UNIVERSITY OF CALIFORNIA SAN DIEGO

Sparse Recovery and Representation Learning

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Mathematics

by

Jingwen Liang

Committee in charge:

Professor Rayan Saab, Chair
Professor Jelena Bradic
Professor Massimo Franceschetti
Professor Philip E. Gill
Professor Tara Javidi

2020

The dissertation of Jingwen Liang is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California San Diego

2020

DEDICATION

To my loving family, friends and my advisor.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

First of all, I would like to express my sincere gratitude to my advisor, Professor Rayan Saab, who has guided me into the math world during the past several years. I benefit so much from his deep insights into the problem, and constant encouragement. Thank him for those hours we spent together in front of your white board, the discussions during our team seminar, and the critiques of my writing and talks. I am very grateful and feel extremely lucky to have him as my advisor.

My sincerely thanks goes to Professor Bredic, Professor Gill, Professor Javidi and Professor Franceschetti for serving as my committee members and offering meaningful discussions.

I have had the honor to work with Han Liu, Yiwei Zhao, Maziar Sanjabi, Mohsen Sardari, Harold Chaput, Navid Aghdaie, Kazi Zaman in Electronic Arts and Aleksandar Bradic, Yenting (Greg) Lin, Yiqing Huang and Yi-Hsuan (Griffey) Kao at SupplyFrame. I would like to thank them for hosting me as an intern and helping me gain industry experience.

I appreciate all the resources and support provided from the Math Department.Thanks to Wilson Cheung, Holly Proudfoot, Scott Rollans, Kelly Guerriero and Debbie Shon for all your patience and assistance. Thanks Brian Preskitt, Eric Lybrand, Aaron Nelson, Anna Ma, Thang Huynh in my research group for interesting ideas and discussions and all the talk. Also I thank all my friends, Yiqing Huang, Jiaying Liu, Yuchao Liu, Chenchen Gu, Hanbo Li, Andrew Ying, Yaqing Wang, Ching-Wei Ho, Marquis Hou, and Selene Xu, Terry Le, Jiayi Li, Zhihua qi for sharing food, interesting ideas, and for gaming time. Greatest thanks to San Diego, the city that I lived for seven years, who always gives the warms welcome to people with its fantastic weather

Last but not least,I would like to give my parents my heartfelt gratitude and appreciation. Thank you both for giving me infinity love and support. Finally, to my dog Pumpkin and my two cats Coffee and Milk, who offered constant company and emotional support when I spent thousands of hours doing research.

Chapter 2, in full is currently being prepared for submission for publication of the material,

co-authored with Saab, Rayan. The dissertation author was the primary investigator and author of this material.

Chapter 3, in full is currently being prepared for submission for publication of the material, co-authored with Saab, Rayan. The dissertation author was the primary investigator and author of this material.

Chapter 4, in full, is a version of the paper “Building Placements in Urban Modeling Using Conditional Generative Latent Optimization.” The paper is co-authered with Han Liu, Yiwei Zhao, Maziar Sanjabi, Mohsen Sardari, Harold Chaput, Navid Aghdaie, Kazi Zaman. The manuscript has been submitted for publication. The dissertation author was the primary investigator and author of this paper.

VITA

2013	Bachelor of Science in Statistics, Southwestern University of Economics and Finance, Chengdu, China
2014-2019	Graduate Teaching Assistant, University of California San Diego
2020	Doctor of Philosophy in Mathematics, University of California San Diego

PUBLICATIONS

Jingwen Liang, Han Liu, Yiwei Zhao, Maziar Sanjabi, Mohsen Sardari, Harold Chaput, Navid Aghdaie, Kazi Zaman. “Building Placements in Urban Modeling Using Conditional Generative Latent Optimization.”, *IEEE International Conference on Image Processing (ICIP 2020)*, submitted.

ABSTRACT OF THE DISSERTATION

Sparse Recovery and Representation Learning

by

Jingwen Liang

Doctor of Philosophy in Mathematics

University of California San Diego, 2020

Professor Rayan Saab, Chair

This dissertation focuses on sparse representation and dictionary learning, with three relative topics. First, in chapter 1, we study the problem of low-rank matrix recovery in the presence of prior information. We first study the recovery of low-rank matrices with a necessary and sufficient condition, called the Null Space Property, for exact recovery from compressively sampled measurements using nuclear norm minimization. Here, we provide an alternative theoretical analysis of the bound on the number of random Gaussian measurements needed for the condition to be satisfied with high probability. We then study low-rank matrix recovery when prior information is available. We analyze an existing algorithm, provide the necessary and sufficient conditions for exact recovery and show that the existing algorithm is limited in certain cases.

We provide an alternative recovery algorithm to deal with the drawback and provide sufficient recovery conditions based on that.

In chapter 2, we study the problem of learning a sparsifying dictionary of a set of data, focusing on learning dictionaries that admit fast transforms. Inspired by the Fast Fourier Transform, we propose a learning algorithm involving $O(N)$ unknown parameters for a $N \times N$ linear transformation matrix. Empirically, our algorithm can produce dictionaries that provide lower numerical sparsity for the sparse representation of images than the Discrete Fourier Transformation (DFT). Additionally, due to its structure, the learned dictionary can recover the original signal from the sparse representation in $O(N \log N)$ computations.

In chapter 3, we study the representation learning problem in a more complex setting. We use the concept of dictionary learning and apply it in a deep generative model. Motivated by an application in the computer gaming industry where designers needs to have an urban layout generation tool that allows fast generation and modification, we present a novel solution to synthesize high quality building placements using conditional generative latent optimization together with adversarial training. The capability of the proposed method is demonstrated in various examples. The inference is nearly in real time, thus it can assist designers to iterate their designs of virtual cities quickly.

Chapter 1

Introduction and Background

1.1 Compressed Sensing and low-rank matrix recovery with prior informations

In signal processing, we often assume that there is a map f mapping from the representation space to the space of signals that we interested:

$$y = f(x) \tag{1.1}$$

In this dissertation, we study three topics relevant to signal representation. First, define the sparsity of a vector as number of non-zero elements of it, we consider sparsity and compressed sensing. For a k-sparse vector $x \in \mathbb{R}^N$, let its measurements $y \in \mathbb{R}^m$ be given by

$$y = Ax \tag{1.2}$$

where $A \in \mathbb{C}^{m \times N}$ is a linear measurement matrix. When $m \leq N$, (1.2) is an underdetermined system with infinitely many solutions (provided that there exists at least one). Nevertheless, compressed sensing theory shows that it is possible to exactly recover all such x from y by solving the ℓ_1 minimization problem

$$\underset{x}{\text{minimize}}\, \|x\|_1 \text{ subject to } y = Ax$$

if the measurement matrix A satisfies certain properties [Don06] [CT06] [CRT06a] and if x is sparse enough. These conditions are satisfied with high probability by many classes of random matrices, for example, matrices with i.i.d. random Gaussian entries with $m \geq Ck \log(N/k)$ for some constant C.

In compressive sensing, we wish to recover sparse signals from compressed measurements. Generalizing the unknown sparse vector x to an unknown low rank matrix X, we have the following problem of recovering matrix $X \in \mathbb{R}^{N_1 \times N_2}$ from

$$y = \mathscr{A}(X)$$

where $\mathscr{A} : \mathbb{R}^{N_1 \times N_2} \to \mathbb{R}^m$. It is been shown that we can successfully recover all low-rank matrices with rank at most r, by solving optimization problem

$$\underset{X}{\text{minimize}}\, \|X\|_* \text{ subject to } y = \mathscr{A}(X), \tag{1.3}$$

if $\mathscr{A}$ satisfies certain properties [RFP10], [CR09], [Rec11], [CT10], [Gro11], where $\|X\|_*$ is the nuclear norm defined as sum of singular values of X.

Define the support of a sparse vector x as the index set indicating non-zero elements of x, and the support of a low-rank matrix X by matrices whose columns are basis vectors that span X's column and row spaces: $\{U, V\}$. While it is shown that ℓ_1 minimization is suitable for recovering all k-sparse signals with arbitrary support set, and the nuclear norm minimization can recover all rank r low-rank matrices with arbitrary support, it is often the case that in many applications, there is useful (but possibly imperfect) prior knowledge about the signals and matrices – for example, as an estimate of the support. In such cases, with more information, one may modify the optimization problem and hopefully weaken the exact recovery conditions.

We will study the problem of solving X from undetermined linear system $y = \mathscr{A}(X)$ given X is a sparse vector or low-rank matrix, as well as a prior estimate of the support. This is the first topic of this dissertation.

1.2 Learning Dictionary with Fast Transforms

The first topic considers the problem of recovering x from $y = f(x)$ assuming that f is a linear transformation A that is given. We consider optimization problems for accomplishing this task, and study the recovery conditions that guarantee exact recovery. However, in some applications, we are only given several instances of y without knowing a transform with which y admits a sparse representation. Our goal here is not signal recovery, but rather learning a sparsifying transform D such that $y = Dx$ and x is sparse. Thus in 2nd topic, we will study the problem of learning a linear transformation (often called dictionary) that can make a set of signals have sparse representation.

Here, we assume that the signal $y \in \mathbb{C}^N$ can be described as

$$y = Dx,$$

where $D \in \mathbb{C}^{N \times n}$ is a linear mapping called a dictionary, $x \in \mathbb{C}^n$ is the representation of the signal y under the dictionary D, and is assumed to be sparse.

When applying the transformation that leads to spare representation, we can either choose a pre-specified set of functions or use the dictionary that is learned from the given set of signals that need to be sparsified. Choosing from pre-constructed dictionaries such as wavelet basis[ABMD92], curvelet basis [DD00] and Fourier basis [BB86] often leads to fast algorithms for the computation of the sparse representation and original signal recovery. For example Fast Fourier Transform (FFT) [Nus81] computes the discrete Fourier Transform (DFT) of a n-dimensional signal with $O(n \log n)$ complexity instead of $O(n^2)$ and the fast Wavelet Transform (FWT) takes only $O(n)$ in certain cases [Cod92].

Another route of choosing dictionaries is to design a learning algorithm and find that dictionary that fit for the given set of signals. Unlike those pre-constructed dictionaries that may be limited in their ability to sparsify the signals they are designed to handle and often restricted to signals or images of a certain type, a dictionary learned from data can potentially be trained to fit arbitrary family of signals of interest and gain better sparsity results. Indeed, it has been shown that using a learned dictionary from training data rather than fixed frames like Fourier or Wavelets basis derives better results in many practical applications such as face recognition [HDRL16], [ZL10], image de-noising [DLZS11], [EA06], image super-resolution [YWL$^+$12], [ZZX$^+$12] and image segmentation [ZZM12]. On the other hand, dictionaries learned from data usually do not exhibit the kind of structure that yields a fast transform. Thus, applying these dictionaries to a vector will usually require $O(n^2)$ operations, which when n is large can be prohibitive, especially if the dictionary needs to also be applied to a large data-set.

Thus in the second topic, we will introduce an approach to dictionary learning which combines the computational efficiency of certain model-based sparsifying transforms – such as the Discrete Fourier Transform, with the advantages of data-driven dictionaries. Similar to traditional dictionary learning algorithms, we will modify the optimization problem

$$\underset{D \in \mathbb{C}^{N \times n}, x_i \in \mathbb{C}^n}{\text{minimize}} \sum_{i=1}^{p} \|y_i - Dx_i\|_{\ell_2} + \lambda \|x_i\|_{\ell_1}. \tag{1.4}$$

by factorizing D into product of a sequence of matrices and learn those factors together with the sparse representation. Our algorithm is based on alternating minimization, where we minimize a sequence of convex functions, and find $n \times n$ dictionaries that can be applied in $O(n \log n)$ time.

