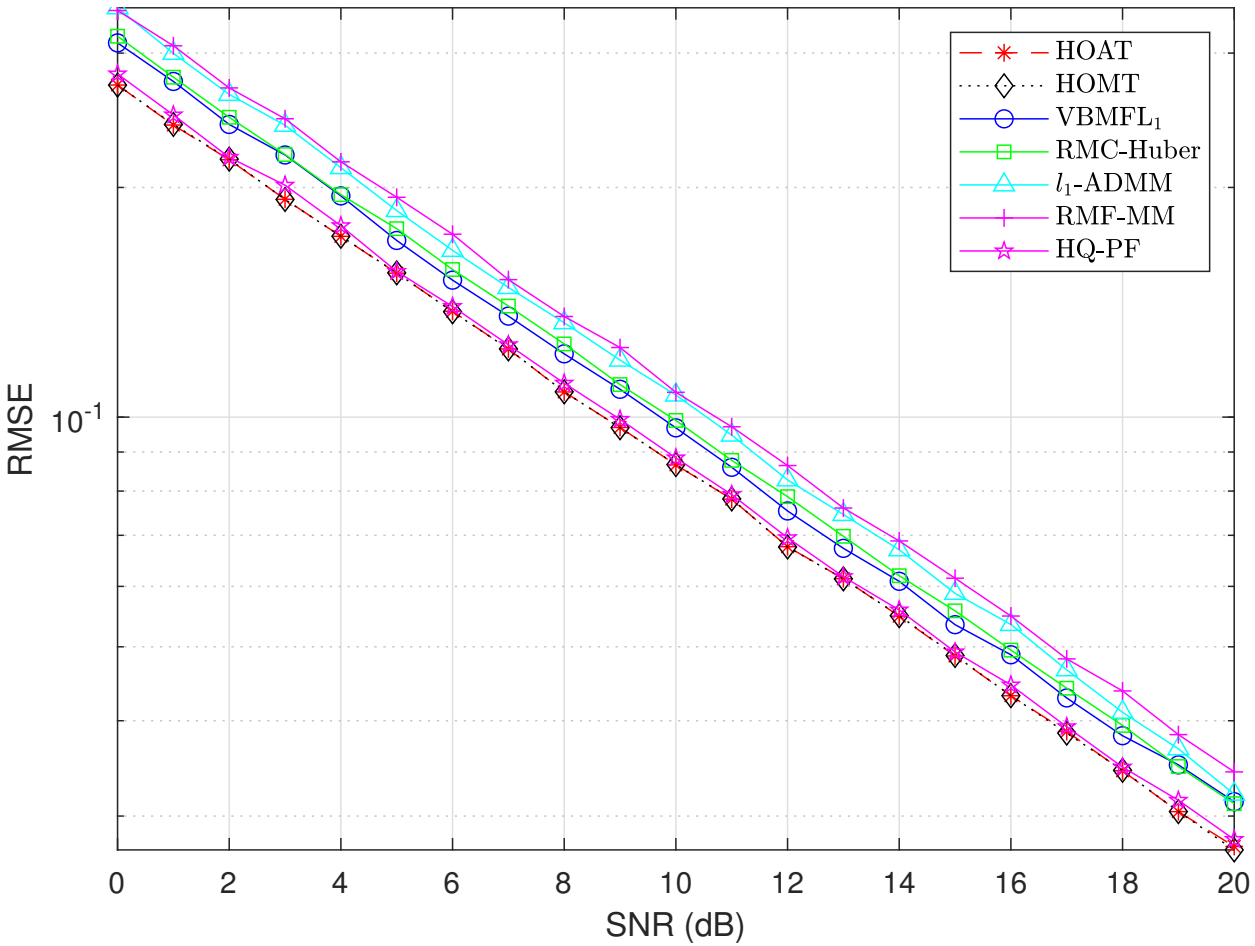


Figure 6: RMSE versus SNR in GMM noise

# Numerical Examples

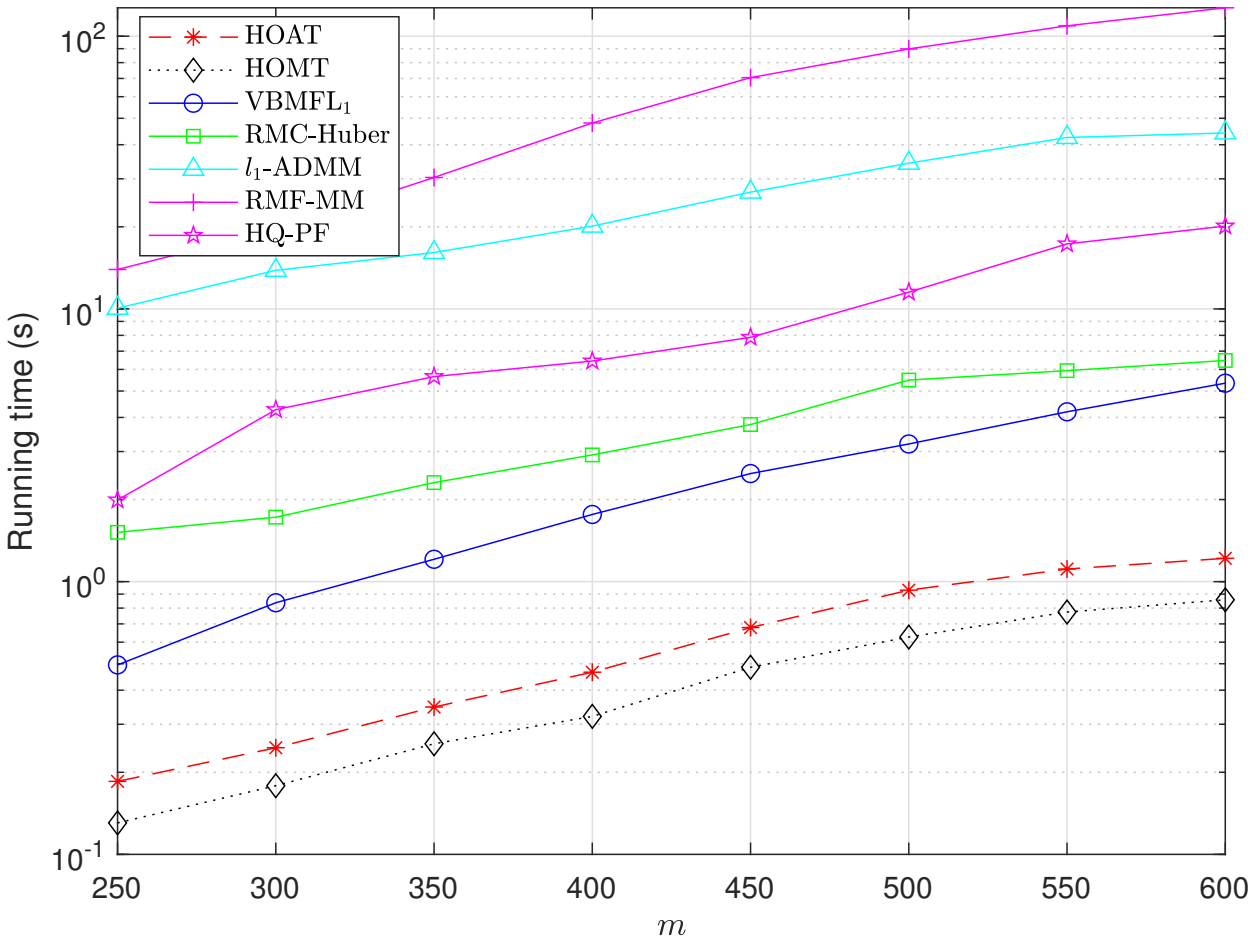


