Simple Atom Selection Strategy for Greedy Matrix Completion

Zebang Shen^1, Hui Qian^{1*}, Tengfei Zhou^1, Song Wang^2
^{1}College of Computer Science and Technology, Zhejiang University, China
^{2}University of South Carolina, U.S.A.
{shenzebang,qianhui,zhoutengfei}@zju.edu.cn, songwang@cse.sc.edu

Abstract

In this paper we focus on the greedy matrix completion problem. A simple atom selection strategy is proposed to find the optimal atom in each iteration by alternating minimization. Based on this per-iteration strategy, we devise a greedy algorithm and establish an upper bound of the approximating error. To evaluate different weight refinement methods, several variants are designed. We prove that our algorithm and three of its variants have the property of linear convergence. Experiments of Recommendation and Image Recovery are conducted to make empirical evaluation with promising results. The proposed algorithm takes only 700 seconds to process Yahoo Music dataset in PC, and achieves a root mean square error 24.5 on the test set.

1 Introduction

Low rank matrix completion is among the most basic problems in machine learning and data analysis. It plays a key role in solving many important problems, such as collaborative filtering [Rennie and Srebro, 2005; Koren *et al.*, 2009; Rendle *et al.*, 2009], dimensionality reduction [Weinberger and Saul, 2006; So and Ye, 2005], clustering [Eriksson *et al.*, 2011; Yi *et al.*, 2012], and multi-class learning [Argyriou *et al.*, 2008; Obozinski *et al.*, 2010; Xu *et al.*, 2013].

Matrix completion can be formulated as seeking the matrix with lowest rank that fits the observed data. However, directly solving such problem is NP-hard and of little practical use [Chistov and Grigor'ev, 1984] which leads to many approximation strategies. One principled approach is to adopt nuclear norm as surrogate for the rank [Cai *et al.*, 2010; Jain *et al.*, 2010; Lin *et al.*, 2010; Mazumder *et al.*, 2010; Toh and Yun, 2010]. Non-convex surrogates, usually complex pseudo norms, have also been brought forth to gain better accuracy or nearly unbiased estimation [Liu *et al.*, 2013]. Although recovery guarantees in these contexts are established [Candès and Recht, 2009; Candès and Tao, 2010; Keshavan *et al.*, 2010], the demand of expensive truncated Singular Value Decomposition (SVD) prevents the applications of nuclear norm based methods to large real-world problems.

Recently, remarkable progress has been made for greedy matrix completion techniques. The idea behind them is to represent matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ as a sparse code over a dictionary of infinite unit rank-one matrices which are referred to as *atoms* [Lee and Bresler, 2010]. Such representation makes low rank matrix completion a natural extension of greedy selection for optimization with sparsity constraint [Mallat and Zhang, 1993; Pati *et al.*, 1993; Tropp, 2004; Shalev-Shwartz *et al.*, 2010; Zhang, 2011] to the matrix case.

Typically, greedy matrix completion algorithms, like GECO [Shalev-Shwartz *et al.*, 2011], R1MP and ER1MP [Wang *et al.*, 2014], BOOST [Zhang *et al.*, 2012], and JS [Jaggi *et al.*, 2010], proceed in two core steps in each iteration. The first step selects a locally optimal atom. The second step refines the weights of all atoms chosen up to this iteration. Since atom selection and weight refinement can be much cheaper than truncated SVD, such a two-step scheme brings us better scalability than nuclear norm based methods.

In the previous works, the second step, i.e., the weight refinement step was the research focus and almost all the existing greedy matrix algorithms differ mainly in their refinement steps[Wang *et al.*, 2014]. For the first step, i.e., the atom selection step, only one strategy, called T1SVD in our paper, was used in current greedy matrix completion literatures, to our best knowledge. The main reason is that the T1SVD strategy corresponds to a Top-1 SVD problem which is numerically easy to solve and has plenty of efficient algorithms.

In this paper, we further explore the atom selection problem and present a simple strategy, called *Optimal Atom (OA)*, to select the best atom. Our research is partially inspired by the work of optimization problem with sparsity constraint [Shalev-Shwartz *et al.*, 2010; Liu *et al.*, 2014]. We directly solve a coordinate optimization problem to find the optimal atom in each step, rather than deal with the first order approximation to the optimal atom choice as T1SVD does. In this line, our *Optimal Atom based Matrix Completion* algorithm (OAMC) adopts an alternating method instead of the common Top-1 SVD solver to conduct the matrix completion. We show that OA is a better strategy to construct the greedy matrix completion algorithm than T1SVD. The major contributions are summarized as follows:

*Corresponding author

- We propose a simple atom selection strategy, called OA, which finds the optimal atom in each iteration by alternating minimization with computational complexity comparable to common Top-1 SVD solvers.
- Under suitable assumptions, we construct an upper bound of the approximating error for OA strategy, which is independent of the largest singular value of underlying residual matrix. Such result is important since it provides a tighter training error bound for OAMC than for the T1SVD baselines.
- A greedy algorithm, called OAMC, is devised to solve the matrix completion problem. Several variants are designed according to different weight refinement methods. We prove that OAMC and three of its variants have the property of linear convergence.

In the experiments, we evaluate the performance of the proposed OAMC algorithm by applying it to the tasks of Recommendation and Image Recovery. We perform matrix completion on three largest publicly available recommendation datasets: MovieLens 10M, NetFlix, and Yahoo Music. In all these experiments, OAMC and its variants significantly outperform their T1SVD-based competitors in terms of both speed and accuracy. Some of these variants are *5 times* faster than existing methods when using efficient random initialization for the alternating minimization scheme. Besides, we are able to process Yahoo Music dataset in about *700* seconds in PC workstation and achieve a root mean square error *24.5* on the test set.

2 Preliminaries

For a matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$, let $\Omega \subset \{1, \cdots, m\} \times \{1, \cdots, n\}$ denote the indices of observed entries. In this paper, we always assume $m \leq n$. We consider the following low rank matrix completion problem:

$$\min_{\substack{\mathbf{X}\in\mathbb{R}^{m\times n}:\\ \operatorname{rank}(\mathbf{X})\leq K}} \mathcal{L}(\mathbf{X}) \triangleq \frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{M}-\mathbf{X})\|_{\mathbf{F}}^2, \tag{1}$$

where $K \ll \min(m, n)$ is a constant, $\|\cdot\|_{\mathbf{F}}$ is the Frobenius norm, and the operator $\mathcal{P}_\Omega : \mathbb{R}^{m\times n} \mapsto \mathbb{R}^{m\times n}$ is defined as follows:

$$[\mathcal{P}_\Omega(\mathbf{A})]_{i,j} = \begin{cases} \mathbf{A}_{i,j} & \text{if } (i,j) \in \Omega, \\ 0 & \text{otherwise.} \end{cases}$$

Usually, we call the Ω as the support of $\mathcal{P}_\Omega(\mathbf{A})$. In practice, problem (1) is relaxed by replacing the $\operatorname{rank}(\mathbf{X})$ with a surrogate function to fit more effective algorithms.

