The official publication is available from now publishers via
http://dx.doi.org/10.1561/2200000058

Chapter 8

Low-rank Matrix Recovery

In this section, we will look at the problem of low-rank matrix recovery in detail. Although simple to motivate as an extension of the sparse recovery problem that we studied in § 7, the problem rapidly distinguishes itself in requiring specific tools, both algorithmic and analytic. We will start our discussion with a milder version of the problem as a warm up and move on to the problem of low-rank matrix completion which is an active area of research.

8.1 Motivating Applications

We will take the following two running examples to motivate the problem of low-rank matrix recovery.

Collaborative Filtering Recommendation systems are popularly used to model the preference patterns of users, say at an e-commerce website, for items being sold on that website, although the principle of recommendation extends to several other domains that demand *personalization* such as education and healthcare. Collaborative filtering is a popular technique for building recommendation systems.

The collaborative filtering approach seeks to exploit co-occurring patterns in the observed behavior across users in order to predict future user behavior. This approach has proven successful in addressing users that interact very sparingly with the system. Consider a set of m users $u_1, \ldots, u_m$, and n items $a_1, \ldots, a_n$. Our goal is to predict the preference score $s_{(i,j)}$ that is indicative of the interest user u_i has in item a_j.

However, we get direct access to (noisy estimates of) actual preference scores for only a few items per user by looking at clicks, purchases etc. That is to say, if we consider the $m \times n$ *preference matrix* $A = [A_{ij}]$ where $A_{ij} = s_{(i,j)}$ encodes the (true) preference of the i^{th} user for the j^{th} item, we get to see only $k \ll m \cdot n$ entries of A, as depicted in Figure 8.1. Our goal is to recover the remaining entries.

The problem of paucity of available data is readily apparent in this setting. In its nascent form, the problem is not even well posed and does not admit a unique solution. A popular way of overcoming these problems is to assume a low-rank structure in the preference matrix.

As we saw in Exercise 3.3, this is equivalent to assuming that there is an r-dimensional vector $\mathbf{u}_i$ denoting the i^{th} user and an r-dimensional vector $\mathbf{a}_j$ denoting the j^{th} such that $s_{(i,j)} \approx \langle \mathbf{u}_i, \mathbf{a}_j \rangle$. Thus, if $\Omega \subset [m] \times [n]$ is the set of entries that have been observed by us, then the problem of recovering the unobserved entries can be cast as the following optimization problem:

$$\min_{\substack{X \in \mathbb{R}^{m \times n} \\ \operatorname{rank}(X) \leq r}} \sum_{(i,j) \in \Omega} (X_{ij} - A_{ij})^2 .$$

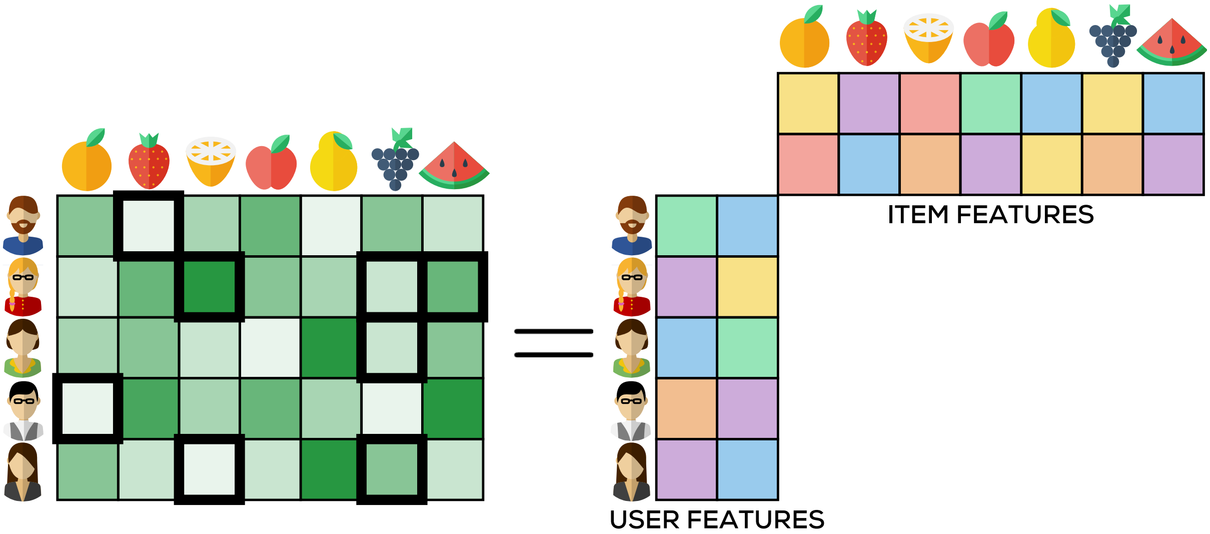


Figure 8.1: In a typical recommendation system, users rate items very infrequently and certain items may not get rated even once. The figure depicts a ratings matrix. Only the matrix entries with a bold border are observed. Low-rank matrix completion can help recover the unobserved entries, as well as reveal hidden features that are descriptive of user and item properties, as shown on the right hand side.

This problem can be shown to be NP-hard [Hardt et al., 2014] but has generated an enormous amount of interest across communities. We shall give special emphasis to this *matrix completion* problem in the second part of this section.

Linear Time-invariant Systems Linear Time-invariant (LTI) systems are widely used in modeling dynamical systems in fields such as engineering and finance. The *response behavior* of these systems is characterized by a model vector $\mathbf{h} = [h(0), h(1), \ldots, h(2N-1)]$. The *order* of such a system is given by the rank of the following Hankel matrix

$$\operatorname{hank}(\mathbf{h}) = \begin{bmatrix} h(0) & h(1) & \ldots & h(N) \\ h(1) & h(2) & \ldots & h(N+1) \\ \vdots & \vdots & \ddots & \vdots \\ h(N-1) & h(N) & \ldots & h(2N-1) \end{bmatrix}$$

Given a sequence of inputs $\mathbf{a} = [a(1), a(2), \ldots, a(N)]$ to the system, the output of the system is given by

$$y(N) = \sum_{t=0}^{N-1} a(N-t)h(t)$$

In order to recover the model parameters of a system, we repeatedly apply i.i.d. Gaussian impulses $a(i)$ to the system for N time steps and then observe the output of the system. This process is repeated, say k times, to yield observation pairs $\{(\mathbf{a}^i, y^i)\}_{i=1}^k$. Our goal now, is to take these observations and identify an LTI vector $\mathbf{h}$ that best fits the data. However, for the sake of accuracy and ease of analysis [Fazel et al., 2013], it is advisable to fit a low-order model to the data. Let the matrix $A \in \mathbb{R}^{k \times N}$ contain the i.i.d. Gaussian impulses applied to the system. Then the problem of fitting a low-order model can be shown to reduce to the following constrained optimization problem with a rank objective and an affine constraint.

$$\begin{array}{ll} \min & \mathrm{rank}(\mathrm{hank}(\mathbf{h})) \\ \text{s.t.} & A\mathbf{h} = \mathbf{y}, \end{array}$$

The above problem is a non-convex optimization problem due to the objective being the minimization of the rank of a matrix. Several other problems in metric embedding and multi-dimensional scaling, image compression, low rank kernel learning and spatio-temporal imaging can also be reduced to low rank matrix recovery problems [Jain et al., 2010, Recht et al., 2010].

