Matrix Completion under Interval Uncertainty

Citation for published version:
Mareek, J, Richtárik, P & Taká, M 2017, 'Matrix Completion under Interval Uncertainty' European Journal of Operational Research, vol. 256, no. 1, pp. 35-43. DOI: 10.1016/j.ejor.2016.07.014

Digital Object Identifier (DOI):
10.1016/j.ejor.2016.07.014

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
European Journal of Operational Research

Accepted Manuscript

Matrix Completion under Interval Uncertainty

Jakub Mareček, Peter Richtárik, Martin Takáč

PII: S0377-2217(16)30551-3
DOI: 10.1016/j.ejor.2016.07.014
Reference: EOR 13840

To appear in: *European Journal of Operational Research*

Received date: 5 September 2015
Revised date: 8 July 2016
Accepted date: 10 July 2016

Please cite this article as: Jakub Mareček, Peter Richtárik, Martin Takáč, Matrix Completion under Interval Uncertainty, *European Journal of Operational Research* (2016), doi: 10.1016/j.ejor.2016.07.014

Highlights

- Matrix completion under interval uncertainty can be encoded using box constraints.
- E.g. in collaborative filtering, a rating of 2 can be replaced by an interval 1-3.
- Present-best results in terms of statistical performance are provided for a benchmark.
- Convergence of randomised coordinate-descent methods to stationary points is proven.

Matrix Completion under Interval Uncertainty

Jakub Mareček[1,], Peter Richtárik[2], Martin Takáč[3]

Abstract

Matrix completion under interval uncertainty can be cast as a matrix completion problem with element-wise box constraints. We present an efficient alternating-direction parallel coordinate-descent method for the problem. We show that the method outperforms any other known method on a benchmark in image in-painting in terms of signal-to-noise ratio, and that it provides high-quality solutions for an instance of collaborative filtering with 100,198,805 recommendations within 5 minutes on a single personal computer.

Keywords: Matrix Completion, Robust Optimization, Collaborative Filtering, Coordinate Descent, Large-Scale Optimization, Non-Convex Optimization

1. Introduction

There has been much recent interest in non-convex optimization problems in statistics, data mining, and machine learning communities. Clearly, non-convex optimization is also at the heart of operations research [35], where considerable advances are being made, e.g., in decomposition approaches to non-convex optimization, and robust optimization [11]. In this paper, we present a decomposition approach to a robust variant of matrix completion, a key problem in data science, with numerous applications ranging from

Email address: `jakub.marecek@ie.ibm.com` (Jakub Mareček)

[1]IBM Research – Ireland, B3 F14 IBM Campus Damastown, Dublin 15, Ireland

[2]University of Edinburgh School of Mathematics, James Clerk Maxwell Building, Peter Guthrie Tait Road, Edinburgh EH9 3FD, Scotland, UK

[3]Lehigh University, Harold S. Mohler Laboratory, 200 West Packer Avenue, Bethlehem, PA 18015-1582, USA

image processing to recommender systems. This shows the value of advances in operations research to data science.

After an informal overview highlighting some key applications, we introduce the problem formally in Section 2. In Section 3, we present our algorithm and its convergence analysis. In Section 4, we present our computational results: In terms of statistical performance, our approach with an explicit consideration of the uncertainty, outperforms a number of previously proposed approaches to matrix completion, on a well-known benchmark. On the computational side, our implementation runs within minutes on a standard laptop even on instances with a $480,189 \times 17,770$ matrix with 100,198,805 non-zero entries, which had been previously [12, 52, 26, 28] solved on substantial clusters of computers in comparable times. We conclude with a variety of suggestions for future work.

1.1. An Informal Overview

When dimensions of a matrix X and some of its elements $X_{i,j}, (i,j) \in \mathcal{E}$ are known, the matrix completion problem is to find the unknown elements. Without imposing any further requirements on X, there are infinitely many solutions. Nevertheless, a matrix completion that minimizes the rank:

$$\min_Y \operatorname{rank}(Y) \quad \text{subject to} \quad Y_{i,j} = X_{i,j}, \quad (i,j) \in \mathcal{E}, \tag{1}$$

provides the simplest explanation for the known elements, in many applications. There is a long history of work on the problem, c.f. [9, 43, 56, 24], with thousands of papers published annually since 2010.

Although we cannot provide a complete overview, let us note that Fazel [10] suggested to replace the rank, which is the sum of non-zero elements of the spectrum, with the nuclear norm, which is the sum of the spectrum. The minimization of the nuclear norm can be cast as a semidefinite programming (SDP) problem and approaches based on the nuclear-norm have proven very successful in theory [6] and very popular in practice. [43, 3] study the Singular Value Thresholding (SVT) algorithm. This, however, required the computation of a singular value decomposition (SVD) in each iteration. A number of other approaches, e.g., augmented Lagrangian methods [53], appeared, but those would require a truncated SVD or a number of iterations [17, 25, 44, 54] of the power method. Even considering the recent progress in randomized methods for approximating SVD, [15], the approximation becomes very time-consuming as the dimensions of matrices grow.

A major computational break-through came in the form of the alternating least squares (ALS) algorithms [47, 38]. Initially, the algorithm has been used as a heuristic for finding stationary points of the non-convex problem [47, 38, 32, 2, 14], where a single iteration had complexity $O(|\mathcal{E}|r^2)$, for $|\mathcal{E}|$ observations and rank r, c.f., p. 60 in [22]. Keshavan et al. [23, 22], however, proved its exponential rate of convergence to the global optimum with high probability, under probabilistic assumptions common in the compressed sensing community. Independently, Cai et al. [3] analyzed matrix completion with an arbitrary convex constraint. Further, more technical analyses of the convergence to the global optimum have been performed by Jain et al. [19].

Many studies of matrix completion consider the uncertainty, in some form. A number of analyses [23, 22, 19] consider the use of the standard rank-minimization for the reconstruction of low-rank $m \times n$ matrix XY^T from $XY^T + W$, where $X \in \mathbb{R}^{m \times r}$, $Y \in \mathbb{R}^{n \times r}$, $W \in \mathbb{R}^{m \times n}$ with elements of W being bounded i.i.d. random variables, which are sub-Gaussian and have bounded expectation. A number of further analyses [55, 4] considered the use of the standard rank-minimization for the reconstruction of low-rank $m \times n$ matrix XY^T from $XY^T + S$, where X, Y are as above and W has a small number of non-zero entries. [8] consider some columns being corrupted. Although we are not aware of any studies of matrix completion under interval uncertainty, interval-based uncertainty has been considered in related problems. Alaiz et al. [1] consider the min-max variant of the problem of finding the nearest correlation matrix, i.e., the problem of finding the closest matrix within the set of symmetric positive definite matrices with the unit diagonal to an uncertainty set, with respect to the Frobenius norm. [27] studied interval uncertainty in certain semidefinite programming problems, which can be used to encode the nuclear-norm minimization.

We present an explicit extension of matrix completion towards interval uncertainty, which has applications in image in-painting, collaborative filtering, and beyond. The algorithm we present for solving the problem can be seen as a coordinate-wise version of the ALS algorithm, which does not require the approximation of the spectrum of the matrix. Before we proceed to describe the actual algorithm, we provide a motivating overview of the possible applications.

