Robust Matrix Completion and Corrupted Columns

Yudong Chen YDCHEN@UTEXAS.EDU
Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712 USA

Huan Xu MPEXUH@NUS.EDU.SG
Department of Mechanical Engineering, National University of Singapore, Singapore 117575

Constantine Caramanis CARAMANIS@MAIL.UTEXAS.EDU
the Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712 USA

Sujay Sanghavi SANGHAVI@MAIL.UTEXAS.EDU
Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712 USA

Abstract

This paper considers the problem of matrix completion, when some number of the columns are arbitrarily corrupted. It is well-known that standard algorithms for matrix completion can return arbitrarily poor results, if even a single column is corrupted. What can be done if a large number, or even a constant fraction of columns are corrupted? In this paper, we study this very problem, and develop an robust and efficient algorithm for its solution. One direct application comes from robust collaborative filtering. Here, some number of users are so-called manipulators, and try to skew the predictions of the algorithm. Significantly, our results hold *without any assumptions on the observed entries of the manipulated columns*.

1. Introduction

Recent work in low-rank matrix completion (Candès & Recht, 2009; Gross, 2009; Keshavan et al., 2009; Negahban & Wainwright, 2010) has demonstrated the following remarkable fact: Given a $p \times n$ matrix of rank r satisfying some technical assumptions (namely, incoherence – we discuss this in detail below), if its entries are sampled uniformly at random, then with high probability, the solution to a convex and in particular tractable optimization problem yields exact reconstruction of the matrix, when only $O((n+p)r\log^2(n+p))$ entries are sampled.

Appearing in *Proceedings of the 28^{th} International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

Yet as our simulations demonstrate, if even a single column of this matrix is corrupted, the output of these algorithms can be arbitrarily skewed from the true matrix. Partial observation makes *a priori* identification of corrupted columns vs good columns, a challenging task.

We approaches this task by regarding the corruption process as the addition of a column-sparse matrix to a low rank matrix. The mathematical problem this paper addresses is as follows. Suppose we are given a *partially observed* matrix M, and we know that the full matrix can be decomposed as

$$M = L_0 + C_0,$$

where L_0 is low-rank and C_0 has only a few non-zero columns. Here both components may have arbitrary magnitude; the rank and column/row space of L_0 as well as the number and positions of non-zero columns of C_0 are unknown. Can we efficiently recover the matrix L_0 on the non-corrupted columns, and also identify the non-zero columns of C_0? And, how does the number of corrupted columns impact the number of observations needed?

We provide an affirmative answer to the first question, and provide finite sample performance bounds that move towards answering the second. We give a convex optimization formulation, and sufficient conditions for when this optimization problem yields exact recovery of L_0, and identification of the corrupted columns. In particular, our results imply the following: if we observe only a vanishing fraction of entries, our convex optimization-based algorithm recovers L_0 exactly even in the face of an increasing number of corrupted columns. If a constant fraction of the columns are corrupted, then our algorithm succeeds in identifying them and recovers L_0 exactly, but now requires a constant fraction of observed entries.

Motivating Applications

A primary motivation for our investigation is Robust Collaborative Filtering. In online commerce and advertisement, companies collect user rankings for products and would like to predict user preferences based these incomplete rankings. Most popular in the news is the so-called Netflix problem, but such recommender systems are of increasing popularity and importance in online commerce. In many of the settings mentioned (again, most well-known in this category is the Netflix problem) this collaborative filtering problem is usually cast as a matrix completion problem, where one tries to recover a low-rank matrix L_0 from its partially observed entries. However, the quality of prediction may be seriously hampered by manipulators – potentially malicious users, who calibrate (possibly in a coordinated way) their rankings *and* the entries they choose to rank in an attempt to skew predictions (Van Roy & Yan, 2010). Under the matrix completion framework, some of the columns of the matrix M are provided by manipulators and thus are corrupted and totally unreliable; these corrupted columns correspond to the column-sparse matrix C_0. Therefore, in order to perform collaborative filtering with robustness to manipulation, we need to identify the non-zero columns of C_0 and at the same time recover L_0, given only a set of incomplete entries.

Another motivation is robust Principal Component Analysis (PCA) with partially observed data. In the robust PCA problem (Xu et al., 2010a;b;c) one is given a data matrix, of which most of the columns correspond to authentic data points and lie in a low-dimensional space – the space of principal components, and the remaining columns are outliers; the goal is to negate the effect of outliers and recover the principal components. In many situations such as medical research (see e.g. (Cesa-Bianchi et al., 2010)), the data matrix is only partially observed, and the question is if we can still perform robust PCA in the presence of missing data entries. Viewing the outliers as columns of C_0 and the authentic data as columns of L_0, the partially observed robust PCA problem can also be cast into our framework. We note that our setting differs significantly from the low-rank-plus-sparse setup in (Candès et al., 2009; Chandrasekaran et al., 2009). We further illustrate this numerically, in Section 5. In Section 3.2, we elaborate on our connections with prior work and the innovations of this paper.

2. Problem Setup

Suppose there is a $p \times n$ data matrix M; among the n columns, a fraction $1-\gamma$ of them span a r-dimensional subspace of $\mathbb{R}^p$, and the remaining γn columns are arbitrarily corrupted. One is given only partial obervation of the matrix M, and the goal is to infer the true subspace of the non-corrupted columns and the identities of the corrupted ones. Notice that neither the true subspace nor its dimension r is known, and no restriction is imposed on the corrupted columns except that the total number of them is controlled – they need not follow any probablistic distributions, and they may be chosen by some adversary who aims to skew one's inference of the non-corrupted columns.