8.2 Problem Formulation

The two problems considered above can actually be cast in a single problem formulation, that of *Affine Rank Minimization* (ARM). Consider a low rank matrix $X^* \in \mathbb{R}^{m\times n}$ that we wish to recover and an affine transformation $\mathcal{A} : \mathbb{R}^{m\times n} \to \mathbb{R}^k$. The transformation can be seen as a concatenation of k real valued affine transformations $\mathcal{A}_i : \mathbb{R}^{m\times n} \to \mathbb{R}$. We are given the transformation $\mathcal{A}$, and its (possibly noisy) action $\mathbf{y} = \mathcal{A}(X^*) \in \mathbb{R}^k$ on the matrix X^* and our goal is to recover this matrix by solving the following optimization problem.

$$\begin{array}{ll} \min & \mathrm{rank}(X) \\ \text{s.t.} & \mathcal{A}(X) = \mathbf{y}, \end{array} \tag{ARM}$$

This problem can be shown to be NP-hard due to a reduction to the sparse recovery problem[1]. The LTI modeling problem can be easily seen to be an instance of ARM with the Gaussian impulses being delivered to the system resulting in a k-dimensional affine transformation of the Hankel matrix corresponding to the system. However, the Collaborative Filtering problem is also an instance of ARM. To see this, for any $(i, j) \in [m] \times [n]$, let $O^{(i,j)} \in \mathbb{R}^{m\times n}$ be the matrix such that its (i, j)-th entry $O^{(i,j)}_{ij} = 1$ and all other entries are zero. Then, simply define the affine transformation $\mathcal{A}_{(i,j)} : X \mapsto \mathrm{tr}(X^\top O^{(i,j)}) = X_{ij}$. Thus, if we observe k user-item ratings, the ARM problem effectively operates with a k-dimensional affine transformation of the underlying rating matrix.

Due to its similarity to the sparse recovery problem, we will first discuss the general ARM problem. However, we will find it beneficial to cast the collaborative filtering problem as a *Low-rank Matrix Completion* problem instead. In this problem, we have an underlying low rank matrix X^* of which, we observe entries in a set $\Omega \subset [m] \times [n]$. Then the low rank matrix completion problem can be stated as

$$\min_{\substack{X\in\mathbb{R}^{m\times n}\\ \mathrm{rank}(X)\leq r}} \left\|\Pi_\Omega(X - X^*)\right\|_F^2, \tag{LRMC}$$

where $\Pi_\Omega(X)$ is defined, for any matrix X as

$$\Pi_\Omega(X)_{i,j} = \begin{cases} X_{i,j} & \text{if } (i, j) \in \Omega \\ 0 & \text{otherwise.} \end{cases}$$

The above formulation succinctly captures our objective to find a *completion* of the ratings matrix that is both, low rank, as well as agrees on the user ratings that are actually observed. As pointed out earlier, this problem is NP-hard [Hardt et al., 2014].

Before moving on to present algorithms for the ARM and LRMC problems, we discuss some matrix design properties that would be required in the convergence analyses of the algorithms.

[1]See Exercise 8.1.

8.3 Matrix Design Properties

Similar to sparse recovery, there exist design properties that ensure that the general NP-hardness of the ARM and LRMC problems can be overcome in well-behaved problem settings. In fact given the similarity between ARM and sparse recovery problems, it is tempting to try and import concepts such as RIP into the matrix-recovery setting.

In fact this is exactly the first line of attack that was adopted in literature. What followed was a beautiful body of work that generalized, both structural notions such as RIP, as well as algorithmic techniques such as IHT, to address the ARM problem. Given the generality of these constructs, as well as the smooth transition it offers having studied sparse recovery, we feel compelled to present them to the reader.

8.3.1 The Matrix Restricted Isometry Property

The generalization of RIP to matrix settings, referred to as matrix RIP, follows in a relatively straightforward manner and was first elucidated by Recht et al. [2010]. Quite in line with the sparse recovery setting, the intuition dictates that recovery should be possible only if the affine transformation does not identify two distinct low-rank matrices. A more robust version dictates that no low-rank matrix should be distorted significantly by this transformation which gives us the following.

Definition 8.1 (Matrix Restricted Isometry Property [Recht et al., 2010]). *A linear map $\mathcal{A} : \mathbb{R}^{m\times n} \to \mathbb{R}^k$ is said to satisfy the matrix restricted isometry property of order r with constant $\delta_r \in [0, 1)$ if for all matrices X of rank at most r, we have*

$$(1-\delta_r)\cdot \|X\|_F^2 \leq \|\mathcal{A}(X)\|_2^2 \leq (1+\delta_r)\cdot \|X\|_F^2.$$

Furthermore, the work of Recht et al. [2010] also showed that linear maps or affine transformations arising in random measurement models, such as those in image compression and LTI systems, do satisfy RIP with requisite constants whenever the number of affine measurements satisfies $k = \mathcal{O}(nr)$ [Oymak et al., 2015]. Note however, that these are settings in which the design of the affine map is within our control. For settings, where the restricted condition number of the affine map is not within our control, more involved analysis is required. The bibliographic notes point to some of these results.

Given the relatively simple extension of the RIP definitions to the matrix setting, it is all the more tempting to attempt to apply gPGD-style techniques to solve the ARM problem, particularly since we saw how IHT succeeded in offering scalable solutions to the sparse recovery problem. The works of [Goldfarb and Ma, 2011, Jain et al., 2010] showed that this is indeed possible. We will explore this shortly.

8.3.2 The Matrix Incoherence Property

We begin this discussion by warning the reader that there are two distinct notions prevalent in literature, both of which are given the same name, that of the matrix incoherence property. The first of these notions was introduced in § 7.6 as a property that can be used to ensure the RIP property in matrices. However a different property, but bearing the same name, finds application in matrix completion problems which we now introduce. We note that the two properties are not known to be strongly related in a formal sense and the coincidental clash of the names seems to be a matter of legacy.

Nevertheless, the intuition behind the second notion of matrix incoherence is similar to that for RIP in that it seeks to make the problem well posed. Consider the matrix $A = \sum_{t=1}^{r} s_t \cdot$

Algorithm 9 Singular Value Projection (SVP)

Input: Linear map $\mathcal{A}$, measurements $\mathbf{y}$, target rank q, step length η
Output: A matrix $\widehat{X}$ with rank at most q

$X^1 \leftarrow \mathbf{0}^{m \times n}$
for $t = 1, 2, \ldots$ **do**
  $Y^{t+1} \leftarrow X^t - \eta \cdot \mathcal{A}^\top(\mathcal{A}(X^t) - \mathbf{y})$
  Compute top q singular vectors/values of Y^{t+1}: U_q^t, Σ_q^t, V_q^t
  $X^{t+1} \leftarrow U_q^t \Sigma_q^t (V_q^t)^\top$
end for
return X^t

$\mathbf{e}_{i_t}\overline{\mathbf{e}}_{j_t}^\top \in \mathbb{R}^{m \times n}$ where $\mathbf{e}_i$ are the canonical orthonormal vectors in $\mathbb{R}^m$ and $\overline{\mathbf{e}}_j$ are the canonical orthonormal vectors in $\mathbb{R}^n$. Clearly A has rank at most r.

However, this matrix A is non-zero only at r locations. Thus, it is impossible to recover the entire matrix uniquely unless these very r locations $\{(i_t, j_t)\}_{t=1,\ldots,r}$ are actually observed. Since in recommendation settings, we only observe a few random entries of the matrix, there is a good possibility that none of these entries will ever be observed. This presents a serious challenge for the matrix completion problem – the low rank structure is not sufficient to ensure unique recovery!

To overcome this and make the LRMC problem well posed with a unique solution, an additional property is imposed. This so-called matrix incoherence property prohibits low rank matrices that are also sparse. A side effect of this imposition is that for incoherent matrices, observing a small random set of entries is enough to uniquely determine the unobserved entries of the matrix.