1.2. Collaborative Filtering under Uncertainty

Collaborative filtering is a well-established application of matrix completion problems [46], largely thanks to the success of the Netflix Prize. There is a matrix, where each row corresponds to one user and each column corresponds to a product or service. Considering that every user rates only a modest number of products or services, there are only a small number of entries of the matrix known. Our extension is motivated by the fact, that one user may provide two different ratings for one and the same product at two different times, depending on the current mood and other circumstances at the two times. One may hence want to consider an interval $[\underline{x}, \overline{x}]$ instead of a fixed value x of the rating, e.g., $[x - \epsilon, x + \epsilon]$. Further, when one knows the scale $[0, M]$ the rating x is chosen from, one can consider $[\max\{0, x - \epsilon\}, \min\{x + \epsilon, M\}]$. Hence, if intervals are known for elements $X_{i,j}$ of a matrix X indexed by $(i, j) \in \mathcal{I}$, one may want to solve:

$$\begin{aligned} &\min\nolimits_{Y_{i,j}\in[0,M]}\max\nolimits_{X_{i,j}\in[\underline{X_{i,j}},\overline{X_{i,j}}]\forall(i,j)\in\mathcal{I}} \operatorname{rank}(Y) \\ &\qquad\qquad\qquad\qquad \text{subject to } Y_{i,j} = X_{i,j}, \quad \forall (i,j) \in \mathcal{I}. \end{aligned} \tag{2}$$

Although numerous extensions of matrix completion problems have been studied, e.g. [31], the use of robustness to interval uncertainty is novel. It can be seen as an extension of robust optimization [45] to matrix completion.

1.3. Image In-Painting

Further applications can be found in image processing. In in-painting problems, a subset of pixels from an image are given and the goal is to fill in the missing pixels. Rank-constrained matrix completion with equalities, where $\mathcal{I}$ is the index set of all known pixels, has been used numerous times [6, 18, 30, 13, 25, 17, 54, 54] in this setting. If the image comes from real sensors, it the corresponding matrix may have full (numerical) rank, but have quickly decreasing singular values in its spectrum. In such a case, instead of solving the equality-constrained problem (1), one should like to find a low-rank approximation Y^* of X, such that the known entry of X is not far away from Y^*, i.e., $\forall (i, j) \in \mathcal{I}$ we have $Y_{i,j} \approx X_{i,j}$. Let us illustrate this with a small matrix

$$X = \begin{pmatrix} 68.16 & 78.12 & 24.04 \\ 78.12 & 90.09 & 30.03 \\ 24.04 & 30.03 & 20.01 \end{pmatrix},$$

which has rank 3 and its singular values $\Sigma = (167.9945, 10.2553, 0.0102)^T$. It is easy to verify that

$$Y^*(2) = \begin{pmatrix} 68.1546 & 78.1250 & 24.0389 \\ 78.1250 & 90.0853 & 30.0310 \\ 24.0389 & 30.0310 & 20.0098 \end{pmatrix}$$

is the best rank 2 approximation of X in Frobenius norm. Observe that no single element of $Y^*(2)$ is identical to X, but that $Y^*(2) \approx X$. It is an easy exercise to show that for any $X \in \mathbb{R}^{m\times n}$ with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min\{m,n\}}$, and $Y^*(r)$ as its best rank-r approximation, we have $|X_{i,j} - (Y^*(r))_{i,j}| \leq \sum_{i=r+1}^{\min\{m,n\}} \sigma_i =: \mathcal{R}(r)$ for all (i,j). Therefore, one should not require equality constrains in (1), but rather inequalities $|Y_{i,j} - X_{i,j}| \leq \mathcal{R}(r), \forall (i,j) \in \mathcal{I}$. Notice that this approach is not the same as minimizing $\sum_{(i,j)\in\mathcal{I}} (X_{i,j} - Y_{i,j})^2$ over all rank r matrices, because we do not penalize the elements of Y, which are already close to X. It is also different from the usual treatment of noise in the observations [5]. One could rather formulate this as the minimization of $\sum_{(i,j)\in\mathcal{I}} \max\{0, |X_{i,j} - Y_{i,j}| - \mathcal{R}(r)\}^2$ over all rank r matrices. Further, one knows the range of values allowed, e.g., $[0,1]$ for common encoding of gray-scale images. This can hence be seen as "side information" which, as we will show in numerical section, improves recovery of a low-rank approximation considerably. Further still, one could assume that the intensity should be at least 0.8, if pixels are missing within a light region of the image, or similar domain-specific heuristics.

A number of other applications, e.g., in the recovery of structured matrices [7], in certain forecasting problems with periodic time series and side information, and in sparse principal component analysis with priors on the principal components can be envisioned. Some are discussed in Section 5. Now, let us introduce our notation and formalize the problem.

2. The Problem

Formally, let X be an $m \times n$ matrix to be reconstructed. Assume that elements $(i,j) \in \mathcal{E}$ of X we wish to fix, for elements $(i,j) \in \mathcal{L}$ we have lower bounds and for elements $(i,j) \in \mathcal{U}$ we have upper bounds. We employ the following natural formulation for the equality and inequality constrained matrix completion problem:

$$\begin{array}{ll} \min\limits_{X\in\mathbb{R}^{m\times n}} & \operatorname{rank}(X) \\ \text{subject to} & X_{ij} = X^{\mathcal{E}}_{ij},\ (i,j)\in\mathcal{E} \\ & X_{ij} \geq X^{\mathcal{L}}_{ij},\ (i,j)\in\mathcal{L} \\ & X_{ij} \leq X^{\mathcal{U}}_{ij},\ (i,j)\in\mathcal{U}. \end{array} \tag{3}$$

We shall enforce the following natural assumption:

Assumption 1. $\mathcal{E}\cap(\mathcal{L}\cup\mathcal{U})=\emptyset$ *and* $X^{\mathcal{L}}_{ij}\leq X^{\mathcal{U}}_{ij}$ *whenever* $(ij)\in\mathcal{L}\cap\mathcal{U}$.

The first condition says that if some element (ij) is already fixed by an equality constraint, it does not (unnecessarily) appear any of the inequality constraints. The second condition says the upper and lower bounds should be consistent.

Problem (3) is NP-hard, even with $\mathcal{U}=\mathcal{L}=\emptyset$ [33, 16]. A number of special cases of (3) have been studied in the literature, e.g., in [43, 37, 21]. A popular heuristic enforces low rank in a synthetic way by writing X as a product of two matrices, $X = LR$, where $L\in\mathbb{R}^{m\times r}$ and $R\in\mathbb{R}^{r\times n}$. Hence, X is of rank at most r [49]. Let $L_{i:}$ and $R_{:j}$ be the i-th row and j-th column of L and R, respectively. Instead of (3), we consider the *smooth, non-convex* problem

$$\min\{f(L,R)\ :\ L\in\mathbb{R}^{m\times r},\ R\in\mathbb{R}^{r\times n}\}, \tag{4}$$

where

$$f(L,R) := \tfrac{\mu}{2}\|L\|_F^2 + \tfrac{\mu}{2}\|R\|_F^2 + f_{\mathcal{E}}(L,R) + f_{\mathcal{L}}(L,R) + f_{\mathcal{U}}(L,R). \tag{5}$$

Above we have

$$\begin{array}{rcl} f_{\mathcal{E}}(L,R) & := & \frac{1}{2}\sum_{(ij)\in\mathcal{E}}(L_{i:}R_{:j} - X^{\mathcal{E}}_{ij})^2 \\ f_{\mathcal{L}}(L,R) & := & \frac{1}{2}\sum_{(ij)\in\mathcal{L}}(X^{\mathcal{L}}_{ij} - L_{i:}R_{:j})_+^2 \\ f_{\mathcal{U}}(L,R) & := & \frac{1}{2}\sum_{(ij)\in\mathcal{U}}(L_{i:}R_{:j} - X^{\mathcal{U}}_{ij})_+^2, \end{array}$$

where $\xi_+ = \max\{0,\xi\}$.