Under the above setup, it is clear that the data matrix M can be decomposed as

$$M = L_0 + C_0.$$

Here L_0 is the matrix corresponding to the non-corrupted columns; thus $\text{rank}(L_0) = r$ and at most $(1-\gamma)n$ of the columns of L_0 are non-zero. C_0 is the matrix corresponding to the corrupted columns; thus at most γn of the columns of C_0 are non-zero. Only some of the entries of M are observed. Let $\Omega \subseteq [p] \times [n]$ be the set of indices of the observed entries, and $\mathcal{P}_\Omega$ be the orthogonal projection onto the linear subspace of matrices supported on Ω, i.e.,

$$\mathcal{P}_\Omega(X) = \begin{cases} X_{ij}, & (i,j) \in \Omega, \\ 0, & (i,j) \notin \Omega. \end{cases}$$

With this notation, our goal is to exactly recover the column space of L_0 and the locations of the non-zero columns of C_0, given $\mathcal{P}_\Omega(M)$.

2.1. Assumptions

In general, it is not always possible to meet our objective of completing a low-rank matrix in the presence of corrupted columns. Indeed, under some circumstances, there are identifiability issues which make the problem ill-posed. For example, if one row or column of L_0 is completely unobserved, there is no hope of recovering that row or column. Likewise, if L_0 has only one non-zero column, it is also impossible to distinguish L_0 from C_0. Finally, if L_0 has only one non-zero row, recovering L_0 is infeasible unless that particular row is fully observed. To avoid such meaningless situations, we will impose that L_0 satisfy the now standard incoherence conditions (Candès & Recht, 2009; Xu et al., 2010c) and observed entries of L_0 are sampled uniformly at random. We note again that we make no assumptions on how the entries of C_0 are sampled, and moreover these entries could be adversarially chosen.

INCOHERENCE CONDITIONS

Suppose the Singular Value Decomposition (SVD) of L_0 is $L_0 = U_0 \Sigma_0 V_0^\top$. Let e_i be the ith standard basis. We assume that the matrix L_0 satisfies the following two incoherence conditions, with parameter μ_0:

$$\begin{aligned} \max_i \left\| U_0^\top e_i \right\|^2 &\leq \mu_0 \frac{r}{p}, \\ \max_j \left\| V_0^\top e_j \right\|^2 &\leq \mu_0 \frac{r}{(1-\gamma)n}. \end{aligned}$$

Given a small incoherence parameter μ_0, the condition asserts that the left singular vectors of L_0 are spread out. Without such a condition, matrix completion does not make sense, since it would be possible for the matrix L_0 to also be row-sparse — one cannot hope to recover a row-sparse matrix with sparse observations, even without outliers. Consequently, this is a standard assumption made in the matrix completion literature (Candès & Recht, 2009; Gross, 2009; Keshavan et al., 2009), and μ_0 is likely to be small for many reasonable models (Candès & Recht, 2009).

The second condition asserts that the right singular vectors of L_0 are incoherent, and it essentially enforces the condition that the information about the column space of L_0 is spread out among the columns. This condition is important in the face of corrupted columns. If, for instance, a column of L_0 were not in the span of all the other columns, one could not hope to recover it or distinguish it from one of the corrupted columns. This condition is standard in the robust PCA literature, and most practical problems have a very small parameter μ_0.

For the corrupted columns, we make only one assumption: they are indeed corrupted. That is, we assume only the following. Suppose an oracle were to provide the true column space, U_0, of the low-rank matrix, L_0. There would be no way to complete the unobserved entries of any of the columns of C_0, so that it lies in the column space of L_0. If this does not hold, then there is no reasonable way to distinguish a corrupted column from an authentic column. Moreover, such entries will not affect the recovery of the unobserved entries in the authentic columns. In terms of the collaborative filtering application, this is akin to saying that we will only call a user a "manipulator" if the corresponding entries indeed would manipulate the entries of the authentic users. In fact, this is better viewed as a *definition* of corruption rather than an assumption. Other than this identifiability requirement, we make no assumptions whatsoever on the corrupted columns. The incoherence assumptions are imposed on the column and row spaces of L_0, not on M, as are the sampling assumptions, and thus the corrupted columns are not restricted in any way by these. One consequence of this is that it is fundamentally impossible to recover the complete corrupted columns, but we are able to recover their identities.

SAMPLING MODEL

Let $\mathcal{I}_0 \subset [n]$ be the set of indices of the corrupted columns. Let $\tilde{\Omega} \subseteq [p] \times \mathcal{I}_0^c$ be the set of indices of observed entries on the non-corrupted columns (i.e. the nonzero columns of L_0). We assume that $\tilde{\Omega}$ is sampled uniformly random from all size-m subsets of $[p] \times \mathcal{I}_0^c$ (this is sometimes called sampling without replacement); so m is the number of observed entries on the non-corrupted columns. Note that *no* assumption whatsoever is imposed on the observed entries on the corrupted columns; the adversary may choose to fill in all entries on columns in $\mathcal{I}_0$ or just a fraction of them, and the locations of these observed entries may be chosen randomly or depending on L_0. On the other hand, as we do not aim at (and there is no hope of) recovering the unobserved entries of C_0, we can assume without loss of generality that all the unobserved entries of C_0 are zero, i.e., $\mathcal{P}_\Omega(C_0) = C_0$.