Definition 8.2 (Matrix Incoherence Property [Candès and Recht, 2009]). *A matrix $A \in \mathbb{R}^{m \times n}$ of rank r is said to be incoherent with parameter μ if its left and right singular matrices have bounded row norms. More specifically, let $A = U\Sigma V^\top$ be the SVD of A. Then μ-incoherence dictates that $\left\|U^i\right\|_2 \leq \frac{\mu\sqrt{r}}{\sqrt{m}}$ for all $i \in [m]$ and $\left\|V^j\right\|_2 \leq \frac{\mu\sqrt{r}}{\sqrt{n}}$ for all $j \in [n]$. A stricter version of this property requires all entries of U to satisfy $|U_{ij}| \leq \frac{\mu}{\sqrt{m}}$ and all entries of V to satisfy $|V_{ij}| \leq \frac{\mu}{\sqrt{n}}$.*

A low rank incoherent matrix is guaranteed to be *far*, i.e., well distinguished, from any sparse matrix, something that is exploited by algorithms to give guarantees for the LRMC problem.

8.4 Low-rank Matrix Recovery via Proj. Gradient Descent

We will now apply the gPGD algorithm to the ARM problem. To do so, first consider the following reformulation of the ARM problem to make it more compatible to the projected gradient descent iterations.

$$\begin{array}{ll} \min & \frac{1}{2}\left\|\mathcal{A}(X) - \mathbf{y}\right\|_2^2 \\ \text{s.t.} & \operatorname{rank}(X) \leq r \end{array} \tag{ARM-2}$$

Applying the gPGD algorithm to the above formulation gives us the *Singular Value Projection* (SVP) algorithm (Algorithm 9). Note that in this case, the projection needs to be carried out onto the set of low rank matrices. However, as we saw in § 3.1, this can be efficiently done by computing the singular value decomposition of the iterates.

SVP offers ease of implementation and speed similar to IHT. Moreover, it applies to ARM problems in general. If the Matrix RIP property is appropriately satisfied, then SVP guarantees a linear rate of convergence to the optimum, much like IHT. All these make SVP a very attractive choice for solving low rank matrix recovery problems.

Below, we give a convergence proof for SVP in the noiseless case, i.e., when $\mathbf{y} = \mathcal{A}(X^*)$. The proof is similar in spirit to the convergence proof we saw for the IHT algorithm in § 7 but differs in crucial aspects since sparsity in this case is apparent not in the signal domain (the matrix is not itself sparse) but the spectral domain (the set of singular values of the matrix is sparse). The analysis can be extended to noisy measurements as well and can be found in [Jain et al., 2010].

8.5 A Low-rank Matrix Recovery Guarantee for SVP

We will now present a convergence result for the SVP algorithm. As before, although the general convergence result for gPGD can indeed be applied here, we will see, just as we did in § 7, that the problem specific analysis we present here is finer and reveals more insights about the ARM problem structure.

Theorem 8.1. *Suppose $\mathcal{A} : \mathbb{R}^{m\times n} \to \mathbb{R}^k$ is an affine transformation that satisfies the matrix RIP property of order $2r$ with constant $\delta_{2r} \le \frac{1}{3}$. Let $X^* \in \mathbb{R}^{m\times n}$ be a matrix of rank at most r and let $\mathbf{y} = \mathcal{A}(X^*)$. Then the SVP Algorithm (Algorithm 9), when executed with a step length $\eta = 1/(1+\delta_{2r})$, and a target rank $q = r$, ensures $\left\|X^t - X^*\right\|_F^2 \le \epsilon$ after $t = \mathcal{O}\left(\log \frac{\|\mathbf{y}\|_2^2}{2\epsilon}\right)$ iterations of the algorithm.*

Proof. Notice that the notions of *sparsity* and *support* are very different in ARM than what they were for sparse regression. Consequently, the exact convergence proof for IHT (Theorem 7.2) is not applicable here. We will first establish an intermediate result that will show, that after $t = \mathcal{O}\left(\log \frac{\|\mathbf{y}\|_2^2}{2\epsilon}\right)$ iterations, SVP ensures $\left\|\mathcal{A}(X^t) - \mathbf{y}\right\|_2^2 \le \epsilon$. We will then use the matrix RIP property (Definition 8.1) to deduce

$$\left\|\mathcal{A}(X^t) - \mathbf{y}\right\|_2^2 = \left\|\mathcal{A}(X^t - X^*)\right\|_2^2 \ge (1-\delta_{2r}) \cdot \left\|X^t - X^*\right\|_F^2,$$

which will conclude the proof. To prove this intermediate result, let us denote the objective function as

$$f(X) = \frac{1}{2}\left\|\mathcal{A}(X) - \mathbf{y}\right\|_2^2 = \frac{1}{2}\left\|\mathcal{A}(X - X^*)\right\|_2^2.$$

An application of the matrix RIP property then gives us

$$\begin{aligned}
& f(X^{t+1}) \\
&= f(X^t) + \left\langle \mathcal{A}(X^t - X^*), \mathcal{A}(X^{t+1} - X^t)\right\rangle + \frac{1}{2}\left\|\mathcal{A}(X^{t+1} - X^t)\right\|_2^2 \\
&\le f(X^t) + \left\langle \mathcal{A}(X^t - X^*), \mathcal{A}(X^{t+1} - X^t)\right\rangle + \frac{(1+\delta_{2r})}{2}\left\|X^{t+1} - X^t\right\|_F^2.
\end{aligned}$$

The following steps now introduce the intermediate variable Y^{t+1} into the analysis in order to link the successive iterates by using the fact that X^{t+1} was the result of a non-convex projection operation.

$$\begin{aligned}
&\left\langle \mathcal{A}(X^t - X^*), \mathcal{A}(X^{t+1} - X^t)\right\rangle + \frac{(1+\delta_{2r})}{2} \cdot \left\|X^{t+1} - X^t\right\|_F^2 \\
&\quad = \frac{1+\delta_{2r}}{2} \cdot \left\|X^{t+1} - Y^{t+1}\right\|_F^2 - \frac{1}{2(1+\delta_{2r})} \cdot \left\|\mathcal{A}^\top(\mathcal{A}(X^t - X^*))\right\|_F^2
\end{aligned}$$

Algorithm 10 AltMin for Matrix Completion (AM-MC)

Input: Matrix $A \in \mathbb{R}^{m \times n}$ of rank r observed at entries in the set Ω, sampling probability p, stopping time T

Output: A matrix $\widehat{X}$ with rank at most r

Partition Ω into $2T+1$ sets $\Omega_0, \Omega_1, \ldots, \Omega_{2T}$ uniformly and randomly

$U^1 \leftarrow \text{SVD}(\frac{1}{p}\Pi_{\Omega_0}(A), r)$, the top r left singular vectors of $\frac{1}{p}\Pi_{\Omega_0}(A)$

for $t = 1, 2, \ldots, T$ **do**

$\quad V^{t+1} \leftarrow \arg\min_{V \in \mathbb{R}^{n \times r}} \left\| \Pi_{\Omega_t}(U^t V^\top - A) \right\|_F^2$