The parameter $\mu > 0$ helps to prevent scaling issues[4]. We could optionally set μ to zero and instead, from time to time, rescale matrices L and R, so

[4]Let $X = LR$, then also $X = (cL)(\frac{1}{c}R)$ as well, but we see that for $c\to 0$ or $c\to\infty$ we have $\|L\|_F^2 + \|R\|_F^2 \ll \|cL\|_F^2 + \|\frac{1}{c}R\|_F^2$.

Input: $\mathcal{E}, \mathcal{L}, \mathcal{U}, X^{\mathcal{E}}, X^{\mathcal{L}}, X^{\mathcal{U}}$, rank r
Output: $m \times n$ matrix LR
1: choose $L \in \mathbb{R}^{m \times r}$ and $R \in \mathbb{R}^{r \times n}$
2: **for** $k = 0, 1, 2, \dots$ **do**
3: choose random subset $\hat{\mathcal{S}}_{\text{row}} \subset \{1, \dots, m\}$
4: **for** $i \in \hat{\mathcal{S}}_{\text{row}}$ **in parallel do**
5: choose $\hat{r} \in \{1, \dots, r\}$ uniformly at random
6: compute $\delta_{i\hat{r}}$ using formula (8)
7: update $L_{i\hat{r}} \leftarrow L_{i\hat{r}} + \delta_{i\hat{r}}$
8: **end for**
9: choose random subset $\hat{\mathcal{S}}_{\text{column}} \subset \{1, \dots, n\}$
10: **for** $j \in \hat{\mathcal{S}}_{\text{column}}$ **in parallel do**
11: choose $\hat{r} \in \{1, \dots, r\}$ uniformly at random
12: compute $\delta_{\hat{r}j}$ using (11)
13: update $R_{\hat{r}j} \leftarrow R_{\hat{r}j} + \delta_{\hat{r}j}$
14: **end for**
15: **end for**

Algorithm 1: MACO: Matrix Completion via Alternating Parallel Coordinate Descent

that their product is not changed [49]. The term $f_{\mathcal{E}}$ (resp. $f_{\mathcal{U}}$, $f_{\mathcal{L}}$) encourages the equality (resp. inequality) constraints to hold.

3. The Method

Coordinate descent algorithms (CDA) are effective in solving large-scale problems, due to their low per-iteration computational cost. Although each iteration of CDA is cheap, many more iterations are required for convergence, compared to second-order algorithms or similar. Recently, the stochastic CDA has received much attention [34, 39] not least due to the parallelizability [42, 41, 50, 51] with almost linear speed-up in regimes with sparse data, when the number of parallel updates τ is much smaller that the dimension of the optimization problem [37]. Distributed variants have also been studied [29, 40].

In Algorithm 1, we present our alternating parallel coordinate descent method for MAtrix COmpletion, henceforth simply "MACO". In Steps 3–8 of our algorithm, we fix R, choose random $\hat{r}$ and a random set $\hat{\mathcal{S}}_{\text{row}}$ of rows of L, and update, in parallel for $i \in \hat{\mathcal{S}}_{\text{row}}$: $L_{i\hat{r}} \leftarrow L_{i\hat{r}} + \delta_{i\hat{r}}$. In Steps 9–14, we fix L, choose random $\hat{r}$ and a random set $\hat{\mathcal{S}}_{\text{column}}$ of columns of R, and update, in parallel for $j \in \hat{\mathcal{S}}_{\text{column}}$: $R_{\hat{r}j} \leftarrow R_{\hat{r}j} + \delta_{\hat{r}j}$.

Let us now comment on the computation of the updates, $\delta_{i\hat{r}}$ and $\delta_{\hat{r}j}$. First, note that while f is not convex jointly in (L, R), it is convex in L for fixed R and in L for fixed R.

3.1. Row Update

If we now fix row $i \in \{1, 2, \ldots, m\}$ and $\hat{r} \in \{1, 2, \ldots, r\}$, and view f as a function of $L_{i\hat{r}}$ only, it has a Lipschitz continuous derivative with constant

$$W_{i\hat{r}} = W_{i\hat{r}}(R) := \mu + \sum_{j\,:\,(ij)\in\mathcal{E}} R^2_{\hat{r}j} + \sum_{j\,:\,(ij)\in\mathcal{L}\cup\mathcal{U}} R^2_{\hat{r}j}. \tag{6}$$

That is, for all L, R and $\delta \in \mathbb{R}$, we have

$$f(L + \delta E_{i\hat{r}}, R) \le f(L, R) + \langle \nabla_L f(L, R), E_{i\hat{r}} \rangle \delta + \frac{W_{i\hat{r}}}{2} \delta^2, \tag{7}$$

where $E_{i\hat{r}}$ is the $n \times r$ matrix with 1 in the $(i\hat{r})$ entry and zeros elsewhere. The minimizer of the right hand side of (7) in δ is given by

$$\delta_{i\hat{r}} := -\langle \nabla_L f(L, R), E_{i\hat{r}} \rangle / W_{i\hat{r}}, \tag{8}$$

where

$$\begin{aligned} \langle \nabla_L f(L, R), E_{i\hat{r}} \rangle = \mu L_{i\hat{r}} \quad &+ \sum_{j\,:\,(ij)\in\mathcal{E}} (L_{i:}R_{:j} - X^{\mathcal{E}}_{ij}) R_{\hat{r}j} \\ &+ \sum_{j\,:\,(ij)\in\mathcal{U}\ \&\ L_{i:}R_{:j} < X^{\mathcal{U}}_{ij}} (L_{i:}R_{:j} - X^{\mathcal{U}}_{ij}) R_{\hat{r}j} \\ &+ \sum_{j\,:\,(ij)\in\mathcal{L}\ \&\ L_{i:}R_{:j} > X^{\mathcal{L}}_{ij}} (L_{i:}R_{:j} - X^{\mathcal{L}}_{ij}) R_{\hat{r}j}. \end{aligned}$$

Note that

$$f(L + \delta_{i\hat{r}} E_{i\hat{r}}, R) \le f(L, R) - \frac{\langle \nabla_L f(L, R), E_{i\hat{r}} \rangle^2}{2W_{i\hat{r}}}. \tag{9}$$

Let $W^{(k)}_{i\hat{r}} := W_{i\hat{r}}(R^{(k)})$ be the value of the Lipschitz constant at iteration k.

3.2. Column Update

Likewise, if we now fix $\hat{r} \in \{1, 2, \ldots, r\}$ and column $j \in \{1, 2, \ldots, n\}$, and view f as a function of $R_{\hat{r}j}$ only, it has a Lipschitz continuous derivative with constant

$$V_{\hat{r}j} = V_{\hat{r}j}(L) := \mu + \sum_{i\,:\,(ij)\in\mathcal{E}} L_{i\hat{r}}^2 + \sum_{i\,:\,(ij)\in\mathcal{U}\cup\mathcal{L}} L_{i\hat{r}}^2.$$