2.2. Notation and Preliminaries

We provide here a brief summary of the notation used in the paper. We will abuse notation by letting $\tilde{\Omega}$ and $\tilde{\Omega}^c$ represent both sets of matrix entries and the linear space of matrices supported on these entries; similarly $\mathcal{I}_0$ and $\mathcal{I}_0^c$ denote both the set of column indices and the linear space of matrices supported on these columns. For a linear subspace $\mathcal{S}$, we let $\mathcal{P}_\mathcal{S}$ denote the orthogonal projection onto $\mathcal{S}$. The SVD of L_0 is $U_0 \Sigma_0 V_0^\top$. Let $\mathcal{P}_{U_0}$ be the projection of each column of a matrix onto the column space, and $\mathcal{P}_{V_0}$ the projection onto the row space, as well as the range of those projections. The subspace $\mathcal{T}_0$ is defined as the span of matrices with the same column or row space as L_0, and the projection is given by $\mathcal{P}_{\mathcal{T}_0}$. The complementary operator $\mathcal{P}_{\mathcal{T}_0^\perp}$ is defined as usual. For a vector x, x_i is its ith entry. For a matrix A, A_i is its i^{th} column and A_{ij} is its $(i,j)^{th}$ entry. Six matrix norms are used: $\|A\|_*$ is the nuclear norm (the sum of singular values), $\|A\|$ is the spectral/operator norm (the largest singular values), $\|A\|_\infty$ is the matrix infinity norm (the largest absolute value of the entries), $\|A\|_{1,2}$ is the sum of ℓ_2 norms of the columns of A, $\|A\|_{\infty,2}$ is the largest ℓ_2 norm of the columns of A, and finally $\|A\|_F$ is the Frobenius norm. For the non-corrupted columns: Let $n_1 = n - |\mathcal{I}_0^c| = (1-\gamma)n$ be the number of non-corrupted columns. To facilitate operations on the $\mathcal{I}_0^c$ portion of a matrix, we define $\mathcal{R}$ to be the operator that extracts the columns in $\mathcal{I}_0^c$, and $\mathcal{R}^{-1}$ the operator that maps back to the full-columns space, inserting all-zero columns into the entries corresponding to $\mathcal{I}_0$. Letting $\tilde{V}_0 = \mathcal{R}(V_0^\top)^\top$, we define analogs of our previous definitions, $\mathcal{P}_{\tilde{V}_0}$, $\mathcal{P}_{\tilde{\mathcal{T}}_0}$ and $\mathcal{P}_{\tilde{\mathcal{T}}_0^\perp}$, where $\tilde{\mathcal{T}}_0$ is defined by U_0 and $\tilde{V}_0$.

3. Main Results and Consequences

Surprisingly, we can simultaneously recover L_0, the non-corrupted columns, and identify $\mathcal{I}_0$, the position of the corrupted columns, based on $\mathcal{P}_\Omega(M)$, a set of incomplete entries, as long as the the number of corrupted columns and unobserved entries are controlled. Moreover, this can be achieved efficiently by solving a *tractable* convex program. Our algorithm is as follows.

We say our algorithm succeeds if we always have $\mathcal{P}_{\mathcal{I}_0^c}(L') = L_0$, $\mathcal{P}_{U_0}(L') = L'$, and $\mathcal{I}' = \mathcal{I}_0$. We recall

Algorithm 1 Manipulator Pursuit

Input $\mathcal{P}_\Omega(M)$, Ω, λ.
Solve for optimum (L^*, C^*):

$$\begin{aligned} \text{minimize}_{L,C} \quad & \|L\|_* + \lambda \|C\|_{1,2} \\ \text{subject to} \quad & \mathcal{P}_\Omega(L + C) = \mathcal{P}_\Omega(M). \end{aligned} \tag{1}$$

Set $\mathcal{I}' = \{j : C^*_{ij} \neq 0 \text{ for some } i\}$, $L' = \mathcal{P}_{\mathcal{I}'^c}(L^*)$.
Output $L', \mathcal{I}'$.

our single restriction on the corrupted columns: they are indeed corrupted, in that they cannot be completed so as to lie in the column space of the true matrix L_0 — failing this, asking for $\mathcal{I}_0$ to be recovered does not make sense, nor is it even clear why such a column should be called "corrupted."

3.1. Main Theorems

Our main theorem states that under some natural conditions, our algorithm exactly recovers the non-corrupted columns and the identities of the corrupted columns with high probability. Here and in what that follows, by *with high probability*, we mean with probability at least $1 - cn^{-5}$ for some constant $c > 0$. Recall that ρ is the fraction of observed entries on the non-corrupted columns and γ is the fraction of corrupted columns.

Theorem 1. *Suppose $n_1 \geq p \geq 32$ and $r \leq \bar{r}$, $\gamma \leq \bar{\gamma}$, $\rho \geq \underline{\rho}$. If $(\bar{r},\ \bar{\gamma},\ \underline{\rho})$ satisfies*

$$\underline{\rho} \geq \eta_1 \frac{\mu_0^2 \bar{r}^2 \log^3(4n_1)}{p} \tag{2}$$

and

$$\frac{\bar{\gamma}}{1 - \bar{\gamma}} \leq \eta_2 \frac{\underline{\rho}^2}{\left(1 + \frac{\mu_0 \bar{r}}{\underline{\rho}\sqrt{p}}\right)^2 \mu_0^3 \bar{r}^3 \log^6(4n_1)}, \tag{3}$$

where η_1 and η_2 are absolute constants, then with high probability Algorithm 1 with $\lambda = \frac{1}{48}\sqrt{\frac{\underline{\rho}}{\bar{\gamma}\bar{r}\mu_0 n \log^2(4n_1)}}$ strictly succeeds.