$\quad U^{t+1} \leftarrow \arg\min_{U \in \mathbb{R}^{m \times r}} \left\| \Pi_{\Omega_{T+t}}(U (V^{t+1})^\top - A) \right\|_F^2$

end for

return $U^\top (V^\top)^\top$

$$
\begin{aligned}
&\leq \frac{1+\delta_{2r}}{2} \cdot \left\| X^* - Y^{t+1} \right\|_F^2 - \frac{1}{2(1+\delta_{2r})} \cdot \left\| \mathcal{A}^\top(\mathcal{A}(X^t - X^*)) \right\|_F^2 \\
&= \left\langle \mathcal{A}(X^t - X^*), \mathcal{A}(X^* - X^t) \right\rangle + \frac{(1+\delta_{2r})}{2} \cdot \left\| X^* - X^t \right\|_F^2 \\
&\leq \left\langle \mathcal{A}(X^t - X^*), \mathcal{A}(X^* - X^t) \right\rangle + \frac{(1+\delta_{2r})}{2(1-\delta_{2r})} \cdot \left\| \mathcal{A}(X^* - X^t) \right\|_2^2 \\
&= -f(X^t) - \frac{1}{2} \left\| \mathcal{A}(X^* - X^t) \right\|_2^2 + \frac{(1+\delta_{2r})}{2(1-\delta_{2r})} \cdot \left\| \mathcal{A}(X^* - X^t) \right\|_2^2 .
\end{aligned}
$$

The first step uses the identity $Y^{t+1} = X^t - \eta \cdot \mathcal{A}^\top(\mathcal{A}(X^t) - \mathbf{y})$ from Algorithm 9, the fact that we set $\eta = \frac{1}{1+\delta_{2r}}$, and elementary rearrangements. The second step follows from the fact that $\left\| X^{t+1} - Y^{t+1} \right\|_F^2 \leq \left\| X^* - Y^{t+1} \right\|_F^2$ by virtue of the SVD step which makes X^{t+1} the best rank-$(2r)$ approximation to Y^{t+1} in terms of the Frobenius norm. The third step simply rearranges things in the reverse order of the way they were arranged in the first step, the fourth step uses the matrix RIP property and the fifth step makes elementary manipulations. This, upon rearrangement, and using $\left\| \mathcal{A}(X^t - X^*) \right\|_2^2 = 2f(X^t)$, gives us

$$
f(X^{t+1}) \leq \frac{2\delta_{2r}}{1-\delta_{2r}} \cdot f(X^t).
$$

Since $f(X^1) = \|\mathbf{y}\|_2^2$, as we set $X^1 = \mathbf{0}^{m \times n}$, if $\delta_{2r} < 1/3$ (i.e., $\frac{2\delta_{2r}}{1-\delta_{2r}} < 1$), we have the claimed convergence result. □

One can, in principle, apply the SVP technique to the matrix completion problem as well. However, on the LMRC problem, SVP is outperformed by gAM-style approaches which we study next. Although the superior performance of gAM on the LMRC problem was well documented empirically, it took some time before a theoretical understanding could be obtained. This was first done in the works of Keshavan [2012], Jain et al. [2013]. These results set off a long line of works that progressively improved both the algorithm, as well as its analysis.

8.6 Matrix Completion via Alternating Minimization

We will now look at the alternating minimization technique for solving the low-rank matrix completion problem. As we have observed before, the LRMC problem admits an equivalent reformulation where the low rank structure constraint is eliminated and instead, the solution is described in terms of two low-rank components

$$\min_{\substack{U \in \mathbb{R}^{m\times k}\\ V\in\mathbb{R}^{n\times k}}} \left\|\Pi_\Omega(UV^\top - X^*)\right\|_F^2. \tag{LRMC*}$$

In this case, fixing either U or V reduces the above problem to a simple least squares problem for which we have very efficient and scalable solvers. As we saw in § 4, such problems are excellent candidates for the gAM algorithm to be applied. The AM-MC algorithm (see Algorithm 10) applies the gAM approach to the reformulated LMRC problem. The AM-MC approach is the choice of practitioners in the context of collaborative filtering [Chen and He, 2012, Koren et al., 2009, Zhou et al., 2008]. However, AM-MC, like other gAM-style algorithms, does require proper initialization and tuning.

8.7 A Low-rank Matrix Completion Guarantee for AM-MC

We will now analyze the convergence properties of the AM-MC method for matrix completion. To simplify the presentation, we will restrict ourselves to the case when the matrix $A \in \mathbb{R}^{m\times n}$ is rank one. This will allow us to present the essential arguments without getting involved with technical details. Let $A = \mathbf{u}^*(\mathbf{v}^*)^\top$ be a μ-incoherent matrix that needs to be recovered, where $\|\mathbf{u}^*\|_2 = \|\mathbf{v}^*\|_2 = 1$. It is easy to see that μ-incoherence implies $\|\mathbf{u}^*\|_\infty \leq \frac{\mu}{\sqrt{m}}$ and $\|\mathbf{v}^*\|_\infty \leq \frac{\mu}{\sqrt{n}}$.

We will also assume the Bernoulli sampling model i.e., that the set of observed indices Ω is generated by selecting each entry (i,j) for inclusion in Ω in an i.i.d. fashion with probability p. More specifically, $(i,j) \in \Omega$ iff $\delta_{ij} = 1$ where $\delta_{ij} = 1$ with probability p and 0 otherwise.

For simplicity, we will assume that each iteration of the AM-MC procedure receives a fresh set of samples Ω from A. This can be achieved in practice by randomly partitioning the available set of samples into as many groups as the iterations of the procedure. The completion guarantee will proceed in two steps. In the first step, we will show that the initialization step in Algorithm 10 itself brings AM-MC within a constant distance of the optimal solution. Next, we will show that this close initialization is sufficient for AM-MC to ensure a linear rate of convergence to the optimal solution.

Initialization: We will now show that the initialization step (Step 2 in Algorithm 10) provides a point $(\mathbf{u}^1, \mathbf{v}^1)$ which is at most a constant $c > 0$ distance away from $(\mathbf{u}^*, \mathbf{v}^*)$. To this we need a Bernstein-style argument which we provide here for the rank-1 case.

Theorem 8.2. *[Tropp, 2012, Theorem 1.6] Consider a finite sequence $\{Z_k\}$ of independent random matrices of dimension $m \times n$. Assume each matrix satisfies $\mathbb{E}[Z_k] = \mathbf{0}$ and $\|Z_k\|_2 \leq R$ almost surely and denote*

$$\sigma^2 := \max\left\{\left\|\textstyle\sum_k Z_k Z_k^\top\right\|_2, \left\|\textstyle\sum_k Z_k^\top Z_k\right\|_2\right\}.$$

Then, for all $t \geq 0$, we have

$$\mathbb{P}\left[\|\textstyle\sum_k Z_k\|_2 \geq t\right] \leq (m+n)\cdot\exp\left(\frac{-t^2}{\sigma^2 + Rt/3}\right).$$

Below we apply this inequality to analyze the initialization step for AM-MC in the rank-1 case. We point the reader to [Recht, 2011] and [Keshavan et al., 2010] for a more precise argument analyzing the initialization step in the general rank-r case.