That is, we use data to learn dictionaries that admit a fast algorithm for the corresponding linear transform. This is the second topic of this dissertation.

1.3 Deep Generative Model Using Representation Learning Techniques

After we learn the dictionary, we can apply them to new data which is assumed to have the same distribution of the training data that we used when learning the dictionary and find the sparse representation. In some applications, however, it is not enough to assume that the relationship between signals and their representations is linear. For example, if we have a dictionary that can sparsely represents natural images. We cannot expect that feeding a random sparse signal to the learnt dictionary can return us a natural image. For applications such as image generation, we need more complex assumptions to model the relationship between the given signal and its representation.

In the last topic, we will study the topic that apply the representation learning techniques in deep learning. We will propose a conditional generative model using representation learning and its application of generating building placement of a certain neighborhood given the information about its environment such as road networks, waterbodies and vegetations. Our work is motivated by the fact that practitioners in the gaming industry often encounter the issue that building layouts on most available maps are largely missing especially in remote areas. We are aiming at training a conditional generative model that can take simple inputs conditions as we listed above and generate the possible placement of buildings around them under different styles given the same input.

In our model, we assume that we have a set of samples $\{(X_i \,|\, C_i)\}$, $i = 1, \cdots, N$, where $X_i \in \mathbb{R}^{n \times n}$ is the i-th sample of $n \times n$ building placement and $C_i \in \mathbb{R}^{5 \times n \times n}$ is the associated input condition, each of it contains 5 $n \times n$ channels i.e. the highways, arterial streets, local roads, waterbodies and vegetations. We are going to train a generator G_ω that takes inputs C and generates a building placement X. Besides this, we also train the model to learn an lower dimensional representation z_i for each training samples X_i, z_i are called latent representation that suppose to encode informations about the building layout of example i and can be used as style source (after training) in the inference for generating the new examples. Thus the generator should have from

$$G_\omega(z\,|\,C) = (X\,|\,C)$$

The generator it self can be trained separately by doing the following optimization

$$\underset{\omega,z}{\text{minimize}} \sum_{i=1}^{N} \ell(G_\omega(z_i|C_i),(X_i|C_i)),$$

where $\omega \in \Omega$ represents the parameters of the generator G_ω, and $\ell(x,x')$ is some predefined distance metric measures the difference between the generative example and the ground truth. In order to enhance the training we also introduce the discriminator. The job of the discriminator is to learn an efficient classifier so that it can tell whether its input is a real data or a generated fake data. In this way, the discriminator can help the generator to learn the map from conditional latent space to the data space bater and faster.

1.4 Contributions

In chapter 2, we study the problem of recovering matrix X from compressed linear sampling

$$y = \mathscr{A}(X)$$

given that X is low-rank matrices as well as the problem when extra prior information about support of interested matrix is provided. First, we study the nuclear norm minimization for low-rank matrix recovery and provide an alternative proof for sampling complexity in regular low-rank matrix recovery without prior information with Random Gaussian Matrix. Then, we studied the low-rank matrices recovery with prior information using weighted nuclear norm minimization with random Gaussian Matrix. We analyze an existing algorithm, provide the necessary and sufficient conditions for exact recovery and show that the existing algorithm is limited in certain cases. We provide an alternative recovery algorithm to deal with the drawback and provide sufficient condition for exact recovery based on that. We show that when using correct prior information, we can dramatically reduce the number of measurements need for exactly recovery. Finally, we present the numerical experiment about the low-rank matrix recovery with prior information to support our analysis.

In chapter 3, we propose an algorithm which can learn the fast factorization of a linear dictionary as well as the approximate sparse representation from the given training data. In our proposed algorithm, we learned a $N \times N$ linear transformation matrix use $O(N)$ degree of freedom. Besides, the output of the algorithm forms a factorization of the dictionary which can obtain fast recovery by $O(N \log N)$ calculations instead of $O(N^2)$. We also provide numerical experiments on image data and show that our algorithm can successfully learn a fast transform that achieves sparse representation that outperform the 2D Discrete Fourier Transform in terms of sparsity.

In chapter 4, firstly, we propose conditional generative model to control the generation on user-specified maps while training representations of given examples. Then we enhance the generator by adversarial training in order to learn more realistic and generic neighborhood styles, and decouple the representation of the target examples from associated input conditions. Last but not least, we formulate the problem of building placements in the scope of image synthesis and format the map data in individual channels. In this way, the data set can also be used for other tasks such as road generation.

Chapter 2

Signal Recovery with Prior Information

2.1 Introduction

Classical data compression as used in such standards as MP3 and JPEG, starts from the characteristics of the data itself, then finds and eliminates redundancy. A signal is called sparse if most of its elements are zero, similarly, a signal is called compressible if it can be well approximated by sparse signals. Suppose one want to compress a signal $x \in \mathbb{R}^N$. A common procedure is to acquire the samples $y \in \mathbb{R}^N$ of the signal, and convert them to some representation where the signal is sparse or compressible, then sort all the samples after the quantization and keep the largest k elements if the signal is k-sparse or approximately k-sparse in that representation. In this procedure, the compression occurs after the data has been collected, N measurements are made (sensing), but only $k \ll N$ coefficients are kept in the compression, which is a waste of resources. Thus one wonders if there is a better way which can compress at the time of sensing. The problem becomes how can we design the sampling and recovery so that one can use less measurements to (simultaneously) compress a signal without losing too much information.

2.1.1 Compressive Sensing

Consider the signal $z \in \mathbb{R}^N$, that is sparse under some representation system (e.g., a basis), so that $z = \Phi x$, and x is sparse. We sample the data z from $y = \tilde{A}z$, where $y \in \mathbb{R}^m$ is the linear measurement of z. Knowing Φ, we can rewrite our equation as $y = \tilde{A}\Phi x =: Ax$. Then, if we can recover x from y, we can simply recover z as $z = \Phi x$. In compressed sensing, we can only collect $m \ll N$ measurements and recover x from y exactly by solving the under-determined system $y = Ax$. Since the system usually has infinitely many solutions, one common way to solve it is through "zero norm" minimization:

$$\underset{z}{\text{minimize}} \, \|z\|_0 \quad \text{subject to} \quad Az = Ax, \tag{2.1}$$

where $\|x\|_{\ell_0}$ norm defined as number of non-zero elements of the vector x. However, ℓ_0 norm minimization is a NP-hard problem. Instead, one often uses ℓ_1 norm (sum of the absolute value of the vector) minimization as a convex relaxation:

$$\underset{z}{\text{minimize}} \, \|z\|_1 \quad \text{subject to} \quad Az = Ax \tag{2.2}$$

The problem (2.2) can then be written as linear programming and be solved efficiently in polynomial time complexity [BV04]. It is shown by [CRT06b] and [Don06], that ℓ_1 minimization can stably and robustly recover x from "incomplete" and inaccurate measurements $y = Ax + e$, when A satisfies certain properties which hold for certain random matrices with high probability, for example, matrices with entries as i.i.d. Gaussian random variables when $m \gtrsim k \log(N/k)$ (where k is the sparsity of the target signal).

One of the above properties of the measurement matrix A that we are interested in this dissertation is called Null Space Property (NSP):

Definition 1 (Null Space Property of order k [CDD09])**.** A matrix $A \in \mathbb{R}^{m \times N}$ is said to have the

Null Space Property of order k and constant $0 < C < 1$ if for any vector $h \in \text{Null}(A) \backslash \{0\}$, and for every index set $T \subset \{1, 2, \cdots, N\}$ with $|T| \leq k$, we have

$$\|h_T\|_1 \leq C\|h_{T^c}\|_1.$$

The NSP requires every vector in null space of A is non-sparse. It is a necessary and sufficient condition for the recovery of all k-sparse vector x from $y = Ax$ [CDD09].

2.1.2 Low-rank matrix recovery

Besides vectors, we are also interested in low-rank matrices measurement and recovery. Low-rank matrices have wide usage in applied mathematics and scientific research, including famously in the Netflix prize problem [Kor09], collaborative filtering [RYRD15], MRI [LHDJ11] and quantum state tomography [KRT17]. With the growth of the size of data, fully observing or sampling the matrix become harder. In this case, we either have highly incomplete observation or we need a more efficient method to compress the matrix. Recovering the full matrix from incomplete data or insufficient data become very important.

In low-rank matrix recovery, we are aiming at finding a low-rank matrix $X \in \mathbb{R}^{n_1 \times n_2}$, $rank(X) = r \ll \min\{n_1, n_2\}$, from $y = \mathscr{A}(X)$. We can write $X = \sum_{i=1}^{r} \sigma_i u_i v_i^T$ where $\sigma_1, \cdots, \sigma_r \geq 0$ are the singular values of X and vectors $u_1, u_2, \cdots, u_r \in \mathbb{R}^{n_1}$ and $v_1, v_2, \cdots, v_r \in \mathbb{R}^{n_2}$ are the left and right singular vectors of X respectively. The collection of all these matrices constitutes a union of subspaces in $\mathbb{R}^{n_1 \times n_2}$: each set of $\{\{u_i\}_{i=1}^{r}, \{v_i\}_{i=1}^{r}\}$ defines a r-dimensional subspace, and the $\{\sigma_i\}$'s are the coefficients corresponding to an element in that subspace. The union contains uncountably many such subspaces since the $\{u_i\}$ and $\{v_i\}$ can vary continuously.

When we count the degrees of freedom in a rank r matrix, it can be represented by $O(r(n_1 + n_2))$ parameters, which is much smaller than $O(n_1 n_2)$ when r is relative small. It has been shown that it is possible to exactly recover X from fewer measurements in [RFP10] [CR09]

[Rec11] [CT10] [Gro11] through $y = \mathscr{A}(X)$.

Here again, one approach to recover the full matrix from the incomplete measurements is through optimization. The task is to recover the matrix $X \in \mathbb{R}^{n_1 \times n_2}$ with rank $r \ll \min\{n_1, n_2\}$, from linear measurements y with $y = \mathscr{A}(X)$, where $\mathscr{A}$ is a linear measurement mapping from $\mathbb{R}^m$ to $\mathbb{R}^{n_1 \times n_2}$ [Faz02], $\mathscr{A}(X) = \sum_{i=1}^m \langle A_i, X \rangle e_i$. This usually lead to non-convex formulations:

$$\underset{X}{\text{minimize}}\, \text{rank}(Z) \quad \text{subject to} \quad \mathscr{A}(Z) = \mathscr{A}(X), \tag{2.3}$$

which are NP hard in general. Researchers have developed convex relaxations that succeed with high probability under appropriate assumptions, to deal with this issue [CP11], [Gro11], [DR16]. Define the nuclear norm $||Z||_*$ as the sum of singular values of a matrix Z. In particular, one can exactly recover all rank r matrices X from y by solving the nuclear norm minimization problem

$$\begin{aligned} \underset{Z}{\text{minimize}} \quad & ||Z||_* \\ \text{subject to} \quad & \mathscr{A}(Z) = \mathscr{A}(X), \end{aligned} \tag{2.4}$$

if $\mathscr{A}$ satisfies certain conditions. For example, $\mathscr{A}$ satisfying the *Null Space Property* (i.e. for every matrix $H \in \text{Null}(\mathscr{A})$ and $H \neq 0$,

$$\sum_{i=1}^{r} \sigma_i(H) < \sum_{i=r+1}^{n} \sigma_i(H),$$

where σ_i represents the ith singular value of H, and $n = \min\{n_1, n_2\}$), is the necessary and sufficient condition for recovering all matrix with rank no larger than r from the nuclear norm minimization (2.4). Moreover, random Gaussian measurement with $m = O(r(n_1 + n_2))$ satisfies the *Null Space Property* with high probability [FR13].

