Chapter 5

Decomposing Low-rank and Sparse Matrices

In the previous chapters, we have studied how either a sparse vector or a low-rank matrix can be recovered from compressive or incomplete measurements. In this chapter, we will show it is also possible to simultaneously recover a sparse signal and a low-rank signal from their superposition or from highly compressive measurements of their superposition. This combination of rank and sparsity gives rise to a broader class of models that can be used to model richer structures of high-dimensional data, as we will see in the examples in this chapter. Nevertheless, we are also faced with new technical challenges about whether and how such structures can be recovered correctly and effectively, from how few observations.

5.1 Robust PCA and Motivating Examples

In this chapter, we study variants of the following problem. We are given a large data matrix $\boldsymbol{Y} \in \mathbb{R}^{n_1 \times n_2}$ which is a superposition of two matrices:

$$\boldsymbol{Y} = \boldsymbol{L}_o + \boldsymbol{S}_o, \tag{5.1.1}$$

where $\boldsymbol{L}_o \in \mathbb{R}^{n_1 \times n_2}$ is a low-rank matrix and $\boldsymbol{S}_o \in \mathbb{R}^{n_1 \times n_2}$ is a sparse matrix. Neither $\boldsymbol{L}_o$, nor $\boldsymbol{S}_o$ is known ahead of time. Can we hope to efficiently recover both $\boldsymbol{L}_o$ and $\boldsymbol{S}_o$?

This problem resembles another classical low-rank matrix recovery problem in which the observed data matrix $\boldsymbol{Y} \in \mathbb{R}^{n_1 \times n_2}$ is a superposition of

two matrices:

$$\boldsymbol{Y} = \boldsymbol{L}_o + \boldsymbol{Z}_o, \tag{5.1.2}$$

where as before $\boldsymbol{L}_o \in \mathbb{R}^{n_1 \times n_2}$ is a low-rank matrix but here $\boldsymbol{Z}_o \in \mathbb{R}^{n_1 \times n_2}$ is assumed to be a dense matrix of small perturbation. For example, $\boldsymbol{Z}_o$ could be a Gaussian random matrix of small magnitude. In other words, one wants to recover a low-rank matrix (or the low-dimensional subspace spanned by the columns of $\boldsymbol{L}_o$) from noisy measurements. The classical *Principal Component Analysis* (PCA) [Jolliffe, 1986, Jollife, 2002] seeks the best rank-r estimate of $\boldsymbol{L}_o$ by solving

$$\min_{\boldsymbol{L}} \|\boldsymbol{Y} - \boldsymbol{L}\|_F \quad \text{s.t.} \quad \operatorname{rank}(\boldsymbol{L}) \leq r. \tag{5.1.3}$$

This problem is also known as the best rank-r approximation problem. As we have seen in Section 4.2.2, it can be solved very efficiently via the *Singular Value Decomposition* (SVD): If $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ is the SVD of the matrix $\boldsymbol{Y}$, the optimal rank-r approximation to $\boldsymbol{Y}$ is

$$\boldsymbol{L}^* = \boldsymbol{U}\boldsymbol{\Sigma}_r\boldsymbol{V}^T,$$

where $\boldsymbol{\Sigma}_r$ keeps only the first r leading singular values of the diagonal matrix $\boldsymbol{\Sigma}$. This solution enjoys a number of optimality properties when the perturbation in matrix $\boldsymbol{Z}_o$ is small or i.i.d. Gaussian.

However, in the new measurement model (5.1.1), the perturbation term $\boldsymbol{S}_o$ can have elements with arbitrary magnitude and hence its ℓ^2 norm can be unbounded. In a sense, the measurement we observe

$$\boldsymbol{Y} = \boldsymbol{L}_o + \boldsymbol{S}_o$$

is a corrupted version of the low-rank matrix $\boldsymbol{L}_o$ – entries of $\boldsymbol{Y}$ where $\boldsymbol{S}_o$ is nonzero carry no information about $\boldsymbol{L}_o$. The problem of recovering the matrix $\boldsymbol{L}_o$ (and the associated low-dimensional subspace) from such highly corrupted measurements can be considered a form of *Robust Principal Component Analysis* (RPCA), as opposed to the classical PCA which is only stable to small noise or perturbation.

5.1.1 Applications of Robust PCA

Many important practical applications confront us with instances of the problem (5.1.1). We here give a few representative examples inspired by some contemporary challenges from computer science and data analysis. Notice that depending on the applications, either the low-rank component or the sparse component could be the object of interest:

- *Video Surveillance.* Given a sequence of surveillance video frames, we often need to identify activities that stand out from the background. If we stack the video frames as columns of a matrix $\boldsymbol{Y}$, then the low-rank component $\boldsymbol{L}_o$ represents the stationary background and the

sparse component $\boldsymbol{S}_o$ captures the moving objects in the foreground. However, since each image frame may have thousands or millions of pixels and each video fragment may contain hundreds or thousands of frames, it would be only possible to decompose $\boldsymbol{Y}$ this way if we have a truly scalable solution to this problem.

- *Face Recognition.* It is well known that images of a convex, Lambertian surface under varying illuminations span a low-dimensional subspace [Basri and Jacobs, 2003b]. That is, if we stack face images of a person as column vectors of a matrix, then this matrix is a low-rank matrix $\boldsymbol{L}_o$. This fact has been a main reason why low-dimensional models are mostly effective for imagery data. In particular, images of a human's face can be well-approximated by a low-dimensional subspace. Being able to correctly retrieve this subspace is crucial in many applications such as face recognition and alignment. However, realistic face images often suffer from self-shadowing, specularities, or saturation in brightness, which make this a difficult task and subsequently compromise the recognition performance. A more careful study shows that the face images are better modeled by a low-rank matrix $\boldsymbol{L}_o$ superposed with a sparse matrix $\boldsymbol{S}_o$ which models such imperfection [Zhang et al., 2013].

- *Latent Semantic Indexing.* Web search engines often need to analyze and index the content of an enormous corpus of documents. A popular scheme is the *Latent Semantic Indexing* (LSI) [Dumais et al., 1988, Papadimitriou et al., 1998]. The basic idea is to gather a document-versus-term matrix $\boldsymbol{Y}$ whose entries typically encode the relevance of a term (or a word) to a document such as the frequency it appears in the document (e.g., term frequency-inverse document frequency, also known as TF-IDF). PCA (or SVD) has traditionally been used to decompose the matrix as a low-rank part plus a residual, which is not necessarily sparse (as we would like). If we were able to decompose $\boldsymbol{Y}$ as a sum of a low-rank component $\boldsymbol{L}_o$ and a sparse component $\boldsymbol{S}_o$, then $\boldsymbol{L}_o$ could capture common words used in all the documents while $\boldsymbol{S}_o$ captures the few keywords that best distinguish each document from others. See Chapter 16, Section 16.1.1, for more details about such a *joint topic-document model* (via a superposition of a low-rank and sparse matrix) [Min et al., 2010].

- *Ranking and Collaborative Filtering.* The problem of anticipating user preference has been an important problem in online commerce and advertisement. Companies now routinely collect user rankings for various products, e.g., movies, books, games, or web tools, among which the Netflix Prize for movie ranking is the best known example. The problem posed in the Netflix Prize is to use very sparse and incomplete rankings provided by the users on some of the products to

predict the preference of any given user on any products, also known as collaborative filtering [Hofmann, 2004b]. This problem is typically cast as a low-rank matrix completion problem. However, as the data collection process often lacks control or is sometimes even *ad hoc* – a small portion of the available rankings could be noisy and even tampered with by malicious users or competitors. The problem is more challenging since we need to simultaneously complete the matrix and correct the errors. That is, we need to infer a low-rank matrix $\boldsymbol{L}_o$ from a set of incomplete and corrupted entries, a problem that methods introduced in the previous chapter are incapable of solving.

All the applications that we have listed above require solving the problem of decomposing a low-rank and sparse matrix possibly of ver high dimension, under various conditions. As it turns out, mathematically, this class of problems are rather fundamental to machine learning and system theory. They are actually the underlying problem for correctly and robustly learning graphical models and identifying dynamical systems, as discussed in the Preface of this book.

5.2 Robust PCA via Principal Component Pursuit

In each of the above problems, the dataset $\boldsymbol{Y}$ can be modeled as a superposition of a low-rank matrix and a sparse matrix:

$$\boldsymbol{Y} = \boldsymbol{L}_o + \boldsymbol{S}_o. \tag{5.2.1}$$

For the majority of this chapter, we will simplify notation by assuming $\boldsymbol{Y} \in \mathbb{R}^{n \times n}$ is a square matrix. Extensions of both the theory and algorithms to the non-square case $\boldsymbol{Y} \in \mathbb{R}^{n_1 \times n_2}$ are for the most part straightforward, some of which will be discussed in Section 5.3 or left to the reader as exercises.

In this chapter, we use $\mathfrak{S}$ and $\boldsymbol{\Sigma}_o$ for the support and signs of $\boldsymbol{S}_o$, respectively:

$$\mathfrak{S} = \operatorname{supp}(\boldsymbol{S}_o) \subseteq [n] \times [n], \tag{5.2.2}$$

$$\boldsymbol{\Sigma}_o = \operatorname{sign}(\boldsymbol{S}_o) \in \{-1, 0, 1\}^{n \times n}. \tag{5.2.3}$$

We note that if we somehow knew the support $\mathfrak{S}$ of $\boldsymbol{S}_o$, we could potentially recover $\boldsymbol{L}_o$ by solving a Matrix Completion problem (as in the previous chapter) using $\mathcal{P}_\Omega[\boldsymbol{L}_o]$ with $\Omega = \mathfrak{S}^c$. But in the problems described above, both $\boldsymbol{L}_o$ and $\boldsymbol{S}_o$ are unknown.

Using this connection to matrix completion, one can show that the Robust PCA problem is NP-Hard in general. The hardness can also be shown directly via a connection to the concept of *matrix rigidity*, which arises in computational complexity and related areas. Exercise 5.11 introduces this concept and guides the interested reader through this connection. For our

purposes, let us simply note that the situation for Robust PCA is analogous to that for low-rank recovery and for sparse recovery: *we should not expect to find an efficient algorithm which works for every problem instance.*

5.2.1 Convex Relaxation for Sparse Low-Rank Separation

Nevertheless, like sparse vector recovery and low-rank matrix recovery, we *might* expect to find efficient algorithms that solve certain well-structured instances of practical importance. Based on our investigation in Chapters 3-4, we should have a very clear idea of how to approach this! A natural idea is to solve a problem with two matrix valued variables of optimization, $\boldsymbol{L}$ and $\boldsymbol{S}$, in which we try to make the nuclear norm of $\boldsymbol{L}$ small and the ℓ^1 norm of $\boldsymbol{S}$ small:

$$\begin{array}{ll} \text{minimize} & \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 \\ \text{subject to} & \boldsymbol{L} + \boldsymbol{S} = \boldsymbol{Y}. \end{array} \tag{5.2.4}$$

Here, $\lambda > 0$ is a positive weight parameter. The linear equality constraint $\boldsymbol{L} + \boldsymbol{S} = \boldsymbol{Y}$ is convex; moreover, since a sum of two convex functions is convex, the objective is also convex. This is a convex program, which we refer to as *Principal Component Pursuit* (PCP).

The relative ease with which we derived this convex relaxation highlights a conceptual advantage of "convex modeling": because convex sets and functions can be combined in nontrivial ways to form new convex sets and functions, it is often straightforward to extend the models to handle new situations of practical interest. Indeed, although it should be straightforward to write down the optimization problem (5.2.4), this opens the door to many new applications, including those listed in the previous section.

Nevertheless, two crucial questions remain. First, since most of these applications involve large data sets, we will need both efficient and scalable algorithms for solving the problem (5.2.4). Second, to deploy the algorithms with confidence, we will need to understand if and when they correctly recover the target low-rank and sparse components $\boldsymbol{L}_o$ and $\boldsymbol{S}_o$. We will address these questions in Sections 5.2.2 and 5.3 respectively. We will then close the chapter with a discussion of several additional extensions to problem with both corruptions *and* missing data, which further highlight the flexibility of the framework and allow us to model additional factors in practical applications.

5.2.2 Solving PCP via Alternating Directions Method

The PCP problem can be solved to very high accuracy in polynomial time using semidefinite programming. Classical polynomial time algorithms for SDP are based on interior point methods [Grant and Boyd, 2014], which converge in very few steps, but have a high per-step cost ($O(n^6)$ for a problem involving $n \times n$ matrices). This complexity limits such methods to

be practical only for problems with small sizes, say $n < 100$. However, for most aforementioned applications of PCP/RPCA, n can be very large. In such situations, a more appropriate goal is to achieve moderate accuracy with algorithms that are both scalable and efficient. In this section, we sketch one way of achieving this, using the technology of Lagrange duality – in particular, the *alternating directions method of multipliers* (ADMM), which will be studied in more details for general cases in Chapter 8.

The main challenge in efficiently solving the PCP problem is coping with the constraint $\boldsymbol{L}+\boldsymbol{S} = \boldsymbol{Y}$. As in the previous chapter on matrix completion, we use the machinery of Lagrange duality. Here, the *Lagrangian* is

$$\mathcal{L}(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) = \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}, \boldsymbol{Y} - \boldsymbol{L} - \boldsymbol{S} \rangle . \tag{5.2.5}$$

which is used both for deriving algorithms and optimality conditions. The *augmented* Lagrangian is

$$\mathcal{L}_\mu(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) = \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}, \boldsymbol{Y} - \boldsymbol{L} - \boldsymbol{S}\rangle + \frac{\mu}{2}\|\boldsymbol{Y} - \boldsymbol{L} - \boldsymbol{S}\|_F^2. \tag{5.2.6}$$

A generic Lagrange multiplier algorithm [Bertsekas, 1982a], like the one we derived for matrix completion, would solve PCP by repeatedly setting

$$(\boldsymbol{L}_k, \boldsymbol{S}_k) = \arg\min_{\boldsymbol{L},\boldsymbol{S}} \mathcal{L}_\mu(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}_k),$$

and then updating the Lagrange multipliers (here as a matrix) via

$$\boldsymbol{\Lambda}_{k+1} = \boldsymbol{\Lambda}_k + \mu(\boldsymbol{Y} - \boldsymbol{L}_k - \boldsymbol{S}_k).$$

For our low-rank and sparse decomposition problem, we can avoid having to solve a sequence of convex programs by recognizing that $\min_{\boldsymbol{L}} \mathcal{L}_\mu(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda})$ and $\min_{\boldsymbol{S}} \mathcal{L}_\mu(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda})$ both have very simple and efficient solutions.

Let $\mathcal{S}_\tau : \mathbb{R} \to \mathbb{R}$ denote the shrinkage operator

$$\mathcal{S}_\tau[x] = \operatorname{sgn}(x) \max(|x| - \tau, 0),$$

and extend it to matrices by applying it to each element. It is easy to show that

$$\arg\min_{\boldsymbol{S}} \mathcal{L}_\mu(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) = \mathcal{S}_{\lambda/\mu}(\boldsymbol{Y} - \boldsymbol{L} + \mu^{-1}\boldsymbol{\Lambda}). \tag{5.2.7}$$

Similarly, for matrices $\boldsymbol{M}$, let $\mathcal{D}_\tau(\boldsymbol{M})$ denote the singular value thresholding operator given by $\mathcal{D}_\tau(\boldsymbol{M}) = \boldsymbol{U}\mathcal{S}_\tau(\boldsymbol{\Sigma})\boldsymbol{V}^*$, where $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$ is any singular value decomposition. It is not difficult to show that

$$\arg\min_{\boldsymbol{L}} \mathcal{L}_\mu(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) = \mathcal{D}_{1/\mu}(\boldsymbol{Y} - \boldsymbol{S} + \mu^{-1}\boldsymbol{\Lambda}). \tag{5.2.8}$$

Thus, a more practical strategy is to first minimize $\mathcal{L}_\mu$ with respect to $\boldsymbol{L}$ (fixing $\boldsymbol{S}$), then minimize $\mathcal{L}_\mu$ with respect to $\boldsymbol{S}$ (fixing $\boldsymbol{L}$), and then finally update the Lagrange multiplier matrix $\boldsymbol{\Lambda}$ based on the residual $\boldsymbol{Y} - \boldsymbol{L} - \boldsymbol{S}$, a strategy that is summarized as Algorithm 5.1 below.