Theorem 8.3. *For a rank one matrix A satisfying the μ-incoherence property, let the observed samples Ω be generated with sampling probability p as described above. Let $\mathbf{u}^1, \mathbf{v}^1$ be the singular* *vectors of* $\frac{1}{p}P_\Omega(A)$ *corresponding to its largest singular value. Then for any* $\epsilon > 0$*, if* $p \geq \frac{45\mu^2}{\epsilon^2} \cdot \frac{\log(m+n)}{\min\{m,n\}}$*, then with probability at least* $1 - 1/(m+n)^{10}$*:*

$$\left\|\frac{1}{p}\Pi_\Omega(A) - A\right\|_2 \leq \epsilon, \quad \left\|\mathbf{u}^1 - \mathbf{u}^*\right\|_2 \leq \epsilon, \quad \left\|\mathbf{v}^1 - \mathbf{v}^*\right\|_2 \leq \epsilon.$$

Moreover, the vectors $\mathbf{u}^1$ *and* $\mathbf{v}^1$ *are also* 2μ*-incoherent.*

Proof. Notice that the statement of the theorem essentially states that once enough entries in the matrix have been observed (as dictated by the requirement $p \geq \frac{45\mu^2}{\epsilon^2} \cdot \frac{\log(m+n)}{\min\{m,n\}}$) an SVD step on the incomplete matrix will yield components $\mathbf{u}^1, \mathbf{v}^1$ that are very close to the components of the complete matrix $\mathbf{u}^*, \mathbf{v}^*$. Moreover, since $\mathbf{u}^*, \mathbf{v}^*$ are incoherent by assumption, the estimated components $\mathbf{u}^1, \mathbf{v}^1$ will be so too.

To apply Theorem 8.2 to prove this result, we will first express $\frac{1}{p}\Pi_\Omega(A)$ as a sum of random matrices. We first rewrite $\frac{1}{p}\Pi_\Omega(A) = \frac{1}{p}\sum_{ij}\delta_{ij}A_{ij}\mathbf{e}_i\mathbf{e}_j^\top = \sum_{ij} W_{ij}$ where $\delta_{ij} = 1$ if $(i,j) \in \Omega$ and 0 otherwise. Note that the Bernoulli sampling model assures us that the random variables $W_{ij} = \frac{1}{p}\delta_{ij}A_{ij}\mathbf{e}_i\mathbf{e}_j^\top$ are independent and that $\mathbb{E}\left[\delta_{ij}\right] = p$. This gives us $\mathbb{E}\left[W_{ij}\right] = A_{ij}\mathbf{e}_i\mathbf{e}_j^\top$. Note that $\sum_{ij} A_{ij}\mathbf{e}_i\mathbf{e}_j^\top = A$.

The matrices $Z_{ij} = W_{ij} - A_{ij}\mathbf{e}_i\mathbf{e}_j^\top$ shall serve as our *random matrices* in the application of Theorem 8.2. Clearly $\mathbb{E}\left[Z_{ij}\right] = \mathbf{0}$. We also have $\max_{ij}\left\|W_{ij}\right\|_2 \leq \frac{1}{p}\max_{ij}\left|A_{ij}\right| \leq \frac{\mu^2}{p\sqrt{mn}}$ due to the incoherence assumption. Applying the triangle inequality gives us $\max_{ij}\left\|Z_{ij}\right\|_2 \leq \max_{ij}\left\|W_{ij}\right\|_2 + \max_{ij}\left\|A_{ij}\mathbf{e}_i\mathbf{e}_j^\top\right\|_2 \leq \left(1 + \frac{1}{p}\right)\frac{\mu^2}{\sqrt{mn}} \leq \frac{2\mu^2}{p\sqrt{mn}}$.

Moreover, as $A_{ij} = \mathbf{u}_i^*\mathbf{v}_j^*$ and $\left\|\mathbf{v}^*\right\|_2 = 1$, we have $\mathbb{E}\left[\sum_{ij} W_{ij}W_{ij}^\top\right] = \frac{1}{p}\sum_i\sum_j A_{ij}^2\mathbf{e}_i\mathbf{e}_i^\top = \frac{1}{p}\sum_i(\mathbf{u}_i^*)^2\mathbf{e}_i\mathbf{e}_i^\top$. Due to incoherence $\left\|\mathbf{u}^*\right\|_\infty \leq \frac{\mu}{\sqrt{m}}$, we get $\left\|\mathbb{E}\left[\sum_{ij} W_{ij}W_{ij}^\top\right]\right\|_2 \leq \frac{\mu^2}{p\cdot m}$, which can be shown to give us

$$\left\|\mathbb{E}\left[\sum_{ij} Z_{ij}Z_{ij}^\top\right]\right\|_2 \leq \left(\frac{1}{p} - 1\right)\cdot\frac{\mu^2}{m} \leq \frac{\mu^2}{p\cdot m}$$

Similarly, we can also get $\left\|\mathbb{E}\left[\sum_{ij} Z_{ij}^\top Z_{ij}\right]\right\|_2 \leq \frac{\mu^2}{p\cdot n}$. Now using Theorem 8.2 gives us, with probability at least $1 - \delta$,

$$\left\|\frac{1}{p}\Pi_\Omega(A) - A\right\|_2 \leq \frac{2\mu^2}{3p\sqrt{mn}}\log\left[\frac{m+n}{\delta}\right] + \sqrt{\frac{\mu^2}{p\cdot\min\{m,n\}}\log\left[\frac{m+n}{\delta}\right]}$$

If $p \geq \frac{45\mu^2\log(m+n)}{\epsilon^2\cdot\min\{m,n\}}$, we have with probability at least $1 - 1/(m+n)^{10}$,

$$\left\|\frac{1}{p}\Pi_\Omega(A) - A\right\|_2 \leq \epsilon.$$

The proof now follows by applying the Davis-Kahan inequality [Golub and Loan, 1996] with the above bound. It can be shown [Jain and Netrapalli, 2015] that the vectors that are recovered as a result of this initialization are incoherent as well. □

Linear Convergence: We will now show that, given the initialization above, the AM-MC procedure converges to the true solution with a linear rate of convergence. This will involve showing a few intermediate results, such as showing that the alternation steps preserve incoherence. Since the Theorem 8.3 shows that $\mathbf{u}^1$ is 2μ-incoherent, this will establish the incoherence of all future iterates. Preserving incoherence will be crucial in showing the next result which shows that successive iterates get increasingly close to the optimum. Put together, these will establish the convergence result. First, recall that in the t^{th} iteration of the AM-MC algorithm, $\mathbf{v}^{t+1}$ is updated as

$$\mathbf{v}^{t+1} = \arg\min_{\mathbf{v}} \sum_{ij} \delta_{ij}(\mathbf{u}_i^t \mathbf{v}_j - \mathbf{u}_i^* \mathbf{v}_j^*)^2,$$

which gives us

$$\mathbf{v}_j^{t+1} = \frac{\sum_i \delta_{ij} \mathbf{u}_i^* \mathbf{u}_i^t}{\sum_i \delta_{ij} (\mathbf{u}_i^t)^2} \cdot \mathbf{v}_j^*. \tag{8.1}$$

Note that this means that if $\mathbf{u}^* = \mathbf{u}^t$, then $\mathbf{v}^{t+1} = \mathbf{v}^*$. Also, note that if $\delta_{ij} = 1$ for all (i,j) which happens when the sampling probability satisfies $p = 1$, we have $\mathbf{v}^{t+1} = \frac{\langle \mathbf{u}^t, \mathbf{u}^* \rangle}{\|\mathbf{u}^t\|_2^2} \cdot \mathbf{v}^*$. This is reminiscent of the *power method* used to recover the leading singular vectors of a matrix. Indeed if we let $\tilde{\mathbf{u}} = \mathbf{u}^t / \|\mathbf{u}^t\|_2$, we get $\|\mathbf{u}^t\|_2 \cdot \mathbf{v}^{t+1} = \langle \tilde{\mathbf{u}}, \mathbf{u}^* \rangle \cdot \mathbf{v}^*$ if $p = 1$.