2.1.3 Prior Information for Compressive Sensing and Low-rank matrix recovery

Define the support of a sparse vector x as the index set indicating non-zero elements of x, and the support of a low-rank matrix X by matrices whose columns are basis vectors that span X's column and row spaces: $\{U,V\}$. In fact, in the compressive sensing problem and the low rank matrix recovery problem, the most challenging part is to identify the support of the target. If the support of a sparse vector is known, one can solve for x directly by least square minimization restricted to the known support, which then only needs k measurements; if the support of a low-rank matrix is known, than one can only search over the space of the row and column span using at most r^2 linear measurements.

In real application, it is often the case that there is useful (but possibly imperfect) prior knowledge about the signals and matrices – for example, as an estimate of the support. In wireless sensor networks [BHSN06], the information of one of the sensors in the network, can be used as an estimate of the support for another sensor. In real time video reconstruction [WLD$^+$06], the support of previous frames can be used as prior information of next frame. In medical imaging processing [LDSP08], MRI imaging information from one slice can be used as prior information for an adjacent slice. In recommender system, similar users share similar interest in products and knowing prior information of a subset of user ratings of a particular item may provide prior subspace information about the ground truth rating matrix. In this case, with more information, we are interested in modifying (2.2) and (2.4) to hopefully weaken the exact recovery conditions.

In this chapter, we analyze the recovery methods for low-rank matrices that incorporate support information. To modify the nuclear norm minimization (2.4), we first define the column space and row space of X by U and V, and the estimate of column space and row space as $\widetilde{U}$ and $\widetilde{V}$, let $\mathcal{P}_{(\cdot)}$ be the orthogonal projection onto the corresponding subspace. We study two modifications

$$\underset{Z}{\text{minimize}}\, \|(w\mathcal{P}_{\widetilde{U}} + \mathcal{P}_{\widetilde{U}^{\perp}})Z(\lambda\mathcal{P}_{\widetilde{V}} + \mathcal{P}_{\widetilde{V}^{\perp}})\|_* \quad \text{subject to } \mathscr{A}(Z) = \mathscr{A}(X), \tag{2.5}$$

and

$$\underset{Z}{\text{minimize}}\, \|(w\mathcal{P}_{\widetilde{U}} + \mathcal{P}_{\widetilde{U}^{\perp}})Z\|_* + \|Z(\lambda\mathcal{P}_{\widetilde{V}} + \mathcal{P}_{\widetilde{V}^{\perp}})\|_* \quad \text{subject to } \mathscr{A}(Z) = \mathscr{A}(X). \tag{2.6}$$

In both optimization problems, we penalize the subspace orthogonal to our support estimate more when $0 < w < 1$, $0 < \lambda < 1$.

2.1.4 Related Work

Compressive Sensing

As we discussed before, compressive sensing appears as an alternative to the traditional sampling theory, endeavoring to reduce the required number of samples for successful signal reconstruction. It studies the recovery algorithms and the conditions for successful recovery of the original signal when taking far fewer measurements then its ambient dimension. We've mentioned that solving ℓ_1 minimization:

$$\underset{z}{\text{minimize}}\, \|z\|_1 \text{ subject to } Az = Ax$$

can stably and robustly recover x from compressed sampled measurements, when the measurement A satisfies certain properties.

For example, for $A \in \mathbb{C}^{m\times N}$ with ℓ_2 normalized columns $a_1, a_2, \cdots, a_N$, the coherence, defined as

$$\mu := \max_{j\neq k} |\langle a_j, a_k\rangle|$$

can be used to measure the quality of A. Compressive Sensing using the incoherence of the measurement matrix [DE03],[CR07] gives a verifiable sufficient condition for the recovery of all k-sparse x from $y = Ax$:

$$(2k-1)\mu \leq 1. \tag{2.7}$$

On the other hand, coherence suffers from a bottleneck, namely that if $A \in \mathbb{R}^{m\times n}$ then $\mu \gtrsim m^{-1/2}$ [FR13], which means that the sparsity of the signals that we can guarantee recovery of scales only like $\sqrt{m}$, where m is the number of measurements.

Restricted isometric property (RIP) is another common tool for the performance analysis of Compressive Sensing recovery algorithms. The restricted isometry constant of a matrix A is defined as the smallest non - negative δ_k such that

$$(1-\delta)\|x\|_2^2 \leq \|Ax\|_2^2 \leq \|(1+\delta)\|x\|_2^2,$$

for all k - sparse x.

Candès and Tao showed in [CT05] that the condition $\delta_k + \delta_{2k} + \delta_{3k} < 1$ guarantees exact k - sparse recovery through ℓ_1 minimization. Further in [CRT06b], Candès et al. showed that to have stable and robust k - sparse recovery via ℓ_1 - minimization, the sufficient condition is $\delta_{3k} + 3\delta_{4k} < 2$. Later in [Can08], Candès further refined the sufficient condition for stable and robust k - sparse signal recovery as $\delta_{2k} < \sqrt{2} - 1 \approx 0.414$. Then this result was improved in a series of papers, e.g., [AS14], [CWX10], [CWX09], [Fou10], [FL09], [ML11]. Many random constructions of measurement matrices satisfy these RIPs with very high probability when $m \gtrsim k \log n$ [FR13], [BDDW08], [MPTJ08], [Rau10], so the number of measurements now scales linearly with the sparsity, which is better than it was with coherence.

The null space property of order k [CDD09] we mentioned before is a necessary and sufficient condition on the measurement matrix A for the success of exact recovery of all k - sparse vectors x from linear measurement $y = Ax$ via ℓ_1 minimization. Basically, it requires that every vector in the null space of A is not sparse.

Compressive Sensing with prior information

While initially assuming sparsity of the target signal as the only characteristic during recovery, researchers have more recently also studied other forms of prior information to enhance the reconstruction in Compressed Sensing. Several works have studied the recovery of compressively sampled signals using prior information. Chen, Tang, and Leng proposed a method that utilized prior information in the form of a similar signal [CTL08]. Their method, the Prior Image Constrained Compressed Sensing (PICCS) algorithm, introduce the prior information as a known similar signal. They modify the objective function of the optimization into two parts, one for measuring the sparsity of the target signal, the other term is for measuring the sparsity of the difference between the prior information signal and the target signal. PICCS is implemented by solving the following constrained minimization

$$\underset{x}{\text{minimize}}\left[\alpha\|D_1(x-x_p)\|_1+(1-\alpha)\|D_2x\|_1\right], \quad \text{such that } Ax=Y$$

where D_1 and D_2 are the sparsifying transforms, and A is the linear measurement matrix, x_p represents the prior signal. Similarly, Mota, Deligiannis and Rodrigues also investigated the problem with the prior information as a known similar signal [MDR17]. The given similar signal is been integrated into the algorithm via minimizations as below:

$$\underset{x}{\text{minimize}}\;\|x\|_1+\beta\|x-x_p\|_1 \quad \text{subject to } Ax=y$$

or

$$\underset{x}{\text{minimize}}\;\|x\|_1+\frac{\beta}{2}\|x-x_p\|_2 \quad \text{subject to } Ax=y$$

where x_p represents the the prior known similar signal and $\beta > 0$ is an estimate of the trade-off between signal sparsity and reliability of the prior information x_p. With the similar prior information, Weizman, Eldar and Bashat proposed a optimization that is close to that of previous two algorithms but with weights in both terms of the objective function [WEBB15]:

$$\underset{x}{\text{minimize}}\, \|W_1 D_1 x\|_1 + \lambda \|W_2(x - x_p)\|_1 \quad \text{subject to } \|Ax - y\| \leq \varepsilon$$

Where W_k is a diagonal matrix, $W_k = diag([w_k^1, w_k^2, \cdots, w_k^N])$, and w_k^i controls the weight given to each element of term 1 or term 2. Adding W_1 to term 1 relaxes the demand for sparsity on the elements in the support of the signal in its sparse transform domain. Adding W_2 to term 2 controls the demand for similarity between x and x_p, enforcing sparsity only in signal regions where x and x_p are similar [WEBB15]. They use the algorithm on longitudinal Magnetic Resonance Imaging (MRI) and assuming that the prior information coming from a previous MRI scan.

Other than using a similar signal as prior information, the work of Khajehnejad, Xu, Avestimehr and Hassibi [KXAH09], [KXAH11], assigns a probability of being zero or nonzero to each entry of the target signal. They assume a non-uniform sparsity model and analyze the weighted ℓ_1 - minimization while allowing for non-uniform weights in noise-free case. In their algorithm, the entries of the target signal are divided into two sets K_1 and K_2 with size n_1 and n_2 that partition $\{1, 2, \cdots, N\}$, the elements in the first and the second set are assigned with probability P_1 and P_2 to be nonzero respectively, $P_1 \neq P_2$. The authors proposed a weighted ℓ_1 - minimization where the ℓ_1 norm for different sets are given different weights W_1 and W_2. The weights are chosen according to the prior probability information. Later in, [KOH12], Khajehnejad et al. extend their result for prior information with more than two sets and provide a heuristic for estimating the optimal weights. They further show that their weighted ℓ_1 - minimization is substantially better than regular ℓ_1 - minimization. Based on the same kind of prior information, the work of [MP15] shows that when the measurements are obtained using a matrix with independent identically distributed Gaussian entries, weighted ℓ_1 - minimization successfully recovers the sparse signal from its measurements with high probability. While in [SED12], Scarlett, Evans and Dey proposed the prior information as the probability of being non zero for each entry. The authors provide the theoretical limits on the number of measurements needed to recover the support set perfectly, and show that significantly fewer measurements can be used if the prior distribution is sufficiently non-uniform.