Actually, we can also depict problem (1) in an infinite vector form [1]. Consider a dictionary like $\mathcal{D} = \mathcal{U} \times \mathcal{V}$, in which $\mathcal{U} = \{\mathbf{u} \in \mathbb{R}^m : \|\mathbf{u}\| = 1\}$, $\mathcal{V} = \{\mathbf{v} \in \mathbb{R}^n : \|\mathbf{v}\| = 1\}$, where we let $\|\cdot\|$ be the 2-norm of vector. A pair $(\mathbf{u}, \mathbf{v}) \in \mathcal{D}$ denotes an atom in this dictionary. Over $\mathcal{D}$, a vector $\lambda \in \mathbb{R}^{|\mathcal{D}|}$ can be used to represent an arbitrary $m \times n$ matrix. That is, given vector $\lambda \in \mathbb{R}^{|\mathcal{D}|}$, we have a correspondent $m\times n$ matrix

$$\mathbf{X}(\lambda) = \sum_{(\mathbf{u},\mathbf{v})\in\mathcal{D}} \lambda_{(\mathbf{u},\mathbf{v})}\mathbf{u}\mathbf{v}^\top,$$

where $\mathbf{X} : \mathbb{R}^{|\mathcal{D}|} \mapsto \mathbb{R}^{m\times n}$ is a linear map, and $\lambda_{(\mathbf{u},\mathbf{v})} \in \mathbb{R}$ denotes the value of λ in the coordinate indexed by the pair $(\mathbf{u}, \mathbf{v})$. From the SVD theorem, if $\operatorname{rank}(\mathbf{X}(\lambda)) \leq r$, then there must be a λ satisfying $\|\lambda\|_0 \leq r$. We also define a standard basis vector $\mathbf{e}^{(\mathbf{u},\mathbf{v})} \in \mathbb{R}^{|\mathcal{D}|}$ over this dictionary as

$$\mathbf{e}^{(\mathbf{u},\mathbf{v})}_{(\mathbf{p},\mathbf{q})} = \begin{cases} 1 & \text{if } \mathbf{p} = \mathbf{u} \text{ and } \mathbf{q} = \mathbf{v}, \\ 0 & \text{otherwise,} \end{cases}$$

where $\mathbf{p} \in \mathcal{U}$, and $\mathbf{q} \in \mathcal{V}$. The difference between $\mathbf{e}^{(\mathbf{u},\mathbf{v})}$ and standard basis vector $\mathbf{e}^i$ in Euclidean space is that $\mathbf{e}^{(\mathbf{u},\mathbf{v})}$ is indexed by pair $(\mathbf{u}, \mathbf{v})$ instead of the number i.

Thus, greedy algorithm for (1) can be developed by resorting to the following equivalent problem:

$$\min_{\substack{\lambda\in\mathbb{R}^{|\mathcal{D}|}:\\ \|\lambda\|_0\leq K}} \mathcal{Q}(\lambda) \tag{2}$$

where $\mathcal{Q}(\lambda) \triangleq \mathcal{L}(\mathbf{X}(\lambda)) = \frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{M} - \mathbf{X}(\lambda))\|_{\mathbf{F}}^2$. For convenience, we also define residual function $\mathbf{R} : \mathbb{R}^{|\mathcal{D}|} \mapsto \mathbb{R}^{m\times n}$ as $\mathbf{R}(\lambda) = \mathcal{P}_\Omega(\mathbf{M} - \mathbf{X}(\lambda))$. We call $\mathbf{M} - \mathbf{X}(\lambda)$ the *underlying matrix* of $\mathbf{R}(\lambda)$. To solve problem (2), the current state-of-the-art greedy methods choose to find the atom using maximum gradient in each iteration. That is, the selected atom comes from following formulation

$$(\hat{\mathbf{u}}, \hat{\mathbf{v}}) = \operatorname*{argmax}_{(\mathbf{u},\mathbf{v})} \left|\frac{\partial \mathcal{Q}(\lambda)}{\partial \lambda_{(\mathbf{u},\mathbf{v})}}\right|. \tag{3}$$

We call this atom selection method *maximum gradient* strategy or *T1SVD* since it is based on solving Top-1 SVD problem.

A few more notations will be useful in our narration. Given a rank r matrix $\mathbf{A}$ with SVD $\sum_{i=1}^r \sigma_i \mathbf{u}_i {\mathbf{v}_i}^\top, \sigma_1 \geq \cdots \geq \sigma_r$. We define $\mathcal{S}_1(\mathbf{A}) = \sigma_1\mathbf{u}_1\mathbf{v}_1^\top$, $\mathcal{S}_2(\mathbf{A}) = \sum_{i=2}^r \sigma_i\mathbf{u}_i\mathbf{v}_i^\top$, and $\sigma_1(\mathbf{A}) = \sigma_1$(the maximum σ_i, for $i = 1, \cdots, r$). We use $\mathbf{A}_i$ to represent the ith column of $\mathbf{A}$ and $\mathbf{A}_{i,:}$ to represent the ith row. $\mathcal{O}$ denotes the big-O notation in mathematics. $\langle\cdot,\cdot\rangle$ represents the inner product of two matrices, and ∇ denotes the Del operator of a function.

3 Methodology

We start from $\lambda^{(0)} = \mathbf{0}$. In the k-th iteration , suppose $\lambda = \lambda^{(k)}$. The locally optimal $\mathbf{u}^*$, $\mathbf{v}^*$, and α^* need to be estimated to approximate the residual $\mathbf{R}(\lambda)$. First of all, for a fixed pair $(\mathbf{u}, \mathbf{v})$, we should find an α that minimizes $\mathcal{Q}(\lambda + \alpha\mathbf{e}^{(\mathbf{u},\mathbf{v})})$ which results in the optimization problem $\min_\alpha \mathcal{Q}(\lambda + \alpha\mathbf{e}^{(\mathbf{u},\mathbf{v})})$. Second, we expect that $(\mathbf{u}, \mathbf{v})$ can make the maximum progress after an increment $\alpha\mathbf{e}^{(\mathbf{u},\mathbf{v})}$ is added into λ. Thus, by combining these two goals, we have the following optimization problem for each iteration.

$$\begin{aligned}(\mathbf{u}^*, \mathbf{v}^*) &= \operatorname*{argmax}_{(\mathbf{u},\mathbf{v})} \left\{\mathcal{Q}(\lambda) - \min_\alpha \mathcal{Q}(\lambda + \alpha\mathbf{e}^{(\mathbf{u},\mathbf{v})})\right\} \\ &= \operatorname*{argmin}_{(\mathbf{u},\mathbf{v})} \min_\alpha \mathcal{Q}(\lambda + \alpha\mathbf{e}^{(\mathbf{u},\mathbf{v})}). \end{aligned} \tag{4}$$

[1] We follow the notation of [Shalev-Shwartz *et al.*, 2011], in which an assumption of finite representation for real number is used to simplify the presentation.

