Optimal Smooth Approximation for Quantile Matrix Factorization

Peng Liu* Yi Liu† Rui Zhu‡ Linglong Kong§ Bei Jiang¶ Di Niu‖

Abstract

Matrix Factorization (MF) is essential to many estimation tasks. Most existing matrix factorization methods focus on least squares matrix factorization (LSMF), which aims to minimize a smooth L_2 loss between observations and their dependent matrix measurement variables. In reality, however, L_1 loss and check loss are widely used in regression to deal with outliers or observations contaminated by skewed or heavy-tailed noise. Although under certain conditions, linear convergence to the global optimality can be established for matrix factorization under the L_2 loss, there is a lack of provably efficient algorithms for solving matrix factorization under non-smooth losses. In this paper, we investigate Quantile Matrix Factorization (QMF), the counterpart of Quantile Regression in matrix estimation, that adopts a tunable check loss and introduces robustness to matrix estimation for skewed and heavy-tailed observations, which are prevalent in reality. To deal with the non-smooth loss, we propose Nesterov-smoothed QMF (NsQMF), extending Nesterov's optimal smooth approximation technique to the matrix factorization setting. We then present an alternating minimization algorithm to solve the smooth NsQMF efficiently. We mathematically prove that solving the smoothed NsQMF is equivalent to solving the original non-smooth QMF problem and that our proposed algorithm achieves linear convergence to the global optimality of QMF. Numerical evaluations verify our theoretical findings and demonstrate that NsQMF significantly outperforms the commonly used LSMF and prior approximate smoothing heuristics for QMF under various noise distributions.

*School of Mathematics, Statistics and Actuarial Science, University of Kent, Canterbury, Kent CT2 7FS, United Kingdom, p.liu@kent.ac.uk

†Department of Mathematical and Statistical Sciences, University of Alberta, Edmonton, AB T6G 2G1, Canada

‡Deepmind, Montreal, Canada

§Department of Mathematical and Statistical Sciences, University of Alberta, Edmonton, AB T6G 2G1, Canada

¶Department of Mathematical and Statistical Sciences, University of Alberta, Edmonton, AB T6G 2G1, Canada

‖Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6J 2W3, Canada

1 INTRODUCTION

Matrix Factorization (MF) is a popular approach to low-rank matrix estimation, which attempts to find two matrices $U \in \mathbb{R}^{r\times m}$ and $V \in \mathbb{R}^{r\times n}$ with $r < \min(m,n)$, such that $M = U^\top V \in \mathbb{R}^{m\times n}$ is low-rank and each $\langle A_i, U^\top V\rangle$ approximates a possibly noisy observation b_i, $i = 1,\dots,p$, where $A_i \in \mathbb{R}^{m\times n}$ is a linear transformation matrix, and the inner product $\langle A_i, M\rangle$ is defined as the sum of element-wise products between two matrices of the same size. MF especially appeals to large-scale or distributed implementation, since it has a lower per-iteration computational cost (Sun & Luo, 2015) and can be solved by simple optimization algorithms. Hence, MF has been widely adopted in recommender systems (Chen, 2016), computer vision (Chen & Suter, 2004), etc.

In recent years, a number of theoretical results have been established for the convergence of MF, especially for the widely adopted *Least Squares Matrix Factorization* (LSMF) (Jain, 2013; Tu, 2016), which aims to minimize an L_2 loss between b_i and corresponding $\langle A_i, U^\top V\rangle$. It is proved that although MF is non-convex in terms of U and V, under certain conditions, some simple algorithms can achieve global optimality.

However, a common limitation of existing MF schemes is that they minimize smooth objective functions, e.g., L_2 loss, which are not robust to outliers. While the use of non-smooth loss functions, such as the check loss and L_1 loss, there lacks provably efficient techniques to handle skewed or heavy-tailed data in matrix observations, which are also prevalent in practice. For example, latencies to web services on the Internet are highly skewed: most measurements are within hundreds of milliseconds while a few outliers could be over several seconds due to congestion (Zheng, 2014). Here LSMF yields $\hat{M}$ such that $\langle A_i, \hat{M}\rangle$ estimates the *conditional mean* of the observation b_i. However, if the observation is contaminated by non-Gaussian noise, the conditional mean may be far away from the central tendency of data. For example, it is better to recommend web services based on the *most probable* latency to each service, which is more appropriately represented by the median latency than by the mean, the latter tending to bias toward large outliers.

QMF (Karatzoglou & Weimer, 2010), Zhu (2017),

the counterpart of Quantile Regression(Koenker & Bassett, 1978) in the context of matrix factorization, has emerged as a promising approach to solving data skewness issues. QMF aims to estimate the τ-th *conditional quantile* of $\langle A_i, \hat{M} \rangle$. QMF not only improves the robustness of MF to outliers, but can also offer a more complete statistical view of matrix variables beyond mean statistics.

In this paper, we propose the first provably optimal algorithm to solve QMF, we focus on the matrix sensing problem, which aims to recover the unknown matrix from a small number of linear measurements Park (2017). Matrix sensing is closely related to the popular matrix completion problemJain (2013), and the latter is a special case of the former. The proposed method is called Nesterov-smoothed QMF (NsQMF) (Nesterov, 2005). Specifically, we present an efficient algorithm to solve the non-convex and non-smooth QMF problem with a theoretical guarantee on linear convergence, while such guarantees were previously known only for smooth and strongly convex MF objective functions. Our main contributions are highlighted as follows:

First, we propose an optimal smooth approximation to the QMF problem based on Nesterov's smoothing method. We establish that Nesterov's method can provide optimal smoothing to QMF. That is, solving NsQMF is equivalent to solving QMF, while NsQMF can be handled much more efficiently with gradient-based optimization techniques.

Second, we propose an alternating minimization algorithm to solve the smooth (yet non-convex) NsQMF problem efficiently. With an initialization procedure that terminates in a constant number of steps, we establish the linear convergence of the proposed algorithm to the global optimality. Since we have proved that solving NsQMF is equivalent to solving the original QMF, we have shown that the proposed NsQMF can efficiently generate optimal solutions to QMF using gradient-descent based optimization.