Another type of prior information that have been studied is in the form of the support estimate. Weighted ℓ_1 minimization, with zero weights on the known support, was proposed by Vaswani and Lu in [VL10] and [LV10]. In their work, they assume to have a support estimate $\widetilde{T}$, and solve the minimization problem

$$\underset{x}{\text{minimize}}\, \|x_{\widetilde{T}}\|_1 \quad \text{subject to } Ax = y$$

where $x_{\widetilde{T}}$ denotes the vector x restricted on $\widetilde{T}$. In the noise-free setting, they derive sufficient conditions on exact recovery that are weaker than the analogous ℓ_1 minimization conditions when $\widetilde{T}$ is an appropriate estimate of the real support. In [FMSY11], Friedlander, Mansour, Saab and Yilmaz study the weighted ℓ_1 - minimization:

$$\underset{z}{\text{minimize}} \sum_i w_i |z_i| \quad \text{subject to } \|Az - y\|_2 \leq \varepsilon \quad \text{where } w_i = \begin{cases} w \in [0,1], & i \in \widetilde{T} \\ 1, & i \in \widetilde{T}^c \end{cases}, \tag{2.8}$$

under the *Resiticted Isometry Property (RIP)* of the sensing matrix, which generate the result of [CRT06b] to weighted ℓ_1 - minimization. They show that if at least 50% of the partial support information is accurate, then weighted ℓ_1 - minimization is stable and robust under weaker sufficient conditions than the analogous conditions for standard ℓ_1 minimization. They also proved that weighted ℓ_1 - minimization provides better upper bounds on the reconstruction error in terms of the measurement noise and the compressibility of the signal to be recovered. Mansour and Saab analyzed

$$\underset{z}{\text{minimize}} \sum_i w_i |z_i| \quad \text{subject to } Az = y \quad \text{where } w_i = \begin{cases} w \in [0,1], & i \in \widetilde{T} \\ 1, & i \in \widetilde{T}^c \end{cases}, \tag{2.9}$$

and derive necessary and sufficient conditions i.e. the *weighted Null Space Property (NSP)*, for exact signal recovery from compressively sampled measurements using weighted ℓ_1 norm minimization [MS17]. They shows that this condition can guarantee recovery even when standard ℓ_1 - minimization fails. They also derive bounds on the number of Gaussian measurements for these conditions to be satisfied, and show that weighted ℓ_1 - minimization requires significantly fewer measurements than standard ℓ_1 - minimization when the support estimate is relatively accurate. Bah and Ward study the minimal number of Gaussian measurement needed for robust recovery from weighted ℓ_1 - minimization using weighted sparsity and weighted NSP in [BW16]. The work of Mansour and Yilmaz in [MY11] extends the result from [FMSY11], study the weighted ℓ_1 - minimization problem with multiple support estimate with distinct weights. In [NSW17], Needell, Saab and Woolf study weighted ℓ_1 norm minimization with completely arbitrary weights under RIP.

Low rank matrix recovery

Similar to compressive sensing, when we study low-rank matrix recovery, we are interested in conditions on the linear measurement matrix $\mathscr{A} : \mathbb{R}^{n_1 \times n_2}$ so that we can successfully recovery all rank - r matrices $X \in \mathbb{R}^{n_2 \times n_2}$ from $y = \mathscr{A}(X)$.

For every r with $1 \leq r \leq \min\{n_1, n_2\}$, the r-restricted isometry constant(RIC) is defined to be the smallest constant δ_r such that the following holds for all rank r matrix $X \in \mathbb{R}^{n_1 \times n_2}$:

$$(1-\delta)\|X\|_F^2 \leq \|\mathscr{A}(X)\|_2^2 \leq (1+\delta)\|X\|_F^2. \tag{2.10}$$

where $\|X\|_F := \sqrt{\langle X, X\rangle}$ is the Frobenius norm of X. Recht, Fazel, and Parrilo show that $\delta_{5r} \leq \frac{1}{10}$ is a sufficient condition for recovering all rank r matrix exactly from the nuclear norm minimization (2.4) [RFP10]. They also prove that the nuclear norm minimization succeeds with high probability when the number of available measurements exceeds a constant times $2nr\log n$ for $n \times n$ matrices. Further in [CP10], Candès and Plan show that if $\delta_{4r} \leq \sqrt{2}-1$ then nuclear norm minimization can recovery all matrices with rank no larger than r. Further more, they show that if $\mathscr{A}$ is a random measurement ensemble obeying the following condition: for any given $X \in \mathbb{R}^{n\times n}$ and any fixed $0 < t < 1$,

$$\mathbb{P}(|\|\mathscr{A}(X)\|_2^2 - \|X\|_F^2| > t\|X\|_F^2) \leq Ce^{-cm}$$

for fixed constant C, c, then if $m \gtrsim O(nr)$ then $\mathscr{A}$ satisfies the condition with high probability. Cai and Zhang [CZ13] show $\delta_r < 1/3$ as the sharp RIP bound in the noiseless case for low rank matrix recovery.

Besides RIP, recovery conditions based on Null Space Property for low-rank matrix recovery have also been studied. Oymak, Mohan, Fazel and Hassibi show how recovery conditions can be extended from Compressive Sensing to low-rank matrix recovery [OMFH11]. Kabanava, Kueng, Rauhut and Terstiege provide a necessary and sufficient condition for low-rank matrix recovery by analyzing the Null Space Property [KKRT16]. They show that $O(r(n_1+n_2))$ measurements are enough to uniformly and stably recover all $n_1 \times n_2$ matrix of rank at most r. The authors of [DF10] give sufficient conditions for the exact recovery of all matrices up to a certain rank, and show that these conditions hold with high probability for operators generated from random Gaussian ensembles by analyzing spherical section property.

Low rank matrix recovery with prior information

We can get prior information about the target low-rank matrices. For example, in [XJZ13] the authors study the case when prior information are in forms of column and row vectors when using nuclear norm minimization to solving matrix completion problem, they assume that two matrices $A = (a_1, a_2, \cdots, a_{r_a}) \in \mathbb{R}^{n_1 \times r_a}$, and $B = (b_1, b_2, \cdots, b_{r_b}) \in \mathbb{R}^{n_2 \times r_b}$ with orthonormal columns are the side information matrices. The column vectors in X lie in the subspace spanned by the column vectors in A, and the row vectors in X lie in the subspace spanned by the column vectors in B. They proposed the optimization problem

$$\underset{Z\in\mathbb{r}_{\mathbb{1}}\times\mathbb{r}_{2}}{\text{minimize}} \|Z\|_* \text{ subject to } \mathcal{R}_\Omega(AZB^T) = \mathcal{R}_\Omega(X) \tag{2.11}$$

where $\mathcal{R}_\Omega$ is the sampling operator in matrix completion problem. They showed that with the side information A and B, the number of measurements required can be reduced to $O(r(r_a + r_b)\ln(r_a + r_b)\ln(n_1 + n_2))$. However their work requires having exact column and row vector information about the target matrix, which is not always the case in real application. In [CDH18], the authors further extended the results from prefect prior information[XJZ13] to noisy prior information and noisy observed low-rank matirx $X = X_0 + S_0$, where X_0 is the underlying low-rank matrix and S_0 is the sparse noise; they proposed to recover X_0 jointly in two parts, one part captures information from the side information space as AZB^T, and the other part N captures the information outside the prior information space.

In [AKM+14], the authors propose a weighted extension of nuclear norm minimization:

$$\underset{Z}{\text{minimize}} \|(w\mathcal{P}_{\widetilde{U}} + \mathcal{P}_{\widetilde{U}^\perp})Z(\lambda\mathcal{P}_{\widetilde{V}} + \mathcal{P}_{\widetilde{V}^\perp})\|_* \quad \text{subject to } \mathscr{A}(Z) = \mathscr{A}(X), \tag{2.12}$$

that allows known subspace information to improve the results of matrix completion formulations. Later, Eftekhari, Yang and Wakin study the optimization (2.12) in [EYW18] for both matrix recovery and matrix completion, and provided that reliable prior knowledge reduces the sample complexity of matrix completion and recovery by a logarithmic factor. In [DAH18] Daei and his co-authors studied the same optimization with prior information and provided an algorithm to find the unique optimal weights that minimize the required number of measurements, and show that with optimization (2.12) and the reliable prior information, the proposed convex program with optimal weights requires substantially fewer measurements than the regular nuclear norm minimization in numerical experiments. However, we show that the optimization problem (2.12) is limited when one side of subspace information is perfectly known. In our work, we first analyze the optimization (2.12), provide a necessary and sufficient condition for exact recovery based on that optimization and then analyze why it is limited when one side of prior support estimate is given. Then we propose a new optimization (2.6) problem when partial information about the column and row subspace of the target matrix is known or estimated and show that sample complexity is reduced when the prior information is reliable.

2.1.5 Contributions

For this topic, we provide 3 main contributions: First, we provide an alternative proof for sampling complexity in regular low-rank matrix recovery without prior information with Random Gaussian Matrix. Then, we study low-rank matrices recovery with prior information by modifying the nuclear norm minimization. We consider two optimization problems and the corresponding weighted rank Null Space Property, analyze and compare the sampling number required for random Gaussian measurements when having correct support estimates. We provide necessary and sufficient recovery condition for the first optimization (2.5) and show that it is limited when support estimate is exact. We provide sufficient condition on the second optimization (2.6). Finally, we present numerical experiments illustrating the performance of low-rank matrix recovery with prior information to augment our analysis.

2.1.6 Overview

The rest of this chapter is organized as followed: Section 2.2 introduced the low-rank matrix recovery and provides an alternative proof on the sampling complexity for Random Gaussian measurements; Section 2.3 considers low-rank matrices recovery with prior information, in terms of the null space property, and discusses sampling complexity in the case of knowing the correct prior information. Section 2.4 is dedicated to the numerical experiments.

2.2 Low-rank Matrices Recovery

2.2.1 Problem Setting and Notation

For low-rank matrix recovery, we want to recover a matrix $X \in \mathbb{R}^{n_1 \times n_2}$ with rank r, $r \ll \min\{n_1, n_2\}$, from linear measurements

$$y = \mathscr{A}(X),$$

where $\mathscr{A} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^m$ is an linear operator and $\mathscr{A}(X) = \sum_{i=1}^m \langle A_i, X \rangle e_i$. Here, e_i represents i-th standard basis for $\mathbb{R}^m$. To recover the matrix X, we looking for the lowest rank X in an affine space. The naive approach of solving this problem

$$\underset{Z \in \mathbb{R}^{n_1 \times n_2}}{\text{minimize}} \, \text{rank}(Z) \text{ subject to } \mathscr{A}(Z) = y$$

is NP-hard. Instead, a heuristic method in the spirit of Compressive Sensing is usually used. Consider the Singular Value Decomposition of X:

$$X = \sum_{i=1}^n \sigma_i u_i v_i^*.$$

Here, $n = \min\{n_1, n_2\}$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0 = \sigma_{r+1} = \cdots = \sigma_n$ are the singular values of X, and $u_i \in \mathbb{R}^{n_1}$, $v_i \in \mathbb{R}^{n_2}$ are the left and right singular vectors of X, respectively. Introduce the nuclear norm of X as sum of singular values:

$$\|X\|_* = \sum_{i=1}^{n} \sigma_i(X)$$

We consider the most common convex relaxation of rank minimization problem, nuclear norm minimization:

$$X^* = \arg\min_{Z} \|Z\|_* \quad \text{subject to } \mathscr{A}(Z) = y \tag{2.13}$$

In this section, we will study the condition on measurement matrices, such that the nuclear norm minimization (2.13) can return the exact result, i.e. $X^* = X$. The condition will be determined by the null space of the measurement matrices $\mathscr{A}$.

2.2.2 Null Space Property of Low-rank Matrix Recovery

Define the Null Space Property for nuclear norm minimization (2.13) as follows (in analogy with the Null Space Property of compressive sensing).

Definition 2 (Null Space Property with rank r [FR13]). Given a linear map $\mathscr{A} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^m$, we say that $\mathscr{A}$ satisfies the Null Space Property with rank r if for all $H \in \mathscr{A} \backslash \{0\}$ with singular values $\sigma_1(H) \geq \sigma_2(H) \geq \cdots \geq \sigma_n \geq 0$, $n := \min\{n_1, n_2\}$,

$$\sum_{i=1}^{r} \sigma_i(H) < \sum_{i=r+1}^{n} \sigma_i(H) \tag{2.14}$$

Then the following theorem states that the Null Space Property with rank r is the necessary and sufficient condition for uniform exact recovery.