Algorithm 1 OAMC

```
Input: $\Omega, \mathcal{P}_\Omega(\mathbf{M}), K$
Output: $\mathbf{X}^{(K)}$
Initialize: Set $\mathbf{X}^{(0)} = 0$ and $\mathbf{R}^{(0)} = \mathcal{P}_\Omega(\mathbf{M})$
for $k = 1, 2, \ldots, K$ do
    $\bar{\mathbf{u}} :=$ INIT()
    repeat
        $\bar{\mathbf{v}} := \mathrm{argmin}_{\mathbf{v}} \|\mathcal{P}_\Omega(\mathbf{R}^{(k-1)} - \bar{\mathbf{u}}\mathbf{v}^\top)\|_\mathbf{F}$
        $\bar{\mathbf{u}} := \mathrm{argmin}_{\mathbf{u}} \|\mathcal{P}_\Omega(\mathbf{R}^{(k-1)} - \mathbf{u}\bar{\mathbf{v}}^\top)\|_\mathbf{F}$
    until converge
    $\mathbf{X}^{(k)} := \mathbf{X}^{(k)} + \bar{\mathbf{u}}\bar{\mathbf{v}}^\top$
    $\mathbf{R}^{(k)} := \mathcal{P}_\Omega(\mathbf{M} - \mathbf{X}^{(k)})$
end for
```

3.1 Optimal Atom

Basically, (4) indicates that we should find an optimal coordinate $\mathbf{e}^{(\mathbf{u}^*,\mathbf{v}^*)}$ and a proper step α^*. From the definition of $\mathcal{Q}(\lambda)$, we have the following objective

$$\begin{aligned}(\mathbf{u}^*, \mathbf{v}^*) &= \operatorname*{argmin}_{(\mathbf{u},\mathbf{v})} \min_\alpha \frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{M} - \mathbf{X}(\lambda + \alpha\mathbf{e}^{(\mathbf{u},\mathbf{v})}))\|_\mathbf{F}^2 \\ &= \operatorname*{argmin}_{(\mathbf{u},\mathbf{v})} \min_\alpha \frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{R}(\lambda) - \alpha\mathbf{u}\mathbf{v}^\top)\|_\mathbf{F}^2. \qquad (5)\end{aligned}$$

Problem (5) has been widely thought to be complicated to solve because the objective is jointly non-convex over $\mathbf{u}$ and $\mathbf{v}$. However, it's easy to know that $\alpha\mathbf{u}\mathbf{v}^\top$ is a rank 1 matrix. To such a rank invariant situation, if we fixed $\mathbf{u}$, the original problem becomes convex, specifically a typical least square problem. The same can be found when $\mathbf{v}$ is fixed. Thus, we can alternately fix one and optimize over the other until convergence. This alternating optimization scheme is a special case of the Alternating Least Square Method [Koren *et al.*, 2009; Jain *et al.*, 2013; Gunasekar *et al.*, 2013].

Practically, proper initialization of $\mathbf{u}$ is crucial for convergence for an alternating optimization procedure. In our method, random and prior-knowledge based initialization methods, are tested (see [Gunasekar *et al.*, 2013] for details), and encapsulated in a macro INIT. In addition, α is redundant. Real implementation can use only two variables: $\bar{\mathbf{u}} \in \mathbb{R}^m$ and $\bar{\mathbf{v}} \in \mathbb{R}^n$ since it can be easily derived that $\mathbf{u} = \bar{\mathbf{u}}/\|\bar{\mathbf{u}}\|, \mathbf{v} = \bar{\mathbf{v}}/\|\bar{\mathbf{v}}\|$ and $\alpha = \|\bar{\mathbf{u}}\|\|\bar{\mathbf{v}}\|$. Furthermore, we use $m \times n$ matrix $\mathbf{X}$ and $\mathbf{R}$ to represent the function $\mathbf{X}(\lambda)$ and $\mathbf{R}(\lambda)$ since λ with infinite dimension is simply used in derivation and analysis. We summarize our pseudo code in Algorithm 1.

Note that, the **first** important difference between OA and T1SVD is that the selected atom in the latter is not derived from (4). Actually T1SVD simplifies (4) by replacing $\mathcal{Q}(\lambda + \alpha\mathbf{e}^{(\mathbf{u},\mathbf{v})})$ with

$$\mathcal{Q}(\lambda) + \langle\nabla\mathcal{Q}(\lambda), \alpha\mathbf{e}^{(\mathbf{u},\mathbf{v})}\rangle, \qquad (6)$$

which is a first-order approximation of $\mathcal{Q}(\lambda + \alpha\mathbf{e}^{(\mathbf{u},\mathbf{v})})$. Since the first item of (6) is irrelevant to $(\mathbf{u}, \mathbf{v})$, by restricting α to be finite, we have

$$(\hat{\mathbf{u}}, \hat{\mathbf{v}}) = \operatorname*{argmin}_{(\mathbf{u},\mathbf{v})} \min_\alpha \langle\nabla\mathcal{Q}(\lambda), \alpha\mathbf{e}^{(\mathbf{u},\mathbf{v})}\rangle$$

instead of problem (4). Apparently, α will always take the reverse sign of $\langle\nabla\mathcal{Q}(\lambda), \mathbf{e}^{(\hat{\mathbf{u}},\hat{\mathbf{v}})}\rangle$, thus we have

$$\begin{aligned}(\hat{\mathbf{u}}, \hat{\mathbf{v}}) &= \operatorname*{argmax}_{(\mathbf{u},\mathbf{v})} |\langle\nabla\mathcal{Q}(\lambda), \mathbf{e}^{(\mathbf{u},\mathbf{v})}\rangle| \\ &= \operatorname*{argmax}_{(\mathbf{u},\mathbf{v})} \left|\frac{\partial\mathcal{Q}(\lambda)}{\partial\lambda_{(\mathbf{u},\mathbf{v})}}\right|,\end{aligned}$$

which is the problem (3). It is reasonable to infer that the error from the linear approximation may amplify the necessity of a fully corrective procedure in current state-of-the-art methods.