We further perform numerical experiments to verify our theoretical findings, algorithm efficiency, and the advantages of the proposed NsQMF method over LSMF and prior heuristic smoothing techniques for QMF, in a wide range of settings. We show that NsQMF significantly outperforms LSMF and other prior smoothing techniques for QMF in a wide range of settings. In the meantime, NsQMF can recover the underlying true low-rank matrix when there is no noise, while achieving linear convergence to the global optimality, which is comparable to the convergence speed of LSMF.

2 RELATED WORK

Recently, there are several works that aim to combine alternating minimization and Nesterov's momentum algorithm, in order to achieve optimal convergence, e.g. (Guminov, 2021; Mitchell, 2020). However, none of them are dealing with matrix recovery or factorization. For MF, Li & Lin (2020) develop an accelerated gradient method for non-convex low-rank optimization. However, the proposed method contains an obstacle that is hard to achieve in practice. Guan *et al.* (2012) utilize Nesterov's smoothing technique for non-negative matrix factorization (NNMF), but have not provided any theoretical characterization on the gap between the smoothed and original problems. We close such a smoothing gap for QMF by showing that solving the smoothed NsQMF is equivalent to solving QMF.

Karatzoglou & Weimer (2010) directly apply gradient descent on the non-smooth objective function, which may cause numerical issues since non-smooth functions do not have gradients[1]. Various algorithms may be used to handle the non-smoothness, e.g., linear programming, interior point and MM algorithms (Barrodale & Roberts, 1973; Hunter & Lange, 2000), and the recently proposed *Mixed Integer Optimization* (MIO) framework (Bertsimas *et al.*, 2014). However, most of these algorithms are computationally intensive or difficult to tune, since it is hard to choose the learning rate around the check point.

As a useful alternative to solve non-smooth problems, smoothing methods are investigated such that gradient based methods can be applied onto the smoothed problem. Zhu (2017) describe the first heuristic smoothing method for QMF, using $\sqrt{x^2+\eta}$ to approximate $|x|$ in the check loss, where η is a smoothing constant, yet without optimality guarantee. Regarding on the application of Nesterov's smoothing method in matrix factorization, Tu *et al.* (2021) compared several ad hoc algorithms, such as Adam and YellowFin, and showed that Nesterov's smoothing method have the best performance in terms of convergence speed and accuracy. To the best of our knowledge, this is the first convergence and optimality result for matrix factorization under a quantile loss function.

Recently, Park (2018) studied matrix factorization problems but adopt a differentiable objective function $f(U^\top V)$, and they showed the linear convergence rate based on gradient descent algorithm. Under non-smooth objective function, Li *et al.* (2020a) considered robust low-rank matrix recovery under subgradient algorithm, which is not efficient. Li *et al.* (2020b) pro-

[1] A subgradient algorithm can be applied, but is 20× slower than our proposed algorithm in experiments.

posed a low-rank matrix recovery under quadratic loss via median-truncated gradient descent algorithm, and obtained the linear convergence rate, but their method is not robust to outliers.

3 PROBLEM FORMULATION

Consider noisy observations made from an underlying low-rank matrix $M \in \mathbb{R}^{m\times n}$ via linear mapping $\mathcal{A}(M) : \mathbb{R}^{m\times n} \to \mathbb{R}^p$. The objective is to recover M based on p observations, which b_i, $i = 1,\dots,p$ are possibly contaminated by noise, here $p \ll m\cdot n$. We assume that the data are generated from the following model:

$$b_i = (\mathcal{A}(M^*))_i + \epsilon_i, \quad i = 1,\dots,p, \tag{3.1}$$

where M^* is the ground truth matrix, ϵ_i is the noise, $(\mathcal{A}(M^*))_i$ denotes the ith component of $\mathcal{A}(M^*)$, assume that $\mathcal{A}(M)$ can be expressed as:

$$(\mathcal{A}(M))_i = \langle A_i, M\rangle, \text{for any } i = 1,\cdots,p \text{ and } A_i \in \mathbb{R}^{m\times n},$$

where $\langle A_i, M\rangle$ is defined as the sum of element-wise products between the two matrices A_i and M.

In MF methods, a true matrix M^* with $\text{rank}(M^*) \le r$ is assumed to be factorizable, i.e., $M^* = U^{*\top}V^*, U^* \in \mathbb{R}^{r\times m}, V^* \in \mathbb{R}^{r\times n}$, where $r \ll \{m,n,p\}$. Under these assumptions, U^* and V^* can be found by solving:

$$\min_{U,V} \frac{1}{p}\sum_{i=1}^{p} \mathcal{L}(b_i, \langle A_i, U^\top V\rangle),$$

where $\mathcal{L}(\cdot,\cdot)$ is a certain loss function. Once the solution to the above problem, i.e., $\hat{U}$ and $\hat{V}$, are found, M^* can be estimated by $\hat{M} = \hat{U}^\top\hat{V}$.

The most common loss function used in MF is the L_2 loss, leading to the LSMF problem:

$$(\hat{U}_{LS}, \hat{V}_{LS}) = \arg\min_{U,V} \frac{1}{p}\sum_{i=1}^{p}(b_i - \langle A_i, U^\top V\rangle)^2. \tag{3.2}$$

Just like *least squares estimates* in linear regression, $\langle A_i, \hat{U}_{LS}^\top\hat{V}_{LS}\rangle$ based on (3.2) can be deemed as estimating the conditional mean function for each observation b_i. Although the conditional mean is the most efficient estimator under symmetric Gaussian noise, yet it is not for skewed or heavy-tailed noise. In these cases, we need new techniques beyond to better capture the central tendency of observations and provide a more complete characterization of the matrix data of interest.

The QMF differs from LSMF in that it replaces the L_2 loss with a check loss, solving the problem:

$$\min_{U,V} \frac{1}{p}\sum_{i=1}^{p} \rho_\tau(b_i - \langle A_i, U^\top V\rangle), \tag{3.3}$$

for simplicity, we denote $\bar{\rho}_\tau(\cdot) = 1/p\sum_{i=1}^{p}\rho_\tau(\cdot)$. The above problem aims to find $\hat{M} = \hat{U}^\top\hat{V}$ such that $\langle A_i, \hat{M}\rangle$ estimates the τ-th conditional *quantile*.