Remark. *Notice that to choose λ, one does not need to know the exact values of ρ, γ, and r, but rather bounds on them.*

We give three corollaries to illustrate the consequence of Theorem 1.

Corollary 1. *If $r \leq \eta_1 \frac{1}{\mu_0}$, $\rho \geq \eta_2 \frac{\log(4n_1)}{p^{1/4}}$, $\gamma \leq \eta_3 \frac{1}{\sqrt{p}}$, then Algorithm 1 with $\lambda = \sqrt{\frac{p^{1/4}}{n}}$ succeeds with high probability..*

Remark. *Notice that the choice λ is universal and does not depende on any unknown quantity. In the case of $p = \Theta(n_1)$, we can recover the non-corrupted columns with a vanishing fraction of entries observed and a growing number of corrupted columns.*

Corollary 2. *If $\rho \geq 0.1$ and $r \leq \bar{r} \leq \eta_1 \frac{\sqrt{p}}{\mu_0 \log^{3/2}(4n_1)}$, then Algorithm 1 with $\lambda = \frac{\mu_0 \bar{r} \log^2(4n_1)}{\sqrt{n}}$ succeeds with high probability if*

$$\gamma \leq \eta_2 \frac{1}{\mu_0^3 \bar{r}^3 \log^6(4n_1)}$$

Remark. *With a constant fraction of entries observed, the fraction of corrupted columns can be as large as one over a poly-logarithm factor. If $\rho = 1$, we partially recovers the result in (Xu et al., 2010c).*

Corollary 3. *If $\gamma = 0$, $r \leq \bar{r}$, and m satisfies*

$$m \geq \eta_1 \mu_0^2 \bar{r}^2 n \log^2(4n)$$

then w.h.p. Algorithm 1 with $\lambda = n$ has a unique solution $(L_0,\ 0)$.

Remark. *This recovers the matrix completion result in (Candès & Tao, 2010; Recht, 2009; Gross, 2009).*

3.2. Connections to Prior Work and Innovation

In this section we briefly discuss the relationship to previous work. Matrix completion – to recover a low rank matrix from a small number of its entries – using convex optimization (Candès & Tao, 2010; Gross, 2009; Recht, 2009) is now standard. This paper significantly extends this line of work and shows that if some columns are completely corrupted , even in a malicious way, one can still recover the non-corrupted columns and identify the corrupted columns.

Our main methodology is to develop and analyze a convex relaxation of a natural yet intractable formulation of decomposing the observation into a low rank matrix and a column-sparse matrix. From a high level, this follows a similar line as works in support recovery (Candès et al., 2006), low-rank matrix recovery and matrix completion papers, (Recht et al., 2010; Candès & Tao, 2010; Gross, 2009; Recht, 2009)) and matrix decomposition (Candès et al., 2009; Chandrasekaran et al., 2009). However, because of the obvious difference in the setup since we allow entire column to be corrupted, previous approaches for matrix completion or matrix completion with sparse corruption fails. Moreover, this setup brings novel challenges to mathematical analysis. First, the optimal solution of the convex problem is in general neither the original low rank matrix L_0, which is supported only on the non-corrupted columns, nor the exact matrix C_0. It is, however, in an appropriately defined equivalence class of the original L_0 and C_0. This is in sharp contrast to all aforementioned works where the intended outcome is known *a priori*. This critical difference

requires the use of an "Oracle Problem" in order to identify an element of the equivalence class for which one can then certify optimality. Second, this problem essentially disentangles three structures: low rank, column-sparse, and element-sparse. This requires to develop several new concentration results in the $\|\cdot\|_{\infty,2}$ norm.

4. Skeleton-Proof of Main Theorem

In this section we provide a proof-skeleton of our main theorem. The full proof details are given in (Chen et al., 2010). The main roadmap to proving a convex optimization problem recovers a desired solution, is to demonstrate that with high probability, one can find a dual certificate of optimality of the desired solution. This basic recipe underlies many of the proofs in sparse recovery and low-rank recovery (Candès et al., 2009; Candès & Recht, 2009; Chandrasekaran et al., 2009). A central roadblock to this approach is that unless the adversary's corrupted columns happen to be perfectly perpendicular to the column space of the true low-rank matrix, *the convex optimization problem given will not precisely recover* L_0. The reason is simple: if the corrupted columns have a non-perpendicular component, then some part of that will be put into the L matrix the optimization recovers. Algorithmically, this matter is irrelevant: as long as the corrupted columns are identified, and the recovered L matches the desired L_0 on the non-corrupted columns, our objective is met, and the problem is solved. The analysis, however, is significantly complicated, since because we do not recover L_0 exactly, we no longer explicitly know for what to write a certificate of optimality.