The **second** difference between OA and T1SVD lies in that the optimization problem of OA involves only the matrix entries in Ω while T1SVD deals with the whole matrix with plenty of zeros indicating the missing entries. For T1SVD, the chain rule of partial derivative allows us to write (3) as

$$\begin{aligned}(\hat{\mathbf{u}}, \hat{\mathbf{v}}) &= \operatorname*{argmax}_{(\mathbf{u},\mathbf{v})} \left|\frac{\partial\mathcal{Q}(\lambda)}{\partial\mathbf{X}(\lambda)} \cdot \frac{\partial\mathbf{X}(\lambda)}{\partial\lambda_{(\mathbf{u},\mathbf{v})}}\right| \\ &= \operatorname*{argmax}_{(\mathbf{u},\mathbf{v})} \left|< \nabla\mathcal{L}(\mathbf{X}(\lambda)), \mathbf{u}\mathbf{v}^\top >\right| \\ &= \operatorname*{argmin}_{(\mathbf{u},\mathbf{v})} \min_\alpha \|\mathbf{R}(\lambda) - \alpha\mathbf{u}\mathbf{v}^\top\|_\mathbf{F}^2, \qquad (7)\end{aligned}$$

in which equation (7) comes from the fact that

$$\begin{aligned}&\min_\alpha \|\mathbf{R}(\lambda) - \alpha\mathbf{u}\mathbf{v}^\top\|_\mathbf{F}^2 \\ =&\|\mathbf{R}(\lambda)\|_\mathbf{F}^2 - \langle\mathbf{R}(\lambda), \mathbf{u}\mathbf{v}^\top\rangle^2 \\ =&\|\nabla\mathcal{L}(\mathbf{X}(\lambda))\|_\mathbf{F}^2 - \langle\nabla\mathcal{L}(\mathbf{X}(\lambda)), \mathbf{u}\mathbf{v}^\top\rangle^2. \qquad (8)\end{aligned}$$

And for equation (8), $\nabla\mathcal{L}(\mathbf{X}(\lambda)) = -\mathbf{R}(\lambda)$ is easy to verify. Comparing (7) with (5), T1SVD chooses the best atom $(\hat{\mathbf{u}}, \hat{\mathbf{v}})$ to approximate the whole $\mathbf{R}(\lambda)$, while OA goes through the same procedure only within Ω which contains the indices of observed entries.

3.2 Variants for Fully Corrective Selection

Our Algorithm (1) is totally non-corrective, that is, at each iteration, we only modify the weights of the current atom. Based on OA, variants for fully corrective selection can also be developed for better accuracy, especially when sufficient computational capacity is available.

In order to compare our OA strategy with other state-of-the-art algorithms, we design OA variants with five mainstream schemes for weight updating and encapsulate them in the form of macros to keep the algorithm succinct. The inputs of these macros are $\{\Omega, \mathcal{P}_\Omega(\mathbf{M}), \mathbf{U} \in \mathbb{R}^{m\times q}, \mathbf{V} \in \mathbb{R}^{m\times q}, \mathbf{\Phi} \in \mathbb{R}^{(q-1)\times(q-1)}\}$, where $\mathbf{U}$ and $\mathbf{V}$ have their column $\mathbf{U}_i \in \mathcal{U}$ and $\mathbf{V}_i \in \mathcal{V}$ respectively, $\mathbf{\Phi} \in \mathbb{R}^{(q-1)\times(q-1)}$ is obtained from the previous iteration and q is the iteration count. The outputs are $\{\mathbf{\Phi} \in \mathbb{R}^{q\times q}, \mathbf{U}, \mathbf{V}\}$. $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{\Phi}$ are in both input and output set. Our variants for fully corrective selection are summarized in Algorithm 2. We briefly explain the weight updating schemes as follows.

ADJ-GECO follows the method in literature [Shalev-Shwartz *et al.*, 2011] that solves the following regression problem

$$\mathbf{S}^* = \operatorname*{argmin}_{\mathbf{S}\in\mathbb{R}^{q\times q}} \mathcal{H}(\mathbf{S}) \triangleq \frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{U}\mathbf{S}\mathbf{V}^\top - \mathbf{M})\|_\mathbf{F}^2.$$

Algorithm 2 OA variants: OA-GECO,OA-R1MP, OA-ER1MP, OA-JS, and OA-BOOST

```
Input: $\Omega, \mathcal{P}_\Omega(\mathbf{M}), K$
Output: $\mathbf{U}^{(K)}\mathbf{\Phi}^{(K)}(\mathbf{V}^{(K)})^\top$
Initialize: Set $\mathbf{U}^{(0)} := \mathbf{V}^{(0)} := [\,]$,and $\mathbf{R}^{(0)} := \mathcal{P}_\Omega(\mathbf{M})$
for $k := 1, 2, \ldots, K$ do
    Step 1:
    $\bar{\mathbf{u}} :=$ INIT()
    repeat
        $\bar{\mathbf{v}} := \text{argmin}_{\mathbf{v}} \|\mathcal{P}_\Omega(\mathbf{R}^{(k-1)} - \bar{\mathbf{u}}\mathbf{v}^\top)\|_{\mathbf{F}}$
        $\bar{\mathbf{u}} := \text{argmin}_{\mathbf{u}} \|\mathcal{P}_\Omega(\mathbf{R}^{(k-1)} - \mathbf{u}\bar{\mathbf{v}}^\top)\|_{\mathbf{F}}$
    until converge
    $\bar{\mathbf{u}} := \frac{\bar{\mathbf{u}}}{\|\bar{\mathbf{u}}\|}$ and $\bar{\mathbf{v}} := \frac{\bar{\mathbf{v}}}{\|\bar{\mathbf{v}}\|}$
    Step 2:
    $\mathbf{U}^{(k)} := [\mathbf{U}^{(k-1)}, \bar{\mathbf{u}}]$ and $\mathbf{V}^{(k)} := [\mathbf{V}^{(k-1)}, \bar{\mathbf{v}}]$
    Let IN be $\{\Omega, \mathcal{P}_\Omega(\mathbf{M}), \mathbf{U}^{(k)}, \mathbf{V}^{(k)}\}$
    Let OUT be $\{\mathbf{\Phi}^{(k)}, \mathbf{U}^{(k)}, \mathbf{V}^{(k)}\}$
    Refinement (Choose one of five):

        OUT := ADJ-GECO(IN),     / * Alg: OA-GECO * /
            := ADJ-R1MP(IN),     / * Alg: OA-R1MP * /
            := ADJ-ER1MP(IN),    / * Alg: OA-ER1MP * /
            := ADJ-JS(IN),       / * Alg: OA-JS * /
         or := ADJ-BOOST(IN)     / * Alg: OA-BOOST * /

    $\mathbf{R}^{(k)} := \mathcal{P}_\Omega(\mathbf{M} - \mathbf{U}^{(k)}\mathbf{\Phi}^{(k)}(\mathbf{V}^{(k)})^\top)$
end for
```

Let $\mathbf{U_S\Phi_S V_S^\top}$ be the SVD of $\mathbf{S}^*$. We construct outputs as $\mathbf{U} := \mathbf{UU_S}$, $\mathbf{V} := \mathbf{VV_S}$ and $\mathbf{\Phi} := \mathbf{\Phi_S}$.

ADJ-R1MP is a simplification of ADJ-GECO with a constraint that $\mathbf{S}$ is diagonal [Wang *et al.*, 2014].

ADJ-ER1MP further simplifies above ADJ-R1MP by setting $\mathbf{S}_{i,i} = a_1\mathbf{\Phi}_{i,i}$ for $i \in \{1, \cdots, (q-1)\}$ and $\mathbf{S}_{q,q} = a_2$ [Wang *et al.*, 2014].

ADJ-JS preprocesses the data as in [Jaggi *et al.*, 2010] and proceeds like ADJ-ER1MP except for an additional constraint $(a_1 + a_2 = 1)$ and a_1 and a_2 are the only variables.