Algorithm 5.1 (PCP by ADMM)

Initialize: $\boldsymbol{S}_0 = \boldsymbol{\Lambda}_0 = 0, \mu > 0$.
while not converged **do**
  Compute $\boldsymbol{L}_{k+1} = \mathcal{D}_{1/\mu}(\boldsymbol{Y} - \boldsymbol{S}_k + \mu^{-1}\boldsymbol{\Lambda}_k)$;
  Compute $\boldsymbol{S}_{k+1} = \mathcal{S}_{\lambda/\mu}(\boldsymbol{Y} - \boldsymbol{L}_{k+1} + \mu^{-1}\boldsymbol{\Lambda}_k)$;
  Compute $\boldsymbol{\Lambda}_{k+1} = \boldsymbol{\Lambda}_k + \mu(\boldsymbol{Y} - \boldsymbol{L}_{k+1} - \boldsymbol{S}_{k+1})$;
end while
Output: $\boldsymbol{L}_\star \leftarrow \boldsymbol{L}_k; \boldsymbol{S}_\star \leftarrow \boldsymbol{S}_k$.

Algorithm 5.1 is a special case of a more general class of augmented Lagrange multiplier algorithms known as *alternating directions* methods of multipliers (ADMM). We will study ADMM in more detail in Section 8.5 of Chapter 8, including its convergence and other matters. Algorithm 5.1 performs excellently on a wide range of instances: as we will see below, relatively small numbers of iterations suffice to achieve good relative accuracy. The dominant cost of each iteration is computing $\boldsymbol{L}_{k+1}$ via singular value thresholding. This requires us to compute those singular vectors of $\boldsymbol{Y} - \boldsymbol{S}_k + \mu^{-1}\boldsymbol{\Lambda}_k$ whose corresponding singular values exceed the threshold μ. Empirically, we have observed that the number of such large singular values is often bounded by $\text{rank}(\boldsymbol{L}_o)$, allowing the next iterate to be computed efficiently via a partial SVD.[1]

Very similar ideas can be used to develop simple and effective augmented Lagrange multiplier algorithms for the robust matrix completion problem (5.6.1) to be introduced in Section 5.6, with similarly good performance.

5.2.3 Numerical Simulations and Experiments of PCP

In this section, we perform numerical simulations and experiments of Algorithm 5.1 for PCP and illustrate several of its many applications in image and video analysis. We first investigate its ability to correctly recover matrices of various rank from errors of various density. We then sketch applications in background modeling from video and removing shadows and specularities from face images.

One important implementation detail in PCP is the choice of λ. As we will see in the next section, the analysis to justify the effectiveness of PCP suggests one natural choice, $\lambda = 1/\sqrt{\max(n_1, n_2)}$, which will be used throughout this section. For practical problems, however, it is often possible to improve performance by choosing λ according to prior knowledge about the solution. For example, if we know that $\boldsymbol{S}$ is very sparse, increasing λ will allow us to recover matrices $\boldsymbol{L}$ of larger rank. For practical

[1] Further performance gains might be possible by replacing this partial SVD with an approximate SVD, as suggested in [Goldfarb and Ma, 2009] for nuclear norm minimization.

Dim. n	rank($\boldsymbol{L}_o$)	$\|\boldsymbol{S}_o\|_0$	rank($\hat{\boldsymbol{L}}$)	$\|\hat{\boldsymbol{S}}\|_0$	$\frac{\|\hat{\boldsymbol{L}}-\boldsymbol{L}_o\|_F}{\|\boldsymbol{L}_o\|_F}$	# SVD	time(s)
500	25	25,000	25	25,000	1.2×10^{-6}	17	4.0
1,000	50	100,000	50	100,000	2.4×10^{-6}	16	13.7
2,000	100	400,000	100	400,000	2.4×10^{-6}	16	64.5
3,000	150	900,000	150	900,000	2.5×10^{-6}	16	191.0

Table 5.1. **Correct recovery for random problems of varying size.** Here, $\boldsymbol{L}_o = \boldsymbol{U}\boldsymbol{V}^* \in \mathbb{R}^{n\times n}$ with $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{n\times r}$; $\boldsymbol{U}, \boldsymbol{V}$ have entries sampled from i.i.d. $\mathcal{N}(0, 1/n)$. $\boldsymbol{S}_o \in \{-1, 0, 1\}^{n\times n}$ has support chosen uniformly at random and independent random signs; $\|\boldsymbol{S}_o\|_0$ is the number of nonzero entries in $\boldsymbol{S}_o$. In all cases, the rank of $\boldsymbol{L}_o$ and ℓ^0-norm of $\boldsymbol{S}_o$ are correctly estimated. Moreover, the number of partial singular value decompositions (# SVD) required to solve PCP is almost constant.

problems, we recommend $\lambda = 1/\sqrt{\max(n_1, n_2)}$ as a good rule of thumb, which can then be adjusted slightly to obtain possibly better results.

Simulation: exact recovery from varying fractions of error.

We first verify how the algorithm does on recovering randomly generated instances, under favorable conditions (i.e., rank of $\boldsymbol{L}$ is very low and $\boldsymbol{S}$ is rather sparse). We consider square matrices of varying dimension $n = 500, \ldots, 3000$. We generate a rank-r matrix $\boldsymbol{L}_o$ as a product $\boldsymbol{L}_o = \boldsymbol{U}\boldsymbol{Y}^*$ where $\boldsymbol{U}$ and $\boldsymbol{V}$ are $n \times r$ matrices with entries independently sampled from a $\mathcal{N}(0, 1/n)$ distribution. $\boldsymbol{S}_o$ is generated by choosing a support set $\mathfrak{S}$ of size k uniformly at random, and setting $\boldsymbol{S}_o = \mathcal{P}_{\mathfrak{S}}\boldsymbol{E}$, where $\boldsymbol{E}$ is a matrix with independent Bernoulli ± 1 entries.

Table 5.1 reports the results for a challenging scenario: $\text{rank}(\boldsymbol{L}_o) = 0.05 \times n$ and $k = 0.10 \cdot n^2$. In all cases, we set $\lambda = 1/\sqrt{n}$. Notice that in all cases, solving the convex PCP gives a result $(\hat{\boldsymbol{L}}, \hat{\boldsymbol{S}})$ with the correct rank and sparsity. Moreover, the relative error $\|\hat{\boldsymbol{L}} - \boldsymbol{L}_o\|_F / \|\boldsymbol{L}_o\|_F$ is small, less than 10^{-5} in all examples considered.[2]

The last two columns of Table 5.1 give the number of partial singular value decompositions computed in the course of the optimization (# SVD) as well as the total computation time.[3] As we see from Algorithm 5.1, the dominant cost in solving the convex program comes from computing one partial SVD per iteration. Strikingly, in Table 5.1, the number of SVD computations is nearly constant regardless of dimension, and in all cases less than 17, suggesting that the ADMM algorithm gives a reasonably practical solver for PCP.

[2]We measure relative error in terms of $\boldsymbol{L}$ only, since we usually view the sparse and low-rank decomposition as recovering a low-rank matrix $\boldsymbol{L}_o$ from gross errors. $\boldsymbol{S}_o$ is of

Experiment: background modeling from surveillance video.

Video is a natural candidate for low-rank modeling, due to the correlation between frames. One of the most basic algorithmic tasks in video surveillance is to estimate a good model for the background variations in a scene. This task is complicated by the presence of foreground objects: in busy scenes, every frame may contain some anomaly. Moreover, the background model needs to be flexible enough to accommodate changes in the scene, for example due to varying illumination. In such situations, it is natural to model the background variations as approximately low rank. Foreground objects, such as cars or pedestrians, generally occupy only a fraction of the image pixels and hence can be treated as sparse errors.

We investigate whether the convex PCP program can separate these sparse errors from the low-rank background. Here, it is important to note that the error support may not be well-modeled as Bernoulli: errors tend to be spatially coherent, and more complicated models such as Markov random fields may be more appropriate [Cevher et al., 2009, Zhou et al., 2009]. Hence, our theorems do not necessarily guarantee the algorithm will succeed with high probability. Nevertheless, as we will see, PCP still gives visually appealing solutions to this practical low-rank and sparse separation problem, without using any additional information about the spatial structure of the error.

We consider two example videos introduced in [Li et al., 2004]. The first is a sequence of 200 grayscale frames taken in an airport. This video has a relatively static background, but significant foreground variations. The frames have resolution 176×144; we stack each frame as a column of our matrix $\boldsymbol{Y} \in \mathbb{R}^{25,344 \times 200}$. We decompose $\boldsymbol{Y}$ into a low-rank term and a sparse term by solving the convex PCP problem (5.2.4) with $\lambda = 1/\sqrt{n_1}$. Figure 5.1(a) shows three frames from the video; (b) and (c) show the corresponding columns of the low rank matrix $\hat{\boldsymbol{L}}$ and sparse matrix $\hat{\boldsymbol{S}}$ (its absolute value is shown here). Notice that $\hat{\boldsymbol{L}}$ correctly recovers the background, while $\hat{\boldsymbol{S}}$ correctly identifies the moving pedestrians. One person appearing in the images in $\hat{\boldsymbol{L}}$ does not move throughout the video, hence it was (correctly) modeled as part of the static background.

We have noticed that the number of iterations for the real data is typically higher than that of the simulations with random matrices given in Table 5.1. The reason for this discrepancy might be that the structures of real data could slightly deviate from the idealistic low-rank and sparse model. Nevertheless, it is important to realize that practical applications

---

course also well-recovered: in this example, the relative error in $\boldsymbol{S}$ is actually smaller than that in $\boldsymbol{L}$.

[3]This experiment was performed in Matlab on a Mac Pro with dual quad-core 2.66 GHz Intel Xenon processors and 16 GB RAM.

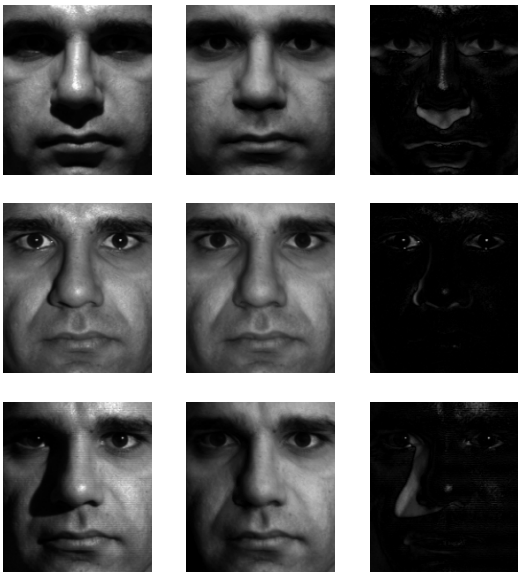
(a) Original frames (b) Low-rank $\hat{\boldsymbol{L}}$ (c) Sparse $\hat{\boldsymbol{S}}$

Figure 5.1. Background modeling from video. Three frames from a 200 frame video sequence taken in an airport [Li et al., 2004]. (a) Frames of original video $\boldsymbol{Y}$. (b)-(c) Low-rank $\hat{\boldsymbol{L}}$ and sparse components $\hat{\boldsymbol{S}}$ obtained by PCP.

such as video surveillance often provide additional information about the signals of interest, e.g. the support of the sparse foreground is spatially piecewise contiguous and temporarily continuous among frames. Or they even impose additional requirements, e.g. the recovered background needs to be non-negative etc. We note that the simplicity of our objective and solution suggests that one can easily incorporate additional constraints and more accurate models of the signals so as to obtain much more efficient and accurate solutions.

Experiment: removing shadows and specularities from face images.

Face recognition is another problem domain in computer vision where low-dimensional linear models have received a great deal of attention. This is mostly due to the work of Basri and Jacobs, who showed that for convex, Lambertian objects, images taken under distant illumination lie approximately in a nine-dimensional linear subspace known as the *harmonic plane* [Basri and Jacobs, 2003b]. However, since faces are neither perfectly convex nor Lambertian, real face images often violate this low-rank model, in part due to cast shadows and specularities. These errors may



(a) $\boldsymbol{Y}$ (b) $\hat{\boldsymbol{L}}$ (c) $\hat{\boldsymbol{S}}$

Figure 5.2. Removing shadows, specularities, and saturations from face images. (a) Cropped and aligned images of a person's face under different illuminations from the Extended Yale B database. The size of each image is 192×168 pixels, a total of 58 different illuminations per person. (b) Low-rank approximation $\hat{\boldsymbol{L}}$ recovered by convex programming. (c) Sparse error $\hat{\boldsymbol{S}}$ corresponding to specularities in the eyes, shadows around the nose region, or brightness saturations on the face. Notice in the bottom left that the sparse term also compensates for errors in image acquisition.

be large in magnitude but sparse in the spatial domain. It is reasonable to believe that if we have enough images of the same face, PCP will be able to remove these errors. As with the previous example, some caveats apply: the theoretical result suggests the performance should be good, but does not guarantee it, since again the error support does not follow a Bernoulli model. Nevertheless, as we will see, the results are visually striking.

Figure 5.2 shows face images of one subject taken from the Yale B face database [Georghiades et al., 2001a]. Here, each image has resolution 192×168; and there are a total of 58 illuminations per subject, which we stack as columns of our matrix $\boldsymbol{Y} \in \mathbb{R}^{32,256 \times 58}$. We again solve PCP with $\lambda = 1/\sqrt{n_1}$.

Figure 5.2 plots the low-rank term $\hat{\boldsymbol{L}}$ and the magnitude of the sparse term $\hat{\boldsymbol{S}}$ obtained as the solution to the convex program. The sparse term $\hat{\boldsymbol{S}}$ compensates for cast shadows and specular regions. In one example (bottom row of Figure 5.2 left), this term also compensates for errors in image ac-

quisition. These results may be useful for conditioning the training data for face recognition, as well as face alignment and tracking under illumination variations.

Simulation: phase transition in rank and sparsity.

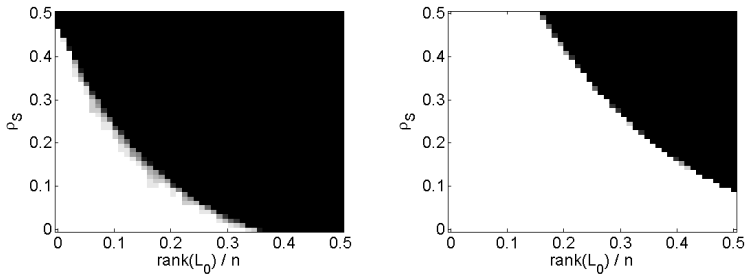
The above simulations and experiments suggest that for well-structured problem instances (datasets that indeed admit a low-rank and sparse decomposition $\boldsymbol{Y} = \boldsymbol{L}_o + \boldsymbol{S}_o$), PCP accurately recovers both $\boldsymbol{L}_o$ and $\boldsymbol{S}_o$. With this as motivation, we next systematically investigate the ability of the algorithm to recover matrices of varying rank from errors of varying sparsity. We consider square matrices of dimension $n_1 = n_2 = 400$. We generate low-rank matrices $\boldsymbol{L}_o = \boldsymbol{U}\boldsymbol{V}^*$ with $\boldsymbol{U}$ and $\boldsymbol{V}$ independently chosen $n \times r$ matrices with i.i.d. Gaussian entries of mean zero and variance $1/n$. For our first experiment, we assume a Bernoulli model for the support of the sparse term $\boldsymbol{S}_o$, with random signs: each entry of $\boldsymbol{S}_o$ takes on value 0 with probability $1 - \rho_s$, and values ± 1 each with probability $\rho_s/2$. For each (r, ρ_s) pair, we generate 10 random problem instances, each of which is solved via the ADMM Algorithm 5.1. We declare a trial to be successful if the recovered $\hat{\boldsymbol{L}}$ satisfies $\|\hat{\boldsymbol{L}} - \boldsymbol{L}_o\|_F / \|\boldsymbol{L}_o\|_F \leq 10^{-3}$.

Figure 5.3 (left) plots the fraction of correct recoveries in grey scale for each pair (r, ρ_s). Notice that there is a large white region in which the recovery is exact. This inspires us to characterize the working conditions of the algorithm in more precise terms in the next section. The simulation already highlights an interesting aspect of PCP: The recovery is correct even though in some cases $\|\boldsymbol{S}_o\|_F \gg \|\boldsymbol{L}_o\|_F$ (e.g., for $r/n = \rho_s$, $\|\boldsymbol{S}_o\|_F$ is $\sqrt{n} = 20$ times larger!). As we shall see in the next section, this is to be expected from the analysis (see Lemma 5.3.2): The optimal solution to PCP is unique and correct only depending on the signs and support of $\boldsymbol{S}_o$ and the orientation of the singular spaces of $\boldsymbol{L}_o$.