Theorem 1. *[FR13] Given a linear map $\mathscr{A} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^m$, every matrix $X \in \mathbb{R}^{n_1 \times n_2}$ of rank at* *most r is the unique solution of*

$$\underset{Z}{\text{minimize}}\, \|Z\|_* \quad \textit{subject to } \mathscr{A}(Z) = y$$

if and only if $\mathscr{A}$ satisfies the Null Space Property with rank r

However, the null space property is not easy to be verified by a direct computation as it requires checking all the vectors in the null-space. One might be curious about what kind of linear measurement satisfies it. Kabanava, Kueng, Rauhut and Terstiege [KKRT16] show that if $m = O(r(n_1 + n_2))$, then with high probability, random Gaussian measurement with m measurements satisfies Null Space Property with rank r.

Our first contribution for this topic is a new proof of the theorem states that Gaussian Random Measurements satisfies Null Space Property with high probability using the technique different from [KKRT16].

Theorem 2. *Let $\mathscr{A} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^m$ be a linear operator such that $\mathscr{A}(X) = \sum_{i=1}^m \langle A_i, X \rangle e_i$, where e_i is the i-th standard basis of $\mathbb{R}^m$, and A_i is the random Gaussian matrices with i.i.d. standard Gaussian entries. Than with probability exceeding $1 - \varepsilon$, $\mathscr{A}$ satisfies Null Space Property for rank r providing*

$$\frac{m}{\sqrt{m+1}} \gtrsim 2\sqrt{r}(\sqrt{n_1} + \sqrt{n_1}) + \sqrt{2\ln(\varepsilon^{-1})}. \tag{2.15}$$

Define the set $\mathscr{H} := \{H : \sum_{i=1}^r \sigma_i(H) \geq \sum_{i=r+1}^n \sigma_i(H), \|H\|_F = 1\}$. In order to show that $\mathscr{A}$ satisfies the Null Space Property, we need to show $\inf_{H \in \mathscr{H}} \|\mathscr{A}(X)\|_2 > 0$. In other words no matrices from $\mathscr{H}$ are in the null-space. Introduce Gorden's Escape Through the Mesh Theorem [Gor88]:

Theorem 3 (Escape Through the Mesh[Gor88])**.** *Let E_m be the expectation of the ℓ_2 norm of m* *dimensional standard Gaussian random vector. Define the Gaussian width of set H by*

$$\ell(H) := \mathbb{E}\sup_{h\in H}\langle h, g\rangle, \tag{2.16}$$

where $g \in \mathbb{R}^N$ is a standard Gaussian random vector. For an $m \times N$ random Gaussian matrix with zero-mean and unit-variance entries and for an arbitrary set $H \subset S^{N-1}$,

$$\mathbb{P}\left(\inf_{h\in H}\|Ah\|_2 \le E_m - \ell(H) - a\right) \le e^{-a^2/2}. \tag{2.17}$$

The theorem leads us to estimate the Gaussian width $\ell(\mathcal{H})$:

$$\ell(\mathcal{H}) = \mathbb{E}\sup_{H\in\mathcal{H}}\langle H, G\rangle_{HS},$$

where G represents the matrix with i.i.d. standard Gaussian entries, and $\langle\cdot,\cdot\rangle_{HS}$ is the Hilbert - Schmidt inner product defined as $\langle X, Y\rangle_{HS} := \mathrm{Tr}(X^*Y)$.

Kabaneva et.al. bound the gaussian width by seeking a simpler set D, such that $\mathcal{H} \subset cD$, with some constant c, then bound the gaussian width of D and hence bound the gaussian width of $\mathcal{H}$. Our proof techniques on the other hand, is a direct estimate of the Gaussian width of $\mathcal{H}$, based on the proof for the Null Space Property of Compressive Sensing using ℓ_1 minimization (Theorem 9.29 [FR13]), with modifications to account for the matrix structure:

Proof. We are aiming at estimate the gaussian width of the set

$$\mathcal{H} := \{H : \sum_{i=1}^{r}\sigma_i(H) \ge \sum_{i=r+1}^{n}\sigma_i(H), \|H\|_F = 1\}.$$

To begin with, we will estimate the supremum $\sup_{H\in\mathcal{H}}\langle H, G\rangle_{HS}$ given G, and then evaluate the expectation with respect to G. Do the singular value decomposition of H as $H =$

$U\Sigma V^*$, we have

$$\begin{aligned}\sup_{H\in\mathcal{H}}\langle H,G\rangle_{HS} &= \sup_{\substack{U,\Sigma,V^*\\ U\Sigma V^*\in\mathcal{H}}} \langle U\Sigma V^*,G\rangle\\ &= \sup_{\substack{U,\Sigma,V^*\\ U\Sigma V^*\in\mathcal{H}}} \langle \Sigma,U^*GV\rangle.\end{aligned}$$

Let $\mathcal{U}_n$ be the set of $n\times n$ unitary matrices. Since $\mathcal{H}$ is the set of matrices whose singular values satisfy $\sum_{i=1}^r \sigma_i(H)\geq\sum_{i=r+1}^n\sigma_i(H)$, it is invariant over the unitary matrices. The optimization over $H=U\Sigma V^*\in\mathcal{H}$ is equivalent to it over $\Sigma\in\mathcal{H}$ and $U,V\in\mathcal{U}_n$. Thus to estimate $\sup_{\substack{U,\Sigma,V^*\\ U\Sigma V^*\in\mathcal{H}}}\langle\Sigma,U^*GV\rangle$, we can fix U and V, let $\widetilde{G}:=U^*GV$, and bound the supremum $\sup_{\sigma\in\mathcal{H}}\langle\Sigma,\widetilde{G}\rangle$ over Σ given $\widetilde{G}$ first, then optimize the result with respect to U and V over the set of unitary matrices $\mathcal{U}_n$:

$$\begin{aligned}\sup_{\substack{U,\Sigma,V^*\\ U\Sigma V^*\in\mathcal{H}}}\langle\Sigma,U^*GV\rangle &= \sup_{U,V\in\mathcal{U}_n}\sup_{\Sigma\in\mathcal{H}}\langle\Sigma,U^*GV\rangle\\ &= \sup_{\substack{U,V\in\mathcal{U}\\ \widetilde{G}=U^*GV}}\sup_{\Sigma\in\mathcal{H}}\langle\Sigma,\widetilde{G}\rangle.\end{aligned}$$

Now we bound the inner supremum $\sup_{\Sigma\in\mathcal{H}}\langle\Sigma,\widetilde{G}\rangle$:

Define $K=\{\sigma\in\mathbb{R}^n:\sum_{i=1}^r\sigma_i\geq\sum_{i=r+1}^n\sigma_i,\sigma_i\geq 0,\|\sigma\|_2=1\}$, then $Q_t=\{z\in\mathbb{R}^n: z_i=t \text{ for } i=1,\cdots,r;\ z_i>-t \text{ for } i=r+1,\cdots,n\}$ is a subset of K's dual cone $K^*:=\{z\in\mathbb{R}^n:\langle z,\sigma\rangle>$

$0, \forall \sigma \in K\}$. Because $\forall z \in Q_t$, $\forall \sigma \in K$, we have

$$\begin{aligned}\langle z, \sigma\rangle &= \sum_{i=1}^{r} z_i\sigma_i + \sum_{i=r+1}^{n} z_i\sigma_i \\ &= \sum_{i=1}^{r} t\sigma_i - \sum_{i=r+1}^{n} t\sigma_i \\ &= t\left(\sum_{i=1}^{r} \sigma_i - \sum_{i=r+1}^{n} \sigma_i\right) \\ &\geq 0\end{aligned}$$

By duality (e.g.[FR13](B.40)), we have:

$$\begin{aligned}\sup_{\Sigma}\langle \Sigma, \widetilde{G}\rangle = \sup_{\Sigma}\sum_{i=1}^{n} \Sigma_{ii}\widetilde{G}_{ii} &= \sup_{\sigma\in K}\langle \sigma, \overrightarrow{\mathrm{diag}(\widetilde{G})}\rangle \\ &\leq \min_{z\in Q_t} \|z + \overrightarrow{\mathrm{diag}(\widetilde{G})}\|_2 \\ &\leq \min_{z\in Q_t} \sqrt{\sum_{i=1}^{r}(t+\widetilde{G}_{ii})^2} + \sqrt{\sum_{i=r+1}^{n}(z_i+\widetilde{G}_{ii})^2} \\ &\leq \sqrt{\sum_{i=1}^{r}(\widetilde{G}_{ii})^2} + t\sqrt{r} + \sqrt{\min_{z_i\geq -t}\sum_{i=r+1}^{n}(z_i+\widetilde{G}_{ii})^2} \\ &\leq \sqrt{\sum_{i=1}^{r}(\widetilde{G}_{ii})^2} + t\sqrt{r} + \sqrt{\sum_{i=r+1}^{n} S_t(\widetilde{G}_{ii})^2}.\end{aligned}$$

where $S_t : \mathbb{R} \to \mathbb{R}$ denotes the the soft-thresholding operator with $S_t(y) := \arg\min_{x\in\mathbb{R}}\{\frac{1}{2}(x-y)^2 + t|x|\}$ Next, we go back to optimize the supremum over $U, V \in \mathcal{U}_n$. Let $U_G\Sigma_G V_G{}^*$ be the SVD of G. Since the objective function is the Hilbert - Schmidt inner product, we should choose U, V as left and right singular matrices of G, i.e. U_G, V_G to minimize the “loss of energy”,

$$\begin{aligned}
&\sup_{\substack{U,V\in\mathcal{U}\\ \widetilde{G}=U^*GV}} \sup_{\Sigma\in\mathcal{H}} \langle\Sigma,\widetilde{G}\rangle\\
&= \sup_{\Sigma\in\mathcal{H}} \sup_{U,V\in\mathcal{U}} \langle\Sigma, U^*U_G\Sigma_G V_G^* V\rangle\\
&\leq \sup_{\Sigma\in\mathcal{H}} \langle\Sigma, U_G^*U_G\Sigma_G V_G^* V_G\rangle\\
&= \sup_{\Sigma\in\mathcal{H}} \langle\Sigma,\Sigma_G\rangle.
\end{aligned}$$

Thus the supremum is bounded by $\sqrt{\sum_{i=1}^r \sigma_i(G)^2} + t\sqrt{r} + \sqrt{\sum_{i=r+1}^n S_t(\sigma_i(G))^2}$. Then,

$$\begin{aligned}
\ell(\mathcal{H}) &\leq \mathbb{E}\left(\sqrt{\sum_{i=1}^r (\sigma_i(G))^2} + t\sqrt{r} + \sqrt{\sum_{i=r+1}^n S_t(\sigma_i(G))^2}\right)\\
&\leq \mathbb{E}\sqrt{\sum_{i=1}^r (\sigma_i(G))^2} + t\sqrt{r} + \mathbb{E}\sqrt{\sum_{i=r+1}^n S_t(\sigma_i(G))^2}\\
&\leq \mathbb{E}\sqrt{r\sigma_1(G)^2} + t\sqrt{r} + \mathbb{E}\sqrt{(n-r)S_t(\sigma_{r+1}(G))^2}\\
&\leq \sqrt{r}\mathbb{E}(\sigma_1(G)) + t\sqrt{r} + \sqrt{(n-r)\mathbb{E}S_t(\sigma_{r+1}(G))^2}
\end{aligned}$$

For the first term in the equation above, the expectation of the largest singular value of G of random gaussian $n_1 \times n_2$ matrix, $\mathbb{E}\sigma_1(G) \lesssim \sqrt{n_1} + \sqrt{n_2}$ up to an absolute constant factor([RV10]).