ADJ-BOOST is similar to ADJ-ER1MP. It adds a regularization term $\beta(a_1\sum_{i=1}^{q-1}\mathbf{\Phi}_{i,i} + a_2)$ into the objective function with constraints: $a_1 \geq 0$ and $a_2 \geq 0$ [Zhang *et al.*, 2012].

4 Analysis

In this section, we investigate how well OA approximates the residual $\mathbf{R}$. The upper bound that we obtain is independent of the largest singular value of the underlying matrix of residual. We also prove that OAMC and three of its variants converge linearly.

We start our analysis with the following definition of incoherence property of matrix.

Definition 1 (μ-incoherent). *Let* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *be a rank* k *matrix with singular value decomposition* $\mathbf{U\Phi V}^\top$. $\mathbf{A}$ *is said to be* μ-incoherent *if there exists a constant* μ *such that* $\max_{i\in\{1,\cdots,m\}} \|\mathbf{U}_{i,:}\| \leq \mu\sqrt{\frac{k}{m}}$ *and* $\max_{i\in\{1,\cdots,n\}} \|\mathbf{V}_{i,:}\| \leq \mu\sqrt{\frac{k}{n}}$.

Lemma 1. *Let* $\mathbf{M} = \sigma\mathbf{uv}^\top$ *be a* μ*-incoherent matrix with* $\mathbf{u} \in \mathcal{U}$ *and* $\mathbf{v} \in \mathcal{V}$ *and* $\mathbf{N} \in \mathbb{R}^{m\times n}$ *be a noise matrix with* $\max_{i,j}\mathbf{N}_{i,j} \leq \frac{c\sigma}{n}$, *where* c *is a constant. Suppose that the support* Ω *of* $\mathbf{R}$ *is obtained by uniformly and independently sampling from* $\{1, \cdots, m\} \times \{1, \cdots, n\}$ *with probability* p. *Let* $G = (\|\mathcal{P}_\Omega(N)\|_{\mathbf{F}}/\sqrt{p})/\sigma$. *If* $G \leq C_1 \leq \sqrt{3}/\mu$ *and*

$$p \geq \frac{C_2\mu^4 \log n \log(1/\mu G)}{\min\{\delta_2, n(\mu G)^4\}m},$$

where C_1, C_2 *and* $\delta_2 \leq (1/64)$ *are constants, then*

$$\sigma\|\mathcal{P}_\Omega(\mathbf{uv}^\top - \bar{\mathbf{u}}\bar{\mathbf{v}}^\top)\|_{\mathbf{F}} \leq C\mu\|\mathcal{P}_\Omega(\mathbf{N})\|_{\mathbf{F}} \tag{9}$$

with probability at least $1 - (1/n^3)$, *where* $(\bar{\mathbf{u}}, \bar{\mathbf{v}})$ *is the solution of step 1 of Algorithm 2 initialized as in [Gunasekar* et al., *2013] with* $\mathcal{O}(\log(1/\mu\mathcal{G}))$ *iterations, and* C *is a constant.*

Lemma 1 will be used to show that our algorithm needs approximately $pmn \geq \frac{C_2\mu^4 n \log n \log\frac{1}{\mu\mathcal{G}}}{\min\{\delta_2, n(\mu\mathcal{G})^4\}}$ samples to fulfill the completion, whose proof is placed in the long version of this paper. Note that $\mathcal{O}(n\log n)$ is the optimum sampling complexity to complete a rank-1 matrix according to the Coupon collector's problem. Thus our atom selection strategy is also optimum in terms of sampling complexity with respect to the matrix size.

Theorem 1. *Let* $\mathbf{L} \in \mathbb{R}^{m\times n}$ *be the underlying rank-r matrix of the residual* $\mathbf{R}$ *and suppose* $\max_{i,j}\mathcal{S}_2(\mathbf{L}) \leq \frac{c\sigma_1(\mathbf{L})}{n}$, *with a constant c. If the support* Ω *of* $\mathbf{R}$ *is obtained by uniformly and independently sampling from* $\{1, \cdots, m\} \times \{1, \cdots, n\}$ *with probability p defined in Lemma 1 using* μ *and G derived from* $\mathcal{S}_1(\mathbf{L})$ *and* $\mathcal{S}_2(\mathbf{L})$, *then*

$$\min_\alpha \|\mathcal{P}_\Omega(\mathbf{R} - \alpha\bar{\mathbf{u}}\bar{\mathbf{v}}^\top)\|_{\mathbf{F}} \leq (1 + C\mu)\|\mathcal{P}_\Omega(\mathcal{S}_2(\mathbf{L}))\|_{\mathbf{F}} \tag{10}$$

with probability at least $1 - \frac{1}{n^3}$, *where* $\bar{\mathbf{u}}\bar{\mathbf{v}}^\top$ *is the atom constructed by step 1 of Algorithm 2 and C is a positive constant.*

Proof. By the subadditivity of Frobenius norm, we have:

$$\begin{aligned}&\min_\alpha \|\mathcal{P}_\Omega(\mathbf{L} - \alpha\bar{\mathbf{u}}\bar{\mathbf{v}}^\top)\|_{\mathbf{F}}\\ \leq &\min_\alpha \left\{\|\mathcal{P}_\Omega(\mathbf{L} - \alpha\mathbf{u}_1\mathbf{v}_1^\top)\|_{\mathbf{F}} + \alpha\|\mathcal{P}_\Omega(\mathbf{u}_1\mathbf{v}_1^\top - \bar{\mathbf{u}}\bar{\mathbf{v}}^\top)\|_{\mathbf{F}}\right\}\end{aligned}$$

where $\mathbf{u}_1$ and $\mathbf{v}_1$ are vectors from $\mathcal{S}_1(\mathbf{L})$. Taking $\alpha = \sigma_1(\mathbf{L})$,

$$\begin{aligned}&\min_\alpha \|\mathcal{P}_\Omega(\mathbf{L} - \alpha\bar{\mathbf{u}}\bar{\mathbf{v}}^\top)\|_{\mathbf{F}}\\ \leq &\|\mathcal{P}_\Omega(\mathcal{S}_2(\mathbf{L}))\|_{\mathbf{F}} + \sigma_1\|\mathcal{P}_\Omega(\mathbf{u}_1\mathbf{v}_1^\top - \bar{\mathbf{u}}\bar{\mathbf{v}}^\top)\|_{\mathbf{F}}.\end{aligned} \tag{11}$$

We bound the second term of (11) by $C\mu\|\mathcal{P}_\Omega(\mathbf{L}_{\geq 2})\|_{\mathbf{F}}$ using Lemma 1. Since $\mathcal{P}_\Omega(\mathbf{R}) = \mathcal{P}_\Omega(\mathbf{L})$, we have the result. □

This bound shows that the result of Algorithm 2 approximates $\mathbf{R}$ with an error independent of the largest singular value of $\mathbf{L}$.