This allows us to rewrite the update (8.1) as a noisy power update.

$$\left\|\mathbf{u}^t\right\|_2 \cdot \mathbf{v}^{t+1} = \langle \tilde{\mathbf{u}}, \mathbf{u}^* \rangle \cdot \mathbf{v}^* - B^{-1}(\langle \tilde{\mathbf{u}}, \mathbf{u}^* \rangle B - C)\mathbf{v}^* \tag{8.2}$$

where $B, C \in \mathbb{R}^{n \times n}$ are diagonal matrices with $B_{jj} = \frac{1}{p}\sum_i \delta_{ij}(\tilde{\mathbf{u}}_i)^2$ and $C_{jj} = \frac{1}{p}\sum_i \delta_{ij}\tilde{\mathbf{u}}_i \mathbf{u}_i^*$. The following two lemmata show that if $\mathbf{u}^t$ is 2μ incoherent and if p is large enough, then: a) $\mathbf{v}^{t+1}$ is also 2μ incoherent, and b) the angular distance between $\mathbf{v}^{t+1}$ and $\mathbf{v}^*$ decreases as compared to that between $\mathbf{u}^t$ and $\mathbf{u}^*$. The following lemma will aid the analysis.

Lemma 8.4. *Suppose* $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ *are two fixed* μ*-incoherent unit vectors. Also suppose* $\delta_i, i \in [n]$ *are i.i.d. Bernoulli random variables such that* $\delta_i = 1$ *with probability* p *and 0 otherwise. Then, for any* $\epsilon > 0$, *if* $p > \frac{27\mu^2 \log n}{n\epsilon^2}$, *then with probability at least* $1 - 1/n^{10}$, $\left|\frac{1}{p}\sum_i \delta_i \mathbf{a}_i \mathbf{b}_i - \langle \mathbf{a}, \mathbf{b} \rangle\right| \leq \epsilon$.

Proof. Define $Z_i = \left(\frac{\delta_i}{p} - 1\right)\mathbf{a}_i \mathbf{b}_i$. Using the incoherence of the vectors, we get $\mathbb{E}[Z_i] = 0$, $\sum_{i=1}^n \mathbb{E}[Z_i^2] = \left(\frac{1}{p} - 1\right)\sum_{i=1}^n (\mathbf{a}_i \mathbf{b}_i)^2 \leq \frac{\mu^2}{pn}$ since $\|\mathbf{b}\|_2 = 1$, and $|Z_i| \leq \frac{\mu^2}{pn}$ almost surely. Applying the Bernstein inequality gives us

$$\mathbb{P}\left[\left|\frac{1}{p}\sum_i \delta_i \mathbf{a}_i \mathbf{b}_i - \langle \mathbf{a}, \mathbf{b} \rangle\right| > t\right] \leq \exp\left(\frac{-3pnt^2}{6\mu^4 + 2\mu^2 t}\right),$$

which upon simple manipulations, gives us the result. □

Lemma 8.5. *With probability at least* $\min\{1 - 1/n^{10}, 1 - 1/m^{10}\}$, *if a pair of iterates* $(\mathbf{u}^t, \mathbf{v}^t)$ *in the execution of the AM-MC procedure are* 2μ*-incoherent, then so are the next pair of iterates* $(\mathbf{u}^{t+1}, \mathbf{v}^{t+1})$.

Proof. Since $\|\tilde{\mathbf{u}}\|_2 = 1$, using Lemma 8.4 tells us that with high probability, for all j, we have $|B_{jj} - 1| \leq \epsilon$ as well as $|C_{jj} - \langle \tilde{\mathbf{u}}, \mathbf{u}^* \rangle| \leq \epsilon$. Also, using triangle inequality, we get $\|\mathbf{u}^t\|_2 \geq 1 - \epsilon$. Using these and the incoherence of $\mathbf{v}^*$ in the update equation for $\mathbf{v}^{t+1}$ (8.2), we have

$$\begin{aligned}
\left|\mathbf{v}_j^{t+1}\right| &= \frac{1}{1-\epsilon}\left|\langle \tilde{\mathbf{u}}, \mathbf{u}^* \rangle \mathbf{v}_j^* - \frac{1}{B_{jj}}(\langle \tilde{\mathbf{u}}, \mathbf{u}^* \rangle B_{jj} - C_{jj})\mathbf{v}_j^*\right| \\
&\leq \frac{1}{1-\epsilon}\left|\langle \tilde{\mathbf{u}}, \mathbf{u}^* \rangle \mathbf{v}_j^*\right| + \frac{1}{1-\epsilon}\left|\frac{1}{B_{jj}}(\langle \tilde{\mathbf{u}}, \mathbf{u}^* \rangle B_{jj} - C_{jj})\mathbf{v}_j^*\right| \\
&\leq \frac{1}{(1-\epsilon)^2}\left(|\langle \tilde{\mathbf{u}}, \mathbf{u}^* \rangle| + |\langle \tilde{\mathbf{u}}, \mathbf{u}^* \rangle(1+\epsilon) - (\langle \tilde{\mathbf{u}}, \mathbf{u}^* \rangle - \epsilon)|\right)\frac{\mu}{\sqrt{n}} \leq \frac{1+2\epsilon}{(1-\epsilon)^2}\frac{\mu}{\sqrt{n}}
\end{aligned}$$

For $\epsilon < 1/6$, the result now holds. □

We note that whereas Lemma 8.4 is proved for fixed vectors, we seem to have inappropriately applied it to $\tilde{\mathbf{u}}$ in the proof of Lemma 8.5 which is not a fixed vector as it depends on the randomness used in sampling the entries of the matrix revealed to the algorithm. However notice that the AM-MC procedure in Algorithm 10 uses fresh samples Ω_t and Ω_{T+t} for each iteration. This ensures that $\tilde{\mathbf{u}}$ does behave like a fixed vector with respect to Lemma 8.4.

Lemma 8.6. *For any $\epsilon > 0$, if $p > \frac{80\mu^2 \log(m+n)}{\epsilon^2 \min\{m,n\}}$ and $\mathbf{u}^t$ is 2μ-incoherent, the next iterate $\mathbf{v}^{t+1}$ satisfies*

$$1 - \left\langle \frac{\mathbf{v}^{t+1}}{\left\|\mathbf{v}^{t+1}\right\|_2}, \mathbf{v}^* \right\rangle^2 \leq \frac{\epsilon}{(1-\epsilon)^3}\left(1 - \left\langle \frac{\mathbf{u}^{t}}{\left\|\mathbf{u}^{t}\right\|_2}, \mathbf{u}^* \right\rangle^2\right)$$

Similarly, for any 2μ-incoherent iterate $\mathbf{v}^{t+1}$, the next iterate satisfies

$$1 - \left\langle \frac{\mathbf{u}^{t+1}}{\left\|\mathbf{u}^{t+1}\right\|_2}, \mathbf{u}^* \right\rangle^2 \leq \frac{\epsilon}{(1-\epsilon)^3}\left(1 - \left\langle \frac{\mathbf{v}^{t+1}}{\left\|\mathbf{v}^{t+1}\right\|_2}, \mathbf{v}^* \right\rangle^2\right).$$