4 NESTEROV's SMOOTHING METHOD for QMF

Despite its statistical advantages, solving (3.3) is significantly more challenging than LSMF due to its non-convex and non-smooth nature. To handle the non-smooth objective function, we propose a smooth approximation to (3.3). Notice that $\rho_\tau(x) = (\tau - 1/2)x + 1/2|x|$, where the first term is smooth and we propose to approximate the $1/2|x|$ with a smoothing function by utilizing Nesterov's smoothing method (Nesterov, 2005). We have following proposition:

PROPOSITION 1. *The smooth approximation for $\rho_\tau(x)$ by utilizing Nesterov's smoothing method is*

$$\rho_{\delta,\tau}(x) = 1/2\psi_\delta(|x|) + (\tau - 1/2)x, \tag{4.4}$$

where

$$\psi_\delta(x) = \begin{cases} \frac{x^2}{2\delta}, & -\delta \le x \le \delta \\ x - \frac{\delta}{2}, & x \ge \delta \text{ or } x \le -\delta \end{cases}. \tag{4.5}$$

Fig. 4.1 illustrates the behavior of the smooth approximation for the check loss function.

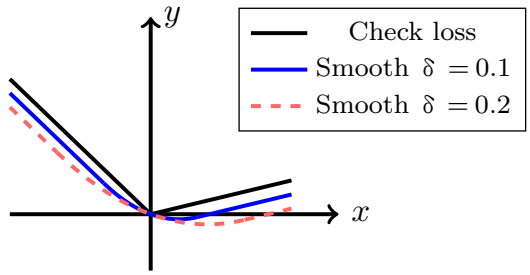


Figure 4.1: Nesterov's smoothing method applied to the check loss function.

Due to Proposition 1, the smooth approximation to the original QMF problem can be written in the following form:

$$\min_{U,V} \frac{1}{p}\sum_{i=1}^{p} \rho_{\delta,\tau}(b_i - \langle A_i, U^\top V\rangle), \tag{4.6}$$

here we denote $\bar{\rho}_{\delta,\tau}(\cdot) = 1/p\sum_{i=1}^{p}\rho_{\delta,\tau}(\cdot)$.

However, in MF, given $M = U^\top V$, we can find infinite number of pairs, PU and $(P^{-1})^\top V$, where $P \in \mathbb{R}^{r\times r}$ could be any invertible matrix. And $(PU)^\top(P^{-1})^\top V$ still equals to M. Therefore, we

consider the optimal solution restricted to a smaller set of 'equally-footed' factorizations (Park, 2017).

$$\begin{aligned}&\mathcal{X}^* = \{(U^*, V^*) : \\ &U^* \in \mathbb{R}^{r\times m}, V^* \in \mathbb{R}^{r\times n}, U^{*\top}V^* = M^*, \\ &\sigma_i(U^*) = \sigma_i(V^*) = \sigma_i(M^*)^{1/2}, i = 1, \ldots, r\}.\end{aligned} \tag{4.7}$$

Here $\sigma_i(\cdot)$ denotes the ith largest eigenvalue. $(U^*, V^*) \in \mathcal{X}^*$ if and only if $U^* = A^*\Sigma^{*1/2}R$, $V^* = B^*\Sigma^{*1/2}R$, where $A^*\Sigma^*B^*$ is the Singular Value Decomposition of M^*, and $R \in \mathbb{R}^{r\times r}$ is an orthogonal matrix.

However, practically it is hard to find a solution (U^*, V^*) that belongs to $\mathcal{X}^*$. As a result, in numerical computation, we introduce a regularizer $\|UU^\top - VV^\top\|_F^2$ when minimizing $\bar{\rho}_{\delta,\tau}(b - \mathcal{A}(U^\top V))$, leading to the following smoothed, regularized problem:

$$\min_{U,V} \bar{\rho}_{\delta,\tau}(b - \mathcal{A}(U^\top V)) + \lambda\|UU^\top - VV^\top\|_F^2, \tag{4.8}$$

denote (4.8) as $\min_{U,V} \tilde{\bar{\rho}}_{\delta,\tau}(b - \mathcal{A}(U^\top V))$. According to Park (2018), (4.8) will lead to the solution of (U, V) lies in $\mathcal{X}^*$, and the regularizer guarantees the putative estimates per iteration are not too ill-conditioned. The regularizer can preserve the relative structures of U and V and can ensure that the two factors satisfy the 'equal-footing' property, which is also referred to as the *orthogonal Procrustes problem* (Schönemann, 1966).

More importantly, introducing the regularizer in (4.8) does not change the problem, in the sense that if U^* and V^* belong to the set $\mathcal{X}^*$, then $\tilde{\bar{\rho}}_{\delta,\tau}(b - \mathcal{A}(U^{*\top}V^*)) = \bar{\rho}_{\delta,\tau}(b - \mathcal{A}(U^{*\top}V^*))$. So the solution to (4.8) is also a solution to (4.6) (Park, 2017).

In our case, the regularizer has another purpose, which is to make the objective function $\tilde{\bar{\rho}}_{\delta,\tau}(\cdot)$ strongly convex and L-smooth. Define

$$Z = \begin{bmatrix} U \\ V \end{bmatrix}, \tilde{M} = ZZ^\top$$

Then $\tilde{\bar{\rho}}_{\delta,\tau}(\tilde{M})$ is strongly $(L + 4\lambda)$-smooth and 2λ strongly convex function over positive semi-definite matrices with $\text{rank}(\tilde{M}) \leq r$, where L is the strongly convex parameter of $\bar{\rho}_{\delta,\tau}(\tilde{M})$.

5 ALGORITHM

Although the loss function $\tilde{\bar{\rho}}_\tau(b - \mathcal{A}(U^\top V))$ is not convex in terms of the joint tuple (U, V), (4.8) has a *bi-convex* structure with respect to U and V, i.e., it is convex in U when V is fixed and vice versa. Leveraging the bi-convex nature of the problem, we employ an alternating optimization method to solve (4.8), which alternately updates U or V in each iteration while keeping the other variables fixed, while in each update, we use the Nesterov's momentum method. The algorithm is presented as follows:

ALGORITHM 5.1. The Alternating Optimization Procedure (Main algorithm)

1: Input b,$\mathcal{A}$
2: Initial point $U_{(0)}$, $V_{(0)}$, iteration number T, subsolver iteration number K
3: For $t = 0, 1, 2, \ldots, T$
4: 1.1. $U_{(t+1)} = \text{NESTMOMENU}(U_{(t)}, V_{(t)})$
5: 1.2. $V_{(t+1)} = \text{NESTMOMENV}(U_{(t+1)}, V_{(t)})$
6: Output $U_{(T)}$, $V_{(T)}$

The alternating optimization idea has been used by several authors in low-rank matrix recovery previously (Jain, 2013; Tu, 2016). However, they only deal with LSMF, with a least squares objective. Here, we make modifications to the usual gradient descent algorithm for the updates of U and V so that we can handle the proposed Nesterov-smoothed objective function in (4.8). This leads to Algorithms 5.2 and 5.3 for updating U and V, respectively, based on Nesterov's momentum method applied to our NsQMF problem. Here we set the momentum term γ to be 2λ, and the learning rate $\alpha = 1/(L + 4\lambda)$. In practical, according to our experience, we can choose the momentum to be 0.9. And we will also examine the practical behavior of different values of the learning rate.