The following Lemma is used to prove the linear convergence of OAMC and three of its variants.

Lemma 2. *Let $(\bar{\mathbf{u}}, \bar{\mathbf{v}})$ be the atom selected by the step 1 of Algorithm 2 and the step 1 is initialized by any pair $(\hat{\mathbf{u}}, \hat{\mathbf{v}})$. We have*

$$\min_{\alpha} \|\mathcal{P}_\Omega(\mathbf{R} - \alpha\bar{\mathbf{u}}\bar{\mathbf{v}}^\top)\|_{\mathbf{F}} \leq \min_{\alpha} \|\mathcal{P}_\Omega(\mathbf{R} - \alpha\hat{\mathbf{u}}\hat{\mathbf{v}}^\top)\|_{\mathbf{F}}. \quad (12)$$

The result of Lemma 2 can be easily obtained since the alternating minimization does not increase the value of the objective in any iteration. Now we present a simple proof to show that OAMC and three of its variants converge linearly with proper initialization. We choose OA-R1MP as an example and use notations conforming to the description of Algorithm 2.

Theorem 2. *The residual $\mathbf{R}^{(k+1)} \in \mathbb{R}^{m\times n}$ of OA-R1MP satisfies*

$$\|\mathbf{R}^{(k+1)}\|_{\mathbf{F}} \leq \gamma^k \|\mathcal{P}_\Omega(\mathbf{M})\|_{\mathbf{F}}, \quad (13)$$

with the ApproxSV defined in [Shalev-Shwartz et al.*, 2011] as the initialization for step 1 in each iteration, where $\gamma \in [0, 1)$ is a constant.*

Proof. From the definition of ADJ-R1MP, we have

$$\|\mathbf{R}^{(k+1)}\|_{\mathbf{F}}^2 = \min_{\mathbf{\Phi}\text{ is diagonal}} \|\mathcal{P}_\Omega(\mathbf{M} - \mathbf{U}^{(k+1)}\mathbf{\Phi}\mathbf{V}^{(k+1)^\top})\|_{\mathbf{F}}^2.$$

Let $\mathbf{\Phi}'$ be in the form of $[[(\mathbf{\Phi}^{(k)})^\top, 0]^\top, [\mathbf{0}, \alpha]^\top]$. We have

$$\begin{aligned}
&\|\mathbf{R}^{(k+1)}\|_{\mathbf{F}}^2 \\
\leq &\min_{\alpha} \|\mathcal{P}_\Omega(\mathbf{M} - \mathbf{U}^{(k+1)}\mathbf{\Phi}'(\alpha)\mathbf{V}^{(k+1)^\top})\|_{\mathbf{F}}^2 \qquad (14)\\
= &\min_{\alpha} \|\mathcal{P}_\Omega(\mathbf{R}^{(k)} - \alpha\bar{\mathbf{u}}\bar{\mathbf{v}}^\top)\|_{\mathbf{F}}^2.
\end{aligned}$$

Let $\hat{\mathbf{u}}\hat{\mathbf{v}}^\top$ be the initial atom constructed by ApproxSV. By applying Lemma 2 to (14) we have

$$\|\mathbf{R}^{(k+1)}\|_{\mathbf{F}}^2 \leq \min_{\alpha} \|\mathcal{P}_\Omega(\mathbf{R}^{(k)} - \alpha\hat{\mathbf{u}}\hat{\mathbf{v}}^\top)\|_{\mathbf{F}}^2. \quad (15)$$

For (15), α has a close form solution

$$\alpha^* = \langle \mathcal{P}_\Omega(\mathbf{R}^{(k)}), \mathcal{P}_\Omega(\hat{\mathbf{u}}\hat{\mathbf{v}}^\top)\rangle / \|\mathcal{P}_\Omega(\hat{\mathbf{u}}\hat{\mathbf{v}}^\top)\|_{\mathbf{F}}^2.$$

Substituting α^* into (15), we have

$$\|\mathbf{R}^{(k+1)}\|_{\mathbf{F}}^2 \leq \|\mathbf{R}^{(k)}\|_{\mathbf{F}}^2 - \frac{\langle \mathcal{P}_\Omega(\mathbf{R}^{(k)}), \mathcal{P}_\Omega(\hat{\mathbf{u}}\hat{\mathbf{v}}^\top)\rangle^2}{\|\mathcal{P}_\Omega(\hat{\mathbf{u}}\hat{\mathbf{v}}^\top)\|_{\mathbf{F}}^2}. \quad (16)$$

We know $\|\mathcal{P}_\Omega(\hat{\mathbf{u}}\hat{\mathbf{v}}^\top)\|_{\mathbf{F}}^2 \leq 1$ and $\langle \mathcal{P}_\Omega(\mathbf{R}^{(k)}), \mathcal{P}_\Omega(\hat{\mathbf{u}}\hat{\mathbf{v}}^\top)\rangle = \langle \mathcal{P}_\Omega(\mathbf{R}^{(k)}), \hat{\mathbf{u}}\hat{\mathbf{v}}^\top\rangle = \langle \mathbf{R}^{(k)}, \hat{\mathbf{u}}\hat{\mathbf{v}}^\top\rangle = \hat{\mathbf{u}}^\top\mathbf{R}^{(k)}\hat{\mathbf{v}} \geq (1-\delta)\sigma_1(\mathbf{R}^{(k)})$, where δ is a constant smaller than 1. The last inequation comes from [Shalev-Shwartz *et al.*, 2011]. Thus,

$$\begin{aligned}
\|\mathbf{R}^{(k+1)}\|_{\mathbf{F}}^2 &\leq \|\mathbf{R}^{(k)}\|_{\mathbf{F}}^2 - (1-\delta)^2\sigma_1^2(\mathbf{R}^{(k)}) \\
&= \|\mathbf{R}^{(k)}\|_{\mathbf{F}}^2 (1 - \frac{(1-\delta)^2\sigma_1^2(\mathbf{R}^{(k)})}{\|\mathbf{R}^{(k)}\|_{\mathbf{F}}^2}).
\end{aligned} \quad (17)$$

Further, we have

$$\|\mathbf{R}^{(k+1)}\| \leq \mathcal{P}_\Omega(\mathbf{M}) \prod_{i=1}^{k} \sqrt{1 - \frac{(1-\delta)^2\sigma_1^2(\mathbf{R}^{(i)})}{\|\mathbf{R}^{(i)}\|_{\mathbf{F}}^2}}. \quad (18)$$

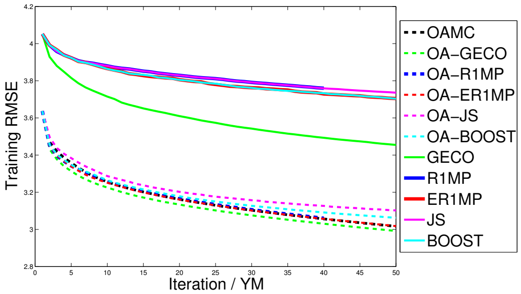

Figure 1: Iteration vs. log(Training RMSE). The line of OAMC overlaps those of OA-ER1MP and OA-R1MP.