That is, for all L, R and $\delta \in \mathbb{R}$,

$$f(L, R + \delta E_{\hat{r}j}) \leq f(L, R) + \langle \nabla_R f(L, R), E_{\hat{r}j}\rangle \delta + \frac{V_{\hat{r}j}}{2}\delta^2, \tag{10}$$

where $E_{\hat{r}j}$ is the $r \times m$ matrix with 1 in the $(\hat{r}j)$ entry and zeros elsewhere. The minimizer of the right hand side of (10) in δ is given by

$$\delta_{\hat{r}j} := -\langle \nabla_R f(L, R), E_{\hat{r}j}\rangle / V_{\hat{r}j}, \tag{11}$$

where

$$\begin{aligned}\langle \nabla_R f(L, R), E_{\hat{r}j}\rangle = \mu R_{\hat{r}j} \quad &+ \sum_{i\,:\,(ij)\in\mathcal{E}} (L_{i:}R_{:j} - X_{ij}^{\mathcal{E}})L_{i\hat{r}} \\ &+ \sum_{i\,:\,(ij)\in\mathcal{L}\ \&\ L_{i:}R_{:j}<X_{ij}^{\mathcal{L}}} (L_{i:}R_{:j} - X_{ij}^{\mathcal{L}})L_{i\hat{r}} \\ &+ \sum_{i\,:\,(ij)\in\mathcal{U}\ \&\ L_{i:}R_{:j}>X_{ij}^{\mathcal{U}}} (L_{i:}R_{:j} - X_{ij}^{\mathcal{U}})L_{i\hat{r}}.\end{aligned}$$

Note that

$$f(L, R + \delta_{\hat{r}j}E_{\hat{r}j}) \leq f(L, R) - \frac{\langle \nabla_R f(L, R), E_{\hat{r}j}\rangle^2}{2V_{\hat{r}j}}. \tag{12}$$

Let $V_{\hat{r}j}^{(k)} := W_{\hat{r}j}(L^{(k)})$ be the value of the Lipschitz constant at iteration k.

3.3. Row and Column Sampling

The random set ("sampling") $\hat{\mathcal{S}}_{\text{row}}$ defined in Step 3 (resp sampling $\hat{\mathcal{S}}_{\text{column}}$ in Step 10) can have an arbitrary distribution as long as it contains every row (resp column) of matrix L (resp R) with positive probability. We shall now formalize this.

Assumption 2. *The samplings* $\hat{\mathcal{S}}_{\text{row}}$ *and* $\hat{\mathcal{S}}_{\text{column}}$ *are* proper*, i.e.,*

$$\mathbf{Prob}(i \in \hat{\mathcal{S}}_{\text{row}}) > 0 \quad \textit{for all} \quad i \in \{1, 2, \ldots, m\},$$

and

$$\mathbf{Prob}(j \in \hat{\mathcal{S}}_{\text{column}}) > 0 \quad \textit{for all} \quad j \in \{1, 2, \ldots, n\}.$$

In particular, we can chose the random sets $\hat{\mathcal{S}}_{\text{row}}$ (resp $\hat{\mathcal{S}}_{\text{column}}$) so that every row (resp column) has equal probability of being chosen. Samplings with this property are called *uniform*, and we use this choice in our experiments. However, our theory also allows for nonuniform samplings. If we have a multicore machine available with τ cores, then a reasonable sampling should have cardinality τ, or some integral multiple of τ, so that every core has a reasonable (not too small to be underutilized, but not too large either, so as to avoid long processing time) load at every iteration.

3.4. The Final Step

Formulae (8) and (11) suggest that the computation of the final step is very computationally demanding. This can, however, be avoided if we define matrices $A \in \mathbb{R}^{m\times r}$ and $B \in \mathbb{R}^{r\times n}$ such that $A_{iv} = W_{iv}$ and $B_{vj} = V_{vj}$. After each update of the solution, we can also update those matrices. Similarly, one can store sparse residuals matrices $\Delta_{\mathcal{E}}$, $\Delta_{\mathcal{L}}$, $\Delta_{\mathcal{U}}$, where

$$(\Delta_{\mathcal{E}})_{i,j} = \begin{cases} L_{i:}R_{:j} - X^{\mathcal{E}}_{ij}, & \text{if } (ij) \in \mathcal{E} \\ 0, & \text{otherwise,} \end{cases}$$

and $\Delta_{\mathcal{U}}$, $\Delta_{\mathcal{L}}$ are defined in similar way. Subsequently, the computation of $\delta_{i\hat{r}}$ or $\delta_{\hat{r}j}$ is reduced to just a few multiplications and additions.

3.5. Convergence Analysis

Due to the non-convex nature of (4), one has to be satisfied with convergence to a stationary point, in general.

Theorem 1. *Let* $\mu > 0$ *and and let* $(L^{(k)}, R^{(k)})$ *be the (random) matrices produced by Algorithm 1 after* k *iterations, assuming that* $\hat{\mathcal{S}}_{\text{row}}$ *and* $\hat{\mathcal{S}}_{\text{column}}$ *are proper. Then for all* $k \geq 0$*,*

$$0 \leq f(L^{(k+1)}, R^{(k+1)}) \leq f(L^{(k)}, R^{(k)}). \tag{13}$$

That is, the method is monotonic. Moreover, with probability 1,

$$\lim_{k\to\infty} \inf \|\nabla_L f(L^{(k)}, R^{(k)})\| = 0,$$

and

$$\lim_{k\to\infty} \inf \|\nabla_R f(L^{(k)}, R^{(k)})\| = 0.$$

We refer to the appendix for the proof.

4. Computational Results and a Discussion

We have conducted a variety of experiments. First, we present the performance in collaborative filtering, next we compare the performance in image in-painting with classical matrix completion techniques with $\mathcal{U} \equiv \mathcal{L} \equiv \emptyset$. We conclude with remarks on the run-time and hardware used.

4.1. Collaborative Filtering

In our computational testing of collaborative filtering, we start with `smallnetflix_mm`, where the training dataset contains $c_{\text{tr}} = 3,298,163$ integers out of $\{1, 2, 3, 4, 5\}$, which describe how $m = 95,526$ users rate $n = 3,561$ movies. Second, we use a well-known data-set, which contains $100,198,805$ ratings on the same scale, obtained from $480,189$ users considering $17,770$ products, as available from CMU[5]. Third, we use Yelp's Academic Dataset[6], from which we have extracted a $252,898 \times 41,958$ matrix with 1,125,458 non-zeros, again on the 1–5 scale.

Although we know some ratings exactly on `smallnetflix_mm`, we consider (4) of (3) with interval uncertainty sets of width 2:

$$\begin{aligned} Y_{i,j} &\leq \min\{5, X_{i,j} + 1\}, (i, j) \in \mathcal{I}, \\ Y_{i,j} &\geq \max\{1, X_{i,j} - 1\}, (i, j) \in \mathcal{I}. \end{aligned} \tag{14}$$

In particular, we complete a 95526×3561 matrix of rank 2 or 3, possibly using width-2 interval uncertainty set and scale of 1 to 5 stars in the ratings. To illustrate the impact of the this change, we present the evolution of Root-Mean-Square Error (RMSE) in Figure 2 (left). Notice that an "epoch", which

[5] http://www.select.cs.cmu.edu/code/graphlab/datasets/

[6] https://www.yelp.co.uk/academic_dataset

is the unit on the horizontal axis, consists of c_{tr} element updates of matrix L and c_{tr} element updates of matrix R.

Let us remark that RMSE is sensitive to the choice of Δ and the rank of the matrix we are looking for. If the underlying matrix has a higher rank than expected, $\Delta > 0$ can lead to smaller values of RMSE. We should also note that for some fixed Δ_1 and Δ_2, RMSE can be better with Δ_1 for a few epochs, but then get worse when compared with Δ_2. Hence, in practice, cross validation should be used to determine suitable value of parameter Δ.

On the Yelp data set, we have performed 10-fold cross-validation on the training set, using varying rank. As we increased the rank from 1 to 2, 4, 8, 16, 32, and 50, the average error decreased from 1.7958 to 1.8284, 1.6464, 1.4590, 1.3395, 1.2702, and 1.2454, respectively. This seems to be comparable to the best results from the 2013 Recommender Systems Challenge[7], where a smaller dataset was used.