ALGORITHM 5.2. Update $U_{(t+1)}$ under Nesterov's momentum algorithm (NESTMOMENU)

1: Input $V_{(t)} \in \mathbb{R}^{r\times n}$, $U_{(t)}^0 \triangleq U_{(t)} \in \mathbb{R}^{r\times m}$
2: For $t^* = 0, 1, 2, \ldots, K - 1$
3: 1.1. $\tilde{U}_{(t)}^{t^*+1} = U_{(t)}^{t^*} - \frac{1}{\beta}\nabla_U \tilde{\bar{\rho}}_{\delta,\tau}(b - \mathcal{A}((U_{(t)}^{t^*})^\top V_{(t)}^{t^*}))$
4: 1.2. $U_{(t)}^{t^*+1} = (1 - \gamma_t)\tilde{U}_{(t)}^{t^*+1} + \gamma\tilde{U}_{(t)}^{t^*}$
5: Output $U_{(t+1)} = U_{(t)}^K$

ALGORITHM 5.3. Update $V_{(t+1)}$ under Nesterov's momentum algorithm (NESTMOMENV)

1: $U_{(t+1)} \in \mathbb{R}^{r\times m}$, $V_{(t)}^0 \triangleq V_{(t)} \in \mathbb{R}^{r\times n}$
2: For $t^\dagger = 0, 1, 2, \ldots, K - 1$
3: 1.1. $\tilde{V}_{(t)}^{t^\dagger+1} = V_{(t)}^{t^\dagger} - \frac{1}{\beta}\nabla_V \tilde{\bar{\rho}}_{\delta,\tau}(b - \mathcal{A}((U_{(t)}^{t^\dagger})^\top V_{(t)}^{t^\dagger}))$
4: 1.2. $V_{(t)}^{t^\dagger+1} = (1 - \gamma_t)\tilde{V}_{(t)}^{t^\dagger+1} + \gamma\tilde{V}_{(t)}^{t^\dagger}$
5: Output $V_{(t+1)} = V_{(t)}^K$

Finally, it is worth noting that in our algorithm, $U_{(0)}$ and $V_{(0)}$ are not randomly initialized. In fact, when the initial values $U_{(0)}$ and $V_{(0)}$ are orthogonal or almost orthogonal to the true search space, there may exist a risk that the optimal values of U and V may never be reached (Jain, 2013). Therefore, before starting running the alternating optimization procedure, we need a initialization (warm-up) to set the proper $U_{(0)}, V_{(0)}$ in order to guarantee the optimality results to be presented in Section 6.

Here we utilize the singular value projection (SVP) for warm-up (Jain, 2010; Tu, 2016). However, we can not utilize their results directly, since in their work, SVP is proposed for the LSMF setting. Therefore, we revise the SVP algorithm to work for the proposed NsQMF setting. The detailed steps of the revised SVP are shown in Algorithm 5.4. The difference between Algorithm 5.4 and the original SVP (Jain, 2010) is that we have modified Step 2 (gradient descent) in the original SVP algorithm to adapt to the new objective in NsQMF.

ALGORITHM 5.4. Initialization by the (revised) SVP algorithm

1: Input $\mathcal{A}$, b, tolerance ε, step size ξ_t for $t = 0, 1, \cdots, M^0 = 0_{m\times n}$
2: Output M^{t+1}
3: Repeat
4: $N^{t+1} \leftarrow M^t - \xi_t \nabla_M \tilde{\bar{\rho}}_{\delta,\tau}(b - \mathcal{A}(M^t))$
5: Compute top r singular vectors of N^{t+1}:U_r, Σ_r, V_r
6: $M^{t+1} \leftarrow U_r \Sigma_r V_r$
7: $t \leftarrow t + 1$
8: Until $\|M^{t+1} - M^t\|_F \le \varepsilon$, denote $T_0 = t + 1$

Algorithm 5.4 can be rewritten in a compact form:

$$M^{t+1} \leftarrow \mathcal{P}_r\left(M^t - \xi_t \nabla_X \bar{\rho}_\tau^\delta(b - \mathcal{A}(M^t))\right),$$

where $\mathcal{P}_r$ represents projection onto the space of matrices of rank r. Interestingly, in the LSMF literature, although SVP can lead to optimal solutions (Jain, 2010), it is only used for initialization to guarantee certain desired theoretical properties of subsequent algorithms, as it demands intensive calculation. Here, we use SVP for the same purpose.

6 OPTIMALITY AND CONVERGENCE

In theoretical analysis, we assume the linear mapping $\mathcal{A}$ has following *restricted isometry property* (RIP):

DEFINITION 1. *(restricted isometry property) A linear mapping $\mathcal{A}$ satisfies the r-RIP with constant η_r if*

$$(1 - \eta_r)\|M\|_F^2 \le \|\mathcal{A}(M)\|_2^2 \le (1 + \eta_r)\|M\|_F^2$$

for all matrices $M \in \mathbb{R}^{m\times n}$ with Rank$(M) = r$.