Now we want to bound the soft-thresholding term $\mathbb{E}S_t(\sigma_{r+1}(G))^2$

$$\begin{aligned}\mathbb{E}S_t(\sigma_{r+1}(G))^2 &\le \mathbb{E}[S_t(\sigma_1(G))^2]\\ &= \mathbb{E}[(0\cdot \mathbb{1}(\sigma_1(G)<t)+\mathbb{E}[(\sigma_1(G)-t)\mathbb{1}(\sigma_1>t)]^2\\ &= \mathbb{E}[(\sigma_1(G)-t)\mathbb{1}(\sigma_1>t)]^2\\ &\le \mathbb{E}[(\sigma_1(G)-t)]^2\mathbb{E}[\mathbb{1}(\sigma_1>t)]^2(\text{Cauchy Schwarz inequality})\\ &= \mathbb{E}[\sigma_1^2(G)-2t\sigma_1(G)+t^2]\mathbb{E}[\mathbb{1}(\sigma_1>t)]^2\\ &= \left(\mathbb{E}[\sigma_1^2(G)]-2t\mathbb{E}[\sigma_1(G)]+t^2\right)\mathbb{E}[\mathbb{1}(\sigma_1>t)]^2\\ &\lesssim \left((\sqrt{n_1}+\sqrt{n_2})^2-2t(\sqrt{n_1}+\sqrt{n_2})+t^2\right)\mathbb{P}(\mathbb{1}(\sigma_1>t))\end{aligned}$$

Notice that the expectation of a random variable X can also been written as $\int_0^\infty -t^2 dS(t)$, where S(t) is the survival function of X (by Darth Vader Rule[MOW12]). The last inequality is bounded as followed:

$$\begin{aligned}\mathbb{E}\sigma_1^2(G)=\mathbb{E}\|G\|^2 &= \int_0^\infty -t^2\phi(t)dt\\ &= \lim_{R\to\infty}\int_0^R -t^2\phi(t)dt\\ &= \lim_{R\to\infty}\left(-R^2F(R^2)+0+\int_0^R 2tF(t)dt\right)\\ &= \int_0^\infty 2tF(t)dt\\ &= \int_0^{\mathbb{E}\|G\|} 2tF(t)dt+\int_{\mathbb{E}\|G\|}^\infty 2sF(s)ds\\ &\le (\mathbb{E}\|G||)^2+\int_0^\infty 2(\mathbb{E}\|G||+s)\exp(\frac{s^2}{\pi^2})ds\\ &\le (\mathbb{E}\|G||)^2+2\mathbb{E}\|G\|\int_0^\infty \exp(-\frac{s^2}{\pi^2})ds+\int_0^\infty 2s\exp(-\frac{s^2}{\pi^2})ds\\ &\le (\mathbb{E}\|G||)^2+C_1 2\mathbb{E}\|G\|+C_2\end{aligned}\tag{2.18}$$

where $\phi(t)=\frac{d}{ds}\mathbb{P}(\|G\|\ge s)\big|_{s=t}$ and F denotes its antiderivative. The third equation is due to integration by parts. The fifth row is because of $F(t) \leq 1$ and $\mathbb{P}(\|G\| > \mathbb{E}\|G\| + s) \leq \exp(-\frac{s^2}{\pi^2})$ (Gaussian Concentration [Ver18])

Now we have

$$\begin{aligned}
\ell(\mathcal{H}) \leq & \sqrt{r}\mathbb{E}(\sigma_1(G)) + t\sqrt{r} + \sqrt{(n-r)\mathbb{E}S_t(\sigma_{r+1}(G))^2} \\
\leq & \sqrt{r}\mathbb{E}(\sigma_1(G)) + t\sqrt{r} + \sqrt{n-r}\sqrt{\left(\mathbb{E}[\sigma_1^2(G)] - 2t\mathbb{E}[\sigma_1(G)] + t^2\right)\mathbb{E}[\mathbb{1}(\sigma_1 > t)]^2} \\
\lesssim & \sqrt{r}(\sqrt{n_1} + \sqrt{n_2}) + t\sqrt{r} \\
& + \sqrt{n-r}\left(((\sqrt{n_1} + \sqrt{n_2})^2 - 2t(\sqrt{n_1} + \sqrt{n_2}) + t^2)\mathbb{P}(\mathbb{1}(\sigma_1 > t))\right)^{1/2}
\end{aligned}$$

where $\mathbb{P}(\mathbb{1}(\sigma_1 > t))$ decays exponentially.

let $t = \sqrt{n_1} + \sqrt{n_2}$, we have $\ell(\mathcal{H}) \lesssim 2\sqrt{r}(\sqrt{n_1} + \sqrt{n_2})$. Let $e^{-a^2/2} = \varepsilon$, since $E_m \geq m/\sqrt{m+1}$,

$$\frac{m}{\sqrt{m+1}} \gtrsim 2\sqrt{r}(\sqrt{n_1} + \sqrt{n_2})$$

ensures that

$$E_m - \ell(\mathcal{H}) - a > 0$$

Apply Gordon's Escape Through the Mesh Theorem,

$$\begin{aligned}
& \mathbb{P}\left(\inf_{h \in H} \|Ah\|_2 > 0\right) \\
\geq & \mathbb{P}\left(\inf_{h \in H} \|Ah\|_2 > E_m - \ell(\mathcal{H}) - a\right) \\
\geq & 1 - e^{-a^2/2} = 1 - \varepsilon
\end{aligned} \tag{2.19}$$

□

This bound is optimal since the $O(r(n_1 + n_2))$ corresponds to the number of degrees of freedom required to describe an $n_1 \times n_2$ matrix of rank r. In contrast to the vector case, there is no logarithmic factor involved.

2.3 Low-rank Matrix Recovery with Prior Information

Consider a matrix X that lives in a union of row and column subspaces denoted by T. Suppose that we are given a subspace $\widetilde{T}$ that is slightly mis-aligned with T. Now the question is: Can we weaken the recovery conditions and reduce the number of measurements needed by penalizing the orthogonal complement of $\widetilde{T}$?

2.3.1 Support of low rank matrices

Every rank r solution X^* of

$$\underset{Z}{\text{minimize}}\, \|Z\|_* \quad \text{subject to } \mathscr{A}(Z) = \mathscr{A}(X)$$

lives in a lower dimension subspace of $\mathbb{R}^{n_1 \times n_2}$ spanned by the $n_1 \times r$ column and $n_2 \times r$ row basis vectors corresponding to the nonzero singular values of X^*. In some situations, it is possible to obtain prior information.

In the vector case, it was shown that prior information on the support (nonzero entries) can be incorporated in the ℓ_1-recovery algorithm by solving the weighted-ℓ_1-minimization problem. In this case, the weights are applied such that solutions with large nonzero entries on the support estimate have a lower cost (weighted-ℓ_1 norm) then solution with large nonzeros outside of the support estimate [MS17], [MY11].

In the matrix case, we first define the support of a low-rank matrix formally. If X is an $n_1 \times n_2$ matrix with rank r, then its full Singular Value Decomposition(SVD) can be written by

$$\begin{aligned} X &= U_X \Sigma_X V_X^* \\ &= \begin{bmatrix} U & U_\perp \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V^* \\ V_\perp^* \end{bmatrix} \end{aligned}$$

where $U_X = \begin{bmatrix} U & U_\perp \end{bmatrix}$ is a $n_1 \times n_1$ unitary matrix, $\Sigma_X = \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} = \operatorname{diag}(\sigma_1, \sigma_2, \cdots, \sigma_n)$ is a $n_1 \times n_2$ diagonal matrix with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0 = \sigma_{r+1} = \cdots = \sigma_n = 0$ for $n = \min\{n_1, n_2\}$, and $V_X^* = \begin{bmatrix} V^* \\ V_\perp^* \end{bmatrix}$ is a $n_2 \times n_2$ unitary matrix.

In the following of this chapter, we define the reduced Singular Value Decomposition(rSVD) of X as

$$X = U\Sigma V^*, \tag{2.20}$$

where U is $n_1 \times r$ matrix with orthonormal columns; $\Sigma = \operatorname{diag}(\sigma_1, \cdots, \sigma_r)$ is $r \times r$ diagonal matrix; and V^* is $r \times n_2$ matrix with orthonormal columns.

Definition 3 (Support of a low-rank matrix)**.** We define the support of a low-rank matrix X by matrices whose columns are basis vectors that span X's column and row spaces: $\{U, V\}$. In particular $\{U, V\}$ is the unique support of the matrix X if and only if $X = P_U X P_V$, with P_U and P_V be the orthogonal projections onto X's column and row space respectively, $P_U = UU^*$ and $P_V = VV^*$.

2.3.2 Weighted Nuclear Norm Minimization

In this section, we will introduce the first modified optimization problem when we have prior support estimate. Now the support estimate can be replaced by $\widetilde{U} \in \mathbb{R}^{n_1 \times r}$ and $\widetilde{V} \in \mathbb{R}^{n_2 \times r}$ that estimate the row and column subspaces bases $U \in \mathbb{R}^{n_1 \times r}$ and $V \in \mathbb{R}^{n_2 \times r}$ of X. Denote $\widetilde{U}_\perp$ and $\widetilde{V}_\perp$ as orthogonal complement of $\widetilde{U}$ and $\widetilde{V}$, respectively. Let $\mathcal{P}_{\widetilde{U}} = \widetilde{U}\widetilde{U}^*$ and $\mathcal{P}_{\widetilde{V}} = \widetilde{V}\widetilde{V}^*$ be the orthogonal projection matrices project onto the subspaces spanned by $\widetilde{U}$ and $\widetilde{V}$, same for $\mathcal{P}_{\widetilde{U}_\perp}$ and $\mathcal{P}_{\widetilde{V}_\perp}$.

