Data-sparse matrix computations
Lecture 25: Low Rank + Sparse Matrix Recovery

Lecturer: Anil Damle
Scribers: Heather Wilber, Lijun Ding, and Andrew Horning

September 12th, 2017

1 Introduction

In the previous lecture, we observed that it is possible to recover a sparse solution to $Ax = b$ by solving a minimization problem involving the 1-norm. In this lecture, we consider a matrix A that can be written as $A = L + S$, where L is a low rank matrix and S is a sparse matrix, and seek a method that recovers L and S. We remark that Lecture 26 forms a sequal to these notes and addresses the technical details related to recovery under the assumption that only a subset of the entries of A are observable. To motivate this work, we begin with two application-based examples.

1.1 Background subtraction

Let $A \in \mathbb{R}^{m\times n}$ be a matrix where each column j is a set of pixels representing the image recorded in a video at time j. Often, the background in a recording is static or varies slowly with time. For this reason, the background can be represented by a low rank matrix L. Motion in the foreground will typically not be low rank. However, once it is separated from the background, the foreground represents only a small fraction of the entries in A, and can encoded as a sparse matrix S (see Figure 1). By finding L and S, expensive computations involving the explicit representation of A can be avoided. A background subtraction method is described in detail in [7], and similar methods are described for a related application involving dynamic MRI data in [6].

1.2 The Netflix Prize

In 2006, the movie-streaming service Netflix offered a prize of $1,000,000 to any team or individual that could develop a collaborative filtering algorithm that outperformed their own algorithm for predicting user reviews of movies. The authors of the prize-winning result argue in [5] that a matrix factorization-based approach is key to the successful development of highly effective recommender systems. This problem appears as an example in a broader overview of matrix recovery problems in [3].

Unlike background subtraction, the Netflix recommender system is an example of a low rank + sparse recovery problem where only a small subset of the entries of A are observable. A definition of the matrix $A \in \mathbb{R}^{m\times n}$ is given in Figure 1. Each row denotes a user, and each column denotes a movie title. The users rate each movie they watch by selecting a number of stars (1 - 4), and these are recorded as entries in A. The matrix A that we seek to recover is dense: It captures a hypothetical underlying truth, with every entry reflecting the true rating that every user would give to every movie. In reality, each user has only watched a small subset of the entire possible set of movies, so only a small subset of entries in A are ever observable. Based on these observations, a predictive recommender system tries to guess what missing entries of A should be. It may not be immediately obvious that A is a low rank + sparse matrix. We give the following explanation:

The low rank component. We assume that a perfect representation of A is well-approximated by a low rank matrix L because we expect that many users behave similarly; we only require k subgroups of

users, where $k \ll m$ to make accurate predicitions. Likewise, we expect that broad subcollections of movies will be rated similarly by groups of users, so they can also be represented by a small collection $j \ll n$.

The sparse component. We can only observe a small portion of the true entries of A, and we realize that some of these observations might be outliers. To account for this and make our recommender system more robust, we write $A = L + S$, where S denotes a small amount of noise associated with the observed entries.

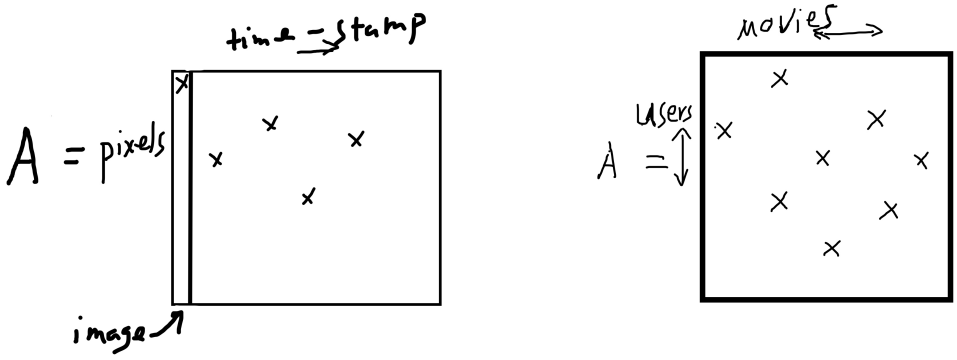

Figure 1: Left: The matrix A encodes the pixels of an image at each time step in a video. The foreground consists of a sparse collection of pixels that vary over time, whereas the background remains nearly static. Right: The matrix A encodes ratings for a collection of movies (column indices) on a 1 to 4 scale for users (row indices). Only a small portion of the entries of A are observable.