Step 1. For the algorithm to succeed, it is sufficient for the recovered pair (L^*, C^*) to have the right column space and correct non-corrupted columns for L^*, and the right column support for C^*. To identify such a solution, we consider the following Oracle Problem; here Γ denotes the space of matrices supported on the set of all entries in the non-corrupted columns plus the observed entries in the corrupted columns.

$$\begin{aligned} \text{minimize}_{L,C} \quad & \|L\|_* + \lambda \|C\|_{1,2} \\ \text{subject to} \quad & \mathcal{P}_\Gamma(L+C) = \mathcal{P}_\Gamma(M_0) \\ & \mathcal{P}_{U_0}(L) = L \\ & \mathcal{P}_{\mathcal{I}_0}(C) = C. \end{aligned}$$

The Oracle Problem is feasible, since the true pair (L_0, C_0) is feasible. Let $(\hat{L}, \hat{C})$ denote the solution to the Oracle Problem. We must identify conditions that a dual certificate must satisfy to guarantee that $(\hat{L}, \hat{C})$ is an optimal solution to Algorithm 1, and that any optimal solution to Algorithm 1 must also have the correct column space and column support.

Step 2. To state these conditions, we need some definitions.

$$\begin{aligned} \hat{U}\hat{\Sigma}\hat{V}^\top &:= \text{the singular value decomposition of } \hat{L} \\ \hat{\mathcal{T}} &= \Big\{ Z \in \mathbb{R}^{p\times n} | Z = \hat{U}X^\top + Y\hat{V}^\top, \\ &\qquad \forall X \in \mathbb{R}^{p\times r}, Y \in \mathbb{R}^{n\times r} \Big\} \\ \hat{\mathcal{I}} &= \text{column support of } \hat{C} \\ \mathfrak{G}(\hat{C}) &= \Big\{ H \in \mathbb{R}^{p\times n} | \mathcal{P}_{\mathcal{I}_0^c}(H) = 0; \\ &\qquad \forall i \in \hat{\mathcal{I}},\ H_i = \frac{\hat{C}_i}{\left\|\hat{C}_i\right\|_2}; \\ &\qquad \forall i \in \mathcal{I}_0 \cap (\hat{\mathcal{I}})^c,\ \|H_i\|_2 \le 1 \Big\}. \end{aligned}$$

It is now straightforward to demonstrate that $\hat{Q}$ is a dual certificate as long as it satisfies the following:

$$\begin{aligned} &(a) \quad \hat{Q} \in \Omega \\ &(b) \quad \mathcal{P}_{\hat{\mathcal{T}}}(\hat{Q}) - \hat{U}\hat{V}^\top = 0 \\ &(c) \quad \left\|\mathcal{P}_{\hat{\mathcal{T}}^\perp}(\hat{Q})\right\| < 1 \\ &(d) \quad \mathcal{P}_{\mathcal{I}_0}(\hat{Q}) \in \lambda \mathfrak{G}(\hat{C}) \\ &(e) \quad \left\|\mathcal{P}_{\mathcal{I}_0^c}(\hat{Q})\right\|_{\infty,2} < \lambda. \end{aligned}$$

We construct a certificate $\hat{Q} \in \Omega$, by first constructing a certificate, Q, that satisfies (b) through (e), and then sampling it according to Ω and scaling appropriately. We then use concentration inequalities to show that the sampling procedure is "close enough" to the identity map. Following this program requires some care. In particular, the equality constraint in (b) must be relaxed, since the concentration inequalities can only guarantee that it is approximately satisfied with high probability. This is done in the next step.

Step 3. Consider any feasible perturbation, $(\hat{L} + \Delta_1, \hat{C} + \Delta_2)$. Given a $\hat{Q}$ that satisfies properties $(a) - (e)$ above, it is immediate to show that $(\hat{L} + \Delta_1, \hat{C} + \Delta_2)$ is suboptimal:

$$\left\|\hat{L}\right\|_* + \lambda \left\|\hat{C}\right\|_{1,2} \le \left\|\hat{L} + \Delta_1\right\|_* + \lambda \left\|\hat{C} + \Delta_2\right\|_{1,2}.$$

Condition (b) above, $\mathcal{P}_{\hat{\mathcal{T}}}(\hat{Q}) - \hat{U}\hat{V}^\top = 0$, comes from the need to show that the above inequality holds for all values of the perturbation, Δ_1, and in particular, its projection onto $\mathcal{P}_{\hat{\mathcal{T}}}$, the column and row space of $\hat{L}$. However, Δ_1 cannot be arbitrary.

Lemma 1. *Suppose* $\Delta_1,\ \Delta_2 \in \mathbb{R}^{p\times n}$ *are feasible perturbations, i.e., they satisfy* $P_\Omega(\Delta_1) + P_\Omega(\Delta_2) = 0$. *Then under the sampling regime in the above results, with high probability,*

$$\left\|\mathcal{P}_{\mathcal{I}_0^c}\mathcal{P}_{\hat{\mathcal{T}}}\Delta_1\right\|_F \le \sqrt{\frac{2pn_1}{m}} \left(\left\|\mathcal{P}_{\hat{\mathcal{T}}^\perp}\Delta_1\right\|_* + \left\|\mathcal{P}_{\mathcal{I}_0^c}\Delta_2\right\|_{1,2} \right).$$

Using this, the equality of condition (b) can be relaxed, leading to alternative conditions that $\hat{Q}$ must satisfy.