Let $Q_{\widetilde{U},w} = w\mathcal{P}_{\widetilde{U}} + \mathcal{P}_{\widetilde{U}_\perp}$, $Q_{\widetilde{V},\lambda} = \lambda\mathcal{P}_{\widetilde{V}} + \mathcal{P}_{\widetilde{V}_\perp}$ be the weighted projection with respect to the support estimate. Define the *weighted nuclear norm of X* as $\|Q_{\widetilde{U},w} X Q_{\widetilde{V},\lambda}\|_*$, with $0 \leq w \leq 1$, $0 \leq \lambda \leq 1$; the *Weighted Nuclear Norm Minimization* can be formulated as follows [AKM$^+$14]:

$$\underset{Z}{\text{minimize}} \, \|Q_{\widetilde{U},w} Z Q_{\widetilde{V},\lambda}\|_* \quad \text{subject to} \quad \mathscr{A}(Z) = \mathscr{A}(X) \tag{2.21}$$

Where

$$\begin{aligned} &\underset{Z}{\text{minimize}} \, \|Q_{\widetilde{U},w} Z Q_{\widetilde{V},\lambda}\|_* \\ =&\underset{Z}{\text{minimize}} \, \|(w\mathcal{P}_{\widetilde{U}} + \mathcal{P}_{\widetilde{U}_\perp}) Z (\lambda \mathcal{P}_{\widetilde{V}} + \mathcal{P}_{\widetilde{V}_\perp})\|_* \\ =&\underset{Z}{\text{minimize}} \, \|w\lambda \mathcal{P}_{\widetilde{U}} Z \mathcal{P}_{\widetilde{V}} + w\mathcal{P}_{\widetilde{U}} Z \mathcal{P}_{\widetilde{V}_\perp} + \lambda \mathcal{P}_{\widetilde{U}_\perp} Z \mathcal{P}_{\widetilde{V}} + \mathcal{P}_{\widetilde{U}_\perp} Z \mathcal{P}_{\widetilde{V}_\perp}\|_*. \end{aligned}$$

Thus minimizing $\|(w\mathcal{P}_{\widetilde{U}} + \mathcal{P}_{\widetilde{U}_\perp}) Z (\lambda \mathcal{P}_{\widetilde{V}} + \mathcal{P}_{\widetilde{V}_\perp})\|_*$ penalizes solutions that live in the subspace which orthogonal to support estimate more when $0 \leq w < 1$, $0 \leq \lambda < 1$. If we take $w = 1$ and $\lambda = 1$ the optimization becomes the regular nuclear norm minimization (2.4), and if we have the prefect estimate of support, i.e. $\widetilde{U} = U$, $\widetilde{V} = V$, then we can set the weights as 0, then the optimization problem becomes

$$\underset{Z}{\text{minimize}} \, \|\mathcal{P}_{U_\perp} Z \mathcal{P}_{V_\perp}\|_* \text{ subject to } \mathscr{A}(Z) = \mathscr{A}(X). \tag{2.22}$$

Null Space Property for Weighted Nuclear Norm Minimization

Our task is to find out when it is possible to reconstruct the rank r matrix X exactly by solving (2.21)

Definition 4. (Null Space Property for weighted nuclear norm minimization) Given a linear operation $\mathscr{A} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^m$, we say that $\mathscr{A}$ satisfies the *Null Space Property for Weighted*

Nuclear Norm Minimization (2.12) if for every matrix $H \neq 0$ such that $\mathscr{A}(H) = 0$,

$$\sum_{i=1}^{r} \sigma_i(Q_{\widetilde{U},w} H Q_{\widetilde{V},\lambda}) < \sum_{i=r+1}^{n} \sigma_i(Q_{\widetilde{U},w} H Q_{\widetilde{V},\lambda}), \tag{2.23}$$

where $n = \min\{n_1, n_2\}$.

Theorem 4. *Given a linear operation $\mathscr{A} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^m$, every matrix X with rank at most r, can be exactly recovered through optimization*

$$\underset{z}{\text{minimize}} \, \|Q_{\widetilde{U},w} Z Q_{\widetilde{V},\lambda}\|_* \quad \textit{subject to} \quad \mathscr{A}(Z) = \mathscr{A}(X)$$

if and only if $\mathscr{A}$ satisfies Null Space Property for Weighted Nuclear Norm Minimization (2.12).

Proof. First, assume $\mathscr{A} : \mathbb{R}^{n_1 \times n_2}$ satisfies *Null Space Property for Weighted Nuclear Norm Minimization*. i.e. $\forall H \in \text{Null}(A) \backslash \{0\}, \sum_{i=1}^{r} \sigma_i(Q_{\widetilde{U},w} H Q_{\widetilde{V},\lambda}) < \sum_{i=r+1}^{n} \sigma_i(Q_{\widetilde{U},w} H Q_{\widetilde{V},\lambda})$, Let $X \in \mathbb{R}^{n_1 \times n_2}$ has rank $\leq r$, and $Z \in \mathbb{R}^{n_1 \times n_2}$ and $Z \neq X$. Suppose $\mathscr{A}(Z) = \mathscr{A}(X)$. Set $H = X - Z \in \text{Null}(\mathscr{A}) \backslash \{0\}$, then by the assumption,

$$\begin{aligned}
\|Q_{\widetilde{U},w} Z Q_{\widetilde{V},\lambda}\|_* &= \|Q_{\widetilde{U},w} (X - H) Q_{\widetilde{V},\lambda}\|_* = \|Q_{\widetilde{U},w} X Q_{\widetilde{V},\lambda} - Q_{\widetilde{U},w} H Q_{\widetilde{V},\lambda}\|_* \\
&\geq \sum_{j=1}^{n} |\sigma_i(Q_{\widetilde{U},w} X Q_{\widetilde{V},\lambda}) - \sigma_i(Q_{\widetilde{U},w} H Q_{\widetilde{V},\lambda})| \\
&= \sum_{j=1}^{r} |\sigma_j(Q_{\widetilde{U},w} X Q_{\widetilde{V},\lambda}) - \sigma_j(Q_{\widetilde{U},w} H Q_{\widetilde{V},\lambda})| + \sum_{j=r+1}^{n} \sigma_j(Q_{\widetilde{U},w} H Q_{\widetilde{V},\lambda}) \\
&\geq \sum_{j=1}^{r} \sigma_i(Q_{\widetilde{U},w} X Q_{\widetilde{V},\lambda}) - \sum_{j=1}^{r} \sigma_i(Q_{\widetilde{U},w} H Q_{\widetilde{V},\lambda}) + \sum_{j=r+1}^{n} \sigma_j(Q_{\widetilde{U},w} H Q_{\widetilde{V},\lambda}) \\
&> \sum_{j=1}^{r} \sigma_i(Q_{\widetilde{U},w} X Q_{\widetilde{V},\lambda}) \\
&= \|Q_{\widetilde{U},w} X Q_{\widetilde{V},\lambda}\|_*.
\end{aligned}$$

The first inequality uses the lemma 11 in [OMFH11].

Conversely, suppose every X of rank r is the unique solution of

$$\underset{Z\in\mathbb{R}^{n_1\times n_2}}{\text{minimize}}\, \|Q_{\widetilde{U},w} Z Q_{\widetilde{V},\lambda}\|_* \quad \text{subject to} \quad \mathscr{A}(Z)=\mathscr{A}(X),$$

$\forall H \in \text{null}(\mathscr{A})\backslash\{0\}$, write $Q_{\widetilde{U},w} H Q_{\widetilde{V},\lambda} = \sum_{i=1}^{n} \sigma_i u_i v_i^*$, Decompose $H = H_1 + H_2$ where

$$H_1 = Q_{\widetilde{U},w}^{-1} \sum_{i=1}^{r} \sigma_i u_i v_i^* Q_{\widetilde{V},\lambda}^{-1}$$
$$H_2 = Q_{\widetilde{U},w}^{-1} \sum_{i=r+1}^{n} \sigma_i u_i v_i^* Q_{\widetilde{V},\lambda}^{-1},$$

Then $\mathscr{A}(H)=0$ implies that $\mathscr{A}(H_1)=\mathscr{A}(-H_2)$. By assumption, H_1 has the minimized weighted nuclear norm

$$\sum_{i=1}^{r} \sigma_i(Q_{\widetilde{U},w} H Q_{\widetilde{V},\lambda}) = \|Q_{\widetilde{U},w} H_1 Q_{\widetilde{V},\lambda}\|_* < \|Q_{\widetilde{U},w} H_2 Q_{\widetilde{V},\lambda}\|_* = \sum_{i=r+1}^{n} \sigma_i(Q_{\widetilde{U},w} H Q_{\widetilde{V},\lambda})$$

□

2.3.3 Drawback of the weighted nuclear norm optimization (2.21)

Let $X \in \mathbb{R}^{n\times n}$ be a rank-r matrix, and $X = U\Sigma V$ be its reduced Singular Value Decomposition, where $U,V \in \mathbb{R}^{n\times r}$ have orthonormal columns that corresponding to the largest r singular values of X. $\widetilde{U} \in \mathbb{R}^{n\times r}$ and $\widetilde{V} \in \mathbb{R}^{n\times r}$ are the estimates of $U \in \mathbb{R}^{n\times r}$ and $V \in \mathbb{R}^{n\times r}$, respectively. Consider $\mathscr{A} : \mathbb{R}^{n_1\times n_2} \to \mathbb{R}^m$ being a random Gaussian measurement operator, $\mathscr{A}(X) := \sum_{i=1}^{m} \langle G_i, X\rangle e_i$, where $G_i \in \mathbb{R}^{n_1\times n_2}$ whose entries are i.i.d. $\mathcal{N}(0,1)$ (zero-mean, unit variance Gaussian). We will see that the optimization (2.21) is limited when we have support estimate being exact.

When the row or column space estimate is perfectly accurate

Since the support estimate $\widetilde{V}$ is exact, we set $\lambda = 0$. This means that as long as there exist a feasible matrix $Z = U_Z\Sigma_Z V_Z^*$ with correct row space $V_Z = V$, then it will be a solution of problem (2.21). Also we can show that in this case, to make the problem have the unique solution, we must have $m = rn$ random Gaussian measurements.

Corollary 1. *Let $\mathscr{A} : \mathbb{R}^{n_1\times n_2} \to \mathbb{R}^m$ being a random Gaussian measurement operator, i.e. $\mathscr{A}(X) := \sum_{i=1}^m \langle G_i, X\rangle e_i$, where $G_i \in \mathbb{R}^{n_1\times n_2}$ whose entries are i.i.d. $\mathcal{N}(0,1)$. If $\widetilde{V} = V$, and $\lambda = 0$. Then every rank r matrix is the unique solution of optimization (2.21) if $m = O(rn)$.*

Proof. let $X = U_X\Sigma_X V^*$ and $Z = U_Z\Sigma_Z V^*$. Thus $\|(w\mathcal{P}_{\widetilde{U}} + \mathcal{P}_{\widetilde{U}^\perp})X(\mathcal{P}_{V^\perp})\|_* = 0$, and $\|(w\mathcal{P}_{\widetilde{U}} + \mathcal{P}_{\widetilde{U}^\perp})Z(\mathcal{P}_{V^\perp})\|_* = 0$. Let $\mathscr{A}(Z) = \mathscr{A}(X)$, then we have

$$\begin{bmatrix} \langle G_1, U_Z\Sigma_Z V\rangle \\ \langle G_2, U_Z\Sigma_Z V\rangle \\ \langle G_3, U_Z\Sigma_Z V\rangle \\ \vdots \\ \langle G_m, U_Z\Sigma_Z V\rangle \end{bmatrix} = \begin{bmatrix} \langle G_1, U_X\Sigma_X V\rangle \\ \langle G_2, U_X\Sigma_X V\rangle \\ \langle G_3, U_X\Sigma_X V\rangle \\ \vdots \\ \langle G_m, U_X\Sigma_X V\rangle \end{bmatrix}.$$

The system has $O(nr)$ degrees of freedom, so the number of measurement m should be at least $O(nr)$ to make the solution unique. □