Finally, inspired by the connection between matrix completion and Robust PCA, we compare the breakdown point of PCP for the low-rank and sparse separation problem to the breakdown behavior of the nuclear-norm heuristic for matrix completion (studied in the previous chapter). By comparing the two heuristics, we can begin to answer the question *how much is gained by knowing the location* $\mathfrak{S}$ *of the corrupted entries*? Here, we again generate $\boldsymbol{L}_o$ as a product of Gaussian matrices. However, we now provide the algorithm with only an incomplete subset $\boldsymbol{M} = \mathcal{P}_{\mathfrak{S}^c}\boldsymbol{L}_o$ of its entries. Each (i, j) may be included in $\mathfrak{S}$ independently with probability $1 - \rho_s$, so rather than a probability of error, here, ρ_s stands for the probability that an entry is omitted. We solve the nuclear norm minimization problem

$$\text{minimize} \quad \|\boldsymbol{L}\|_* \quad \text{subject to} \quad \mathcal{P}_{\mathfrak{S}^c}\boldsymbol{L} = \mathcal{P}_{\mathfrak{S}^c}\boldsymbol{M}$$

using an augmented Lagrange multiplier algorithm very similar to the one discussed in the above section. We again declare $\boldsymbol{L}_o$ to be successfully recovered if $\|\hat{\boldsymbol{L}} - \boldsymbol{L}_o\|_F / \|\boldsymbol{L}_o\|_F < 10^{-3}$. Figure 5.3 (right) plots the fraction



(a) Principal Component Pursuit (b) Matrix Completion

Figure 5.3. **Correct recovery for varying rank and sparsity.** Fraction of correct recoveries across 10 trials, as a function of $\text{rank}(\boldsymbol{L}_o)$ (x-axis) and sparsity of $\boldsymbol{S}_o$ (y-axis). Here, $n_1 = n_2 = 400$. In all cases, $\boldsymbol{L}_o$ is a product of independent $n \times r$ i.i.d. $\mathcal{N}(0, 1/n)$ matrices. Trials are considered successful if $\|\hat{\boldsymbol{L}} - \boldsymbol{L}_o\|_F / \|\boldsymbol{L}_o\|_F < 10^{-3}$. Left: low-rank and sparse decomposition, $\boldsymbol{\Sigma}_o = \text{sign}(\boldsymbol{S}_o)$ random. Right: matrix completion. For matrix completion, ρ_s is the probability that an entry is omitted from the observation.

of correct recoveries for varying r, ρ_s. Notice that nuclear norm minimization successfully recovers $\boldsymbol{L}_o$ over a wider range of (r, ρ_s). The difference between breakdown points can be viewed as the price of not knowing ahead of time which entries are unreliable.

5.3 Identifiability and Exact Recovery

Simulations and real examples of the previous section suggest a similar phenomenon of the RPCA problem to Matrix Completion: when the solution is sufficiently structured (i.e. sufficient low-rank or sparse), the convex relaxation (and the associated algorithm) succeeds. Our next goal will be to understand this phenomenon at a more mathematical level and to provide a theory that delineates when the convex optimization PCP solves the RPCA problem.

5.3.1 Identifiability Conditions

At first sight, the RPCA problem (5.1.1) of separating a matrix into a low-rank one and a sparse one may seem impossible to solve. In general, there is not enough information to perfectly disentangle the low-rank and the spare components since the number of unknowns to infer $\boldsymbol{L}_o \in \mathbb{R}^{n \times n}$ and $\boldsymbol{S}_o \in \mathbb{R}^{n \times n}$ is twice as many as the observations given in $\boldsymbol{Y} \in \mathbb{R}^{n \times n}$.

Clearly, we will need both $\boldsymbol{L}_o$ and $\boldsymbol{S}_o$ to be well structured, in the sense that $\boldsymbol{L}_o$ is sufficiently low-rank, and $\boldsymbol{S}_o$ is sufficiently sparse.

However, identifiability issues arise even for very structured examples. For instance, suppose the matrix $\boldsymbol{Y}$ is equal to $\boldsymbol{e}_1\boldsymbol{e}_1^*$ (this matrix has a one in the top left corner and zeros everywhere else). Then since $\boldsymbol{Y}$ is both sparse and low-rank, how can we decide whether it is low-rank or sparse? To make the problem meaningful, we need to impose that the low-rank component $\boldsymbol{L}_o$ is *not* sparse so it can be differentiated from $\boldsymbol{S}_o$.[4]

Incoherence Conditions on $\boldsymbol{L}_o$.

In the matrix completion problem of the previous chapter (Section 4.4), we have introduced the notion of ν-incoherence to ensure that a low-rank matrix is not too sparse. Let us write the singular value decomposition of $\boldsymbol{L}_o \in \mathbb{R}^{n\times n}$ as

$$\boldsymbol{L}_o = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^* = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^*,$$

where r is the rank of the matrix, $\sigma_1, \dots, \sigma_r$ are the positive singular values, and $\boldsymbol{U} = [\boldsymbol{u}_1, \dots, \boldsymbol{u}_r]$, $\boldsymbol{V} = [\boldsymbol{v}_1, \dots, \boldsymbol{v}_r]$ are the matrices of left- and right-singular vectors. Then according to (4.4.13) and (4.4.14), $\boldsymbol{L}_o$ is ν-incoherent

$$\max_i \|\boldsymbol{e}_i^*\boldsymbol{U}\|_2^2 \le \frac{\nu r}{n}, \quad \max_j \|\boldsymbol{e}_j^*\boldsymbol{V}\|_2^2 \le \frac{\nu r}{n}. \tag{5.3.1}$$

For technical reasons that we will see later, unlike the matrix completion case, we here need a stronger notion of incoherence for the low-rank sparse separation problem. In addition to the above two incoherence conditions, we need to further require:

$$\|\boldsymbol{U}\boldsymbol{V}^*\|_\infty \le \frac{\sqrt{\nu r}}{n}. \tag{5.3.2}$$

Here and below, $\|\boldsymbol{M}\|_\infty = \max_{i,j} |\boldsymbol{M}_{ij}|$, i.e. is the ℓ^∞ norm of $\boldsymbol{M}$ seen as a long vector. This incoherence condition asserts that for small values of ν, the singular vectors are reasonably spread out.

One can show that the above incoherence conditions are not atypical and they hold with high probability for low-rank matrices that generated with random orthogonal factors $\boldsymbol{U}$ and $\boldsymbol{V}$.

Randomness of $\boldsymbol{S}_o$.

Another identifiability issue arises if the sparse matrix has low rank. This will occur if, say, all the nonzero entries of $\boldsymbol{S}_o$ occur in a column or in a few columns. Suppose for instance, that the first column of $\boldsymbol{S}_o$ is the opposite

[4] In Chapter 14, we will study the case when a matrix is simultaneously low-rank and sparse, when the goal is to recover it as a whole instead of separating low-rank and sparse components.

of that of $\boldsymbol{L}_o$, and that all the other columns of $\boldsymbol{S}_o$ vanish. Then it is clear that we would not be able to recover $\boldsymbol{L}_o$ and $\boldsymbol{S}_o$ by any method whatsoever since $\boldsymbol{Y} = \boldsymbol{L}_o + \boldsymbol{S}_o$ would have a column space equal to, or included in that of $\boldsymbol{L}_o$. To avoid such meaningless situations, we may assume that the sparsity pattern of the sparse component $\boldsymbol{S}_o$ is selected independently and identically according to a Bernoulli distribution

$$\mathfrak{S} \sim \mathrm{Ber}(\rho_s).$$

Under this model, the expected number of nonzero entries in $\boldsymbol{S}_o$ is $\mathbb{E}\left[|\mathfrak{S}|\right] = \rho_s \cdot n^2$.

Uniqueness.

The incoherence conditions only ensure that one would not confuse the low-rank component $\boldsymbol{L}_o$ with the sparse one $\boldsymbol{S}_o$, but it does not ensure the decomposition $\boldsymbol{Y} = \boldsymbol{L} + \boldsymbol{S}$ would be unique. For instance, any pair $(\boldsymbol{L}', \boldsymbol{S}')$ of the form:

$$\boldsymbol{L}' = \boldsymbol{L}_o + \boldsymbol{e}_i \boldsymbol{e}_j^*, \quad \boldsymbol{S}' = \boldsymbol{S}_o - \boldsymbol{e}_i \boldsymbol{e}_j^*$$

would be a feasible decomposition of $\boldsymbol{Y}$. The only way we could hope for a unique decomposition is to chose the pair $(\boldsymbol{L}^*, \boldsymbol{S}^*)$ that is in some sense the *simplest*. In our context, we would desire $\boldsymbol{L}^*$ to have the lowest possible rank and $\boldsymbol{S}^*$ the sparsest. Or more precisely, we wish to minimize certain measure of "simplicity" that encourages a decomposition such that $\boldsymbol{L}$ is low-rank and $\boldsymbol{S}$ is sparse. Thus, if the ground truth is such that the rank of $\boldsymbol{L}_o$ is low enough and $\boldsymbol{S}_o$ is sparse enough, then they will be the only optimal solution that minimizes such a measure. In this section, we will try to show that for a properly chosen $\lambda \in \mathbb{R}_+$

$$\|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1$$

is precisely such a measure of model simplicity.

Similar to the case with recovering a sparse vector (Chapter 3) or with recovering a low-rank matrix (Chapter 4), could we expect that under reasonable conditions, the above convex program PCP can actually recover the correct low-rank $\boldsymbol{L}_o$ and sparse $\boldsymbol{S}_o$?

In fact, under the minimal conditions discussed in the identifiability section above, the solution to the convex PCP program exactly recovers the low-rank and sparse components, provided that the rank of $\boldsymbol{L}_o$ is not too large and $\boldsymbol{S}_o$ is reasonably sparse. To be more precise, the following statement is true:

Theorem 5.3.1 (Principal Component Pursuit)**.** *Suppose $\boldsymbol{L}_o$ is $n \times n$, obeys (5.3.1)–(5.3.2). Suppose that the support $\mathfrak{S}$ of $\boldsymbol{S}_o$ follows the Bernoulli model with parameter $\rho < \rho_s$, and that the signs of the nonzero entries of $\boldsymbol{S}_o$ are chosen independently from the uniform distribution on $\{\pm 1\}$. Then, there is a numerical constant C such that with probability at*

least $1 - Cn^{-10}$ *(over the choice of signs and support of* $\boldsymbol{S}_o$*), PCP* (5.2.4) *with* $\lambda = 1/\sqrt{n}$ *is* exact*, i.e.* $\hat{\boldsymbol{L}} = \boldsymbol{L}_o$ *and* $\hat{\boldsymbol{S}} = \boldsymbol{S}_o$*, provided that*

$$\operatorname{rank}(\boldsymbol{L}_o) \leq C_r \frac{n}{\nu \log^2 n}. \tag{5.3.3}$$

Above, C_r *and* ρ_s *are positive numerical constants.*

Extension: Non-square matrices.

In the general rectangular case where $\boldsymbol{L}_o$ is $n_1 \times n_2$, PCP with $\lambda = 1/\sqrt{n_{(1)}}$ succeeds with probability at least $1 - cn_{(1)}^{-10}$, provided that $\operatorname{rank}(\boldsymbol{L}_o) \leq \rho_r n_{(2)}\, \nu^{-1} (\log n_{(1)})^{-2}$ and $m \leq \rho_s\, n_1 n_2$. A rather remarkable fact is that there is no tuning parameter in solving the PCP program. Under the assumption of the theorem, minimizing

$$\|\boldsymbol{L}\|_* + \frac{1}{\sqrt{n_{(1)}}} \|\boldsymbol{S}\|_1, \quad n_{(1)} = \max(n_1, n_2)$$

always returns the correct answer. This is surprising because one might have expected that one would have to choose the right scalar λ to balance the two terms in $\|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1$ appropriately (perhaps depending on their relative size). This is, however, clearly not the case. In this sense, the choice $\lambda = 1/\sqrt{n_{(1)}}$ is universal. Further, it is not a priori very clear why $\lambda = 1/\sqrt{n_{(1)}}$ is a correct choice no matter what $\boldsymbol{L}_o$ and $\boldsymbol{S}_o$ are. It is the mathematical analysis which reveals the correctness of this value. In fact, the proof of the theorem gives a whole range of correct values, and we have selected a sufficiently simple value in that range.

5.3.2 Correctness of Principal Component Pursuit

In this section, we prove Theorem 5.3.1.

Dual certificates for optimality.

As for each optimization problem we have encountered thus far, we begin by writing down an optimality condition. To prove that the target pair $(\boldsymbol{L}_o, \boldsymbol{S}_o)$ is the unique optimal solution to the convex program, we then must prove that under our assumptions, this condition is satisfied with high probability.

The key tool for obtaining optimality conditions is the KKT conditions of convex optimization; these conditions are naturally phrased in terms of the subdifferential of the objective function. Recall the sub differentials of the ℓ^1 norm

$$\partial \|\cdot\|_1 (\boldsymbol{S}_o) = \{\boldsymbol{\Sigma}_o + \boldsymbol{F} \mid \mathcal{P}_{\Theta} \boldsymbol{F} = \boldsymbol{0},\ \|\boldsymbol{F}\|_\infty \leq 1\}, \tag{5.3.4}$$

and the nuclear norm

$$\partial \|\cdot\|_* (\boldsymbol{L}_o) = \{\boldsymbol{U}\boldsymbol{V}^* + \boldsymbol{W} \mid \mathcal{P}_T \boldsymbol{W} = \boldsymbol{0},\ \|\boldsymbol{W}\| \leq 1\}. \tag{5.3.5}$$

Here, $\boldsymbol{U}$ and $\boldsymbol{V}$ are matrices of left and right singular vectors of $\boldsymbol{L}_o$, corresponding to nonzero singular values, and

$$T \doteq \left\{ \boldsymbol{U}\boldsymbol{R}^* + \boldsymbol{Q}\boldsymbol{V}^* \mid \boldsymbol{R}, \boldsymbol{Q} \in \mathbb{R}^{n \times r} \right\}$$

is the tangent space to the variety of rank r matrices at $\boldsymbol{L}_o$.

To the optimization problem

$$\min_{\boldsymbol{L},\boldsymbol{S}} \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 \quad \text{s.t.} \quad \boldsymbol{L} + \boldsymbol{S} = \boldsymbol{Y}, \tag{5.3.6}$$

associate a matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{n \times n}$ of Lagrange multipliers, and the Lagrangian

$$\mathcal{L}(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) = \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}, \boldsymbol{Y} - \boldsymbol{L} - \boldsymbol{S} \rangle . \tag{5.3.7}$$

The KKT conditions imply that $(\boldsymbol{L}_\star, \boldsymbol{S}_\star)$ are optimal if there exists $\boldsymbol{\Lambda}_\star$ such that $\boldsymbol{0} = \partial_{\boldsymbol{L}} \mathcal{L}(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda})$ *and* $\boldsymbol{0} \in \partial_{\boldsymbol{S}} \mathcal{L}(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda})$. Thus,

$$\boldsymbol{\Lambda} \in \partial \|\cdot\|_* (\boldsymbol{L}_\star) \quad \text{and} \quad \boldsymbol{\Lambda} \in \lambda \partial \|\cdot\|_1 (\boldsymbol{S}_\star). \tag{5.3.8}$$

To show optimality, *it is enough to find a matrix* $\boldsymbol{\Lambda}$ *that is in both the subdifferential of the nuclear norm, and the subdifferential of the* ℓ^1 *norm, at the same time.*

From the KKT conditions to usable optimality conditions.

Although the KKT conditions are a useful guide, the form that we have derived is neither strong enough nor robust enough for our purposes. We need to strengthen them to guarantee *unique* optimality, so that we can eventually ensure that the true pair $(\boldsymbol{L}_o, \boldsymbol{S}_o)$ is the only solution to the PCP problem. Moreover, as in matrix completion, it will be easier to demonstrate that a modified condition is satisfied, in which we merely guarantee that there exists $\boldsymbol{\Lambda}$ which is *close to* the two subdifferentials, rather than lying exactly within them.

We introduce a simple condition for the pair $(\boldsymbol{L}_o, \boldsymbol{S}_o)$ to be the unique optimal solution to PCP. These conditions, given in the following lemma, are stated in terms of a dual vector, the existence of which certifies optimality.