Further, one can illustrate the effects in a matrix-recovery experiment. We use random matrices $X \in \mathbb{R}^{20\times 20}$ of rank 8. We sample $p\%$ of entries of the matrix and store their indices in $\mathcal{I}$. We solve (4) with just the inequality constrains, i.e., $\mathcal{E} \equiv \emptyset, \mathcal{U} \equiv \mathcal{L} \equiv \mathcal{I}$, $X^{\mathcal{U}} = X - \Delta\mathbf{1}$ and $X^{\mathcal{L}} = X + \Delta\mathbf{1}$, where $\mathbf{1} \in \mathbb{R}^{m\times n}$ is a matrix with all elements equals to 1. Let us denote by $Y^*(\Delta)$ the solution of that optimization problem after 10^5 serial iterations ($|\hat{\mathcal{S}}| = 1$) and with $\mu = 10^{-5}$. Figure 1 shows the dependence of error defined as follows $\text{Error}(\Delta) = \frac{\|Y^*(\Delta) - X(7)\|_F}{\|X(7)\|_F}$, where $X(r)$ is the best rank r approximation of X obtain using SVD decomposition of the whole matrix. Figure 1 clearly suggest that, e.g., if 50% of elements are observed then by allowing each entry $\in \mathcal{I}$ of reconstructed matrix to lie in Δ neighborhood of observed values, we can decrease the relative error of reconstruction from approximately 1.22 to 0.4 for $\Delta \approx \mathcal{R}(r)$. In this case, the value of $\|X(7)\|_F$ was 21.3245 and $\mathcal{R}(r) = 0.1075$.

4.2. Image In-Painting

Further, we provide a comparison on the in-painting benchmark of [54]. Table 1 details the performance of SVT [6], SVP [18], SoftImpute [30], LMaFit [13], ADMiRA, [25], JS [17], OR1MP [54], and EOR1MP [54] on 10 well-known gray-scale images (Barbara, Cameraman, Clown, Couple, Crowd, Girl, Goldhill, Lenna, Man, Peppers) of 512×512 pixels each. 50% of pix-

[7] `https://www.kaggle.com/c/yelp-recsys-2013`

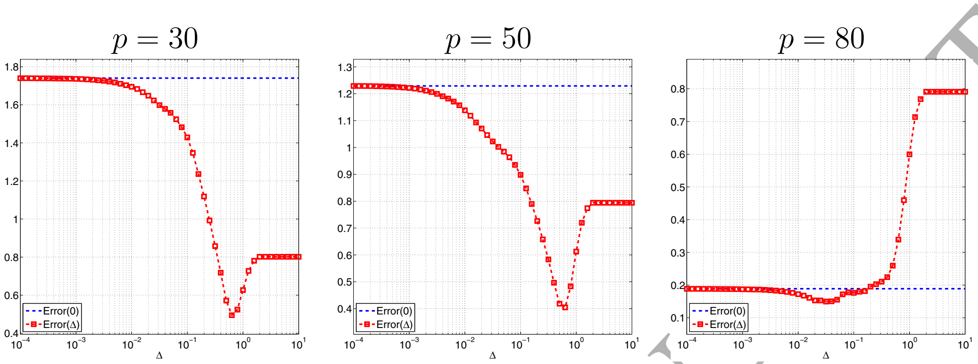

Figure 1: Dependence of Error on Δ for various $p \in \{30, 50, 80\}$ in matrix reconstruction.

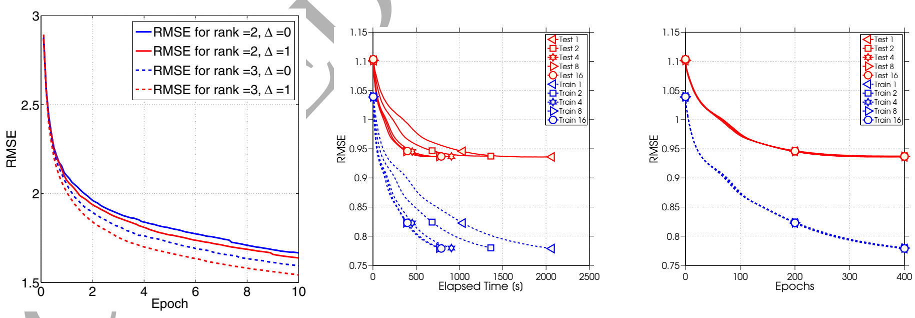

Figure 2: Left: The effect of adding inequalities ($\Delta = 1$) to the equality-constrained problem ($\Delta = 0$) on `smallnetflix`, for $r = 2, 3$, $\mu = 10^{-3}$. Center and right: RMSE as a function of the number of iterations and wall-clock time, respectively, on a well-known 480189×17770 matrix, for $r = 20$ and $\mu = 16$.

els were removed uniformly at random, and the image was reconstructed using rank 50. The performance was measured in terms of PSNR, which is $10 \log_{10}(255^2/E)$ for mean squared error E, with the results of all approaches but ours cited from the literature [54]. Our approach with inequalities $0 \leq Y_{i,j} \leq 255$ dominates all other approaches on 7 out of the 10 images. On the remaining 3 images, one would have to use the extrema of the observed elements, e.g., a subinterval of 12–246 for Barbara.

To illustrate the aggregate results further, we undertook the following experiment. We took a 512×512 gray scale image (Lenna) and chose 50% of the pixels randomly, indexed as $\mathcal{I}$. Then, we ran Algorithm 1 for 10^7 serial iterations ($|\hat{\mathcal{S}}| = 1$). We obtained solutions $X_E(\text{rank})$ and $X_{IN}(\text{rank})$, where $X_E(\text{rank})$ was obtained when we used only equality constrains ($\mathcal{E} = \mathcal{I}, \mathcal{U} \equiv \mathcal{L} \equiv \emptyset$) and $X_{IN}(\text{rank})$ was obtained when we used also inequality constrains ($\mathcal{E} = \mathcal{I}$, $\mathcal{U} \equiv \mathcal{L} \equiv -\mathcal{I}$, $X^{\mathcal{L}} = \mathbf{0} \in \mathbb{R}^{512\times512}$, $X^{\mathcal{U}} = \mathbf{1} \in \mathbb{R}^{512\times512}$ and $-\mathcal{I}$ is a set of all elements of X except those in $\mathcal{I}$). Figure 3 shows for different rank $\in \{30, 50, 100\}$ the best rank approximation obtained by SVD ($X(\text{rank})$) and solutions $X_E(\text{rank})$ and $X_{IN}(\text{rank})$. The benefit of obvious inequality constrains is nicely visible, e.g., at rank $= 100$, where the relative error of reconstruction is more than twice smaller. Further, the image is more smooth, upon visual inspection.

rank	$X(\text{rank})$	$X_E(\text{rank})$	$X_{IN}(\text{rank})$
30	 $\|X(\text{rank})\|_F = 223.9999$	 $Error = 13.1394$	 $Error = 12.6303$
50	 $\|X(\text{rank})\|_F = 224.6876$	 $Error = 18.2070$	 $Error = 13.1859$
100	 $\|X(\text{rank})\|_F = 225.2117$	 $Error = 39.1631$	 $Error = 15.2551$

Figure 3: Adding obvious constraints can help to get better solution. Error is defined as $Error := \|X(rank) - X\|_F$.

Table 1: Comparison with other solvers on the image recovery in terms of the peak signal-to-noise ratio (PSNR), citing the experiments of Wang et al. and adding results considering $0 \leq Y_{i,j} \leq 255$ under "MACO".