Proposition 1 (Condition of Success)**.** *Suppose $\lambda < 1$. Then with high probability, under the sampling regime of the results, $(\hat{L},\ \hat{C})$ is an optimal solution to* (1) *if there exists $\hat{Q}$ such that*

$$(a) \quad \hat{Q} \in \Omega,$$

$$(b') \quad \mathcal{P}_{\hat{\mathcal{T}}}(\hat{Q}) - \hat{U}\hat{V}^\top = \mathcal{P}_{\hat{\mathcal{T}}}\mathcal{R}^{-1}(D), \quad \text{for some } D \text{ with } \|D\|_F \leq \frac{1}{2}\sqrt{\frac{m}{2pn_1}}\lambda,$$

$$(c') \quad \left\|\mathcal{P}_{\hat{\mathcal{T}}^\perp}(\hat{Q})\right\| \leq \frac{1}{2},$$

$$(d) \quad \mathcal{P}_{\mathcal{I}_0}(\hat{Q}) \in \lambda\mathfrak{G}(\hat{C})$$

$$(e') \quad \left\|\mathcal{P}_{\mathcal{I}_0^c}(\hat{Q})\right\|_{\infty,2} \leq \frac{\lambda}{2}.$$

If both inequalities are strict, and $\mathcal{P}_{\mathcal{I}_0} \cap \mathcal{P}_{\hat{V}} = \{0\}$, then any optimal solution $(L',\ C')$ to (1) *satisfies $\mathcal{P}_{\mathcal{I}_0^c}(L') = L_0$, $\mathcal{P}_{U_0}(L') = L'$, and $\mathcal{P}_{\mathcal{I}_0 \cap \Omega}(C') = C'$, which means Algorithm 1 succeeds.*

Step 4. The next step requires constructing a dual certificate Q, that satisfies properties $(b)-(e)$, and also $(b')-(e')$. This is the Q that we then sample. The sampling procedure is described next.

Step 5. The final step requires us to sample Q to obtain $\hat{Q}$, and then show using concentration inequalities, that the resulting $\hat{Q}$ satisfies $(a') - (e')$ with high probability. The naive approach does not quite work, and thus requires a different sampling scheme. We do this using a modification of the approach coined "The Golfing Scheme" (Gross, 2009; Recht, 2009). We sample Ω by a modified batched sampling-with replacement scheme. The final step requires showing that Bernstein's inequality still holds under this scheme (since the sampled entries are no longer all independent).

The Oracle Problem approach, the conditions on Δ_1 and Δ_2 in the Lemma above, the alternative conditions for the certificate that we present here, and the validation of our choice of the certificate, are new. Moreover, because our objective involves a $\|\cdot\|_{1,2}$–term, our results require us to obtain new concentration results for the dual $\|\cdot\|_{\infty,2}$ bound, that are previously not known (at least to us).

5. Implementation and Simulations

To facilitate fast and efficient solution, we use a family of algorithms called Augmented Lagrange Multiplier (ALM) methods (see e.g., (Lin et al., 2009)), shown to be effective on problems involving nuclear norm minimization. We have adapted this method to our $\|\cdot\|_* + \lambda\|\cdot\|_{1,2}$-type problem; see Algorithm 2.

Algorithm 2 The ALM Algorithm for Robust Matrix Completion

input: $\mathcal{P}_\Omega(M) \in \mathbb{R}^{p\times n}$, $\Omega \subseteq [p]\times[n]$, λ (assuming $P_{\Omega^c}(M) = 0$)
initialize: $Y^{(0)} = 0$; $L^{(0)} = 0$; $C^{(0)} = 0$; $E^{(0)} = 0$; $u_0 > 0$; $\alpha > 1$; $k = 0$.
while not converged **do**
  $(U, S, V) = \text{svd}(M - E^{(k)} - C^{(k)} + u_k^{-1}Y^{(k)})$;
  $L^{(k+1)} = U\mathfrak{L}_{u_k^{-1}}(S)V^\top$;
  $C^{(k+1)} = \mathfrak{C}_{\lambda u_k^{-1}}(M - E^{(k)} - L^{(k+1)} + u_k^{-1}Y^{(k)})$;
  $E^{(k+1)} = P_{\Omega^c}(M - L^{(k+1)} - C^{(k+1)} + u_k^{-1}Y^{(k)})$;
  $Y^{(k+1)} = Y^{(k)} + u_k(M - E^{(k+1)} - L^{(k+1)} - C^{(k+1)})$;
  $u_{k+1} = \alpha u_k$;
  $k = k + 1$;
end while
return $(L^{(k+1)}, C^{(k+1)})$

Here $\mathfrak{L}_\epsilon(S)$ is the entry-wise soft-thresholding operator: if $|S_{ij}| \leq \epsilon$, then set it to zero, otherwise, let $S_{ij} := S_{ij} - \epsilon S_{ij}/|S_{ij}|$. Similarly, $\mathfrak{C}_\epsilon(C)$ is the column-wise soft-thresholding operator: if $\|C_i\|_2 \leq \epsilon$, then set it to zero, otherwise let $C_i := C_i - \epsilon C_i/\|C_i\|_2$. In our experiments, we choose $u_0 = \left(\|M\|_{1,2}\right)^{-1}$ and $\alpha = 1.1$, and the criterion for convergence is