In the following, we will show results on the equivalence between the smoothed problem (4.8) and the original non-smooth QMF problem (3.3) in terms of the objective functions attained as well as the optimal solutions. Let $\hat{U}^\delta, \hat{V}^\delta$ and $\hat{U}, \hat{V}$ be the optimal solutions to $\bar{\rho}_{\delta,\tau}(b - \mathcal{A}(U^\top V))$ and $\bar{\rho}_\tau(b - \mathcal{A}(U^\top V))$, respectively, where $(\hat{U}^\delta, \hat{V}^\delta) \in \mathcal{X}^*$, $(\hat{U}, \hat{V}) \in \mathcal{X}^*$. Then we have the following theorem:

THEOREM 6.1. *For any $b \in \mathbb{R}^p, U \in \mathbb{R}^{r\times m}, V \in \mathbb{R}^{r\times n}$,*

$$-\frac{\delta}{4} \le \bar{\rho}_{\delta,\tau}(b - \mathcal{A}(U^\top V)) - \bar{\rho}_\tau(b - \mathcal{A}(U^\top V)) \le 0. \tag{6.9}$$

Moreover,

$$-\frac{\delta}{4} \le \bar{\rho}_{\delta,\tau}(b - \mathcal{A}(\hat{U}^{\delta\top}\hat{V}^\delta)) - \bar{\rho}_\tau(b - \mathcal{A}(\hat{U}^\top\hat{V})) \le 0. \tag{6.10}$$

$$-\frac{\delta}{4} \le \tilde{\bar{\rho}}_{\delta,\tau}(b - \mathcal{A}(\hat{U}^{\delta\top}\hat{V}^\delta)) - \tilde{\bar{\rho}}_\tau(b - \mathcal{A}(\hat{U}^\top\hat{V})) \le 0. \tag{6.11}$$

As $\delta \to 0^+$, we have $\hat{U}^{\delta\top}\hat{V}^\delta \to \hat{U}^\top\hat{V}$.

Theorem 6.1 means that we can use NsQMF to solve original QMF, and they are equivalent when $\delta \to 0^+$. The advantage of NsQMF is that it is a smooth optimization problem which can be solved by gradient-based algorithms. In the meantime, the optimal solution to NsQMF will converge to the optimal solution to the original QMF problem when the smoothing parameter δ approaches zero. It is worth noting that while smooth approximations to the check loss have been considered by several authors (Aravkin *et al.*, 2014; Chen, 2007). However, to the best of our knowledge, no literature has formally proved that the smoothed problem will converge to the original non-smooth problem. As a result, the optimal solution of the QMF (3.3) is equivalent to the optimal solution of NsQMF (4.8).

Furthermore, we can establish a linear convergence rate for Algorithm 5.1. Define the distance between the two matrices $\begin{bmatrix} U_{(0)} \\ V_{(0)} \end{bmatrix}$ and $\begin{bmatrix} U \\ V \end{bmatrix}$ as

$$\begin{aligned} &\operatorname{dist}\left(\begin{bmatrix} U_{(0)} \\ V_{(0)} \end{bmatrix}, \begin{bmatrix} U \\ V \end{bmatrix}\right) \\ &= \min_{R\in\mathbb{R}^{r}: R^\top R = I_r} \left\| \begin{bmatrix} U_{(0)} \\ V_{(0)} \end{bmatrix} - \begin{bmatrix} U \\ V \end{bmatrix} R \right\|_F. \end{aligned}$$

Then we have the following theorem:

THEOREM 6.2. *Let $X \in \mathbb{R}^{m\times n}$ be a matrix with rank r. Let $X = A^\top \Sigma B$. Define $U = \Sigma^{1/2} A \in \mathbb{R}^{r\times m}$, $V = \Sigma^{1/2} B \in \mathbb{R}^{r\times n}$. $\mathcal{A}$ satisfies a rank-6r RIP condition with RIP constant $\sigma_{6r} < \frac{1}{25}$, $\xi_t = \frac{1}{p}$. Then starting from any initial solution that satisfies*

$$dist\left(\begin{bmatrix} U_{(0)} \\ V_{(0)} \end{bmatrix}, \begin{bmatrix} U \\ V \end{bmatrix}\right) \le \frac{1}{4}\sigma_r(U), \tag{6.12}$$

the t-th iteration of Algorithm 5.1 satisfies

$$dist\left(\begin{bmatrix} U_{(t)} \\ V_{(t)} \end{bmatrix}, \begin{bmatrix} U \\ V \end{bmatrix}\right) \le \frac{1}{4}\left(1 - \frac{\sqrt{2\lambda}}{\sqrt{L + 4\lambda}}\right)^t \sigma_r(U).$$

Theorem 6.2 states that linear convergence rate can be established for NsQMF. However, since the above results rely on an initial solution satisfying the condition (6.12), in the following, we show that Algorithm 5.4 is able to produce an initial solution that satisfies (6.12):

THEOREM 6.3. *Let $X \in \mathbb{R}^{m\times n}$ be a matrix with rank r, denote the singular values of X as $\sigma_1(X) \geq \sigma_2(X) \geq \cdots \geq \sigma_r(X)$, let $\kappa = \sigma_1(X)/\sigma_r(X)$ be the conditional number, let $X = A^\top \Sigma B$. Define $U = \Sigma^{1/2}A \in \mathbb{R}^{r\times m}$, $V = \Sigma^{1/2}B \in \mathbb{R}^{r\times n}$. $\mathcal{A}$ satisfies a rank-$6r$ RIP condition with RIP constant $\sigma_{6r} < \frac{1}{25}$, $\xi_t = \frac{1}{p}$, then performing $T_0 \geq 3\log(\sqrt{r}\kappa) + 5$ iterations of the initialization phase of Procrustes Flow yields a solution $U_{(0)}, V_{(0)}$ that satisfies*

$$dist\left(\begin{bmatrix} U_{(0)} \\ V_{(0)} \end{bmatrix}, \begin{bmatrix} U \\ V \end{bmatrix}\right) \leq \frac{1}{4}\sigma_r(U). \tag{6.13}$$

Theorem 6.3 implies that we can run T_0 iterations of Algorithm 5.4 to make the distance between $[U_{(0)}, V_{(0)}]^\top$ and $[U, V]^\top$ smaller than $1/4\sigma_r(U)$. The number of iterations T_0 can be chosen as the smallest positive integer that is larger than $3\log(\sqrt{r}\kappa) + 5$.

By combining Theorem 6.1 and 6.2, we are able to establish linear convergence rate in matrix factorization under a non-smooth check loss (or l_1 loss), whereas Park (2018) achieved linear convergence rate in MF only when the objective function is L-smooth and μ-restricted strongly convex.