Methods and theoretical results related to finding L and S in problems like this one are given in the notes for Lecture 26.

2 Solving the recovery problem via Principle Component Pursuit.

2.1 Recovery method

We now outline a method for recovering the matrices L and S in the background subtraction example, and additionally discuss the assumptions needed to theoretically guarantee that L and S can be recovered. In order to sketch a method for solving this problem, we require definitions for a few key concepts.

Definition 1. *For a matrix $M \in \mathbb{R}^{m\times n}$, let $\|M\|_* = \sum_{i=1}^{\min(m,n)} \sigma_i(A)$ be the nuclear norm of M, where $\sigma_i(M)$ are the singular values of M. One can also view the nuclear norm as the l_1 norm applied to the vector of M's singular values.*

Definition 2. *For a matrix $M \in \mathbb{R}^{m\times n}$, let $|||M|||_1 = \sum_{i,j} |M_{ij}|$ be the element-wise l_1 norm of M.*

Equipped with these two notions, we state the following optimization problem:

$$(P1) \quad \underset{L,S}{\text{minimize}} \quad \|L\|_* + \lambda |||S|||_1$$
$$\text{subject to} \quad A = L + S,$$

where λ is a positive tuning parameter. Intuitively, the minimization of L in nuclear norm encourages the rank of L to be small. Minimizing S with respect to the $|||\cdot|||$ norm promotes sparsity in S. Seeking the low rank and sparse components of A through solving this optimization problem is known as principle component pursuit, and this strategy is analyzed in greater detail in [4, 2].

Suppose that $\hat{L}$ and $\hat{S}$ are minimizers of the problem $(P1)$. If the true solution is given by $A = L + S$, we want to know whether $\hat{L} = L$ and $\hat{S} = S$.

Without additional assumptions on L and S, this problem may not have a unique solution, since there may be issues related to *identifiability*. Consider, for example, the matrix

$$A = e_i e_j^T + e_{i'} e_{j'}^T,$$

where $(i,j) \neq (i',j')$. This scenario is problematic, since we cannot determine whether $L = e_i e_j^T$ and $S = e_{i'} e_{j'}$, or, for example, $L = e_i e_j^T + e_{i'} e_{j'}^T$ and $S = 0$. Other combinations are also possible. The trouble here is that matrices like $e_i e_j^T$ are both low rank and sparse. To avoid the identifiability issue, we must impose additional conditions on L and S ensuring that they cannot be simultaneously low rank and sparse.

We enforce the condition that L is not sparse through the use of an *incoherence condition*, defined below:

Definition 3. *Let $L = U\Sigma V^T = \sum_{i=1}^k \sigma_i u_i v_i^T$ be the singular value decomposition of L. L satisfies an incoherence condition with parameter μ if the following conditions are satisfied.*

1. $\max_i \|U^T e_i\|_2^2 \leq \frac{\mu k}{n}$.
2. $\max_i \|V^T e_i\|_2^2 \leq \frac{\mu k}{n}$.
3. $\left\|\left|UV^T\right|\right\|_\infty = \max_{ij} |(UV^T)_{ij}| \leq \sqrt{\frac{\mu k}{n^2}}$.

Essentially, the first two conditions imply that the rows of U and V are poorly correlated with the standard basis vectors when μ is small. Taken together, the three items in the incoherence condition ensure that the singular vectors of L are not too sparse.

It is also necessary to assume that the sparsity pattern of S is not too structured. To see why, consider the decomposition of A into a low-rank component L and a sparse component S. If S is a sparse matrix consisting of a column whose entries are identical in magnitude and opposite in sign to the corresponding column in L, then we face another identifiability issue. To ensure that this does not occur, we assume that the sparsity pattern of S is distributed uniformly over sets of cardinality p, where p is the number of nonzeros in S.