It is well known that $\sigma_1^2(\mathbf{R}^{(k)}) \geq \frac{\|\mathbf{R}^{(k)}\|_{\mathbf{F}}^2}{\text{rank}(\mathbf{R}^{(k)})} \geq \frac{\|\mathbf{R}^{(k)}\|_{\mathbf{F}}^2}{\min(m,n)}$, from which we have $\frac{(1-\delta)^2\sigma_1(\mathbf{R}^{(k)})^2}{\|\mathbf{R}^{(k)}\|_{\mathbf{F}}^2} < 1$. Thus, there must be a constant $\gamma < 1$ that satisfies $\|\mathbf{R}^{(k+1)}\|_{\mathbf{F}} \leq \gamma^k \|\mathcal{P}_\Omega(\mathbf{M})\|_{\mathbf{F}}$. □

Note that OAMC, OA-GECO, and OA-ER1MP can also be found convergent linearly by slightly modifying (14) together with a proper initialization of the alternating minimization step.

5 Experiment

To make empirical evaluation, we conduct experiments of Recommendation and Image Recovery. OAMC and its variants are compared with five state-of-the-art T1SVD based competitors including GECO, R1MP, ER1MP, JS, and BOOST. All experiments are conducted on the same PC (Windows Server 2012 R2, Intel Xeon E5 2690v2*2 CPU, and 128G RAM).

We call the PROPACK to solve Top-1 SVD for T1SVD strategy. As for alternating minimization, we use random and the ApproxSV initialization respectively and set the maximum number of iterations to be ten. It turns out that random initialization does little harm to the convergence rate and the accuracy in our experiments. We only report the experiments with random initialization due to limited space.

For the parameter setting, we set the same maximum number of iteration for all the algorithms. And λ, the regularization parameter for BOOST, is selected by 3-fold cross validation. Additionally, JS requires a regularization parameter t, which is set to a doubled value of the nuclear norm solved by OA-ER1MP (This value is close in all OA variants).

To measure the performance, Root Mean Square Errors (RMSE) on both training set and testing set are calculated. We also record their running time (in seconds). Experimental results show that using OA strategy speeds up the

Table 1: Statistics for CF datasets

Dataset	#row	#column	#rating
MovieLens10M	69878	10677	1×10^7
NetFlix	461444	17770	1×10^8
Yahoo Music	1000990	624961	2.5×10^8

Table 2: Test RMSE (10^{-1}) and running time (sec) for 5 Images. We present the running time in square brackets next to RMSE. OA-JS achieves the best performance.

Image	OAMC	OA-GECO	OA-R1MP	OA-ER1MP	OA-JS	OA-BOOST	GECO	R1MP	ER1MP	JS	BOOST
Barbra	1.08[1.33]	1.07[48.2]	1.07[12.5]	1.07[1.11]	**1.05**[1.19]	1.07[1.11]	1.11[54.1]	1.13[19.7]	1.14[9.01]	1.09[8.87]	1.13[8.31]
Clown	0.84[1.08]	0.83[49.0]	0.83[12.3]	0.83[1.14]	**0.79**[1.19]	0.86[1.11]	0.89[54.3]	0.89[19.4]	0.89[8.79]	0.83[8.78]	0.88[8.40]
Couple	0.84[1.11]	0.83[47.7]	0.83[12.1]	0.84[1.11]	**0.81**[1.24]	0.83[1.13]	0.86[54.2]	0.89[19.6]	0.89[8.79]	0.84[8.52]	0.88[7.93]
Crowd	1.05[1.12]	1.02[47.8]	1.02[12.5]	1.03[1.12]	**1.00**[1.23]	1.03[1.23]	1.08[53.5]	1.09[19.5]	1.10[8.67]	1.05[9.03]	1.10[8.84]
Lenna	0.82[1.11]	0.81[47.4]	0.82[12.4]	0.82[1.10]	**0.77**[1.21]	0.81[1.13]	0.84[55.2]	0.88[19.3]	0.88[8.88]	0.82[8.33]	0.88[7.81]

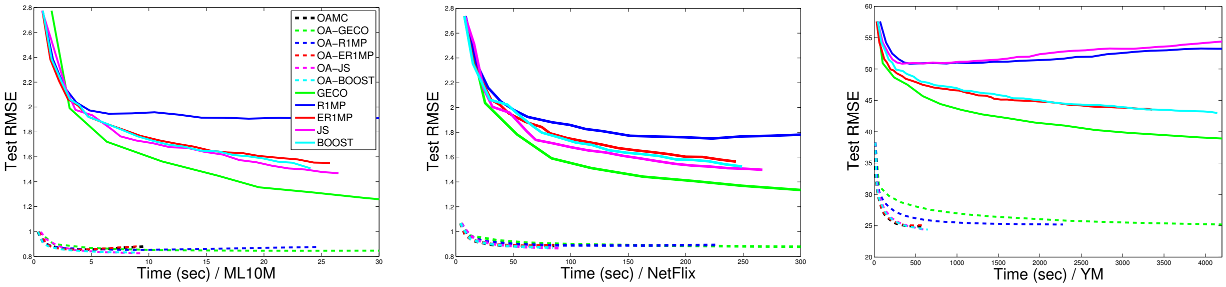

Figure 2: Time (sec) vs. Test RMSE. Lines terminate when the corresponding algorithms stop and results beyond the predefined time limit are not reported.

convergence and reduces the test error. Additionally, non-corrective algorithm OAMC performs well along with its variants, which implies that OA strategy reduces the importance of weight refinement procedure.

5.1 Recommendation

We use three largest publicly available datasets: MovieLens10M, NetFlix, and Yahoo Music to test the matrix completion based recommendation. The statistics of these datasets are listed in Table 1. All datasets are randomly split into **equal-sized** training and testing parts. The maximum iteration K for them are $\{20, 20, 50\}$ respectively. λ is chosen from $\{10^i, i \in \{-1, \cdots, 3\}\}$ by 3-fold cross validation.

In Figure 1, we plot the logarithm of training error of Yahoo Music dataset in each iteration to compare the convergence rates of different algorithms. We can observe two phases in all OA based methods in the figure. The first few iterations drastically reduce the training error, which we attribute to the approximation guarantee of OA: Theorem 1 shows that, when the gap between the singular values of the underlying matrix is large, OA can remove the impact of the largest singular value on the training error. In the following iterations, the training error decreases slower, but still linearly. This can be explained by Theorem 2: convergence is linear regardless of the distribution of the singular values. Additionally, all OA based methods have smaller training error than T1SVD based methods, even just after one iteration. Further, OAMC, OA-GECO, OA-R1MP, and OA-ER1MP have similar performance. This suggests the inequality (14) is indeed tight, which we attribute to the construction of OA. As for T1SVD based methods, GECO outperforms the rest by much. Such contrast makes it reasonable to infer that OA strategy substantially diminishes the contribution of weight update procedure.