7 EXPERIMENTS

In this section, we conduct numerical experiments to evaluate the recovery performance and convergence behaviour of the NsQMF in comparison to various baselines. We repeat the experiment 100 times. Each curve plots the median of them with IQR region.

7.1 Exact Recovery from Noiseless Observations First, we evaluate the performance of NsQMF on matrix recovery when observations are made noiseless. We intend to recover a ground-truth matrix M^* from observations $\{b_i, i = 1, \ldots, 2000\}$, where $b_i = \langle A_i, M^*\rangle$, and each A_i is a random observation matrix. The ground-truth matrix $M^* \in \mathbb{R}^{64\times 64}$ is randomly generated, where $M^* = U^{*\top}V^*$, $U^* \in \mathbb{R}^{6\times 64}$ and $V^* \in \mathbb{R}^{6\times 64}$. Each component of U^* and V^* is drawn randomly from a Gaussian distribution.

Fig. 7.1 shows the performance of exact recovery of NsQMF under $\tau = 0.2, 0.5, 0.8$, which corresponds to lower, median, and upper quantile, respectively. From Fig. 7.1 we can see that under the three scenarios, $\|\hat{M} - M^*\|_F$ converges to zero, which implies exact recovery under different quantile levels.

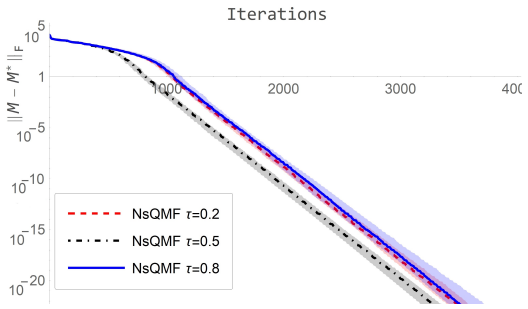


Figure 7.1: Convergence of $\|\hat{M} - M^*\|_F$ in logarithmic scale under noiseless observations.

Another phenomenon we observe in Fig. 7.1 is that there is a critical spot in all three curves. To be specific, when $\tau = 0.5$, $\|\hat{M} - M^*\|_F$ decreases slowly at the beginning and, after approximately 500 iterations, drops linearly to zero. In fact, $\|\hat{M} - M^*\|_F$ is in the order of 10^{-2} after 1000 iterations, and about 10^{-10} after 2000 iterations. Similarly, we can find such critical spot around 1000 iterations when $\tau = 0.2$ and 0.8. This phenomenon is largely due to our smoothing technique: after NsQMF runs for a certain number of iterations, solutions enter the smoothing area, which is a *locally* strongly convex valley, this phenomenon corroborates our theoretical result in Theorem 6.2, where linear convergence rate is established for NsQMF given a good initial solution satisfying (6.12).

7.2 Recovery From Heavy-Tailed Noise We now evaluate the matrix recovery performance of NsQMF when observations are made with heavy-tailed noise. In particular, we assume $b_i = \langle A_i, M^*\rangle + \epsilon_i$, where each A_i is a random observation matrix, and ϵ_i follows a heavy-tailed distribution. We compare our method with the following methods:

1. **LSMF**: the conventional MF scheme using L_2 loss, which aims at estimating the conditional mean;
2. **QMF** (Zhu, 2017): a QMF heuristic using $\sqrt{x^2 + \eta}$ for smooth approximation, where $\eta > 0$ is a smoothing constant. For simplicity, we call this method rough-smoothed QMF (RsQMF).

We use *relative recovery loss* (RRL), which is defined as $\|\hat{M} - M^*\|_F / \|M^*\|_F$, to compare the performance of these MF methods. We use RRL instead of $\|\hat{M} - M^*\|_F$ as the performance metric, because RRL for different methods are on the same scale. Similarly, we repeat each experiment for 100 times.

First, we assume the noise ϵ_i follows the Standard Cauchy distribution. The results are shown in Fig. 7.2. Although the RRL of LSMF decreases at a faster rate

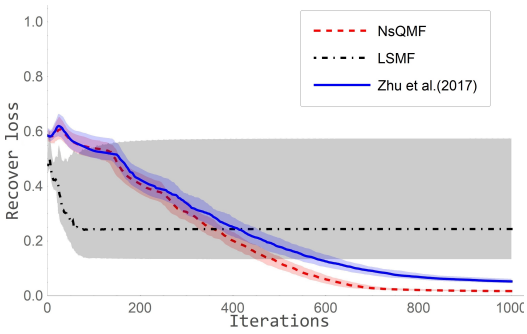


Figure 7.2: Convergence of RRL under Cauchy noise.

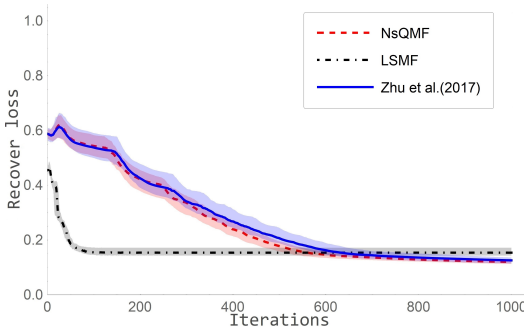


Figure 7.3: Convergence of RRL under log-normal noise.

initially, decreasing from around 0.5 to 0.23 in 20 iterations, the RRL of LSMF can not further decrease after 100 iterations, which shows that LSMF can not recover the ground-truth matrix M^* well under this type of noise. In contrast, the RRL of RsQMF or NsQMF approaches 0 at iteration 1000. Therefore, an L_2 loss is not suitable for recovering a matrix from observations with heavy-tailed noise. In this case, QMF is a more appealing alternative. Comparing the two QMF schemes, NsQMF and RsQMF, we can see that NsQMF converges faster than RsQMF. Specifically, it takes 600 iterations for the RRL of NsQMF to reduce from 0.6 to 0.05, while it takes RsQMF approximately 1000 iterations to achieve the same reduction in RRL.

We also performed experiments when ϵ_i follows a *log-normal* distribution, which is asymmetrically skewed and heavy-tailed. The results are shown in Fig. 7.3. We can see that the behavior of the three methods is similar to their behavior for Standard Cauchy noise. Specifically, the RRL of LSMF decreases fast initially but stays at a higher error, while NsQMF outperforms all other methods as it achieves the fastest decrease and convergence in RRL.