Proof. Using the modified form of the update for $\mathbf{u}^{t+1}$ (8.2), we get, for any unit vector $\mathbf{v}_\perp$ such that $\langle \mathbf{v}_\perp, \mathbf{v}^* \rangle = 0$,

$$\begin{aligned}
\left\|\mathbf{u}^t\right\|_2 \cdot \left\langle \mathbf{v}^{t+1}, \mathbf{v}_\perp \right\rangle &= \left\langle \mathbf{v}_\perp, B^{-1}(\langle \tilde{\mathbf{u}}, \mathbf{u}^* \rangle B - C)\mathbf{v}^* \right\rangle \\
&\leq \left\|B^{-1}\right\|_2 \left\|(\langle \tilde{\mathbf{u}}, \mathbf{u}^* \rangle B - C)\mathbf{v}^*\right\|_2 \\
&\leq \frac{1}{1-\epsilon}\left\|(\langle \tilde{\mathbf{u}}, \mathbf{u}^* \rangle B - C)\mathbf{v}^*\right\|_2,
\end{aligned}$$

where the last step follows from an application of Lemma 8.4. To bound the other term let $Z_{ij} = \frac{1}{p}\delta_{ij}(\langle \tilde{\mathbf{u}}, \mathbf{u}^* \rangle (\tilde{\mathbf{u}}_i)^2 - \tilde{\mathbf{u}}_i\mathbf{u}^*_i)\mathbf{v}^*_j\mathbf{e}_j \in \mathbb{R}^n$. Clearly $\sum_{i=1}^m \sum_{j=1}^n Z_{ij} = (\langle \tilde{\mathbf{u}}, \mathbf{u}^* \rangle B - C)\mathbf{v}^*$. Note that due to fresh samples being used by Algorithm 10 at every step, the vector $\tilde{\mathbf{u}}$ appears as a constant vector to the random variables δ_{ij}. Given this, note that

$$\begin{aligned}
\mathbb{E}\left[\sum_{i=1}^m Z_{ij}\right] &= \sum_{i=1}^m (\langle \tilde{\mathbf{u}}, \mathbf{u}^* \rangle (\tilde{\mathbf{u}}_i)^2 - \tilde{\mathbf{u}}_i\mathbf{u}^*_i)\mathbf{v}^*_j\mathbf{e}_j \\
&= \langle \tilde{\mathbf{u}}, \mathbf{u}^* \rangle \sum_{i=1}^m (\tilde{\mathbf{u}}_i)^2\mathbf{v}^*_j\mathbf{e}_j - \sum_{i=1}^m \tilde{\mathbf{u}}_i\mathbf{u}^*_i\mathbf{v}^*_j\mathbf{e}_j \\
&= (\langle \tilde{\mathbf{u}}, \mathbf{u}^* \rangle \|\tilde{\mathbf{u}}\|_2^2 - \langle \tilde{\mathbf{u}}, \mathbf{u}^* \rangle)\mathbf{v}^*_j\mathbf{e}_j = \mathbf{0},
\end{aligned}$$

since $\|\tilde{\mathbf{u}}\|_2 = 1$. Thus $\mathbb{E}\left[\sum_{ij} Z_{ij}\right] = \mathbf{0}$ as well. Now, we have $\max_i(\tilde{\mathbf{u}}_i)^2 = \frac{1}{\|\mathbf{u}^t\|_2^2} \cdot \max_i(\mathbf{u}^t_i)^2 \leq \frac{4\mu^2}{m\|\mathbf{u}^t\|_2^2} \leq \frac{\mu^2}{m(1-\epsilon)}$ since $\left\|\mathbf{u}^t - \mathbf{u}^*\right\|_2 \leq \epsilon$. This allows us to bound

$$\begin{aligned}
\left|\mathbb{E}\left[\sum_{ij} Z_{ij}^\top Z_{ij}\right]\right| &= \frac{1}{p}\sum_{i=1}^m \sum_{j=1}^n (\langle \tilde{\mathbf{u}}, \mathbf{u}^* \rangle (\tilde{\mathbf{u}}_i)^2 - \tilde{\mathbf{u}}_i\mathbf{u}^*_i)^2(\mathbf{v}^*_j)^2 \\
&= \frac{1}{p}\sum_{i=1}^m (\tilde{\mathbf{u}}_i)^2(\langle \tilde{\mathbf{u}}, \mathbf{u}^* \rangle \tilde{\mathbf{u}}_i - \mathbf{u}^*_i)^2 \\
&\leq \frac{\mu^2}{pm(1-\epsilon)}\sum_{i=1}^m \langle \tilde{\mathbf{u}}, \mathbf{u}^* \rangle^2 (\tilde{\mathbf{u}}_i)^2 + (\mathbf{u}^*_i)^2 - 2\langle \tilde{\mathbf{u}}, \mathbf{u}^* \rangle \tilde{\mathbf{u}}_i\mathbf{u}^*_i \\
&\leq \frac{8\mu^2}{pm}(1 - \langle \tilde{\mathbf{u}}, \mathbf{u}^* \rangle^2),
\end{aligned}$$

where we set $\epsilon = 0.5$. In the same way we can show $\left\| \mathbb{E}\left[\sum_{ij} Z_{ij}^\top Z_{ij}\right] \right\|_2 \leq \frac{8\mu^2}{pm}(1 - \langle \widetilde{\mathbf{u}}, \mathbf{u}^* \rangle^2)$ as well. Using a similar argument we can show $\|Z_{ij}\|_2 \leq \frac{4\mu^2}{p\sqrt{mn}} \sqrt{1 - \langle \widetilde{\mathbf{u}}, \mathbf{u}^* \rangle^2}$. Applying the Bernstein inequality now tells us, that for any $\epsilon > 0$, if $p > \frac{80\mu^2 \log(m+n)}{\epsilon^2 \min\{m,n\}}$, then with probability at least $1 - 1/n^{10}$, we have

$$\|(\langle \widetilde{\mathbf{u}}, \mathbf{u}^* \rangle B - C)\mathbf{v}^*\|_2 \leq \epsilon \cdot \sqrt{1 - \langle \widetilde{\mathbf{u}}, \mathbf{u}^* \rangle^2}.$$

Since $\|\mathbf{u}^t\|_2 \geq 1 - \epsilon$ is guaranteed by the initialization step, we now get,

$$\left\langle \mathbf{v}^{t+1}, \mathbf{v}_\perp \right\rangle \leq \frac{\epsilon}{(1-\epsilon)^2} \sqrt{1 - \langle \widetilde{\mathbf{u}}, \mathbf{u}^* \rangle^2}.$$

If $\mathbf{v}_\perp^{t+1}$ and $\mathbf{v}_\parallel^{t+1}$ be the components of $\mathbf{v}^{t+1}$ perpendicular and parallel to $\mathbf{v}^*$. Then the above guarantees that $\left\|\mathbf{v}^\perp\right\|_2 \leq c \cdot \sqrt{1 - \langle \widetilde{\mathbf{u}}, \mathbf{u}^* \rangle^2}$. This gives us, upon applying the Pythagoras theorem,

$$\left\|\mathbf{v}^{t+1}\right\|_2^2 = \left\|\mathbf{v}_\perp^{t+1}\right\|_2^2 + \left\|\mathbf{v}_\parallel^{t+1}\right\|_2^2 \leq \frac{\epsilon}{(1-\epsilon)^2}\left(1 - \langle \widetilde{\mathbf{u}}, \mathbf{u}^* \rangle^2\right) + \left\|\mathbf{v}_\parallel^{t+1}\right\|_2^2$$

Since $\left\|\mathbf{v}_\parallel^{t+1}\right\|_2 = \langle \mathbf{v}^{t+1}, \mathbf{v}^* \rangle$ and $\|\mathbf{v}^{t+1}\|_2 \geq 1 - \epsilon$ as $\|\mathbf{v}^{t+1} - \mathbf{v}^*\| \leq \epsilon$ due to the initialization, rearranging the terms gives us the result. □

Using these results, it is easy to establish the main theorem.