Figure 7: Running time versus $m$

# Numerical Examples

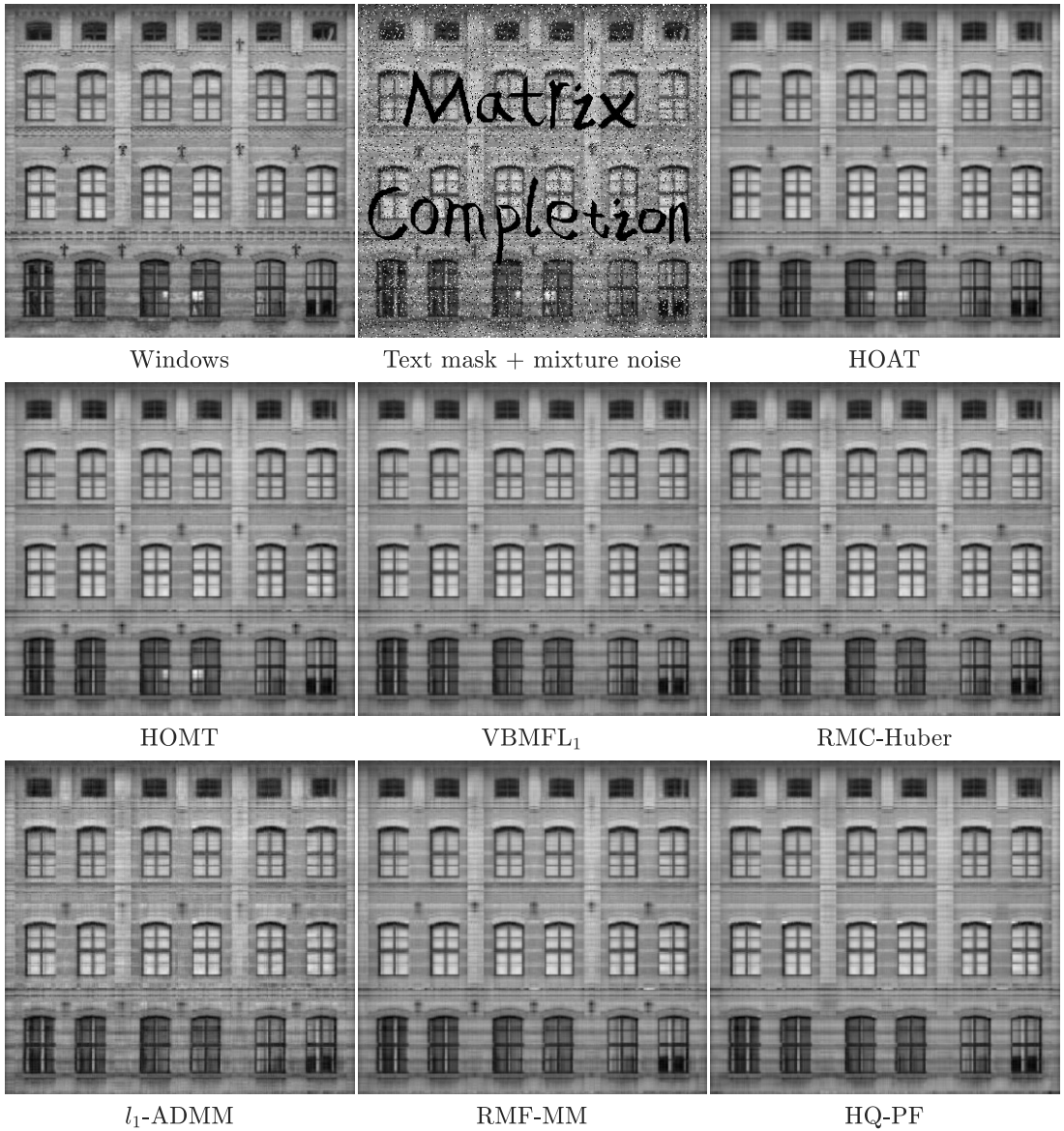


Figure 8: Image recovery results with text mask by different algorithms

# Numerical Examples

Table 1: Performance comparison for text mask: average PSNR, SSIM and runtime.