Lemma 5.3.2 (Unique Optimality)**.** *Assume that* $\|\mathcal{P}_{\mathfrak{S}} \mathcal{P}_T\| < 1$ *(or equivalently* $\mathfrak{S} \cap T = \{\boldsymbol{0}\}$*). Then* $(\boldsymbol{L}_o, \boldsymbol{S}_o)$ *is the unique optimal solution to the PCP problem if there exists* $\boldsymbol{\Lambda}$ *such that*

$$\textit{[Subdifferential of } \|\cdot\|_*\textit{]:} \quad \mathcal{P}_T \boldsymbol{\Lambda} = \boldsymbol{U}\boldsymbol{V}^*, \quad \|\mathcal{P}_{T^\perp} \boldsymbol{\Lambda}\| < 1, \tag{5.3.9}$$

and

$$\textit{[Subdifferential of } \lambda \|\cdot\|_1\textit{]:} \quad \mathcal{P}_{\mathfrak{S}} \boldsymbol{\Lambda} = \lambda \boldsymbol{\Sigma}_o, \quad \|\mathcal{P}_{\mathfrak{S}^c} \boldsymbol{\Lambda}\|_\infty < \lambda. \tag{5.3.10}$$

There are two aspects of this lemma which deserve comment. First, compared to the KKT condition, it has the extra requirement that $\|\mathcal{P}_{\mathfrak{S}} \mathcal{P}_T\| < 1$. This condition means that the subspace of matrices supported on $\mathfrak{S}$ does not intersect the tangent space T to the low-rank matrices at $\boldsymbol{L}_o$. Second,

compared to the KKT condition, which just requires that $\boldsymbol{\Lambda}$ lie in the subdifferentials of $\|\cdot\|_*$ and $\lambda\|\cdot\|_1$, this condition requires that $\boldsymbol{\Lambda}$ lie within the *relative interiors* of these two sets, by requiring $\|\mathcal{P}_{T^\perp}\boldsymbol{\Lambda}\|$ to be strictly less than one and $\|\mathcal{P}_{\mathfrak{S}^c}\boldsymbol{\Lambda}\|_\infty$ to be strictly less than λ. Under these stronger conditions, we can guarantee that $(\boldsymbol{L}_o, \boldsymbol{S}_o)$ is the *unique* optimal solution to the PCP problem.

Proof. We consider a feasible perturbation $(\boldsymbol{L}_o + \boldsymbol{H}, \boldsymbol{S}_o - \boldsymbol{H})$ and show that the objective increases whenever $\boldsymbol{H} \neq \boldsymbol{0}$, hence proving that $(\boldsymbol{L}_o, \boldsymbol{S}_o)$ is the unique optimal solution. To do this, let $\mathcal{P}_{T^\perp}\boldsymbol{H} = \bar{\boldsymbol{U}}\bar{\boldsymbol{\Sigma}}\bar{\boldsymbol{V}}^*$ denote the reduced singular value decomposition of $\mathcal{P}_{T^\perp}\boldsymbol{H}$. Set $\boldsymbol{W} \doteq \bar{\boldsymbol{U}}\bar{\boldsymbol{V}}^* \in T^\perp$, and notice that

$$\langle \boldsymbol{W}, \boldsymbol{H}\rangle = \langle \boldsymbol{W}, \mathcal{P}_{T^\perp}\boldsymbol{H}\rangle = \|\mathcal{P}_{T^\perp}\boldsymbol{H}\|_*. \tag{5.3.11}$$

Note further that $\boldsymbol{U}\boldsymbol{V}^* + \boldsymbol{W} \in \partial\|\cdot\|_*(\boldsymbol{L}_o)$.

Similarly, set $\boldsymbol{F} \doteq -\mathrm{sign}(\mathcal{P}_{\mathfrak{S}^c}\boldsymbol{H})$, notice that $\lambda(\boldsymbol{\Sigma}_o + \boldsymbol{F}) \in \partial\lambda\|\cdot\|_1(\boldsymbol{S}_o)$, and that $-\lambda\langle \boldsymbol{F}, \boldsymbol{H}\rangle = \lambda\|\mathcal{P}_{\mathfrak{S}^c}\boldsymbol{H}\|_1$.

Using the subgradient inequality for both $\|\cdot\|_*$ and $\lambda\|\cdot\|_1$, we obtain that

$$\begin{aligned}
&\|\boldsymbol{L}_o + \boldsymbol{H}\|_* + \lambda\|\boldsymbol{S}_o - \boldsymbol{H}\|_1 \\
&\geq \|\boldsymbol{L}_o\|_* + \lambda\|\boldsymbol{S}_o\|_1 + \langle \boldsymbol{U}\boldsymbol{V}^* + \boldsymbol{W}, \boldsymbol{H}\rangle - \lambda\langle \mathrm{sgn}(\boldsymbol{S}_o) + \boldsymbol{F}, \boldsymbol{H}\rangle \\
&= \|\boldsymbol{L}_o\|_* + \lambda\|\boldsymbol{S}_o\|_1 + \|\mathcal{P}_{T^\perp}\boldsymbol{H}\|_* + \lambda\|\mathcal{P}_{\mathfrak{S}^c}\boldsymbol{H}\|_1 + \langle \boldsymbol{U}\boldsymbol{V}^* - \lambda\mathrm{sgn}(\boldsymbol{S}_o), \boldsymbol{H}\rangle, \\
&= \|\boldsymbol{L}_o\|_* + \lambda\|\boldsymbol{S}_o\|_1 + \|\mathcal{P}_{T^\perp}\boldsymbol{H}\|_* + \lambda\|\mathcal{P}_{\mathfrak{S}^c}\boldsymbol{H}\|_1 + \langle \mathcal{P}_T[\boldsymbol{\Lambda}] - \lambda\mathcal{P}_{\mathfrak{S}}[\boldsymbol{\Lambda}], \boldsymbol{H}\rangle \\
&= \|\boldsymbol{L}_o\|_* + \lambda\|\boldsymbol{S}_o\|_1 + \|\mathcal{P}_{T^\perp}\boldsymbol{H}\|_* + \lambda\|\mathcal{P}_{\mathfrak{S}^c}\boldsymbol{H}\|_1 + \langle \mathcal{P}_{T^\perp}[\boldsymbol{\Lambda}] - \lambda\mathcal{P}_{\mathfrak{S}^c}[\boldsymbol{\Lambda}], \boldsymbol{H}\rangle \\
&\geq \|\boldsymbol{L}_o\|_* + \lambda\|\boldsymbol{S}_o\|_1 + \|\mathcal{P}_{T^\perp}\boldsymbol{H}\|_* + \lambda\|\mathcal{P}_{\mathfrak{S}^c}\boldsymbol{H}\|_1 \\
&\quad - \|\mathcal{P}_{T^\perp}[\boldsymbol{\Lambda}]\|\,\|\mathcal{P}_{T^\perp}\boldsymbol{H}\|_* - \lambda\|\mathcal{P}_{\mathfrak{S}^c}[\boldsymbol{\Lambda}]\|_\infty\|\mathcal{P}_{\mathfrak{S}^c}\boldsymbol{H}\|_1 \\
&\geq \|\boldsymbol{L}_o\|_* + \lambda\|\boldsymbol{S}_o\|_1 + (1-\beta)\{\|\mathcal{P}_{T^\perp}\boldsymbol{H}\|_* + \lambda\|\mathcal{P}_{\mathfrak{S}^c}\boldsymbol{H}\|_1\},
\end{aligned}$$

where $\beta = \max\{\|\mathcal{P}_{T^\perp}\boldsymbol{\Lambda}\|, \lambda^{-1}\|\mathcal{P}_{\mathfrak{S}^c}\boldsymbol{\Lambda}\|_\infty\} < 1$. Since by assumption, $\mathfrak{S} \cap T = \{\boldsymbol{0}\}$, we have $\|\mathcal{P}_{T^\perp}\boldsymbol{H}\|_* + \lambda\|\mathcal{P}_{\mathfrak{S}^c}\boldsymbol{H}\|_1 > 0$ unless $\boldsymbol{H} = \boldsymbol{0}$. □

This lemma gives a sufficient condition for $(\boldsymbol{L}_o, \boldsymbol{S}_o)$ to be the *unique* optimal solution. It is still challenging to work with, because it demands that $\boldsymbol{\Lambda}$ is an element of both $\partial\|\cdot\|_*(\boldsymbol{L}_o)$ and $\partial\lambda\|\cdot\|_1(\boldsymbol{S}_o)$. This forces $\boldsymbol{\Lambda}$ to exactly satisfy the equalities $\mathcal{P}_T\boldsymbol{\Lambda} = \boldsymbol{U}\boldsymbol{V}^*$ and $\mathcal{P}_{\mathfrak{S}}\boldsymbol{\Lambda} = \lambda\boldsymbol{\Sigma}_o$. As we did for matrix completion, it will be helpful to state a modified optimality condition, which accepts $\boldsymbol{\Lambda}$ that satisfied these equalities *approximately*. We state this new condition as follows:

Lemma 5.3.3. *Assume $\|\mathcal{P}_{\mathfrak{S}}\mathcal{P}_T\| \leq 1/2$ and $\lambda < 1$. Then with the same notation, $(\boldsymbol{L}_o, \boldsymbol{S}_o)$ is the unique solution if there exists $\boldsymbol{\Lambda}$ such that*

$$\textit{[Approx. subgradient of } \|\cdot\|_*\textit{]:} \quad \|\mathcal{P}_T\boldsymbol{\Lambda} - \boldsymbol{U}\boldsymbol{V}^*\|_F \leq \tfrac{\lambda}{8}, \quad \|\mathcal{P}_{T^\perp}\boldsymbol{\Lambda}\| < \tfrac{1}{2}, \tag{5.3.12}$$

and

$$\textit{[Approx. subgradient of } \lambda\|\cdot\|_1\textit{]:} \quad \|\mathcal{P}_{\mathfrak{S}}\boldsymbol{\Lambda} - \lambda\boldsymbol{\Sigma}_o\|_F \le \tfrac{\lambda}{8}, \quad \|\mathcal{P}_{\mathfrak{S}^c}\boldsymbol{\Lambda}\|_\infty < \tfrac{\lambda}{2}. \tag{5.3.13}$$

Proof. Consider any nonzero $\boldsymbol{H} \in \mathbb{R}^{n\times n}$. We demonstrate that in a particular sense, $\boldsymbol{H}$ cannot be simultaneously concentrated on T and $\mathfrak{S}$. Observe that

$$\begin{aligned}\|\mathcal{P}_{\mathfrak{S}}\boldsymbol{H}\|_F &\le \|\mathcal{P}_{\mathfrak{S}}\mathcal{P}_T\boldsymbol{H}\|_F + \|\mathcal{P}_{\mathfrak{S}}\mathcal{P}_{T^\perp}\boldsymbol{H}\|_F \\ &\le \tfrac{1}{2}\|\boldsymbol{H}\|_F + \|\mathcal{P}_{T^\perp}\boldsymbol{H}\|_F \\ &\le \tfrac{1}{2}\|\mathcal{P}_{\mathfrak{S}}\boldsymbol{H}\|_F + \tfrac{1}{2}\|\mathcal{P}_{\mathfrak{S}^c}\boldsymbol{H}\|_F + \|\mathcal{P}_{T^\perp}\boldsymbol{H}\|_F,\end{aligned}$$

and, therefore,

$$\|\mathcal{P}_{\mathfrak{S}}\boldsymbol{H}\|_F \le \|\mathcal{P}_{\mathfrak{S}^c}\boldsymbol{H}\|_F + 2\|\mathcal{P}_{T^\perp}\boldsymbol{H}\|_F.$$

Symmetric reasoning establishes that

$$\|\mathcal{P}_T\boldsymbol{H}\|_F \le \|\mathcal{P}_{T^\perp}\boldsymbol{H}\|_F + 2\,\|\mathcal{P}_{\mathfrak{S}^c}\boldsymbol{H}\|_F\,. \tag{5.3.14}$$

With these observations in hand, we proceed in a similar spirit to the proof of Lemma 5.3.2. Notice that

$$\begin{aligned}\boldsymbol{U}\boldsymbol{V}^* &= \mathcal{P}_T\boldsymbol{\Lambda} + (\boldsymbol{U}\boldsymbol{V}^* - \mathcal{P}_T\boldsymbol{\Lambda}) \\ &= \boldsymbol{\Lambda} - \mathcal{P}_{T^\perp}\boldsymbol{\Lambda} + (\boldsymbol{U}\boldsymbol{V}^* - \mathcal{P}_T\boldsymbol{\Lambda})\,, \qquad (5.3.15)\\ \lambda\boldsymbol{\Sigma}_o &= \mathcal{P}_{\mathfrak{S}}\boldsymbol{\Lambda} + (\lambda\boldsymbol{\Sigma}_o - \mathcal{P}_{\mathfrak{S}}\boldsymbol{\Lambda}) \\ &= \boldsymbol{\Lambda} - \mathcal{P}_{\mathfrak{S}^c}\boldsymbol{\Lambda} + (\lambda\boldsymbol{\Sigma}_o - \mathcal{P}_{\mathfrak{S}}\boldsymbol{\Lambda})\,, \qquad (5.3.16)\end{aligned}$$

and so

$$\boldsymbol{U}\boldsymbol{V}^* - \lambda\boldsymbol{\Sigma}_o \;=\; -\mathcal{P}_{T^\perp}\boldsymbol{\Lambda} + \mathcal{P}_{\mathfrak{S}^c}\boldsymbol{\Lambda} \;+\; (\boldsymbol{U}\boldsymbol{V}^* - \mathcal{P}_T\boldsymbol{\Lambda}) \;+\; (\lambda\boldsymbol{\Sigma}_o - \mathcal{P}_{\mathfrak{S}}\boldsymbol{\Lambda})\,.$$

Following the proof of Lemma 5.3.2, we have

$$\begin{aligned}&\|\boldsymbol{L}_o + \boldsymbol{H}\|_* + \lambda\|\boldsymbol{S}_o - \boldsymbol{H}\|_1 \\ &\quad\ge \|\boldsymbol{L}_o\|_* + \lambda\,\|\boldsymbol{S}_o\|_1 + \|\mathcal{P}_{T^\perp}\boldsymbol{H}\|_* + \lambda\,\|\mathcal{P}_{\mathfrak{S}^c}\boldsymbol{H}\|_1 + \langle \boldsymbol{U}\boldsymbol{V}^* - \lambda\boldsymbol{\Sigma}_o, \boldsymbol{H}\rangle \\ &\quad\ge \|\boldsymbol{L}_o\|_* + \lambda\|\boldsymbol{S}_o\|_1 + \tfrac{1}{2}\Big(\|\mathcal{P}_{T^\perp}\boldsymbol{H}\|_* + \lambda\|\mathcal{P}_{\mathfrak{S}^c}\boldsymbol{H}\|_1\Big) \\ &\qquad -\tfrac{\lambda}{8}\,\|\mathcal{P}_T\boldsymbol{H}\|_F - \tfrac{\lambda}{8}\,\|\mathcal{P}_{\mathfrak{S}}\boldsymbol{H}\|_F \\ &\quad\ge \|\boldsymbol{L}_o\|_* + \lambda\,\|\boldsymbol{S}_o\|_1 + \underbrace{\left(\tfrac{1}{2} - \tfrac{\lambda}{8} - \tfrac{\lambda}{4}\right)}_{\ge 1/8}\|\mathcal{P}_{T^\perp}\boldsymbol{H}\|_* + \underbrace{\left(\tfrac{\lambda}{2} - \tfrac{\lambda}{4} - \tfrac{\lambda}{8}\right)}_{\ge \lambda/8}\|\mathcal{P}_{\mathfrak{S}^c}\boldsymbol{H}\|_1 \\ &\quad> \|\boldsymbol{L}_o\|_* + \lambda\,\|\boldsymbol{S}_o\|_1\,, \qquad (5.3.17)\end{aligned}$$

where the final (strict) inequality holds because $\boldsymbol{H} \neq \boldsymbol{0}$ and $\mathfrak{S} \cap T = \{\boldsymbol{0}\}$. □

Showing that the optimality conditions can be satisfied.

We next show that under our conditions, with high probability the conditions of Lemma 5.3.3 can be satisfied. To do this, we have to show two things:

1. $\|\mathcal{P}_{\mathfrak{S}}\mathcal{P}_T\| < 1/2$;
2. existence of a near dual certificate $\mathbf{\Lambda}$ as in Lemma 5.3.3.

Let $\Omega = \mathfrak{S}^c$. These are the *clean* entries. Notice that if $\mathfrak{S} \sim \text{Ber}(\rho_s)$, $\Omega \sim \text{Ber}(1-\rho_s)$. We are going to show 1 and 2 by building on machinery developed in Chapter 4 for *matrix completion*. In particular, in that chapter we showed that if

$$\rho_{\text{clean}} = 1 - \rho_s > C_0 \frac{\nu r \log n}{n}, \tag{5.3.18}$$

with high probability

$$\left\| \mathcal{P}_T - \rho_{\text{clean}}^{-1} \mathcal{P}_T \mathcal{P}_{\mathfrak{S}^c} \mathcal{P}_T \right\| \ < \ \tfrac{1}{8}. \tag{5.3.19}$$

Under this condition,

$$\begin{aligned} \|\mathcal{P}_T \mathcal{P}_{\mathfrak{S}} \mathcal{P}_T\| &= \|\mathcal{P}_T - \mathcal{P}_T \mathcal{P}_{\mathfrak{S}^c} \mathcal{P}_T\| \\ &\le \|\rho_{\text{clean}} \mathcal{P}_T - \mathcal{P}_T \mathcal{P}_{\mathfrak{S}^c} \mathcal{P}_T\| + \|(1-\rho_{\text{clean}})\mathcal{P}_T\| \\ &= \rho_{\text{clean}} \left\| \mathcal{P}_T - \rho_{\text{clean}}^{-1} \mathcal{P}_T \mathcal{P}_{\mathfrak{S}^c} \mathcal{P}_T \right\| + 1 - \rho_{\text{clean}} \\ &\le \tfrac{\rho_{\text{clean}}}{8} + 1 - \rho_{\text{clean}} \\ &< \tfrac{1}{4}, \end{aligned} \tag{5.3.20}$$

provided $\rho_{\text{clean}} > \frac{6}{7}$. This implies that

$$\|\mathcal{P}_{\mathfrak{S}} \mathcal{P}_T\| \ = \ \|\mathcal{P}_T \mathcal{P}_{\mathfrak{S}} \mathcal{P}_T\|^{1/2} \ \le \ \tfrac{1}{2}. \tag{5.3.21}$$

This establishes statement 1. By exactly the same reasoning, for any constant $\sigma > 0$, there exists a constant $\rho_{\text{clean},\star}(\sigma) < 1$ such that if $\rho_{\text{clean}} > \rho_{\text{clean},\star}$, with high probability $\|\mathcal{P}_{\mathfrak{S}} \mathcal{P}_T\| < \sigma$.