Instance / Algo.	SVT	SVP	SoftImpute	LMaFit	ADMiRA	JS	OR1MP	EOR1MP	MACO
Barbara	**26.9635**	25.2598	25.6073	25.9589	23.3528	23.5322	26.5314	26.4413	23.8015
Cameraman	25.6273	25.9444	26.7183	24.8956	26.7645	24.6238	27.8565	27.8283	**28.9670**
Clown	28.5644	19.0919	26.9788	27.2748	25.7019	25.2690	28.1963	28.2052	**29.0057**
Couple	23.1765	23.7974	26.1033	25.8252	25.6260	24.4100	27.0707	27.0310	**27.1824**
Crowd	**26.9644**	22.2959	25.4135	26.0662	24.0555	18.6562	26.0535	26.0510	26.1705
Girl	29.4688	27.5461	27.7180	27.4164	27.3640	26.1557	30.0878	30.0565	**30.4110**
Goldhill	28.3097	16.1256	27.1516	22.4485	26.5647	25.9706	28.5646	28.5101	**28.6265**
Lenna	28.1832	25.4586	26.7022	23.2003	26.2371	24.5056	28.0115	27.9643	**28.3581**
Man	**27.0223**	25.3246	25.7912	25.7417	24.5223	23.3060	26.5829	26.5049	26.5990
Peppers	25.7202	26.0223	26.8475	27.3663	25.8934	24.0979	28.0781	28.0723	**28.8469**

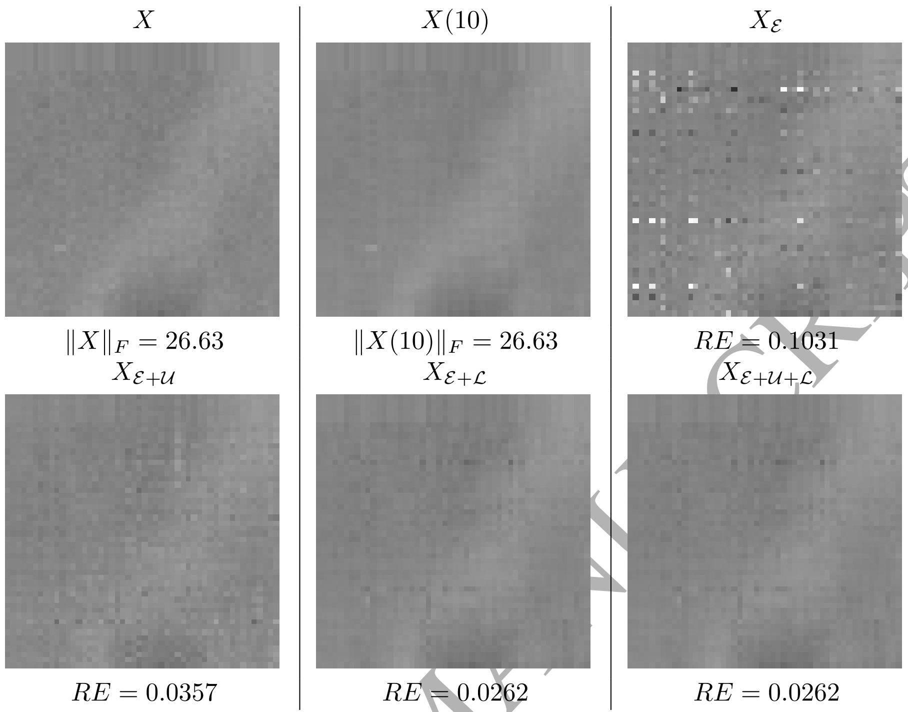

Figure 4: Original 50×50 image, the best rank 10 approximation and reconstruction using Algorithm 1 with different settings. The RE is a relative error defined as $RE(X.) = \|X. - X(10)\|_F / \|X(10)\|$.

Further, we took a 50×50 image and sampled randomly 50% of pixels. (The image is the top-left corner of the Lenna image.) Figure 4 shows the original image X and the best rank 10 approximation $X(10)$. The solutions $X_{\mathcal{E}}$, $X_{\mathcal{E}+\mathcal{U}}$, $X_{\mathcal{E}+\mathcal{L}}$ and $X_{\mathcal{E}+\mathcal{U}+\mathcal{L}}$ were obtained by running Algorithm 1 for 3×10^5 serial iterations ($|\hat{\mathcal{S}}| = 1$), where $\mathcal{E}$ contains the observed pixels and $\mathcal{U}$ and $\mathcal{L}$ contains all other pixels. We have used $X^{\mathcal{L}} = \mathbf{0}$ and $X^{\mathcal{U}} = \mathbf{1}$. The result again suggest that adding simple and obvious constrains leads to better low rank reconstruction and helps to keep reconstructed elements of matrix in expected bounds.

4.3. The Run-Time

Finally, in order to illustrate the run-time and efficiency of parallelization of Algorithm 1, Figure 2 (right) presents the evolution of RMSE over time on the well-known $480,189 \times 17,770$ matrix of rank 20. There is an almost linear speed-up visible from 1 to 4 cores and marginally worse speed-up between 4 and 8 cores. Considering that most other algorithms proposed in the literature cannot cope with instances of this size, we cannot compare the performance directly to SVT [6], SVP [18], SoftImpute [30], LMaFit [13], ADMiRA, [25], JS [17], OR1MP [54], EOR1MP [54], and similar. We can, however, compare the run-time on the 512×512 instances, detailed in Table 1.

5. Conclusions

We have studied the matrix completion problem under interval uncertainty and an efficient algorithm, which converges to stationary points of the NP-Hard, non-convex optimization problem, without ever trying to approximate the spectrum of the matrix. In our computational experiments, we have shown that even the seemingly most trivial inequality constraints are useful in a number of applications. This opens numerous avenues for further research:

- Forecasting with Side Information: A related application comes from the forecasting of seasonal data, e.g. sales. Let us assume that in process $\{X_t\}$, one knows $k + 1 = \tau$ such that $F_X(x_{t_1+\tau}, \ldots, x_{t_k+\tau}) = F_X(x_{t_1}, \ldots, x_{t_k})$ for the cumulative distribution function $F_X(x_{t_1+\tau}, \ldots, x_{t_k+\tau})$ of the joint distribution of $\{X_t\}$ at times $t_1+\tau, \ldots, t_k+\tau$. One can then formulate the forecasting into the future as a matrix completion problem, where there the historical datum at time t is at row $\lfloor t/\tau \rfloor$, column $t \mod k$ specified by an equality or a pair of inequalities, and where inequalities represent side information. For example in sales forecasts, one often has bookings for many months in advance and knows that the sales for the respective months will not be less than the bookings taken. On the other hand, there clearly are [36] instances, where this approach may fail.

- Non-negative matrix factorization: The coordinate descent algorithm for the problem (4) is easy to extend, e.g., toward non-negative factorization. It is sufficient to modify lines 7 and 13 in Algorithm 1 as follows: $L_{i,\hat{r}} = \max\{0, L_{i,\hat{r}} + \delta_{i,\hat{r}}\}$, $R_{\hat{r},j} = \max\{0, R_{\hat{r},j} + \delta_{\hat{r},j}\}$. One could consider extensions beyond box constraints on the individual elements as well.

- Auto-tuning μ: If we have some *a priori* bound on the largest eigenvalue of the matrix to reconstruct, let us denote it ζ, then we can modify lines 7 and 13 in Algorithm 1 as follows $L_{i,\hat{r}} = \max\{\min\{\zeta, L_{i,\hat{r}} + \delta_{i,\hat{r}}\}, -\zeta\}$, $R_{\hat{r},j} = \max\{\min\{0, R_{\hat{r},j} + \delta_{\hat{r},j}\}, -\zeta\}$.