We then plot test error over the running time to show the accuracy and efficiency of our methods in Figure 2. It shows that OA based methods achieve small test error with quite little known entries while T1SVD based methods fail in these situations. One reason is that the alternating minimization has recovery guarantee [Jain *et al.*, 2013; Gunasekar *et al.*, 2013] which ensures the better reconstruction of underlying matrix. We can also see that all OA based methods are more efficient than their T1SVD competitors and some OA variants are even **5 times faster**. It is worth emphasizing that some of our algorithms take only *700* seconds to process Yahoo Music dataset in PC, yet achieve a root mean square error *24.5*. We attribute this to the efficiency of alternating minimization procedure.

5.2 Image Recovery

In Image Recovery, we use five 512×512 sized gray-scale benchmark images[2]. Since images are typically high rank, we set rank $K = 200$. We uniformly retain 20% pixels as known entries. By conducting 10 independent trials for each image, average RMSE and running time are presented in Table 2.

We first compare the test error of two methods using the same weight refinement strategy but with different atom selection methods. The advantage of OA over T1SVD is clear, as all OA based methods outperform their T1SVD based competitor. Furthermore, we can see that even without further weight refinement, OAMC outperforms T1SVD based algorithms in most cases. Besides, OA based methods are also much faster, for example, OA-ER1MP is at least **7 times faster** than ER1MP on every image. Explanation for such observation is similar to the one in Recommendation.

6 Conclusion

We propose a novel atom selection strategy for greedy matrix completion called Optimal Atom, based on which several

[2] http://www.utdallas.edu/ cxc123730/mh_bcs_spl.html

algorithms are derived as well. Both approximation guarantee of OA and the convergence rate of our variants are established. Through two applications, Recommendation and Image Recovery, we demonstrate the superiority of our methods over existing T1SVD based algorithms. In the future work, we will further investigate the weight refinement step for OA.

Acknowledgments

This work is partially supported by National Natural Science Foundation of China (Grant No: 61472347). Hui Qian is supported by China Scholar Council (Grant No: 201406325003).

References

[Argyriou *et al.*, 2008] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Mach. Learn.*, 2008.

[Cai *et al.*, 2010] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Opt.*, 2010.

[Candès and Recht, 2009] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 2009.

[Candès and Tao, 2010] Emmanuel J. Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Trans. Inf. Theor.*, 2010.

[Chistov and Grigor'ev, 1984] Alexander L Chistov and D Yu Grigor'ev. Complexity of quantifier elimination in the theory of algebraically closed fields. In *Mathematical Foundations of Computer Science*. 1984.

[Eriksson *et al.*, 2011] Brian Eriksson, Laura Balzano, and Robert Nowak. High-rank matrix completion and subspace clustering with missing data. *CoRR*, 2011.

[Gunasekar *et al.*, 2013] Suriya Gunasekar, Ayan Acharya, Neeraj Gaur, and Joydeep Ghosh. Noisy matrix completion using alternating minimization. In *ECML*. 2013.

[Jaggi *et al.*, 2010] Martin Jaggi, Marek Sulovsk, et al. A simple algorithm for nuclear norm regularized problems. In *ICML*, 2010.

[Jain *et al.*, 2010] Prateek Jain, Raghu Meka, and Inderjit S Dhillon. Guaranteed rank minimization via singular value projection. In *NIPS*, 2010.

[Jain *et al.*, 2013] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *STOC*, 2013.

[Keshavan *et al.*, 2010] Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *Inf. Theor., IEEE Trans.*, 2010.

[Koren *et al.*, 2009] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 2009.

[Lee and Bresler, 2010] Kiryung Lee and Yoram Bresler. Admira: Atomic decomposition for minimum rank approximation. *Inf. Theor., IEEE Trans.*, 2010.

[Lin *et al.*, 2010] Zhouchen Lin, Minming Chen, and Yi Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.

[Liu *et al.*, 2013] Dehua Liu, Tengfei Zhou, Hui Qian, Congfu Xu, and Zhihua. A nearly unbiased matrix completion approach. In *ECML/PKDD 2013*, 2013.

[Liu *et al.*, 2014] Ji Liu, Jieping Ye, and Ryohei Fujimaki. Forward-backward greedy algorithms for general convex smooth functions over a cardinality constraint. In *ICML*, 2014.

[Mallat and Zhang, 1993] S.G. Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *Trans. Sig. Proc.*, 1993.

[Mazumder *et al.*, 2010] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *JMLR*, 2010.

[Obozinski *et al.*, 2010] Guillaume Obozinski, Ben Taskar, and Michael I. Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 2010.

[Pati *et al.*, 1993] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad. Orthonormal matching pursuit : recursive function approximation with applications to wavelet decomposition. In *Proceedings of the* 27^{th} *Annual Asilomar Conf. on Signals, Systems and Computers*, 1993.

[Rendle *et al.*, 2009] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, 2009.

[Rennie and Srebro, 2005] Jasson D. M. Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *ICML*, 2005.

[Shalev-Shwartz *et al.*, 2010] Shai Shalev-Shwartz, Nathan Srebro, and Tong Zhang. Trading accuracy for sparsity in optimization problems with sparsity constraints. *SIAM Journal on Opt.*, 2010.

[Shalev-Shwartz *et al.*, 2011] Shai Shalev-Shwartz, Alon Gonen, and Ohad Shamir. Large-scale convex minimization with a low-rank constraint. *arXiv preprint arXiv:1106.1622*, 2011.

[So and Ye, 2005] Anthony Man-Cho So and Yinyu Ye. Theory of semidefinite programming for sensor network localization. In *SODA*, 2005.

[Toh and Yun, 2010] Kim-Chuan Toh and Sangwoon Yun. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of Opt.*, 2010.

[Tropp, 2004] Joel A. Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Trans. Inf. Theor.*, 2004.

[Wang *et al.*, 2014] Zheng Wang, Ming-Jun Lai, Zhaosong Lu, and Jieping Ye. Orthogonal rank-one matrix pursuit for low rank matrix completion. *arXiv preprint arXiv:1404.1377*, 2014.

[Weinberger and Saul, 2006] K.Q. Weinberger and L.K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *IJCV*, 2006.

[Xu *et al.*, 2013] Miao Xu, Rong Jin, and Zhi-Hua Zhou. Speedup matrix completion with side information: Application to multi-label learning. In *NIPS*, 2013.

[Yi *et al.*, 2012] Jinfeng Yi, Tianbao Yang, Rong Jin, Anil K. Jain, and Mehrdad Mahdavi. Robust ensemble clustering by matrix completion. In *ICDM*, 2012.

[Zhang *et al.*, 2012] Xinhua Zhang, Dale Schuurmans, and Yaoliang Yu. Accelerated training for matrix-norm regularization: A boosting approach. In *NIPS*, 2012.

[Zhang, 2011] Tong Zhang. Adaptive forward-backward greedy algorithm for learning sparse representations. *Inf. Theor., IEEE Trans.*, 2011.