Constructing the certificate $\mathbf{\Lambda}$.

To show that $(\boldsymbol{L}_o, \boldsymbol{S}_o)$ is the unique optimal solution, we further need to establish statement 2. That is, there exists a matrix $\mathbf{\Lambda}$ that is simultaneously close to the subdifferential $\partial \|\cdot\|_* (\boldsymbol{L}_o)$ and the subdifferential $\partial \lambda \|\cdot\|_1 (\boldsymbol{S}_o)$ as in Lemma 5.3.3.

In the previous paragraph, we saw that the clean entries $\Omega = \mathfrak{S}^c$ are distributed as a Bernoulli subset, with parameter

$$\rho_{\text{clean}} \doteq 1 - \rho_s. \tag{5.3.22}$$

This is exactly the same model of randomness as in our analysis of matrix completion! We use this fact as a starting point for our construction.

Proposition 4.4.8 implies that as long as the rank of $\boldsymbol{L}_o$ is not too large, i.e.,

$$r < \frac{\rho_{\text{clean}} n}{C_0 \nu \log^2 n}, \tag{5.3.23}$$

with high probability there exists a matrix $\boldsymbol{\Lambda_L}$ supported only on the clean set Ω satisfying

(i) $\|\mathcal{P}_T \boldsymbol{\Lambda_L} - \boldsymbol{U}\boldsymbol{V}^*\|_F \leq \frac{1}{4n}$,

(ii) $\|\mathcal{P}_{T^\perp} \boldsymbol{\Lambda_L}\| \leq \frac{1}{4}$,

(iii) $\|\boldsymbol{\Lambda_L}\|_\infty < \frac{C \log n}{\rho_{\text{clean}}} \|\boldsymbol{U}\boldsymbol{V}^*\|_\infty$.

This certificate $\boldsymbol{\Lambda_L}$ lies close enough to the subdifferential of the nuclear norm. Furthermore, let us further verify that it satisfies the condition $\|\mathcal{P}_{\mathfrak{S}^c} \boldsymbol{\Lambda_L}\|_\infty < \frac{\lambda}{2}$ in Lemma 5.3.3. This is because $\boldsymbol{U}\boldsymbol{V}^*$ is ν-incoherent: from (5.3.2), $\|\boldsymbol{U}\boldsymbol{V}^*\|_\infty \leq \frac{\sqrt{\nu r}}{n}$ and with the assumption on the rank r of the matrix $\boldsymbol{L}_o$, we have

$$\|\boldsymbol{\Lambda_L}\|_\infty < \frac{C \log n}{\rho_{\text{clean}}} \frac{\sqrt{\nu r}}{n} \leq \frac{C}{\sqrt{C_0 \rho_{\text{clean}} \nu}} \frac{1}{\sqrt{n}} = \frac{C}{\sqrt{\rho_{\text{clean}} C_0 \nu}} \lambda. \tag{5.3.24}$$

By properly choosing the constant C_0 and C we could make the coefficient $\frac{C}{\sqrt{\rho_{\text{clean}} C_0 \nu}} < 1/2$.

But $\boldsymbol{\Lambda_L}$ is not yet close to the the subdifferential of the ℓ^1 norm – in particular, elements of the subdifferential of the ℓ^1 norm should satisfy $\mathcal{P}_{\mathfrak{S}} \boldsymbol{\Lambda} = \lambda \boldsymbol{\Sigma}_o$, but $\mathcal{P}_{\mathfrak{S}} \boldsymbol{\Lambda_L} = \boldsymbol{0}$. To correct this, we choose

$$\boldsymbol{\Lambda} = \boldsymbol{\Lambda_L} + \boldsymbol{\Lambda_S},$$

where the second element $\boldsymbol{\Lambda_S}$ satisfies $\mathcal{P}_{\mathfrak{S}} \boldsymbol{\Lambda_S} = \lambda \boldsymbol{\Sigma}_o$. We need to show that we can choose $\boldsymbol{\Lambda_S}$ such that this combined certificate $\boldsymbol{\Lambda}$ remains close to the subdifferential of the nuclear norm at $\boldsymbol{L}_o$, and is also close to the subdifferential of $\lambda \|\cdot\|_1$ at $\boldsymbol{S}_o$. The following lemma shows that this is possible:

Lemma 5.3.4. *Under the conditions of the Theorem 5.3.1, with high probability, there exists* $\boldsymbol{\Lambda_S}$ *such that*

(i) $\mathcal{P}_{\mathfrak{S}} \boldsymbol{\Lambda_S} = \lambda \boldsymbol{\Sigma}_o$,

(ii) $\|\mathcal{P}_{\mathfrak{S}^c} \boldsymbol{\Lambda_S}\|_\infty < \frac{\lambda}{4}$,

(iii) $\mathcal{P}_T \boldsymbol{\Lambda_S} = \boldsymbol{0}$,

(iv) $\|\mathcal{P}_{T^\perp} \boldsymbol{\Lambda_S}\| < \frac{1}{4}$.

Under the conditions of the theorem, we have in total for $\mathbf{\Lambda} = \mathbf{\Lambda_L} + \mathbf{\Lambda_S}$:

$$\begin{aligned}
\|\mathcal{P}_T \mathbf{\Lambda} - \boldsymbol{U}\boldsymbol{V}^*\|_F &= \|\mathcal{P}_T \mathbf{\Lambda_L} - \boldsymbol{U}\boldsymbol{V}^*\|_F \leq \frac{1}{4n} && (5.3.25)\\
\|\mathcal{P}_{T^\perp} \mathbf{\Lambda}\| &\leq \|\mathcal{P}_{T^\perp} \mathbf{\Lambda_L}\| + \|\mathcal{P}_{T^\perp} \mathbf{\Lambda_S}\| \leq \frac{1}{2} && (5.3.26)\\
\mathcal{P}_{\mathfrak{S}} \mathbf{\Lambda} &= \mathcal{P}_{\mathfrak{S}} \mathbf{\Lambda_S} = \lambda \mathbf{\Sigma}_o && (5.3.27)\\
\|\mathcal{P}_{\mathfrak{S}^c} \mathbf{\Lambda}\|_\infty &\leq \|\mathcal{P}_{\mathfrak{S}^c} \mathbf{\Lambda_L}\|_\infty + \|\mathcal{P}_{\mathfrak{S}^c} \mathbf{\Lambda_S}\|_\infty \\
&\leq \|\mathbf{\Lambda_L}\|_\infty + \frac{\lambda}{4} \\
&\leq \frac{C \log n}{\rho_{\text{clean}}} \sqrt{\frac{\nu r}{n}} + \frac{\lambda}{4} \\
&\leq \frac{\lambda}{2}, && (5.3.28)
\end{aligned}$$

where in the final inequality we have used that $\lambda > c/\sqrt{n}$, $\rho_{\text{clean}} > c$ and

$$r < c \frac{n}{\nu \log^2 n}. \tag{5.3.29}$$

Constructing the dual certificate $\mathbf{\Lambda_S}$ using least squares.

To finish our proof, we need to verify Lemma 5.3.4, by showing that we can indeed construct $\mathbf{\Lambda_S}$ that satisfies the requisite properties. To do this, we resort to a strategy that has proved useful at several points over the past few chapters: the method of least squares (minimum energy). Namely, we choose $\mathbf{\Lambda_S}$ to satisfy the constraints $\mathcal{P}_{\mathfrak{S}} \mathbf{\Lambda_S} = \lambda \mathbf{\Sigma}_o$ and $\mathcal{P}_T \mathbf{\Lambda_S} = \mathbf{0}$, but have the smallest possible energy: formally,

$$\mathbf{\Lambda_S} = \arg\min_{\tilde{\mathbf{\Lambda}}} \left\| \tilde{\mathbf{\Lambda}} \right\|_F^2 \quad \text{s.t.} \quad \mathcal{P}_{\mathfrak{S}} \tilde{\mathbf{\Lambda}} = \lambda \mathbf{\Sigma}_o,\ \mathcal{P}_T \tilde{\mathbf{\Lambda}} = \mathbf{0}. \tag{5.3.30}$$

This optimization problem is feasible, provided $\mathfrak{S} \cap T = \{\mathbf{0}\}$. The constraints ensure that $\mathbf{\Lambda_S}$ satisfies criteria (i) and (iii) of Lemma 5.3.4 automatically.

To check that criteria (ii) and (iv) are satisfied, i.e., that $\mathcal{P}_{\mathfrak{S}^c} \mathbf{\Lambda_S}$ has small ℓ^∞ norm and $\mathcal{P}_{T^\perp} \mathbf{\Lambda_S}$ has small operator norm, we utilize the scalar and operator Bernstein inequalities, respectively. These calculations are facilitated by the existence of a closed-form solution to (5.3.30):

$$\mathbf{\Lambda_S} = \lambda \mathcal{P}_{T^\perp} \sum_{k=0}^{\infty} \left(\mathcal{P}_{\mathfrak{S}} \mathcal{P}_T \mathcal{P}_{\mathfrak{S}} \right)^k \mathbf{\Sigma}_o. \tag{5.3.31}$$

Exercise 5.4 asks you to check that this construction indeed satisfies the constraints, and that it is indeed the solution to the energy minimization problem (5.3.30).

Proof. (of Lemma 5.3.4). Let $\mathcal{E}$ be the event that $\|\mathcal{P}_T\mathcal{P}_{\mathfrak{S}}\| \leq \sigma$. This holds with high probability in the support set $\mathfrak{S}$. Notice that on the event $\mathcal{E}$,

$$\sum_{k=0}^{\infty} \left\| (\mathcal{P}_S\mathcal{P}_T\mathcal{P}_S)^k \right\| \leq \sum_{k=0}^{\infty} \sigma^{2k} = \frac{1}{1-\sigma^{2k}} < \infty. \tag{5.3.32}$$

So, on $\mathcal{E}$, the summation in (5.3.31) converges, and

$$\boldsymbol{\Lambda}_{\boldsymbol{S}} = \lambda \mathcal{P}_{T^\perp} \sum_{k=0}^{\infty} (\mathcal{P}_{\mathfrak{S}}\mathcal{P}_T\mathcal{P}_{\mathfrak{S}})^k \boldsymbol{\Sigma}_o \tag{5.3.33}$$

is well-defined. Property (iii), which states that $\mathcal{P}_T\boldsymbol{\Lambda}_{\boldsymbol{S}} = \mathbf{0}$ follows immediately, since $\mathcal{P}_T\mathcal{P}_{T^\perp} = 0$. Property (i), which states that $\mathcal{P}_{\mathfrak{S}}\boldsymbol{\Lambda}_{\boldsymbol{S}} = \lambda\boldsymbol{\Sigma}_o$, is a consequence of the construction of $\boldsymbol{\Lambda}_{\boldsymbol{S}}$ as the solution to a least squares problem (5.3.31). To verify this property, we can note that

$$\begin{aligned} \mathcal{P}_{\mathfrak{S}}\boldsymbol{\Lambda}_{\boldsymbol{S}} &= \lambda \sum_{k=0}^{\infty} (\mathcal{P}_{\mathfrak{S}}\mathcal{P}_T\mathcal{P}_{\mathfrak{S}})^k \boldsymbol{\Sigma}_o - \lambda \sum_{k=1}^{\infty} (\mathcal{P}_{\mathfrak{S}}\mathcal{P}_T\mathcal{P}_{\mathfrak{S}})^k \boldsymbol{\Sigma}_o \\ &= \lambda\boldsymbol{\Sigma}_o, \end{aligned} \tag{5.3.34}$$

as desired. Properties (iv) and (ii) state that $\boldsymbol{\Lambda}_{\boldsymbol{S}}$ is small, in two appropriate senses. These require a bit more work.

Verifying (iv)**.** Write

$$\boldsymbol{\Lambda}_{\boldsymbol{S}} = \underbrace{\lambda\mathcal{P}_{T^\perp}[\boldsymbol{\Sigma}_o]}_{\boldsymbol{\Lambda}_{\boldsymbol{S}}^{(1)}} + \underbrace{\lambda\mathcal{P}_{T^\perp} \sum_{k=1}^{\infty} (\mathcal{P}_{\mathfrak{S}}\mathcal{P}_T\mathcal{P}_{\mathfrak{S}})^k [\boldsymbol{\Sigma}_o]}_{\boldsymbol{\Lambda}_{\boldsymbol{S}}^{(2)}}. \tag{5.3.35}$$

For the second term, we introduce the more concise notation

$$\mathcal{R} = \mathcal{P}_{T^\perp} \sum_{k=1}^{\infty} (\mathcal{P}_{\mathfrak{S}}\mathcal{P}_T\mathcal{P}_{\mathfrak{S}})^k, \tag{5.3.36}$$

so that

$$\boldsymbol{\Lambda}_{\boldsymbol{S}}^{(2)} = \lambda\mathcal{R}[\boldsymbol{\Sigma}_o]. \tag{5.3.37}$$

Notice that

$$\|\mathcal{R}\| \leq \frac{\sigma^2}{1-\sigma^2}. \tag{5.3.38}$$

The norm of $\boldsymbol{\Lambda}_{\boldsymbol{S}}^{(1)}$ can be controlled by noting that

$$\left\| \boldsymbol{\Lambda}_{\boldsymbol{S}}^{(1)} \right\| = \lambda \|\mathcal{P}_{T^\perp}\boldsymbol{\Sigma}_o\| \leq \lambda \|\boldsymbol{\Sigma}_o\|. \tag{5.3.39}$$

With high probability,

$$\|\boldsymbol{\Sigma}_o\| \leq C\sqrt{\rho m}, \tag{5.3.40}$$

whence for $\rho < \rho_\star$ a small constant, $\left\|\boldsymbol{\Lambda}_{\boldsymbol{S}}^{(1)}\right\| \leq \frac{1}{16}$. To control the norm of $\boldsymbol{\Lambda}_{\boldsymbol{S}}^{(2)}$, let N be a $\frac{1}{2}$ net for $\mathbb{S}^{n-1}$. By Lemma 3.4.5, such a net exists, with size $|N| \leq 6^n$. Moreover,

$$\begin{aligned}
\left\|\boldsymbol{\Lambda}_{\boldsymbol{S}}^{(2)}\right\| &= \sup_{\boldsymbol{u},\boldsymbol{v}\in\mathbb{S}^{n-1}} \boldsymbol{u}^* \boldsymbol{\Lambda}_{\boldsymbol{S}}^{(2)} \boldsymbol{v} \\
&\leq 4 \max_{\boldsymbol{u},\boldsymbol{v}\in N} \boldsymbol{u}^* \boldsymbol{\Lambda}_{\boldsymbol{S}}^{(2)} \boldsymbol{v} \\
&= 4 \max_{\boldsymbol{u},\boldsymbol{v}} \left\langle \boldsymbol{u}\boldsymbol{v}^*, \lambda \mathcal{R}\left[\boldsymbol{\Sigma}_o\right] \right\rangle \\
&= 4 \max_{\boldsymbol{u},\boldsymbol{v}} \left\langle \lambda \mathcal{R}[\boldsymbol{u}\boldsymbol{v}^*], \boldsymbol{\Sigma}_o \right\rangle \\
&= 4 \max_{\boldsymbol{u},\boldsymbol{v}} \left\langle \boldsymbol{X}_{\boldsymbol{u},\boldsymbol{v}}, \boldsymbol{\Sigma}_o \right\rangle . \qquad (5.3.41)
\end{aligned}$$