- Additional analyses: A variety of conditions, e.g., [20], are known under which one can reconstruct the optimum to a non-convex problem using a convex relaxation. In some cases [6, 18, 55, 22, 23, 19, 48], these can be used to analyse algorithms for matrix-completion. Perhaps, one could develop similar analyses for matrix completion under interval uncertainty as well?

We would be delighted to share our code with other researchers interested in these and related problems. Currently, the code is available from `http://optml.github.io/ac-dc/`. Should it become unavailable, for any reason, we encourage researchers to contact us.

Acknowledgements. The authors are grateful for the numerous suggestions of the anonymous reviewers as well as the editor that have helped them to improve both the presentation and contents of the paper. In addition, the first author acknowledges funding from the European Union Horizon 2020 Programme (Horizon2020/2014-2020), under grant agreement number 688380. The second author would like to acknowledge support from the EPSRC Grant EP/K02325X/1, *Accelerated Coordinate Descent Methods for Big Data Optimization.*

References

[1] C. M. Alaíz, F. Dinuzzo, S. Sra, Correlation matrix nearness and completion under observation uncertainty, IMA Journal of Numerical Analysis 35 (1) (2013) 325–340.

[2] R. Bell, Y. Koren, Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights, in: IEEE 7th International Conference on Data Mining, IEEE, Piscataway, NJ, USA, 43–52, 2007.

[3] J.-F. Cai, E. J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion, SIAM Journal on Optimization 20 (4) (2010) 1956–1982.

[4] E. J. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis?, Journal of the ACM 58 (3) (2011) 11.

[5] E. J. Candès, Y. Plan, Matrix completion with noise, Proceedings of the IEEE 98 (6) (2010) 925–936.

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

[7] Y. Chen, Y. Chi, Spectral Compressed Sensing via Structured Matrix Completion, in: Proceedings of the 30th International Conference on Machine Learning (ICML 13), ACM, New York, NY, USA, 414–422, 2013.

[8] Y. Chen, H. Xu, C. Caramanis, S. Sanghavi, Robust Matrix Completion with Corrupted Columns, in: Proceedings of the 28th International Conference on Machine Learning (ICML 11), ACM, New York, NY, USA, 873–880, 2011.

[9] A. L. Chistov, D. Grigoriev, Complexity of Quantifier Elimination in the Theory of Algebraically Closed Fields, in: Proceedings of the Mathematical Foundations of Computer Science 1984, Springer-Verlag, London, UK, 17–31, 1984.

[10] M. Fazel, Matrix Rank Minimization with Applications, Ph.D. thesis, Stanford University, 2002.

[11] V. Gabrel, C. Murat, A. Thiele, Recent advances in robust optimization: An overview, European Journal of Operational Research 235 (3) (2014) 471–483.

[12] R. Gemulla, E. Nijkamp, P. J. Haas, Y. Sismanis, Large-scale matrix factorization with distributed stochastic gradient descent, in: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, New York, NY, USA, 69–77, 2011.

[13] D. Goldfarb, S. Ma, Z. Wen, Solving low-rank matrix completion problems efficiently, in: 47th Annual Allerton Conference on Communication, Control, and Computing, IEEE, Piscataway, NJ, USA, 1013–1020, 2009.

[14] J. Haldar, D. Hernando, Rank-Constrained Solutions to Linear Matrix Equations Using PowerFactorization, IEEE Signal Processing Letters 16 (7) (2009) 584–587.

[15] N. Halko, P.-G. Martinsson, J. A. Tropp, Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions, SIAM Review 53 (2) (2011) 217–288.

[16] N. J. Harvey, D. R. Karger, S. Yekhanin, The complexity of matrix completion, in: Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1103–1111, 2006.

[17] M. Jaggi, M. Sulovský, A simple algorithm for nuclear norm regularized problems, in: Proceedings of the 27th International Conference on Machine Learning (ICML 10), ACM, New York, NY, USA, 471–478, 2010.

[18] P. Jain, R. Meka, I. S. Dhillon, Guaranteed rank minimization via singular value projection, in: Advances in Neural Information Processing Systems 23 (NIPS 2010), Curran Associates, Inc., Red Hook, NY, USA, 937–945, 2010.

[19] P. Jain, P. Netrapalli, S. Sanghavi, Low-rank Matrix Completion Using Alternating Minimization, in: Proceedings of the forty-fifth annual ACM symposium on Theory of computing, New York, NY, USA, 665–674, 2013.

[20] V. Jeyakumar, S. Srisatkunarajah, Geometric conditions for Kuhn–Tucker sufficiency of global optimality in mathematical programming, European Journal of Operational Research 194 (2) (2009) 363–367.

[21] R. Kannan, M. Ishteva, H. Park, Bounded Matrix Low Rank Approximation, in: 2012 IEEE 12th International Conference on Data Mining, IEEE, Piscataway, NJ, USA, 319–328, 2012.

[22] R. H. Keshavan, Efficient Algorithms for Collaborative Filtering, Ph.D. thesis, Stanford University, 2012.

[23] R. H. Keshavan, A. Montanari, S. Oh, Matrix completion from a few entries, IEEE Transactions on Information Theory 56 (6) (2010) 2980–2998.

[24] Y. Koren, R. Bell, C. Volinsky, Matrix Factorization Techniques for Recommender Systems, Computer Journal 42 (8) (2009) 30–37.

[25] K. Lee, Y. Bresler, ADMiRA: Atomic Decomposition for Minimum Rank Approximation, IEEE Transactions on Information Theory 56 (9) (2010) 4402–4416.

[26] B. Li, S. Tata, Y. Sismanis, Sparkler: Supporting large-scale matrix factorization, in: Proceedings of the 16th International Conference on Extending Database Technology, ACM, New York, NY, USA, 625–636, 2013.

[27] G. Li, A. K. C. Ma, T. K. Pong, Robust least square semidefinite programming with applications, Computational Optimization and Applications 58 (2) (2014) 347–379.

[28] F. Makari, C. Teflioudi, R. Gemulla, P. Haas, Y. Sismanis, Shared-memory and Shared-nothing Stochastic Gradient Descent Algorithms for Matrix Completion, Knowledge and Information Systems 42 (3) (2015) 493–523.

[29] J. Mareček, P. Richtárik, M. Takáč, Distributed Block Coordinate Descent for Minimizing Partially Separable Functions, in: Numerical Analysis and Optimization, Springer International Publishing, Cham, Switzerland, 261–288, Springer Proceedings in Math. and Statistics, Vol. 134, 2015.

[30] R. Mazumder, T. Hastie, R. Tibshirani, Spectral regularization algorithms for learning large incomplete matrices, Journal of Machine Learning Research 11 (Aug) (2010) 2287–2322.

[31] B. Mehta, T. Hofmann, W. Nejdl, Robust Collaborative Filtering, in: Proceedings of the 2007 ACM Conference on Recommender Systems, ACM, New York, NY, USA, 49–56, 2007.

[32] A. Mnih, R. Salakhutdinov, Probabilistic matrix factorization, in: Advances in Neural Information Processing Systems 20, Curran Associates, Inc., Red Hook, NY, USA, 1257–1264, 2007.

[33] B. K. Natarajan, Sparse approximate solutions to linear systems, SIAM Journal on Computing 24 (2) (1995) 227–234.

[34] Y. Nesterov, Efficiency of coordinate descent methods on huge-scale optimization problems, SIAM Journal on Optimization 22 (2) (2012) 341–362.

[35] S. Olafsson, X. Li, S. Wu, Operations research and data mining, European Journal of Operational Research 187 (3) (2008) 1429–1448.