| Method | Gaussian noise | | | Mixture noise | | |
|---|---|---|---|---|---|---|
| | PSNR | SSIM | Runtime | PSNR | SSIM | Runtime |
| HOAT | **24.202** | **0.6520** | 4.6316 | **28.116** | **0.8523** | 1.129 |
| HOMT | 24.200 | 0.6517 | **4.139** | 28.083 | 0.8522 | **0.729** |
| VBMFL$_1$ [3] | 23.920 | 0.6387 | 6.663 | 27.747 | 0.8477 | 5.889 |
| RMC-Huber [4] | 23.064 | 0.5933 | 16.632 | 27.804 | 0.8469 | 19.456 |
| $\ell_1$-ADMM [5] | 22.323 | 0.5509 | 44.026 | 26.597 | 0.8073 | 27.982 |
| RMF-MM [6] | 22.532 | 0.5658 | 389.137 | 27.670 | 0.8435 | 148.132 |
| HQ-PF [7] | 24.142 | 0.6024 | 127.571 | 28.045 | 0.8449 | 40.394 |

# Conclusions

1. Based on the matrix factorization approach, truncated-quadratic loss function is adopted to achieve robust low-rank matrix recovery:
   1. Data uncontaminated by outliers are not affected.
   2. Outlier-contaminated residuals will be bounded.
   3. Half quadratic optimization is utilized to transform the problem in additive and multiplicative forms, allowing computationally efficient alternating minimization.
2. The developed HOAT and HOMT outperform a number of state-of-the-art techniques in terms of RMSE, PSNR, SSIM and runtime for both synthetic and real-world data.
3. One future work is to apply our sparse-inducing regularizer $\varphi_A(\cdot)$ to robust PCA to achieve low-rank and sparse decomposition.

Z.-Y. Wang, X. P. Li, H. C. So, "Robust matrix completion based on factorization and truncated-quadratic loss function," to appear in IEEE Transactions on Circuits and Systems for Video Technology

Code: https://github.com/bestzywang

# References

[1] M. Nikolova and M. K. Ng, "Analysis of half-quadratic minimization methods for signal and image recovery," SIAM J. Sci. Comput., vol. 27, no. 3, pp. 937–966, Jan. 2005.

[2] R. A. Maronna, D. R. Martin and V. J. Yohai, Robust Statistics: Theory and Methods, England: Wiley, 2006.

[3] Q. Zhao, D. Meng, Z. Xu,W. Zuo, and Y. Yan, "L1-Norm low-rank matrix factorization by variational Bayesian method," IEEE Trans. Neural Netw. Learn. Syst., vol. 26, no. 4, pp. 825–839, Apr. 2015.

[4] M. Muma, W. Zeng, and A. M. Zoubir, "Robust M-estimation based matrix completion," in 2019 IEEE Int. Conf. on Acoust., Speech and Signal Process., Brighton, UK, May 2019, pp. 5476–5480.

[5] W. Zeng and H. C. So, "Outlier-robust matrix completion via $\ell_p$-minimizatio IEEE Trans. Signal Process., vol. 66, no. 5, pp. 1125–1140, Mar. 2018.

[6] Z. Lin, C. Xu and H. Zha, "Robust matrix factorization by majorization minimization," IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 1, pp. 208–220, Jan. 2018.

# References

[7] Y. He, F. Wang, Y. Li, J. Qin, and B. Chen, "Robust matrix completion via maximum correntropy criterion and half-quadratic optimization," IEEE Trans. Signal Process., vol. 68, pp. 181–195, Nov. 2019.