Conditioned on the support $\mathfrak{S}$ of the sparse error term, we can observe that the random variable $\langle \boldsymbol{X}_{\boldsymbol{u},\boldsymbol{v}}, \boldsymbol{\Sigma}_o \rangle$ is a linear combination of Rademacher (± 1) random variables. Hoeffding's inequality gives

$$\mathbb{P}\Big[\left\langle \boldsymbol{X}_{\boldsymbol{u},\boldsymbol{v}}, \boldsymbol{\Sigma}_o \right\rangle > t \;\Big|\; \mathfrak{S} \Big] \leq \exp\left(-\frac{t^2}{2\left\|\boldsymbol{X}_{\boldsymbol{u},\boldsymbol{v}}\right\|_F^2} \right). \qquad (5.3.42)$$

On $\mathcal{E}$, using the bound (5.3.38), we can control the norm of $\boldsymbol{X}_{\boldsymbol{u},\boldsymbol{v}}$, via

$$\left\|\boldsymbol{X}_{\boldsymbol{u},\boldsymbol{v}}\right\|_F \leq \frac{\lambda\sigma^2}{1-\sigma^2}. \qquad (5.3.43)$$

So, for each $\boldsymbol{u}, \boldsymbol{v}$,

$$\mathbb{P}\Big[\left\langle \boldsymbol{X}_{\boldsymbol{u},\boldsymbol{v}}, \boldsymbol{\Sigma}_o \right\rangle > t \;\Big|\; \mathcal{E} \Big] \leq \exp\left(-\frac{t^2}{2\left\|\boldsymbol{X}_{\boldsymbol{u},\boldsymbol{v}}\right\|_F^2} \right). \qquad (5.3.44)$$

Hence,

$$\begin{aligned}
&\mathbb{P}\Big[\left\|\boldsymbol{\Lambda}_{\boldsymbol{S}}^{(2)}\right\| > t \Big] \\
&\leq \mathbb{P}\Big[\max_{\boldsymbol{u},\boldsymbol{v}\in N} \left\langle \boldsymbol{X}_{\boldsymbol{u},\boldsymbol{v}}, \boldsymbol{\Sigma}_o \right\rangle > \tfrac{t}{4} \Big] \\
&\leq \mathbb{P}\Big[\max_{\boldsymbol{u},\boldsymbol{v}\in N} \left\langle \boldsymbol{X}_{\boldsymbol{u},\boldsymbol{v}}, \boldsymbol{\Sigma}_o \right\rangle > \tfrac{t}{4} \mid \mathcal{E} \Big] + \mathbb{P}\Big[\mathcal{E}^c\Big] \\
&\leq (\#N)^2 \times \max_{\boldsymbol{u},\boldsymbol{v}\in N} \mathbb{P}\Big[\left\langle \boldsymbol{X}_{\boldsymbol{u},\boldsymbol{v}}, \boldsymbol{\Sigma}_o \right\rangle > \tfrac{t}{4} \mid \mathcal{E} \Big] + \mathbb{P}\Big[\mathcal{E}^c\Big] \\
&\leq 6^{2n} \times \exp\left(-\frac{t^2(1-\sigma^2)^2}{2\lambda^2\sigma^4} \right) + \mathbb{P}\left[\mathcal{E}^c\right]. \qquad (5.3.45)
\end{aligned}$$

Setting $t = \frac{1}{8}$, and ensuring that σ is appropriately small, we obtain that with high probability $\left\|\boldsymbol{\Lambda}_{\boldsymbol{S}}^{(2)}\right\| \leq \frac{1}{8}$; combining with our bound on $\left\|\boldsymbol{\Lambda}_{\boldsymbol{S}}^{(1)}\right\|$, we obtain that $\|\boldsymbol{\Lambda}_{\boldsymbol{S}}\| < \frac{1}{4}$ with high probability, as desired.

Verifying (ii)**.** We finish by verifying that with high probability, $\|\mathcal{P}_{\mathfrak{S}^c}\boldsymbol{\Lambda}_{\boldsymbol{S}}\|_\infty < \frac{\lambda}{4}$. For this, notice that

$$\begin{aligned}\mathcal{P}_{\mathfrak{S}^c}\boldsymbol{\Lambda}_{\boldsymbol{S}} &= \lambda\mathcal{P}_{\mathfrak{S}^c}\mathcal{P}_{T^\perp}\sum_{k=0}^{\infty}\left(\mathcal{P}_{\mathfrak{S}}\mathcal{P}_T\mathcal{P}_{\mathfrak{S}}\right)^k\boldsymbol{\Sigma}_o \\ &= \lambda\mathcal{P}_{\mathfrak{S}^c}\mathcal{P}_T\mathcal{P}_{\mathfrak{S}}\sum_{k=0}^{\infty}\left(\mathcal{P}_{\mathfrak{S}}\mathcal{P}_T\mathcal{P}_{\mathfrak{S}}\right)^k\boldsymbol{\Sigma}_o \\ &\doteq \lambda\mathcal{H}[\boldsymbol{\Sigma}_o]. \end{aligned} \tag{5.3.46}$$

On $\mathcal{E}$, for any $(i,j)\in\mathfrak{S}^c$, we have

$$\begin{aligned}\left\|\mathcal{H}^*[\boldsymbol{e}_i\boldsymbol{e}_j^*]\right\|_F &= \left\|\left[\sum_{k=0}^{\infty}(\mathcal{P}_{\mathfrak{S}}\mathcal{P}_T\mathcal{P}_{\mathfrak{S}})^k\right]\mathcal{P}_{\mathfrak{S}}\mathcal{P}_T[\boldsymbol{e}_i\boldsymbol{e}_j^*]\right\|_F \\ &\le \left\|\left[\sum_{k=0}^{\infty}(\mathcal{P}_{\mathfrak{S}}\mathcal{P}_T\mathcal{P}_{\mathfrak{S}})^k\right]\mathcal{P}_{\mathfrak{S}}\mathcal{P}_T\right\|\left\|\mathcal{P}_T[\boldsymbol{e}_i\boldsymbol{e}_j^*]\right\|_F \\ &\le \frac{\sigma}{1-\sigma^2}\times\sqrt{\frac{2\nu r}{n}} \\ &\le C\sqrt{\log n}. \end{aligned} \tag{5.3.47}$$

Notice that

$$\begin{aligned}\|\mathcal{P}_{\mathfrak{S}^c}\boldsymbol{\Lambda}_{\boldsymbol{S}}\|_\infty &= \lambda\max_{i,j}|\boldsymbol{e}_i^*\mathcal{H}[\boldsymbol{\Sigma}_o]\boldsymbol{e}_j| \\ &= \lambda\max_{ij}\left|\left\langle\mathcal{H}[\boldsymbol{e}_i\boldsymbol{e}_j^*],\boldsymbol{\Sigma}_o\right\rangle\right|. \end{aligned} \tag{5.3.48}$$

Write

$$\boldsymbol{Y}_{ij} = \left\langle\mathcal{H}[\boldsymbol{e}_i\boldsymbol{e}_j^*],\boldsymbol{\Sigma}_o\right\rangle. \tag{5.3.49}$$

Using Hoeffding's inequality again, we have

$$\mathbb{P}\Big[|\boldsymbol{Y}_{ij}|>t\ \mid\ \mathfrak{S}\Big] \le 2\exp\left(-\frac{t^2}{2\left\|\mathcal{H}[\boldsymbol{e}_i\boldsymbol{e}_j^*]\right\|_F^2}\right). \tag{5.3.50}$$

Hence,

$$\mathbb{P}\Big[|\boldsymbol{Y}_{ij}|>t\ \mid\ \mathcal{E}\Big] \le 2n^{-12} \tag{5.3.51}$$

We have

$$\begin{aligned}
\mathbb{P}\Big[\|\mathcal{P}_{\mathfrak{S}^c}\boldsymbol{\Lambda}_{\boldsymbol{S}}\|_\infty \geq \tfrac{\lambda}{4}\Big] \quad &\leq \quad \mathbb{P}\Big[\max_{ij}|\boldsymbol{Y}_{ij}| > \tfrac{1}{4}\Big] \\
&\leq \quad \mathbb{P}\Big[\max_{ij}|\boldsymbol{Y}_{ij}| > \tfrac{1}{4} \;\mid\; \mathcal{E}\Big] \quad + \quad \mathbb{P}\Big[\,\mathcal{E}^c\,\Big] \\
&\leq \quad \sum_{ij}\mathbb{P}\Big[|\boldsymbol{Y}_{ij}| > \tfrac{1}{4} \;\mid\; \mathcal{E}\Big] \quad + \quad \mathbb{P}\Big[\,\mathcal{E}^c\,\Big] \\
&\leq \quad n^2 \times 2n^{-12} \quad + \quad \mathbb{P}\left[\mathcal{E}^c\right] \\
&\leq \quad 2n^{-10} \quad + \quad \mathbb{P}\left[\mathcal{E}^c\right]. \qquad (5.3.52)
\end{aligned}$$

This completes the proof. □

With the so constructed $\boldsymbol{\Lambda}_{\boldsymbol{S}}$ and $\boldsymbol{\Lambda}_{\boldsymbol{L}}$, it can be easily show that the combined

$$\boldsymbol{\Lambda} = \boldsymbol{\Lambda}_{\boldsymbol{L}} + \boldsymbol{\Lambda}_{\boldsymbol{S}}$$

satisfies all conditions of Lemma 5.3.3 under the assumptions of Theorem 5.3.1. This completes the proof of Theorem 5.3.1.

5.4 Noise Stability of Principal Component Pursuit

The PCP model and result (Theorem 5.3.1) is limited to the low-rank component being exactly low-rank and the sparse component being exactly sparse. However, in real world applications the observations are often perturbed by noise, which may be stochastic or deterministic, affecting every entry of the data matrix. For example, in face recognition that we mentioned earlier, the human face is not a strictly convex and Lambertian surface hence the low-rank model (due to photometric properties) is only approximately low-rank. In ranking and collaborative filtering, user's ratings could be noisy because of the lack of control in the data collection process. Therefore, for the PCP method to be applicable to wider range of real world problems, we need to exam if it can handle small entry-wise (dense) noise.

In the presence of noise, the new measurement model becomes

$$\boldsymbol{Y} = \boldsymbol{L}_o + \boldsymbol{S}_o + \boldsymbol{Z}_o, \qquad (5.4.1)$$

where $\boldsymbol{Z}_o$ is a noise term – say i.i.d. noise on each entry of the matrix. However, all we assume about $\boldsymbol{Z}_o$ here is that $\|\boldsymbol{Z}_o\|_F \leq \delta$ for some $\delta > 0$.

5.4.1 Recovery Results for Deterministic Noise

To recover the unknown matrices $\boldsymbol{L}_o$ and $\boldsymbol{S}_o$, one may consider solving the following optimization problem, as a relaxed version to PCP (5.2.4):

$$\min_{\boldsymbol{L},\boldsymbol{S}} \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 \quad \text{subject to} \quad \|\boldsymbol{Y} - \boldsymbol{L} - \boldsymbol{S}\|_F \leq \delta. \tag{5.4.2}$$

where we choose $\lambda = 1/\sqrt{n}$. Note that with this choice, we typically have $\lambda < 1/2$ for large n. Our main result is that under the same conditions as PCP, the above convex program gives a stable estimate of $\boldsymbol{L}_o$ and $\boldsymbol{S}_o$:

Theorem 5.4.1 (Stability of PCP to Deterministic Noise)**.** *Under the same assumptions of Theorem 5.3.1, that is, $\boldsymbol{L}_o$ obeys the incoherence conditions and the support of $\boldsymbol{S}_o$ is uniformly distributed of size m. Then if $\boldsymbol{L}_o$ and $\boldsymbol{S}_o$ satisfy*

$$\operatorname{rank}(\boldsymbol{L}_o) \leq \frac{\rho_r n}{\nu \log^2 n} \quad \textit{and} \quad m \leq \rho_s n^2, \tag{5.4.3}$$

with $\rho_r, \rho_s > 0$ being sufficiently small numerical constants, with high probability in the support of $\boldsymbol{S}_o$, for any $\boldsymbol{Z}_o$ with $\|\boldsymbol{Z}_o\|_F \leq \delta$, the solution $(\hat{\boldsymbol{L}}, \hat{\boldsymbol{S}})$ to the convex program (5.4.2) satisfies

$$\|\hat{\boldsymbol{L}} - \boldsymbol{L}_o\|_F^2 + \|\hat{\boldsymbol{S}} - \boldsymbol{S}_o\|_F^2 \leq C\delta^2, \tag{5.4.4}$$

where the constant $C = \left(16\sqrt{5}n + \sqrt{2}\right)^2$.

Here, we would like to point out two ways to view the significance of this result. To some extent, the model (5.4.2) unifies the classical PCA and the robust PCA by considering both gross sparse errors and small entry-wise noise in the measurements. So on the one hand, the above theorem says that the low-rank and sparse decomposition via PCP is stable in the presence of small entry-wise noise, hence making PCP more widely applicable to practical problems where the low-rank structure is not exact. On the other hand, the theorem convincingly justifies that the classical PCA can now be made robust to sparse gross corruptions via certain convex program. Since this convex program can be solved very efficiently via algorithms similar to Algorithm 5.1, at a cost not so much higher than the classical PCA, this model and result can be applied to many practical problems where both small noise and gross corruption are present simultaneously.

Before we set out to prove the above result, let us first introduce some new notation. For any matrix pair $\boldsymbol{X} = (\boldsymbol{L}, \boldsymbol{S})$ let

$$\|\boldsymbol{X}\|_F \doteq \left(\|\boldsymbol{L}\|_F^2 + \|\boldsymbol{S}\|_F^2\right)^{1/2}, \quad \|\boldsymbol{X}\|_\diamond = \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1.$$

Define a projection operator

$$\mathcal{P}_T \times \mathcal{P}_{\mathfrak{S}} : (\boldsymbol{L}, \boldsymbol{S}) \mapsto (\mathcal{P}_T \boldsymbol{L}, \mathcal{P}_{\mathfrak{S}} \boldsymbol{S}).$$

Also we define the subspaces $\Gamma \doteq \{(\boldsymbol{Q}, \boldsymbol{Q}) \mid \boldsymbol{Q} \in \mathbb{R}^{n\times n}\}$ and $\Gamma^\perp \doteq \{(\boldsymbol{Q}, -\boldsymbol{Q}) \mid \boldsymbol{Q} \in \mathbb{R}^{n\times n}\}$, and let $\mathcal{P}_\Gamma$ and $\mathcal{P}_{\Gamma^\perp}$ denote their respective projection operators.

Lemma 5.4.2. *Suppose that* $\|\mathcal{P}_T\mathcal{P}_\mathfrak{S}\| \le 1/2$. *Then for any pair* $\boldsymbol{X} = (\boldsymbol{L}, \boldsymbol{S})$, $\|\mathcal{P}_\Gamma(\mathcal{P}_T \times \mathcal{P}_\mathfrak{S})(\boldsymbol{X})\|_F^2 \ge \frac{1}{4}\|(\mathcal{P}_T \times \mathcal{P}_\mathfrak{S})(\boldsymbol{X})\|_F^2$.