Theorem 8.7. *Let $A = \mathbf{u}^*(\mathbf{v}^*)^\top$ be a unit rank matrix where $\mathbf{u}^* \in \mathbb{R}^m$ and $\mathbf{v}^* \in \mathbb{R}^n$ are two μ-incoherent unit vectors. Let the matrix be observed at a set of indices $\Omega \subseteq [m] \times [n]$ where each index is observed with probability p. Then if $p \geq C \cdot \frac{\mu^2 \log(m+n)}{\epsilon^2 \min\{m,n\}}$ for some large enough constant C, then with probability at least $1 - 1/\min\{m,n\}^{10}$, AM-MC generates iterates which are 2μ-incoherent. Moreover, within $\mathcal{O}\left(\log \frac{1}{\epsilon}\right)$ iterations, AM-MC also ensures that $\left\|\frac{\mathbf{u}^t}{\|\mathbf{u}^t\|_2} - \mathbf{u}^*\right\|_2 \leq \epsilon$ and $\left\|\frac{\mathbf{v}^t}{\|\mathbf{v}^t\|_2} - \mathbf{v}^*\right\|_2 \leq \epsilon$.*

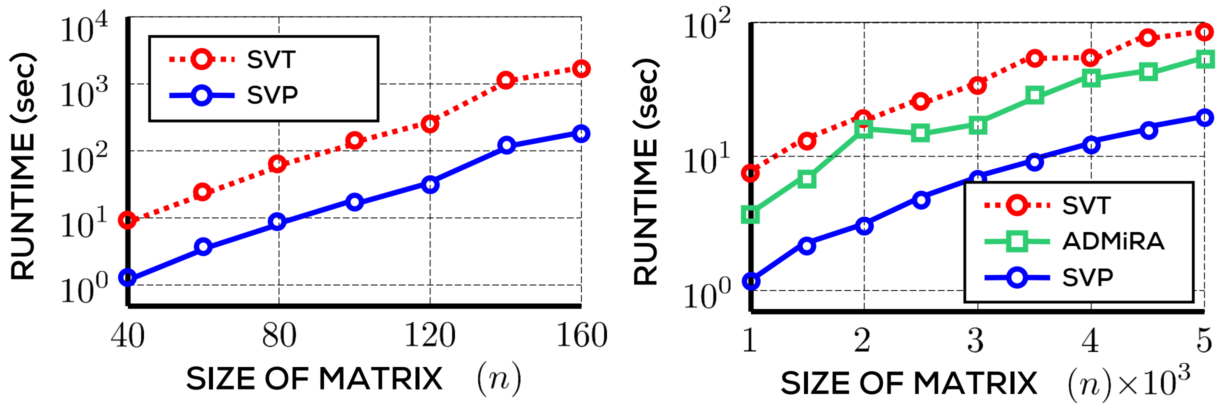
8.8 Other Popular Techniques for Matrix Recovery

As has been the case with the other problems we have studied so far, the first approaches to solving the ARM and LRMC problems were relaxation based approaches [Candès and Recht, 2009, Candès and Tao, 2009, Recht et al., 2010, Recht, 2011]. These approaches relax the non-convex rank objective in the ((ARM)) formulation using the (convex) *nuclear norm*

$$\begin{array}{ll} \min & \|X\|_* \\ \text{s.t.} & \mathcal{A}(X) = \mathbf{y}, \end{array}$$

where the nuclear norm of a matrix $\|X\|_*$ is the sum of all singular values of the matrix X. The nuclear norm is known to provide the tightest convex envelope of the rank function, just as the ℓ_1 norm provides a relaxation to the sparsity norm $\|\cdot\|_0$ [Recht et al., 2010]. Similar to sparse recovery, under matrix-RIP settings, these relaxations can be shown to offer exact recovery [Recht et al., 2010, Hastie et al., 2016].

Also similar to sparse recovery, there exist pursuit-style techniques for matrix recovery, most notable among them being the ADMiRA method [Lee and Bresler, 2010] that extends the orthogonal matching pursuit approach to the matrix recovery setting. However, this method can be a bit sluggish when recovering matrices with slightly large rank since it discovers a matrix with larger and larger rank incrementally.



(a) Matrix Recovery (b) Matrix Completion

Figure 8.2: An empirical comparison of run-times offered by the SVT, ADMiRA and SVP methods on synthetic matrix recovery and matrix completion problems with varying matrix sizes. The SVT method due to Cai et al. [2010] is an efficient implementation of the nuclear norm relaxation technique. ADMiRA is a pursuit-style method due to Lee and Bresler [2010]. For the ARM task in Figure 8.2a, the rank of the true matrix was set to $r = 5$ whereas it was set to $r = 2$ for the LRMC task in Figure 8.2b. SVT is clearly the most scalable of the methods in both cases whereas the relaxation-based SVT technique does not scale very well to large matrices. Note however, that for the LRMC problem, AM-MC (not shown in the figure) outperforms even SVP. Figures adapted from [Meka et al., 2008].

Before concluding, we present the reader with an empirical performance of these various methods. Figure 8.2 provides a comparison of these methods on synthetic matrix recovery and matrix completion problems with increasing dimensionality of the (low-rank) matrix being recovered. The graphs indicate that non-convex optimization methods such as IHT are far more scalable, often by an order of magnitude, than relaxation-based methods.

8.9 Exercises

Exercise 8.1. *Show that low-rank matrix recovery is NP-hard.*
Hint: Take the sparse recovery problem in ((SP-REG)) *and reduce it to the reformulation* ((ARM-2)) *of the matrix recovery problem.*

Exercise 8.2. *Show that the matrix RIP constant is monotonic in its order i.e., if a linear map $\mathcal{A}$ satisfies matrix RIP of order r with constant δ_r, then it also satisfies matrix RIP for all orders $r' \leq r$ with $\delta_{r'} \leq \delta_r$.*

8.10 Bibliographic Notes

There are a lot of aspects of low-rank matrix recovery that we have not covered in our discussion. Here we briefly point out some of these.

Similar to the sparse regression setting, the problem of ill-conditioned problems requires special care in the matrix recovery setting as well. For the general ARM problem, the work of Jain et al. [2014] does this by first proposing appropriate versions of RSC-RSS properties (see Definition 7.4) for the matrix case, and the suitably modifying SVP-style techniques to function even in high condition number settings. The final result is similar to the one for sparse recovery (see Theorem 7.4) wherein a more "relaxed" projection step is required by using a rank $q > r$ while executing the SVP algorithm.

It turns out to be challenging to prove convergence results for SVP for the LRMC problem. This is primarily due to the difficulty in establishing the matrix RIP property for the affine transformation used in the problem. The affine map simply selects a few elements of the matrix and reproduces them which makes establishing RIP properties harder in this setting. Specifically, even though the initialization step can be shown to yield a matrix that satisfies matrix RIP [Jain et al., 2010], if the underlying matrix is low-rank and incoherent, it becomes challenging to show that RIP-ness is maintained across iterates. Jain and Netrapalli [2015] overcome this by executing the SVP algorithm in a *stage-wise* fashion which resembles ADMiRA-like pursuit approaches.

Several works have furthered the alternating minimization approach itself by reducing its sample complexity [Hardt, 2014], giving recovery guarantees independent of the condition number of the problem [Hardt and Wootters, 2014, Jain and Netrapalli, 2015], designing universal sampling schemes for recovery [Bhojanapalli and Jain, 2014], as well as tackling settings where some of the revealed entries of the matrix may be corrupted [Chen et al., 2016, Cherapanamjeri et al., 2017].

Another interesting line of work for matrix completion is that of [Ge et al., 2016, Sun and Lu, 2015] which shows that under certain regularization assumptions, the matrix completion problem does not have any non-optimal stationary points once one gets close-enough to the global minimum. Thus, one can use any method for convex optimization such as alternating minimization, gradient descent, stochastic gradient descent, and its variants we studied in § 6, once one is close enough to the global minimum.