[36] F. Petropoulos, S. Makridakis, V. Assimakopoulos, K. Nikolopoulos, 'Horses for Courses' in demand forecasting, European Journal of Operational Research 237 (1) (2014) 152–163.

[37] B. Recht, C. Ré, S. J. Wright, F. Niu, Hogwild: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent, in: Advances in Neural Information Processing Systems 24 (NIPS 2011), Curran Associates, Inc., Red Hook, NY, USA, 693–701, 2011.

[38] J. D. M. Rennie, N. Srebro, Fast Maximum Margin Matrix Factorization for Collaborative Prediction, in: Proceedings of the 31st International Conference on Machine Learning (ICML 14), ACM, New York, NY, USA, 713–719, 2005.

[39] P. Richtárik, M. Takáč, Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function, Mathematical Programming 144 (2) (2014) 1–38.

[40] P. Richtárik, M. Takáč, Distributed coordinate descent method for learning with big data, Journal of Machine Learning Research (2016) to appear.

[41] P. Richtárik, M. Takáč, On optimal probabilities in stochastic coordinate descent methods, Optimization Letters (2016) to appear.

[42] P. Richtárik, M. Takáč, Parallel coordinate descent methods for big data optimization, Mathematical Programming 156 (1) (2016) 433–484.

[43] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Application of dimensionality reduction in recommender system-a case study, in: Proceedings of Web Mining for E-Commerce – Challenges and Opportunities, August 20, 2000, Boston, MA, Held in conjunction with the ACM-SIGKDD Conference on Knowledge Discovery in Databases, ACM, New York, NY, USA, without pagination, 2000.

[44] S. Shalev-Shwartz, A. Gonen, O. Shamir, Large-Scale Convex Minimization with a Low-Rank Constraint, in: Proceedings of the 28th International Conference on Machine Learning (ICML 11), ACM, New York, NY, USA, 329–336, 2011.

[45] A. L. Soyster, Convex Programming with Set-Inclusive Constraints and Applications to Inexact Linear Programming, Operations Research 21 (5) (1973) 1154–1157.

[46] N. Srebro, Learning with Matrix Factorizations, Ph.D. thesis, MIT, 2004.

[47] N. Srebro, J. Rennie, T. S. Jaakkola, Maximum-margin matrix factorization, in: Advances in Neural Information Processing Systems 17 (NIPS 2004), Curran Associates, Inc., Red Hook, NY, USA, 1329–1336, 2004.

[48] R. Sun, Z. Q. Luo, Guaranteed Matrix Completion via Nonconvex Factorization, in: IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS), IEEE, Piscataway, NJ, USA, 270–289, 2015.

[49] J. Tanner, K. Wei, Normalized Iterative Hard Thresholding for Matrix Completion, SIAM Journal on Scientific Computing 35 (5).

[50] R. Tappenden, P. Richtárik, B. Büke, Separable approximations and decomposition methods for the augmented Lagrangian, Optimization Methods and Software 30 (3) (2015) 463–668.

[51] R. Tappenden, M. Takáč, P. Richtárik, On the complexity of parallel coordinate descent, arXiv preprint arXiv:1503.03033 .

[52] C. Teflioudi, F. Makari, R. Gemulla, Distributed Matrix Completion, in: 2012 IEEE 12th International Conference on Data Mining, IEEE, Piscataway, NJ, USA, 655–664, 2012.

[53] R. Tomioka, T. Suzuki, M. Sugiyama, H. Kashima, A fast augmented lagrangian algorithm for learning low-rank matrices, in: Proceedings of the 27th Annual International Conference on Machine Learning (ICML 2010), ACM, New York, NY, USA, 1087–1094, 2010.

[54] Z. Wang, M.-J. Lai, Z. Lu, W. Fan, H. Davulcu, J. Ye, Rank-One Matrix Pursuit for Matrix Completion, in: Proceedings of the 31st International Conference on Machine Learning (ICML 14), ACM, New York, NY, USA, 91–99, 2014.

[55] J. Wright, A. Ganesh, S. Rao, Y. Peng, Y. Ma, Robust Principal Component Analysis: Exact Recovery of Corrupted Low-Rank Matrices via Convex Optimization, in: Advances in Neural Information Processing Systems 22 (NIPS 2009), Curran Associates, Inc., Red Hook, NY, USA, 2080–2088, 2009.

[56] J. Ye, Generalized low rank approximations of matrices, Machine Learning Journal 61 (1–3) (2005) 167–191.

Appendix A. Proof of Theorem 1

Proof. From (9) and (12) we see that for all k we have

$$f(L^{(k)}, R^{(k)}) \overset{(9)}{\geq} f(L^{(k+1)}, R^{(k)}) \overset{(12)}{\geq} f(L^{(k+1)}, R^{(k+1)}) \geq 0,$$

where the last inequality follows from the fact that all parts of f defined in (5) are non-negative.

Monotonicity (13) together with the fact that $\mu > 0$ imply that the level set

$$\Omega_0 := \{(L, R) \ : \ f(L, R) \leq f(L^{(0)}, R^{(0)})\}$$

is bounded. Now, for all $i \in \{1, 2, \ldots, m\}$, $v \in \{1, 2, \ldots, r\}$ and any iteration counter k we have

$$\mu \overset{(6)}{\leq} W_{iv}^{(k)} \overset{(6)}{\leq} \mu + \|R^{(k)}\|_F^2 \overset{(5)}{\leq} \mu + \frac{2}{\mu} f(L^{(k)}, R^{(k)}) \leq \mu + \frac{2}{\mu} f(L^{(0)}, R^{(0)}). \quad \text{(A.1)}$$

In the second inequality we have used Assumption 1, and in the last inequality we have used monotonicity. The same lower and upper bounds can be established for $V_{vj}^{(k)}$.

We shall now establish that $\liminf \|\nabla_L f(L^{(k)}, R^{(k)})\|_F^2 = 0$ with probability 1 (the claim $\liminf \|\nabla_R f(L^{(k)}, R^{(k)})\|_F^2 = 0$ can be proved in an analogous way). Since

$$\|\nabla_L f(L^{(k)}, R^{(k)})\|_F^2 = \sum_{i=1}^{m} \sum_{v=1}^{r} \langle \nabla_L f(L^{(k)}, R^{(k)}), E_{iv} \rangle^2,$$

it is enough to show that for $\Delta_{iv}^{(k)} := \langle \nabla_L f(L^{(k)}, R^{(k)}), E_{iv} \rangle$ we have $\liminf (\Delta_{iv}^{(k)})^2 = 0$ with probability 1 for all $i \in \{1, 2, \ldots, m\}$ and $v \in \{1, 2, \ldots, r\}$. Fix any i and v. Since $\hat{\mathcal{S}}_{\text{row}}$ is proper, and since $\hat{r}$ is chosen uniformly at random in each iteration, there is an infinite sequence of iterations, indexed by K_{iv}, in which the pair (i, v) is sampled.

In view of (9) and (A.1), for all $k \in K_{iv}$ we have

$$f(L^{(k+1)}, R^{(k+1)}) \leq f(L^{(k+1)}, R^{(k)}) \leq f(L^{(k)}, R^{(k)}) - \frac{(\Delta_{iv}^{(k)})^2}{C},$$

where $C = 2(\mu + \frac{2}{\mu} f(L^{(0)}, R^{(0)}))$. Since $f(L, U)$ is nonnegative, it must be the case that $\sum_{k \in K_{iv}} (\Delta_{iv}^{(k)})^2$ is finite. This means that, with probability 1, $\lim_{k \to \infty} \inf (\Delta_{iv}^{(k)})^2 = 0$, as desired.

□