Proof. For any matrix pair $\boldsymbol{X}' = (\boldsymbol{L}', \boldsymbol{S}')$, $\mathcal{P}_\Gamma(\boldsymbol{X}') = \left(\frac{\boldsymbol{L}'+\boldsymbol{S}'}{2}, \frac{\boldsymbol{L}'+\boldsymbol{S}'}{2}\right)$ and so $\|\mathcal{P}_\Gamma(\boldsymbol{X}')\|_F^2 = \frac{1}{2}\|\boldsymbol{L}' + \boldsymbol{S}'\|_F^2$. So,

$$\begin{aligned}\|\mathcal{P}_\Gamma(\mathcal{P}_T \times \mathcal{P}_\mathfrak{S})(\boldsymbol{X})\|_F^2 &= \tfrac{1}{2}\|\mathcal{P}_T(\boldsymbol{L}) + \mathcal{P}_\mathfrak{S}(\boldsymbol{S})\|_F^2 \\ &= \frac{1}{2}\left(\|\mathcal{P}_T(\boldsymbol{L})\|_F^2 + \|\mathcal{P}_\mathfrak{S}(\boldsymbol{S})\|_F^2 + 2\langle \mathcal{P}_T(\boldsymbol{L}), \mathcal{P}_\mathfrak{S}(\boldsymbol{S})\rangle\right).\end{aligned}$$

Now,

$$\begin{aligned}\langle \mathcal{P}_T(\boldsymbol{L}), \mathcal{P}_\mathfrak{S}(\boldsymbol{S})\rangle &= \langle \mathcal{P}_T(\boldsymbol{L}), (\mathcal{P}_T\mathcal{P}_\mathfrak{S})\mathcal{P}_\mathfrak{S}(\boldsymbol{S})\rangle \\ &\ge -\|\mathcal{P}_T\mathcal{P}_\mathfrak{S}\|\|\mathcal{P}_T(\boldsymbol{L})\|_F\|\mathcal{P}_\mathfrak{S}(\boldsymbol{S})\|_F.\end{aligned}$$

Since $\|\mathcal{P}_T\mathcal{P}_\mathfrak{S}\| \le 1/2$,

$$\begin{aligned}&\|\mathcal{P}_\Gamma(\mathcal{P}_T \times \mathcal{P}_\mathfrak{S})(\boldsymbol{X})\|_F^2 \\ \ge\ & \tfrac{1}{2}\left(\|\mathcal{P}_T(\boldsymbol{L})\|_F^2 + \|\mathcal{P}_\mathfrak{S}(\boldsymbol{S})\|_F^2 - \|\mathcal{P}_T(\boldsymbol{L})\|_F\|\mathcal{P}_\mathfrak{S}(\boldsymbol{S})\|_F\right) \\ \ge\ & \tfrac{1}{4}\left(\|\mathcal{P}_T(\boldsymbol{L})\|_F^2 + \|\mathcal{P}_\mathfrak{S}(\boldsymbol{S})\|_F^2\right) = \tfrac{1}{4}\|(\mathcal{P}_T \times \mathcal{P}_\mathfrak{S})(\boldsymbol{X})\|_F^2,\end{aligned}$$

where we have used that for any a, b, $a^2 + b^2 - ab \ge (a^2 + b^2)/2$. □

The proof for the noisy case largely relies on the method and results we have developed before for proving the noiseless case of PCP. From the proof of Theorem 5.3.1, we know that, with high probability, there exists a dual certificate $\boldsymbol{\Lambda}$ satisfying the conditions in Lemma (5.3.3):

$$\begin{cases} \|\mathcal{P}_T\boldsymbol{\Lambda} - \boldsymbol{U}\boldsymbol{V}^*\|_F \le \frac{\lambda}{8}, & \|\mathcal{P}_{T^\perp}\boldsymbol{\Lambda}\| < \frac{1}{2}, \\ \|\mathcal{P}_\mathfrak{S}\boldsymbol{\Lambda} - \lambda\boldsymbol{\Sigma}_o\|_F \le \frac{\lambda}{8}, & \|\mathcal{P}_{\mathfrak{S}^c}\boldsymbol{\Lambda}\|_\infty < \frac{\lambda}{2}. \end{cases} \tag{5.4.5}$$

Our proof uses two crucial properties of $\hat{\boldsymbol{X}} = (\hat{\boldsymbol{L}}, \hat{\boldsymbol{S}})$. First, since $\boldsymbol{X}_o$ is also a feasible solution to (5.4.2), we have $\|\hat{\boldsymbol{X}}\|_\diamond \le \|\boldsymbol{X}_o\|_\diamond$. Second, we use triangle inequality to get

$$\begin{aligned}&\|\hat{\boldsymbol{L}} + \hat{\boldsymbol{S}} - \boldsymbol{L}_o - \boldsymbol{S}_o\|_F \\ \le\ & \|\hat{\boldsymbol{L}} + \hat{\boldsymbol{S}} - \boldsymbol{Y}\|_F + \|\boldsymbol{L}_o + \boldsymbol{S}_o - \boldsymbol{Y}\|_F \le 2\delta.\end{aligned} \tag{5.4.6}$$

Furthermore, set $\hat{\boldsymbol{X}} = \boldsymbol{X}_o + \boldsymbol{H}$ where $\boldsymbol{H} = (\boldsymbol{H}_L, \boldsymbol{H}_S)$. We want to bound the norm of the perturbation $\|\boldsymbol{H}\|_F^2 = \|\boldsymbol{H}_L\|_F^2 + \|\boldsymbol{H}_S\|_F^2$, Notice that unlike the noise-free case, here $\boldsymbol{H}_L + \boldsymbol{H}_S$ is not necessarily equal to zero. So in order to leverage results from the noise-free case, we decompose the perturbation into the two orthogonal components in Γ and $\Gamma^\perp$ respectively:

$\boldsymbol{H}^{\Gamma} = \mathcal{P}_{\Gamma}(\boldsymbol{H})$ and $\boldsymbol{H}^{\Gamma^{\perp}} = \mathcal{P}_{\Gamma^{\perp}}(\boldsymbol{H})$. Then $\|\boldsymbol{H}\|_F^2$ can be expanded as

$$\begin{aligned}\|\boldsymbol{H}\|_F^2 &= \|\boldsymbol{H}^{\Gamma}\|_F^2 + \|\boldsymbol{H}^{\Gamma^{\perp}}\|_F^2 \\ &= \|\boldsymbol{H}^{\Gamma}\|_F^2 + \|(\mathcal{P}_T \times \mathcal{P}_{\Omega})(\boldsymbol{H}^{\Gamma^{\perp}})\|_F^2 + \|(\mathcal{P}_{T^{\perp}} \times \mathcal{P}_{\mathfrak{S}^c})(\boldsymbol{H}^{\Gamma^{\perp}})\|_F^2. \end{aligned} \tag{5.4.7}$$

Since (5.4.6) gives us $\|\boldsymbol{H}^{\Gamma}\|_F = \big(\|(\boldsymbol{H}_L+\boldsymbol{H}_S)/2\|_F^2+\|(\boldsymbol{H}_L+\boldsymbol{H}_S)/2\|_F^2\big)^{1/2} \le \sqrt{2}/2 \times 2\delta = \sqrt{2}\delta$, it suffices to bound the second and third terms on the right-hand-side of (5.4.7).

a. Bound the third term of (5.4.7). Let $\boldsymbol{\Lambda}$ be a dual certificate satisfying (5.4.5). Then we have

$$\|\boldsymbol{X}_o + \boldsymbol{H}\|_{\diamond} \ge \|\boldsymbol{X}_o + \boldsymbol{H}^{\Gamma^{\perp}}\|_{\diamond} - \|\boldsymbol{H}^{\Gamma}\|_{\diamond}. \tag{5.4.8}$$

Since $\boldsymbol{H}_L^{\Gamma^{\perp}} + \boldsymbol{H}_S^{\Gamma^{\perp}} = 0$, following the proof of Lemma 5.3.3, we have

$$\begin{aligned} & \|\boldsymbol{X}_o + \boldsymbol{H}^{\Gamma^{\perp}}\|_{\diamond} \\ \ge\ & \|\boldsymbol{X}_o\|_{\diamond} + 1/8\|\mathcal{P}_{T^{\perp}}(\boldsymbol{H}_L^{\Gamma^{\perp}}) + \lambda/8\|_*\|\mathcal{P}_{\mathfrak{S}^c}(\boldsymbol{H}_S^{\Gamma^{\perp}})\|_1 \\ \ge\ & \|\boldsymbol{X}_o\|_{\diamond} + \frac{1}{8}\Big(\|\mathcal{P}_{T^{\perp}}(\boldsymbol{H}_L^{\Gamma^{\perp}})\|_* + \lambda\|\mathcal{P}_{\mathfrak{S}^c}(\boldsymbol{H}_S^{\Gamma^{\perp}})\|_1\Big), \end{aligned}$$

which implies that

$$\|\mathcal{P}_{T^{\perp}}(\boldsymbol{H}_L^{\Gamma^{\perp}})\|_* + \lambda\|\mathcal{P}_{\mathfrak{S}^c}(\boldsymbol{H}_S^{\Gamma^{\perp}})\|_1 \le 8\|\boldsymbol{H}^{\Gamma}\|_{\diamond}. \tag{5.4.9}$$

For any matrix $\boldsymbol{Y} \in \mathbb{R}^{n\times n}$, we have the following inequalities:

$$\|\boldsymbol{Y}\|_F \le \|\boldsymbol{Y}\|_* \le \sqrt{n}\|\boldsymbol{Y}\|_F, \quad \frac{1}{\sqrt{n}}\|\boldsymbol{Y}\|_F \le \lambda\|\boldsymbol{Y}\|_1 \le \sqrt{n}\|\boldsymbol{Y}\|_F,$$

where we assume $\lambda = \frac{1}{\sqrt{n}}$. Therefore

$$\begin{aligned} & \|(\mathcal{P}_{T^{\perp}} \times \mathcal{P}_{\mathfrak{S}^c})(\boldsymbol{H}^{\Gamma^{\perp}})\|_F \\ \le\ & \|\mathcal{P}_{T^{\perp}}(\boldsymbol{H}_L^{\Gamma^{\perp}})\|_F + \|\mathcal{P}_{\mathfrak{S}^c}(\boldsymbol{H}_S^{\Gamma^{\perp}})\|_F \\ \le\ & \|\mathcal{P}_{T^{\perp}}(\boldsymbol{H}_L^{\Gamma^{\perp}})\|_* + \lambda\sqrt{n}\|\mathcal{P}_{\mathfrak{S}^c}(\boldsymbol{H}_S^{\Gamma^{\perp}})\|_1 \\ \le\ & 8\sqrt{n}\|\boldsymbol{H}^{\Gamma}\|_{\diamond} = 8\sqrt{n}\big(\|\boldsymbol{H}_L^{\Gamma}\|_* + \lambda\|\boldsymbol{H}_S^{\Gamma}\|_1\big) \\ \le\ & 8n(\|\boldsymbol{H}_L^{\Gamma}\|_F + \|\boldsymbol{H}_S^{\Gamma}\|_F) \le 8\sqrt{2n}\|\boldsymbol{H}^{\Gamma}\|_F \le 16n\delta, \end{aligned} \tag{5.4.10}$$

where the last equation uses the fact that $\boldsymbol{H}_L^{\Gamma} = \boldsymbol{H}_S^{\Gamma}$.

b. Bound the second term of (5.4.7). By Lemma 5.4.2,

$$\|\mathcal{P}_{\Gamma}(\mathcal{P}_T \times \mathcal{P}_{\mathfrak{S}})(\boldsymbol{H}^{\Gamma^{\perp}})\|_F^2 \ge \frac{1}{4}\|(\mathcal{P}_T \times \mathcal{P}_{\mathfrak{S}})(\boldsymbol{H}^{\Gamma^{\perp}})\|_F^2.$$

But since $\mathcal{P}_{\Gamma}(\boldsymbol{H}^{\Gamma^{\perp}}) = 0 = \mathcal{P}_{\Gamma}(\mathcal{P}_T \times \mathcal{P}_{\mathfrak{S}})(\boldsymbol{H}^{\Gamma^{\perp}}) + \mathcal{P}_{\Gamma}(\mathcal{P}_{T^{\perp}} \times \mathcal{P}_{\mathfrak{S}^c})(\boldsymbol{H}^{\Gamma^{\perp}})$, we have

$$\begin{aligned} \|\mathcal{P}_{\Gamma}(\mathcal{P}_T \times \mathcal{P}_{\mathfrak{S}})(\boldsymbol{H}^{\Gamma^{\perp}})\|_F &= \|\mathcal{P}_{\Gamma}(\mathcal{P}_{T^{\perp}} \times \mathcal{P}_{\mathfrak{S}^c})(\boldsymbol{H}^{\Gamma^{\perp}})\|_F \\ &\le \|(\mathcal{P}_{T^{\perp}} \times \mathcal{P}_{\mathfrak{S}^c})(\boldsymbol{H}^{\Gamma^{\perp}})\|_F. \end{aligned}$$

Combining the previous two inequalities, we have

$$\|(\mathcal{P}_T \times \mathcal{P}_{\mathfrak{S}})(\boldsymbol{H}^{\Gamma^\perp})\|_F^2 \quad \leq \quad 4\|(\mathcal{P}_{T^\perp} \times \mathcal{P}_{\mathfrak{S}^c})(\boldsymbol{H}^{\Gamma^\perp})\|_F^2,$$

which, together with (5.4.10), gives us the desired result,

$$\|\boldsymbol{H}^{\Gamma^\perp}\|_F^2 \leq 5\|(\mathcal{P}_{T^\perp} \times \mathcal{P}_{\mathfrak{S}^c})(\boldsymbol{H}^{\Gamma^\perp})\|_F^2 \leq 5 \times 16^2 n^2 \delta^2. \tag{5.4.11}$$

5.5 Compressive Principal Component Pursuit

From the above sections, we saw that under fairly broad conditions, via convex optimization, a low-rank matrix $\boldsymbol{L}_o$ and the sparse matrix $\boldsymbol{S}_o$ can be recovered correctly if we observe fully their superposition $\boldsymbol{Y} = \boldsymbol{L}_o + \boldsymbol{S}_o$.

This makes us wonder if we would be able to recover both low-rank and sparse components from a small set of general linear measurements of $\boldsymbol{Y}$. That is, whether we are able to "compressive sensing" a low-rank structure and a sparse model superimposed together. Mathematically, we may assume the observations have the form:

$$\boldsymbol{Y} \;\doteq\; \mathcal{P}_Q[\boldsymbol{L}_o + \boldsymbol{S}_o], \tag{5.5.1}$$

where $Q \subseteq \mathbb{R}^{n_1 \times n_2}$ is a linear subspace, and $\mathcal{P}_Q$ denotes the projection operator onto that subspace. In fact, this problem may arise whenever we observe a "deformed" version of certain 2D array $\boldsymbol{M}$, say $\boldsymbol{M} \circ \tau = \boldsymbol{L}_o + \boldsymbol{S}_o$ where τ is certain domain deformation. One natural approach to recover the deformation τ and the low-rank and sparse components is to linearize the above equation respect to τ and obtain the differential of the above equation at a given τ_o:

$$\boldsymbol{M} \circ \tau_o + \boldsymbol{J} \circ d\tau = \boldsymbol{L}_o + \boldsymbol{S}_o,$$

where $\boldsymbol{J}$ is the Jacobian matrix and $d\tau$ is the infinitesimal deformation. To eliminate the unknown $d\tau$, let Q be the left kernel of the Jacobian $\boldsymbol{J}$, i.e., Q is a subspace spanned by all matrices $Q \doteq \{\boldsymbol{Q} \mid \langle \boldsymbol{Q}, \boldsymbol{J} \rangle = 0\}$. So we have

$$\boldsymbol{Y} \doteq \mathcal{P}_Q[\boldsymbol{M} \circ \tau_o] = \mathcal{P}_Q[\boldsymbol{L}_o + \boldsymbol{S}_o].$$

Can we simultaneously recover the low-rank and sparse components correctly from highly compressive measurements via the natural convex program

$$\text{minimize} \quad \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 \quad \text{subject to} \quad \mathcal{P}_Q[\boldsymbol{L} + \boldsymbol{S}] = \boldsymbol{Y} \;? \tag{5.5.2}$$

We call this convex program *Compressive Principal Component Pursuit*, or shortly *CPCP*. In this section, we study when this program can correctly recover $\boldsymbol{L}_o$ and $\boldsymbol{S}_o$. As before, throughout this section, we assume the low-rank matrix $\boldsymbol{L}_o$ is ν-incoherent and the support of the sparse component $\boldsymbol{S}_o$, say $\mathfrak{S}$, is (Bernoulli) random.

To recover both $\boldsymbol{L}_o$ and $\boldsymbol{S}_o$ correctly, we must require measurements Q to be incoherent with *both* the low-rank and the sparse component. To ensure the incoherence property, we may assume that Q is a randomly chosen subspace in the matrix space $\mathbb{R}^{n_1 \times n_2}$.

More precisely, suppose the dimension of the subspace Q is q, and we assume Q is distributed according to the Haar measure on the Grassmannian $\mathbb{G}(\mathbb{R}^{m \times n}, q)$. On a more intuitive level, this means that Q is equal in distribution to the linear span of a collection of q independent iid $\mathcal{N}(0,1)$ matrices. In notation more familiar from compressive sensing, we may let $\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_q$ denote such a set of matrices, and define an operator $\mathcal{Q} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^q$ via

$$\mathcal{Q}[\boldsymbol{M}] \;=\; \big(\langle \boldsymbol{Q}_1, \boldsymbol{M} \rangle, \ldots, \langle \boldsymbol{Q}_q, \boldsymbol{M} \rangle\big)^* \;\in\; \mathbb{R}^q. \tag{5.5.3}$$

Our analysis also pertains to the equivalent convex program:

$$\text{minimize} \quad \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 \quad \text{subject to} \quad \mathcal{Q}[\boldsymbol{L} + \boldsymbol{S}] = \mathcal{Q}[\boldsymbol{L}_o + \boldsymbol{S}_o]. \tag{5.5.4}$$

Since $\mathcal{Q}$ has full rank q almost surely, (5.5.4) and (5.5.2) are completely equivalent.