Next, we evaluate how RRL will change with respect to different quantile levels. Here we only consider the scenario when ϵ_i follows log-normal distribution. The results are presented in Fig. 7.4. The figure shows that the RRL is larger in upper and lower tails and is smaller in the middle part. In addition, RRL achieves minimum when $\tau = 0.55$. This is due to the fact that the center of the distribution is around its 55% quantile.

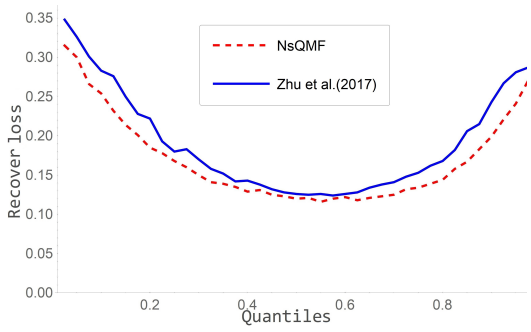


Figure 7.4: Relative recovery loss (RRL) after 1000 iterations under log-normal noise for different quantile levels. Each curve represents the median of 25 runs.

Then we investigate how different quantile levels affect recovery performance for NsQMF and RsQMF, by considering three quantile levels $\tau = 0.2, 0.5, 0.8$, when the noise is either Standard Cauchy distribution or log-normal distribution. Under log-normal distribution, we multiply the noise term by 15 to ensure that the noise is not negligible compared to the observations. The results are shown in Table 7.1. To make both methods comparable, we select the results at the 1000th iteration for RsQMF and NsQMF, respectively. From Table 7.1, we can see that NsQMF has achieved smaller recovery error than RsQMF under all the quantile levels. We also do further comparisons under more quantile levels under log-normal noise. The results are shown in Fig. 7.4, which shows that NsQMF outperforms RsQMF.

τ	Noise	NsQMF	RsQMF
0.2	*(a)*	0.0509	0.1176
	(b)	0.1881	0.2183
0.5	*(a)*	0.0215	0.0539
	(b)	0.1197	0.1282
0.8	*(a)*	0.0383	0.1134
	(b)	0.1483	0.1770

Table 7.1: RRL. *(a)*: Cauchy, *(b)*: log-normal.

Finally, we examine the effect of different learning rates under different noises. Here M is a $50\times$ 50 matrix, and $\text{rank}(M) = 5$. We run each simulation for 20,000 steps, the final error is measured by the relative recovery loss. For ith iteration, here we consider four different step sizes, (I) Small constant: 10^{-5}. (II) Small

decreasing: $10^{-5}\frac{100}{100+i}$. (III) Large constant: 10^{-4}. (IV) Large decreasing: $10^{-4}\frac{100}{100+i}$.

p/mn	outlier	I	II	III	IV
2	0	0	0	0	0
2	0.1	5.466	5.437	8.413	5.610
2	0.2	9.032	8.794	1299.257	8.878
2	0.3	11.843	13.002	1834.958	250.332
2	0.4	18.019	17.317	2181.498	673.469
2	0.5	28.017	27.134	2315.424	1191.906
8	0	0	0	0	0
8	0.1	2.417	2.446	102.623	2.281
8	0.2	3.640	3.613	329.327	3.881
8	0.3	5.376	5.119	453.162	5.154
8	0.4	7.473	7.132	652.517	7.251
8	0.5	9.515	9.809	742.683	9.240

Table 7.2: Convergence under different step sizes, each of the values in column 2-6 is multiplied by $\times 10^{-5}$

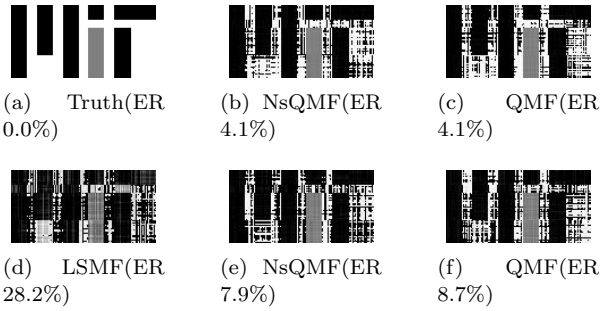
From Table 7.2, under different outlier percentages, we can see that when the sample size is small ($p/mn = 2$), smaller step size are preferred, and there is no significant difference between a constant step size and a decreasing one. When the sample size is large ($p/mn = 8$), apart from small constant and small decreasing step sizes, the performance of large decreasing step size is also comparable to previous two small step sizes as well.

8 MIT Logo Experiment

In this experiment, we use the MIT logo (a 128×64 grey-scale image) as the ground truth matrix. Our goal is to recover this image using NsQMF under noisy observations. The model to generate the observation is $b_i = \langle A_i, M^* \rangle + 2\epsilon_i$, where ϵ_i follows *Chi-squared* distribution with a degree of freedom of 3.

We also compare the performance of NsQMF with that of LSMF. The recovery results are shown in Fig. 8.1, where Fig. 8.1(a) represents the true image. Fig. 8.1(b) and (e) represent the recovered images for NsQMF under different quantile levels $\tau = 0.1$ and $\tau = 0.9$. Fig. 8.1(d) represents the image recovered by LSMF. We also present the images recovered by the original non-smooth QMF based on a *subgradient* algorithm[2]. The results are shown in Fig. 8.1(c) and (f).

From the figure, we can observe that LSMF has a high error rate of 28.2%. For NsQMF, they are 4.1%/7.9% when $\tau = 0.1/0.9$, which are significantly smaller than LSMF. This means the proposed NsQMF can handle heavy-tailed noise much better. For the non-smooth QMF based on subgradient descent, when $\tau = 0.1/0.9$, an error rate of 4.1%/8.7% are achieved, which is the same as/slightly higher than NsQMF.



(a) Truth(ER 0.0%) (b) NsQMF(ER 4.1%) (c) QMF(ER 4.1%)

(d) LSMF(ER 28.2%) (e) NsQMF(ER 7.9%) (f) QMF(ER 8.7%)

Figure 8.1: Recovery results on noisy real data. ER: error rate. (a) True image, (b/e) NsQMF, $\tau = 0.1/0.9$, (c/f) QMF with subgradient, $\tau = 0.1/0.9$, (d) LSMF.