$$\left\|M - E^{(k)} - L^{(k)} - C^{(k)}\right\|_F / \|M\|_F \leq 10^{-6}.$$

The first set of experiments demonstrates the power of the manipulator, as we show that even a single adversarially corrupted column can arbitrarily skew the prediction of standard matrix completion algorithms. In our experiments, we fix $n = p = 400$, and $\gamma = 1/400$. For different ρ and r, we generate the low-rank matrix L_0 by forming the product $L_0 = AB^\top$. The matrices $A \in \mathbb{R}^{p\times r}$ and $B \in \mathbb{R}^{n(1-\gamma)\times r}$, have i.i.d. standard Gaussian entries. The single corrupted column $C_0 \in \mathbb{R}^{p\times 1}$ is chosen identical to first column of L_0 except for the last entry, which is assigned a large value (10 in our experiments). In Collaborative Filtering this corresponds to a manipulator trying to promote the last movie. The set of observed entries in the uncorrupted columns is chosen uniformly at random from all subsets of $[p]\times[n]$ of size $\rho \times pn_1$. Set $M = \begin{bmatrix} L_0 & C_0 \end{bmatrix}$. $\mathcal{P}_\Omega(M)$ and Ω are then given as input. We apply both our algorithm and standard nuclear-norm-based matrix completion. As shown in Figure 1, standard matrix completion fails essentially for all values of ρ and r, while our algorithm is almost unaffected. Here for each pair of $(\rho,\ \gamma)$ we run the experiment for 5 times, and plot the frequency of success. Our figures show the number of successes by grayscale, where white denotes all success and black denotes all failure.

Next, we investigate our algorithm's performance under different numbers of corrupted columns, and neutral and adversarial corruption. In the first case, each entry of C_0 is i.i.d. Gaussian. In the second case, the corrupted columns are constructed as follows. For $1 \leq i \leq \gamma n$, corrupted column i copies the observed entries of clean column i and fills other entries with i.i.d. Gaussian noise. We fix $r = 10$ and vary $(\rho,\ \gamma)$. In both cases, each entry in C_0 is observed with probability ρ independently. Other settings are the same as in the first set of experiments. The results for our algorithm and standard matrix completion are shown in the left and right panes of Figure 2 for the first corruption scheme, and in Figure 3 for the second corruption scheme.

Comparison to Low-rank Plus Sparse. When only a small fraction of the entries are observed, the corrupted columns $\mathcal{P}_\Omega(C_0)$ can be viewed as a sparse matrix. Therefore, to separate L_0 from $\mathcal{P}_\Omega(C_0)$, one might think it is possible to apply the techniques in (Candès et al., 2009; Chandrasekaran et al., 2009), dubbed $L+S$ approach, which decompose a low-rank matrix and a sparse matrix from their sum. In particular, given input $\mathcal{P}_\Omega(M)$, one attempts to decompose it by solving the following convex program:

$$\begin{aligned} \min \quad & \|L\|_* + \lambda \|S\|_1 \\ \text{s.t.} \quad & \mathcal{P}_\Omega\left(L+S\right) = \mathcal{P}_\Omega\left(M\right). \end{aligned} \tag{4}$$

However, note that a central assumption of the $L + S$ approach, namely, the support of the sparse matrix is uniformly random, is violated in the setup considered in this paper. In contrast, our approach specifically deals with corrupted columns, in order to deal with persistent corruption. Therefore, it is no surprise that using the above algorithm instead should not be successful. Indeed, this is the case, and we illustrate this numerically in Figures 2 and 3, using the same synthetic data described above.

Acknowledgements

Y. Chen and C. Caramanis acknowledge support from NSF grant EFRI-0735905, and DTRA grant HDTRA1-08-0029. C. Caramanis further acknowledges support from NSF grant CNS- 0831580. H. Xu acknowledges support from NUS startup grant R-265-000-384-133. S. Sanghavi acknowledges support from NSF CAREER grant 0954059.

References

Candès, E. J., Romberg, J., and Tao, T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

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

Candès, E.J. and Tao, T. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010. ISSN 0018-9448.

Candès, E.J., Li, X., Ma, Y., and Wright, J. Robust principal component analysis? *Arxiv preprint arXiv:0912.3599*, 2009.

Cesa-Bianchi, N., Shalev-Shwartz, S., and Shamir, O. Efficient learning with partially observed attributes. In *Proceedings of the 27th international conference on Machine learning*, pp. 216–223, Haifa, Israel, 2010. ACM.

Chandrasekaran, V., Sanghavi, S., Parrilo, P.A., and Willsky, A.S. Rank-Sparsity Incoherence for Matrix Decomposition. *Arxiv preprint arXiv:0906.2220*, 2009.

Chen, Y., Xu, H., Caramanis, C., and Sanghavi, S. Robust Matrix Completion with Corrupted Columns. *Arxiv preprint arXiv:1102.2254*, 2010.

Gross, D. Recovering low-rank matrices from few coefficients in any basis. *CoRR, abs/0910.1879*, 2009.

Keshavan, R.H., Oh, S., and Montanari, A. Matrix Completion from a Few Entries. *Arxiv preprint arXiv:0901.3150*, 2009.

Lin, Z., Chen, M., Wu, L., and Ma, Y. The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices. *UIUC Technical Report UILU-ENG-09-2215*, 2009.

Negahban, S. and Wainwright, M.J. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *Arxiv preprint arXiv:1009.2118*, 2010.

Recht, B., Fazel, M., and Parrilo, P.A. Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization. *SIAM Review*, 52(471), 2010.