With these assumptions, the following theorem gives a tight bound on the number of (random) measurements required to correctly recover the pair $(\boldsymbol{L}_o, \boldsymbol{S}_o)$ from $\mathcal{P}_Q[\boldsymbol{L}_o + \boldsymbol{S}_o]$ via CPCP:

Theorem 5.5.1 (Compressive PCP Recovery). *Let $\boldsymbol{L}_o, \boldsymbol{S}_o \in \mathbb{R}^{n_1 \times n_2}$, with $n_1 \geq n_2$, and suppose that $\boldsymbol{L}_o \neq \boldsymbol{0}$ is a rank-r, ν-incoherent matrix with*

$$r \;\leq\; \frac{c_r n_2}{\nu \log^2 n_1}, \tag{5.5.5}$$

and $\operatorname{sign}(\boldsymbol{S}_o)$ is iid Bernoulli-Rademacher with nonzero probability $\rho < c_\rho$. Let $Q \subset \mathbb{R}^{n_1 \times n_2}$ be a random subspace of dimension

$$\dim(Q) \;\geq\; C_Q \cdot (\rho n_1 n_2 + n_1 r) \cdot \log^2 n_1 \tag{5.5.6}$$

distributed according to the Haar measure, probabilistically independent of $\operatorname{sign}(\boldsymbol{S}_o)$. Then with probability at least $1 - C n_1^{-9}$ in $(\operatorname{sign}(\boldsymbol{S}_o), Q)$, the solution to

$$\textit{minimize} \quad \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 \quad \textit{subject to} \quad \mathcal{P}_Q[\boldsymbol{L} + \boldsymbol{S}] = \mathcal{P}_Q[\boldsymbol{L}_o + \boldsymbol{S}_o] \tag{5.5.7}$$

with $\lambda = 1/\sqrt{n_1}$ is unique, and equal to $(\boldsymbol{L}_o, \boldsymbol{S}_o)$. Above, c_r, c_ρ, C_Q, C are positive numerical constants.

Here, the magnitudes of the nonzeros in $\boldsymbol{S}_o$ are arbitrary, and no randomness is assumed in $\boldsymbol{L}_o$. The randomness in this result occurs in the sign and support pattern of $\boldsymbol{S}_o$ and in the measurements Q. The bounds on r and ρ essentially match those of PCP for the fully observed case, possibly with different constants. So, again, r and $\|\boldsymbol{S}_o\|_0$ can be rather large. On the other hand, when these quantities are small, the bound on

$\dim(Q)$ ensures that the number of measurements needed for accurate recovery is also commensurately small. In fact, this result can be obtained via general arguments that can also be applied to other compressive sensing and decomposition problems of a family of low-dimensional structures in high-dimensional space (as we will introduce in the next Chapter). Since the approach and techniques of the proof are rather similar to that for the PCP, we here do not elaborate and instead point interested readers to [Wright et al., 2013] for a complete and rigorous proof.

5.6 Matrix Completion with Corrupted Entries

We have seen that the main result on PCP (Theorem 5.3.1) asserts that it is possible to recover a low-rank matrix even though a significant fraction of its entries are corrupted. Furthermore, the above section reveals that both the low-rank and sparse components can be recovered even if only a small number of general linear measurements of the corrupted matrix $\boldsymbol{Y}$ are given.

In many applications, however, the (linear) measurements of the corrupted matrix available to us are not general and have very peculiar structures. For instance, we only get to see a small fraction of the entires of $\boldsymbol{Y}$, and the remaining of the entries may be missing. For instance, in the case of taking face images under different illuminations, we can use random corruptions to model pixels associated with surfaces that violate the Lambertian property (such as specular surfaces); and we may assume the intensities of pixels which are blocked from light sources (in the shadow areas) are missing. Hence the data (matrix) have both corrupted and missing entries. Can we still expect to recovery the low-rank matrix? As the observations are no longer general (e.g., they are not incoherent with the sparse term $\boldsymbol{S}_o$), results from the above section do not directly apply to the situations here. This section addresses this problem.

To be precise, as before, we assume $\boldsymbol{Y} = \boldsymbol{L}_o + \boldsymbol{S}_o$ is a low-rank matrix $\boldsymbol{L}_o$ corrupted by a sparse matrix $\boldsymbol{S}_o$ whose support $\mathfrak{S}$ is distributed as $\mathfrak{S} \sim Ber(\rho_s)$ for some small constant $\rho_s < 1$.

We further assume we only observe a small fraction ρ_o of the entries of $\boldsymbol{Y}$. Let O be a support distributed as $O \sim Ber(\rho_o)$, where O stands for "observed" entries. We may assume $\mathfrak{S}$ and O are independent Bernoulli variables.

Let $\mathcal{P}_O$ be the orthogonal projection onto the linear space of matrices supported on $O \subset [n_1] \times [n_2]$,

$$\mathcal{P}_O[\boldsymbol{X}] = \begin{cases} X_{ij}, & (i,j) \in O, \\ 0, & (i,j) \notin O. \end{cases}$$

Then imagine we only have available those entries of $\boldsymbol{L}_o + \boldsymbol{S}_o$ such that $(i,j) \in O \subset [n_1] \times [n_2]$, which we conveniently write as

$$\mathcal{P}_O([\boldsymbol{Y}]) = \mathcal{P}_O[\boldsymbol{L}_o + \boldsymbol{S}_o] = \mathcal{P}_O[\boldsymbol{L}_o] + \boldsymbol{S}'_o.$$

This models the following problem: we wish to recover $\boldsymbol{L}_o$ but only see a few entries about $\boldsymbol{L}_o$, and among those a fraction happens to be corrupted, and we of course do not know which one. As is easily seen, this is an extension to the Matrix Completion problem of the previous chapter, which seeks to recover $\boldsymbol{L}_o$ from under-sampled but otherwise perfect data $\mathcal{P}_O[\boldsymbol{L}_o]$; and this is also an extension to the RPCA problem as there we only see a small fraction of the corrupted matrix $\boldsymbol{Y}$.

We propose recovering $\boldsymbol{L}_o$ (and $\boldsymbol{S}'_o$) by solving the following problem:

$$\begin{array}{ll} \text{minimize} & \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 \\ \text{subject to} & \mathcal{P}_O[\boldsymbol{L} + \boldsymbol{S}] = \mathcal{P}_O[\boldsymbol{Y}]. \end{array} \tag{5.6.1}$$

In words, among all decompositions matching the available data, Principal Component Pursuit, finds the one that minimizes the weighted combination of the nuclear norm, and of the ℓ^1 norm. Our observation is that under some conditions, this simple approach recovers the low-rank component exactly. In fact, the techniques developed here establish this result:

Theorem 5.6.1. *Suppose $\boldsymbol{L}_o$ is $n \times n$, obeys the conditions (5.3.1)–(5.3.2). Suppose $\rho_0 > C_0 \frac{\nu r \log^2 n}{n}$ and $\rho_s \leq C_s$, and let $\lambda = \frac{1}{\sqrt{\rho_0 n \log n}}$. Then the optimal solution to the convex program (5.6.1) is exactly $\boldsymbol{L}_o$ and $\boldsymbol{S}'_o$ with probability at least $1 - Cn^{-3}$ for some constant C, provided the constants C_0 is large enough and C_s is small enough.*

In short, perfect recovery from incomplete and corrupted entries is possible by convex optimization. The approach and techniques of the proof are similar to that of the PCP, we refer interested readers for a complete and rigorous proof to the work of [Li, 2013].

On the one hand, this result extends the RPCA result in the following way: If all the entries are available, i.e. $\rho_0 = 1$, the above theorem guarantees perfect recovery as long as $1 > C_0 \frac{\nu r \log^2 n}{n}$ or $r < C_0^{-1} n \nu^{-1} (\log n)^2$ for small enough C_0^{-1}, which is exactly Theorem 5.3.1. On the other hand, it extends the Matrix Completion results developed in the previous chapter too. Indeed, if $\rho_s = 0$, we have a pure matrix completion problem from about ρ_0 fraction of entries, and the above theorem guarantees perfect recovery as long as $\rho_0 > C_0 \frac{\nu r \log^2 n}{n}$ for large enough C_0, which is exactly Theorem 4.4.3.

We remark that the recovery is exact, however, via a different algorithm. To be sure, in matrix completion one typically minimizes the nuclear norm $\|\boldsymbol{L}\|_*$ subject to the constraint $\mathcal{P}_O[\boldsymbol{L}] = \mathcal{P}_O[\boldsymbol{L}_o]$. Here, our program would

solve

$$\begin{array}{ll} \text{minimize} & \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 \\ \text{subject to} & \mathcal{P}_O[\boldsymbol{L}+\boldsymbol{S}] = \mathcal{P}_O[\boldsymbol{L}_o], \end{array} \tag{5.6.2}$$

and return $\hat{\boldsymbol{L}} = \boldsymbol{L}_o$, $\hat{\boldsymbol{S}} = 0$! In this context, Theorem 5.6.1 proves that matrix completion is stable vis a vis gross errors.

5.7 Summary

5.8 Notes and References

Algorithms for Solving PCP.

For small problem sizes, Principal Component Pursuit,

$$\begin{array}{ll} \text{minimize} & \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 \\ \text{subject to} & \boldsymbol{L}+\boldsymbol{S} = \boldsymbol{Y} \end{array}$$

can be performed using off-the-shelf tools such as interior point methods [Grant and Boyd, 2014]. This was suggested for rank minimization in [**?**, **?**] and for low-rank and sparse decomposition [**?**] (see also [**?**]). However, despite their superior convergence rates, interior point methods are typically limited to small problems, say $n < 100$, due to the $O(n^6)$ complexity of computing a step direction.

The limited scalability of interior point methods has inspired a recent flurry of work on first-order methods. Exploiting an analogy with iterative thresholding algorithms for ℓ^1-minimization [Hale et al., 2008, Yin et al., 2008], Cai et. al. developed an algorithm that performs nuclear-norm minimization by repeatedly shrinking the *singular values* of an appropriate matrix, essentially reducing the complexity of each iteration to the cost of an SVD [Cai et al.,]. However, for our low-rank and sparse decomposition problem, this form of iterative thresholding converges slowly, requiring up to 10^4 iterations. Ma et. al. [Goldfarb and Ma, 2009, **?**] suggest improving convergence using continuation techniques, and also demonstrate how Bregman iterations [**?**] can be applied to nuclear norm minimization.

The convergence of iterative thresholding has also been greatly improved using ideas from Nesterov's optimal first-order algorithm for smooth minimization [Nesterov, 1983], which was extended to non-smooth optimization in [**?**, **?**], and applied to ℓ^1-minimization in [**?**, **?**, **?**]. Based on [**?**], Toh et. al. developed a proximal gradient algorithm for matrix completion which they termed *Accelerated Proximal Gradient (APG)*. A very similar APG algorithm was suggested for low-rank and sparse decomposition in [**?**]. That algorithm inherits the optimal $O(1/k^2)$ convergence rate for this class of problems. Empirical evidence suggests that these algorithms can solve the

convex PCP problem at least 50 times faster than straightforward iterative thresholding (for more details and comparisons, see [**?**]).

However, despite its good convergence guarantees, the practical performance of APG depends strongly on the design of good continuation schemes. Generic continuation does not guarantee good accuracy and convergence across a wide range of problem settings.[5] In this chapter, we have chosen to instead solve the convex PCP problem (5.2.4) using an augmented Lagrange multiplier (ALM) algorithm introduced in [**?**, Yuan and Yang, 2009]. In our experience, ALM achieves much higher accuracy than APG, in fewer iterations. It works stably across a wide range of problem settings with no tuning of parameters. Moreover we observe an appealing (empirical) property: the rank of the iterates often remains bounded by $\operatorname{rank}(\boldsymbol{L}_o)$ throughout the optimization, allowing them to be computed especially efficiently. APG, on the other hand, does not have this property.

5.9 Exercises

5.1 (RPCA as an underdetermined linear inverse problem)**.** *Consider the space* V *of pairs* $(\boldsymbol{L}, \boldsymbol{S}) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times n}$. *This is a vector space over* $\mathbb{R}$. *Consider the function*

$$\|\cdot\|_{\diamondsuit} : \mathsf{V} \to \mathbb{R} \tag{5.9.1}$$

via

$$\|(\boldsymbol{L}, \boldsymbol{S})\|_{\diamondsuit} = \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 . \tag{5.9.2}$$

Show that $\|\cdot\|_{\diamondsuit}$ *is a norm on* V*, by showing that it satisfies the axioms of a norm. For* $\boldsymbol{x} = (\boldsymbol{L}, \boldsymbol{S})$ *in* V*, let* $\mathcal{A}[\boldsymbol{x}] = \boldsymbol{L} + \boldsymbol{S}$. *Interpret the PCP problem as*

$$\min \|\boldsymbol{x}\|_{\diamondsuit} \quad st \quad \mathcal{A}[\boldsymbol{x}] = \boldsymbol{Y}. \tag{5.9.3}$$

5.2 (A Monotonicity Property of PCP)**.** *Call* $\boldsymbol{S}'$ *a* trimmed version *of* $\boldsymbol{S}$ *if* $\operatorname{supp}(\boldsymbol{S}') \subset \operatorname{supp}(\boldsymbol{S})$ *and* $\boldsymbol{S}'_{ij} = \boldsymbol{S}_{ij}$ *whenever* $\boldsymbol{S}'_{ij} \neq 0$. *Prove that whenever* $(\boldsymbol{L}_o, \boldsymbol{S}_o)$ *is the unique optimal solution to the PCP problem with data* $\boldsymbol{Y}_o = \boldsymbol{L}_o + \boldsymbol{S}_o$, $(\boldsymbol{L}_o, \boldsymbol{S}')$ *is the unique optimal solution to the PCP problem with data* $\boldsymbol{Y}' = \boldsymbol{L}_o + \boldsymbol{S}'$.

5.3 (Derandomizing the signs)**.** *In this exercise, we "derandomize" the signs in the RPCA problem, using the elimination property from Exercise 5.2. Suppose that for a given* $\boldsymbol{L}_o$*, RPCA succeeds with high probability when* $\operatorname{sign}(\boldsymbol{S}_o)$ *is a* Bernoulli(ρ_s)*-Rademacher matrix. Prove that with at least the*

[5] In our experience, the optimal choice may depend on the relative magnitudes of the $\boldsymbol{L}$ and $\boldsymbol{S}$ terms and the sparsity of the corruption.

same probability, RPCA succeeds when $\mathfrak{S} \sim_{\text{iid}} \text{Ber}(\rho_s/2)$*, and* $\text{sign}(\boldsymbol{S}_o) = \mathcal{P}_{\mathfrak{S}}\bar{\boldsymbol{\Sigma}}$ *for some fixed matrix of signs* $\boldsymbol{\Sigma} \in \{\pm 1\}^{n\times n}$.

5.4 (Least Squares Solution for the Dual Certificate)**.** *Here we study the solution to* (5.3.30) *in the form of* (5.3.31).

5.5 (Dense Errors with Random Signs)**.** *Prove that with an appropriate choice of* λ*, PCP can handle any constant fraction* $\rho_s < 1$ *of errors.*

5.6 (Very sparse, deterministic errors)**.**

5.7 (Planted Clique via RPCA)**.** *In the* planted clique *problem, we are given a large graph. Determine (i)* $\text{rank}(\boldsymbol{L}_o)$*, (ii)* $\nu(\boldsymbol{L}_o)$*, (iii)* $\nu_\infty(\boldsymbol{L}_o)$*. How big does the clique* C *need to be for RPCA to succeed?*

5.8 (Lower Bounds from Planted Clique)**.**

5.9 (Finding Planted Cliques)**.** *Develop an experiment to test your ....*

5.10 (Find Maximum Low-rank Matrix)**.** *Given an* $n \times n$ *matrix* $\boldsymbol{M}$*, find the largest submatrix* $\boldsymbol{S}$ *such that*

$$\text{rank}\,(\boldsymbol{S}) \leq r$$

for some given small rank r.

5.11 (Matrix Rigidity and NP-Hardness of Low-Rank and Sparse Separation)**.**

5.12 (Sparse Subspace Clustering*)**.** *Sparse Subspace Clustering (SSC) Program a MATLAB function for SSC.*

5.13 (Low-Rank Representation*)**.** *Low-Rank Representation (LRR) (Liu et al, PAMI 2012) is an extension of RPCA:*

$$\|\boldsymbol{Z}\|_* + \lambda\|\boldsymbol{E}\|_{2,1} \quad s.t.\ \boldsymbol{X} = \boldsymbol{X}\boldsymbol{Z} + \boldsymbol{E}.$$

Program a MATLAB function for LRR and use it for noise removal on frontal face images.

5.14 (Background Subtraction*)**.** *Code a MATLAB program that utilizes Robust PCA to separate the foreground images and background images in video sequences captured by stationary cameras.*

5.15 (Robust Texture Inpainting*)**.** *Code a MATLAB program that utilizes Robust PCA to perform texture in painting to compensate corrupted texture images without knowing the location of the corrupted pixels:*

```
(I_hat, E) = robust_inpainting(I),
```

where `I` *is the input texture image,* `I_hat` *is the recovered texture image, and* `E` *is the detected corruption in the same image space.*