2.2 Theoretical guarantees

Under the above assumptions on L and S, a guarantee on the exact recovery of L and S can be stated. We provide the formal result for square matrices as follows:

Theorem 1 (Candes et al [2])**.** *Let $A = L+S$. Suppose that $L \in \mathbb{R}^{n\times n}$ is μ incoherent and fix any $M \in \mathbb{R}^{n\times n}$ whose entries have values ± 1. Now, suppose the index support set Ω of S is uniformly distributed over all sets of cardinality p, where $p = \text{nnz}(S)$, and, furthermore, suppose $sign(S_{ij}) = M_{ij}$ for all $i,j \in \Omega$. Then, there exists a constant c such that the following holds with probability at least $1 - cn^{-10}$. When $\lambda = 1/\sqrt{n}$, the minimizers $\hat{L}, \hat{S}$ of $(P1)$ are exact, i.e. $\hat{L} = L$ and $\hat{S} = S$, provided that $rank(L) \leq c_1 n \mu^{-1} (log(n))^{-2}$ and $p \leq c_2 n^2$, where c_1, c_2 are constants.*

The proof of this result is lengthy and beyond the scope of this course. It relies on the formulation of a dual problem for which estimates can be obtained. For those who are interested, see [2]. Remarkably, the requirement on the low-rank component L is mild - the rank can scale linearly (apart from a logarithmic factor) with the dimension n of A. Note that the constant c will depend on the constants c_1 and c_2.

As a final note, we mention that the PCP optimization problem may be solved using a variety of approaches. Several efficient algorithms have been derived from the classical augmented Lagrangian methods (ALM). See [1] for a comprehensive survey and performance comparisons.

3 Stable Principle Component Pursuit

Classical principle component analysis (PCA) seeks a low rank decomposition of a matrix $A \in \mathbb{R}^{n\times n}$ whose entries have been corrupted by small levels of noise. Mathematically, the noise takes the form of a matrix with very small i.i.d. Gaussian entries. PCA is widely used in statistical data analysis and dimensionality reduction. However, it is fragile with respect to larger noise levels, even if only a few data points have been

severely corrupted. The principle component pursuit (PCP) formulation described in Section 2 provides a scalable method for recovering the low rank decomposition of A in the presence of highly corrupted data, provided that the corrupted data makes up a small fraction of the total data set.

In real-world applications, such as background/foreground subtraction, there is often a mixture of low-level noise spread across the data set and significant corruption supported on a small subset of the data. In 2010, Zhou et al proposed a stable principle component pursuit (SPCP) formulation which can be viewed as a stabilization of PCA to significant data corruption or as a stabilization of PCP to persistant low-level noise [8].

Suppose that $A \in \mathbb{R}^{n\times n}$ now has the decomposition $A = L + S + Z$ where L and S are the same as in Section two (low rank and sparse, respectively), and Z is a 'noise term' satisfying $\|Z\|_F < \delta$ for some $\delta > 0$. To recover L and S, Zhou et al form a relaxed version of (P1),

$$\begin{aligned} (P2) \quad & \underset{L,S}{\text{minimize}} \quad \|L\|_* + \lambda |||S|||_1 \\ & \text{subject to} \quad \|A - L - S\|_F < \delta. \end{aligned}$$

An anologue of Theorem 1 can be established in this regime, and is formally given by the following:

Theorem 2 (Zhou et al [8])**.** *Let $A = L + S + Z$, with L, S satisfying the hypotheses of Theorem 1, and $\|Z\|_F < \delta$ for some $\delta > 0$. If the numerical constants c_1, c_2 from Theorem 1 are sufficiently small then, with high probability in the support of S, the solution $(\hat{L}, \hat{S})$ to (P2) satisfies*

$$\|\hat{L} - L\|_F^2 + \|\hat{S} - S\|_F^2 < Cn^2\delta^2,$$

where C is a numerical constant.

The SPCS optimization problem in (P2) can be solved efficiently using convex optimization techniques and is not much more expensive than the solution of (P1) [8].

References

[1] Thierry Bouwmans and El Hadi Zahzah. Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance. *Computer Vision and Image Understanding*, 122(Supplement C):22 – 34, 2014.

[2] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

[3] Emmanuel J Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

[4] Venkat Chandrasekaran, Sujay Sanghavi, Pablo A Parrilo, and Alan S Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.

[5] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.

[6] Ricardo Otazo, Emmanuel Candès, and Daniel K Sodickson. Low-rank plus sparse matrix decomposition for accelerated dynamic MRI with separation of background and dynamic components. *Magnetic Resonance in Medicine*, 73(3):1125–1136, 2015.

[7] John Wright, Arvind Ganesh, Shankar Rao, Yigang Peng, and Yi Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in neural information processing systems*, pages 2080–2088, 2009.

[8] Zihan Zhou, Xiaodong Li, John Wright, Emmanuel Candes, and Lei Yu. Stable principal component pursuit, 07 2010.