Recht, Benjamin. A Simpler Approach to Matrix Completion. *Arxiv preprint arXiv:0910.0651*, 2009.

Van Roy, B. and Yan, X. Manipulation Robustness of Collaborative Filtering. 2010.

Xu, H., Caramanis, C., and Mannor, S. Principal component analysis with contaminated data: The high dimensional case. *Arxiv preprint arXiv:1002.4658*, 2010a.

Xu, H., Caramanis, C., and Mannor, S. Principal component analysis with contaminated data: The high dimensional case. In *The 23rd Annual Conference on Learning Theory (COLT)*, 2010b.

Xu, H., Caramanis, C., and Sanghavi, S. Robust PCA via outlier pursuit. In *Advances in Neural Information Processing Systems*, 2010c.

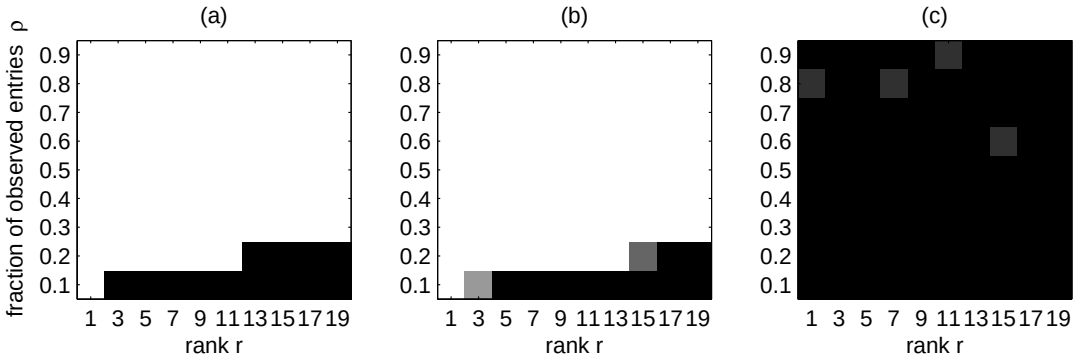

Figure 1. Experiment results for 400×400 matrix with one corrupted column. We plot the probability of successful recovery of the low rank matrix. Panes (a) and (b) show the results of our approach with and without the corrupted column, respectively. Pane (c) shows the essentially complete failure of standard matrix completion, due to the corrupted column.

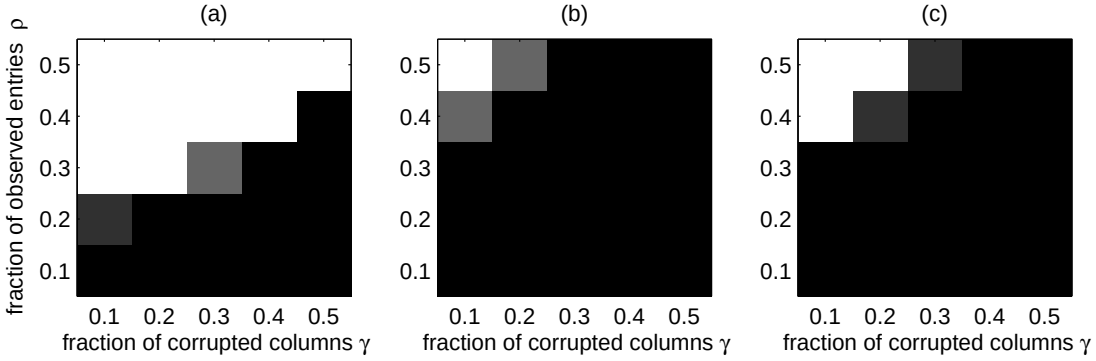

Figure 2. Experiment results for 400×400 rank-10 matrix with different fraction of observed entries ρ and fraction of corrupted columns γ. Corrupted columns are generated neutrally random. Panes (a) and (c) show the results of our approach and standard matrix completion, repectively. Pane (b) shows the results of minimizing a convex combination of the nuclear norm and the matrix ℓ_1 norm.

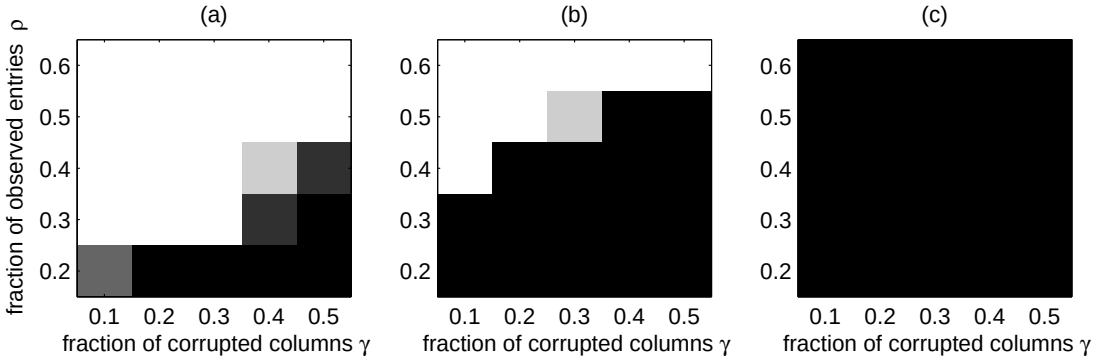

Figure 3. Experiment results for adversarial corruptions. Other settings are the same as in Fig. 2.