9 CONCLUDING REMARKS

In this paper, we present an optimal solution to Quantile Matrix Factorization, the counterpart of Quantile Regression in the setting of matrix estimation, which is able to handle skewed observations and outliers in real data. We propose the NsQMF technique to solve quantile matrix factorization, which is a challenging non-smooth and non-convex matrix factorization problem. To handle the non-smoothness in the objective function, we use the Nesterov's smoothing method to obtain a smooth approximation of the check loss function in the original objective. We then apply an alternating minimization algorithm onto the smoothed approximation to approach the original non-smooth QMF problem. We theoretically show the equivalence of the smoothed NsQMF problem to the original QMF problem in terms of the objective functions attained as well as their optimal solutions. Specifically, we show that when the smoothing parameter δ in Nesterov's method approaches zero, the optimal solution to NsQMF will converge to the optimal solution to the original problem. Furthermore, under a simple initialization procedure, we establish the linear convergence rate for the proposed alternating minimization algorithm for NsQMF and that the convergence rate for NsQMF is known to be optimal. Numerical experiments have verified our theoretical findings and the algorithm efficiency, as well as the benefits of the proposed NsQMF method in a range of settings. Recently, Zhou *et al.* (2008) and Meira *et al.* (2018) studied scalable matrix factorization methods based on the alternative least squares, in the future, we intend to extend the existing methodology to allow for scalability and adaptive to large-scale computation.

[2]Since the objective function of QMF is non-smooth, gradient descent cannot be direclty applied. A subgradient descent algorithm is used instead.

References

Aravkin, Aleksandr Y, Kambadur, Anju, Lozano, Aurelie C, & Luss, Ronny. 2014. Sparse quantile Huber regression for efficient and robust estimation. *arXiv preprint arXiv:1402.4624*.

Barrodale, I, & Roberts, F. 1973. An improved algorithm for discrete l_1 linear approximation. *SIAM J Numer Anal*, **10**(5), 839–48.

Bertsimas, D, *et al.* 2014. Least quantile regression via modern optimization. *Ann Stat*, **42**(6), 2494–525.

Chen, C. 2007. A finite smoothing algorithm for quantile regression. *J Comput Graph Stat*, **16**(1), 136–64.

Chen, H, et al. 2016. Separating-Plane Factorization Models: Scalable Recommendation from One-Class Implicit Feedback. *Pages 669–78 of: CIKM'16*.

Chen, P, & Suter, D. 2004. Recovering the missing components in a large noisy low-rank matrix. *IEEE Trans Pattern Anal Mach Intell*, **26**(8), 1051–63.

Guan, Naiyang, Tao, Dacheng, Luo, Zhigang, & Shawe-Taylor, John. 2012. MahNMF: Manhattan non-negative matrix factorization. *Submitted to Journal of Machine Learning Research*.

Guminov, S et al. 2021. On a combination of alternating minimization and Nesterov's momentum. *Pages 3886–98 of: ICML*.

Hunter, D, & Lange, K. 2000. Quantile regression via an MM algorithm. *J Comput Graph Stat*, **9**(1), 60–77.

Jain, P et al. 2010. Guaranteed rank minimization via singular value projection. *Pages 937–45 of: NeurIPS*.

Jain, P et al. 2013. Low-rank matrix completion using alternating minimization. *Pages 665–74 of: STOC*.

Karatzoglou, A, & Weimer, M. 2010. Quantile matrix factorization for collaborative filtering. *Pages 253–64 of: ICEC*.

Koenker, R, & Bassett, G. 1978. Regression quantiles. *Econometrica*, 33–50.

Li, H, & Lin, Z. 2020. Provable accelerated gradient method for nonconvex low rank optimization. *Machine Learning*, **109**(1), 103–34.

Li, Xiao, Zhu, Zhihui, Man-Cho So, Anthony, & Vidal, Rene. 2020a. Nonconvex robust low-rank matrix recovery. *SIAM Journal on Optimization*, **30**(1), 660–686.

Li, Yuanxin, Chi, Yuejie, Zhang, Huishuai, & Liang, Yingbin. 2020b. Non-convex low-rank matrix recovery with arbitrary outliers via median-truncated gradient descent. *Information and Inference: A Journal of the IMA*, **9**(2), 289–325.

Meira, Dânia, Viterbo, José, & Bernardini, Flavia. 2018. An experimental analysis on scalable implementations of the alternating least squares algorithm. *Pages 351–359 of: 2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE.

Mitchell, D, et al. 2020. Nesterov acceleration of alternating least squares for canonical tensor decomposition: Momentum step size selection and restart mechanisms. *Numer Linear Algebra Appl*, **27**(4), e2297.

Nesterov, Y. 2005. Smooth minimization of non-smooth functions. *Mathematical programming*, **103**(1), 127–52.

Park, D, et al. 2017. Non-square matrix sensing without spurious local minima via the Burer-Monteiro approach. *Pages 65–74 of: Artificial Intelligence and Statistics*.

Park, D, et al. 2018. Finding low-rank solutions via nonconvex matrix factorization, efficiently and provably. *SIAM J Imaging Sci*, **11**(4), 2165–204.

Schönemann, P. 1966. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, **31**(1), 1–10.

Sun, R, & Luo, Z. 2015. Guaranteed matrix completion via nonconvex factorization. *Pages 270–89 of: FOCS*.

Tu, S, et al. 2016. Low-rank Solutions of Linear Matrix Equations via Procrustes Flow. *Pages 964–73 of: ICML*.

Tu, W, Liu, P, Liu, Yi, Li, G, Jiang, B, Kong, L, Yao, H, & Jui, S. 2021. Nonsmooth low-rank matrix recovery: methodology, theory and algorithm. *Pages 848–862 of: Proceedings of the Future Technologies Conference*. Springer.

Zheng, Z, et al. 2014. Investigating QoS of real-world web services. *IEEE Trans Serv Comput*, **7**(1), 32–39.

Zhou, Yunhong, Wilkinson, Dennis, Schreiber, Robert, & Pan, Rong. 2008. Large-scale parallel collaborative filtering for the netflix prize. *Pages 337–348 of: International conference on algorithmic applications in management*. Springer.

Zhu, R., et al. 2017. Robust web service recommendation via quantile matrix factorization. *Pages 1–9 of: INFOCOM*.