When both the row and column space estimates are perfectly accurate

Corollary 2. *Let $\mathscr{A} : \mathbb{R}^{n_1\times n_2} \to \mathbb{R}^m$ be a random Gaussian measurement operator, i.e. $\mathscr{A}(X) := \sum_{i=1}^m \langle G_i, X\rangle e_i$, where $G_i \in \mathbb{R}^{n_1\times n_2}$ whose entries are i.i.d. $\mathcal{N}(0,1)$. If $\widetilde{U} = U$ and $\widetilde{V} = V$, every rank r matrix is the unique solution solution of optimization (2.21) with $w = 0$ and $\lambda = 0$ if $m = O(rn)$, $n = \min\{n_1, n_2\}$.*

Proof. Suppose the $Z^* = X + H$ is the minimizer of the optimization problem

$$\underset{Z}{\text{minimize}} \, \|\mathcal{P}_{U_\perp} Z \mathcal{P}_{V_\perp}\|_* \text{ subject to } \mathscr{A}(Z) = \mathscr{A}(X),$$

Then we should have $\|\mathcal{P}_{U_\perp}(X+H)\mathcal{P}_{V_\perp}\|_* = 0$, i.e.

$$\mathcal{P}_{U_\perp} H \mathcal{P}_{V_\perp} = 0,$$

this means that H belongs to the space

$$T := \{H \in \mathbb{R}^{n_2 \times n_2} : H = \mathcal{P}_U H + H \mathcal{P}_V - \mathcal{P}_U H \mathcal{P}_V\} \tag{2.24}$$

which is a $O(n \times r)$ dimensional subspace, $n = \min\{n_1, n_2\}$. To make Z feasible, H must be the only solution to $\mathscr{A}(H) = 0$, i.e. $H = 0$, implies that $\mathscr{A}$ has at least $O(n \times r)$ measurements. □

The analysis above means that having correct subspace estimates doesn't help us on reducing number of measurement needed for exact recovery, which means that the optimization problem (2.21) is perhaps not ideal when we have partial support information on X.

2.3.4 Sum of Two Weighted Nuclear Norm Minimization

Let us consider a new optimization problem, based on *Minimizing the Sum of Two Weighted Nuclear Norms*:

$$\underset{Z}{\text{minimize}} \, \|(w\mathcal{P}_{\widetilde{U}} + \mathcal{P}_{\widetilde{U}^\perp})Z\|_* + \|Z(\lambda \mathcal{P}_{\widetilde{V}} + \mathcal{P}_{\widetilde{V}^\perp})\|_* \qquad \text{subject to} \quad \mathscr{A}(Z) = \mathscr{A}(X) \tag{2.25}$$

This optimization will allow us to penalize solutions that live in the orthogonal complement spaces of our estimates when w and λ smaller than 1. Also it will allow us to separately consider correctness of estimates of row and column spaces. For example, if we have a correct estimate for the row space: $\widetilde{V} = V$, thus we can set $\lambda = 0$. And even though λ is zero, when doing the optimization, we still need to consider the column space.

Most importantly, as we will cover in detail later, it prevents the case where having the prefect subspace estimate requires more measurements by minimizing

$$\underset{Z}{\text{minimize}} \, \|\mathcal{P}_{U_\perp} Z\|_* + \|Z \mathcal{P}_{V_\perp}\|_* \text{ subject to } \mathscr{A}(Z) = \mathscr{A}(X). \tag{2.26}$$

Null Space Property for Minimizing the Sum of Two Weighted Nuclear Norms

In this section, we propose a sufficient condition for exact recovery base on the null space of the linear measurement $\mathscr{A}$ for *Sum of Two Weighted Nuclear Norm Minimization*. Our proof is based on the technique that were first proposed by Eftekhari, Yang and Wakin [EYW18]. In the following analysis, we consider $X \in \mathbb{R}^{n \times n}$ for convenience, i.e. $n_1 = n_2 = n$.

Lemma 1. *[EYW18] Consider a rank r matrix $X \in \mathbb{R}^{n \times n}$, let $U \in \mathbb{R}^{n \times r}$ be the matrix with orthonormal columns such that span(U) = span(X). Let $\widetilde{U} \in \mathbb{R}^{n \times r}$ be the estimate of U, such that $U^* \widetilde{U} = \cos \Theta_L$, where*

$$\cos \Theta_L = \begin{bmatrix} \cos \theta_r & & & \\ & \cos \theta_{r-1} & & \\ & & \ddots & \\ & & & \cos \theta_1 \end{bmatrix} \in \mathbb{R}^{r \times r}$$

for $1 \geq \cos\theta_r \geq \cos\theta_{r-1} \geq \cdots \geq \cos\theta_1 \geq 0$ *with* $\{\theta_i\}$ *being the principle angles between* U *and* $\widetilde{U}$*. Then there exist* $U', \widetilde{U}' \in \mathbb{R}^{n\times r}$ *and* $U'' \in \mathbb{R}^{n\times(n-2r)}$*, such that*

$$\begin{aligned} B_L &= \begin{bmatrix} U & U' & U'' \end{bmatrix} \in \mathbb{R}^{n\times n} \\ \widetilde{B}_L &= \begin{bmatrix} \widetilde{U} & \widetilde{U}' & U'' \end{bmatrix} \in \mathbb{R}^{n\times n} \end{aligned} \tag{2.27}$$

are both orthonormal bases for $\mathbb{R}^n$*, where* $U' = -(I - UU^*)\widetilde{U}\sin^{-1}\Theta_L$ *and* $\widetilde{U}' = (I - \widetilde{U}\widetilde{U}^*)U\sin^{-1}\Theta_L$*, with* $\sin^{-1}\Theta_L$ *denotes the inverse of the matrix* $\sin\Theta_L$ *Moreover, it holds that*

$$B_L^*\widetilde{B}_L = \begin{bmatrix} \cos\Theta_L & \sin\Theta_L & \\ -\sin\Theta_L & \cos\Theta_L & \\ & & I_{n-2r} \end{bmatrix}. \tag{2.28}$$

A similar construction exists for V *such* $span(V) = span(X^*)$ *where we form the orthonormal bases* B_R *and* $\widetilde{B}_R$ *such that*

$$B_R^*\widetilde{B}_R = \begin{bmatrix} \cos\Theta_R & \sin\Theta_R & \\ -\sin\Theta_R & \cos\Theta_R & \\ & & I_{n-2r} \end{bmatrix}. \tag{2.29}$$

As before, the diagonal of $\Theta_R \in \mathbb{R}^{r\times r}$ *contains the principal angles between* V *and* $\widetilde{V}$ *in non-increasing order.*

Using such decomposition, we have

$$\widetilde{U} = B_L \begin{bmatrix} \cos\Theta_L \\ \sin\Theta_L \\ 0_{(n-2r)\times r} \end{bmatrix}$$

and thus,

$$\begin{aligned} \mathcal{P}_{\widetilde{U}} &= \widetilde{U}^* \widetilde{U} \\ &= B_L \begin{bmatrix} \cos^2\Theta_L & -\sin\Theta_L \cdot \cos\Theta_L & \\ -\sin\Theta_L \cdot \cos\Theta_L & \sin^2\Theta_L & \\ & & 0_{n-2r} \end{bmatrix} B_L^*, \end{aligned}$$

$$\begin{aligned} \mathcal{P}_{\widetilde{U}_\perp} &= \widetilde{U}_\perp^* \widetilde{U}_\perp = I_n - \mathcal{P}_{\widetilde{U}} \\ &= B_L \begin{bmatrix} \sin^2\Theta_L & \sin\Theta_L \cdot \cos\Theta_L & \\ \sin\Theta_L \cdot \cos\Theta_L & \cos^2\Theta_L & \\ & & I_{n-2r} \end{bmatrix} B_L^*. \end{aligned}$$

It follows that

$$Q_{\widetilde{U},w} = w\mathcal{P}_{\widetilde{U}} + \mathcal{P}_{\widetilde{U}_\perp} \tag{2.30}$$

$$= B_L \begin{bmatrix} w\cos^2\Theta_L + \sin^2\Theta_L & (1-w)\sin\Theta_L \cdot \cos\Theta_L & \\ (1-w)\sin\Theta_L \cdot \cos\Theta_L & w\sin^2\Theta_L + \cos^2\Theta_L & \\ & & I_{n-2r} \end{bmatrix} B_L^* \tag{2.31}$$

Define the orthonormal basis $O_L \in \mathbb{R}^{n\times n}$ as

$$O_L := \begin{bmatrix} (w\cos^2\Theta_L+\sin^2\Theta_L)\Delta_L^{-1} & -(1-w)\sin\Theta_L\cdot\cos\Theta_L\cdot\Delta_L^{-1} & 0_{r\times(n-2r)} \\ (1-w)\sin\Theta_L\cdot\cos\Theta_L\cdot\Delta_L^{-1} & (w\cos^2\Theta_L+\sin^2\Theta_L)\Delta_L^{-1} & 0_{r\times(n-2r)} \\ 0_{(n-2r)\times r} & 0_{(n-2r)\times r} & I_{n-2r} \end{bmatrix}$$

where

$$\Delta_L := (w^2\cos^2\Theta_L+\sin^2\Theta_L)^{\frac{1}{2}} \tag{2.32}$$

Δ_L is invertible because $w>0$, by assumption. (It is easily verify that indeed $O_L^* O_L = I_n$.) We then rewrite (2.31) as

$$\begin{aligned} Q_{\widetilde{U},w} &= B_L(O_L O_L*) \begin{bmatrix} w\cos^2\Theta_L+\sin^2\Theta_L & (1-w)\sin\Theta_L\cdot\cos\Theta_L & \\ (1-w)\sin\Theta_L\cdot\cos\Theta_L & w\sin^2\Theta_L+\cos^2\Theta_L & \\ & & I_{n-2r} \end{bmatrix} B_L^* \\ &= B_L O_L \begin{bmatrix} \Delta_L & (1-w^2)\sin\Theta_L\cdot\cos\Theta_L\cdot\Delta_L^{-1} & \\ & w\Delta_L^{-1} & \\ & & I_{n-2r} \end{bmatrix} B_L^* \\ &= B_L O_L L B_L* \end{aligned} \tag{2.33}$$

where $L \in \mathbb{R}^{n\times n}$ is an upper-triangular matrix with blocks $L_{11}, L_{22}, L_{12} \in \mathbb{R}^{r\times r}$ and defined as

$$L := \begin{bmatrix} L_{11} & L_{12} & \\ & L_{22} & \\ & & I_{n-2r} \end{bmatrix} = \begin{bmatrix} \Delta_L & (1-w^2)\sin\Theta_L \cdot \cos\Theta_L \cdot \Delta_L^{-1} & \\ & w\Delta_L^{-1} & \\ & & I_{n-2r} \end{bmatrix}. \tag{2.34}$$

Perform the same calculations for the row spaces and, in particular, define $R \in \mathbb{R}^{n\times n}$ as

$$R := \begin{bmatrix} R_{11} & R_{12} & \\ & R_{22} & \\ & & I_{n-2r} \end{bmatrix} = \begin{bmatrix} \Delta_R & (1-\lambda^2)\sin\Theta_R \cdot \cos\Theta_R \cdot \Delta_R^{-1} & \\ & \lambda\Delta_R^{-1} & \\ & & I_{n-2r} \end{bmatrix}, \tag{2.35}$$

with $\Delta_R = (\lambda \cos^2\Theta_R + \sin^2\Theta_R)^{\frac{1}{2}} \in \mathbb{R}^{r\times r}$

Definition 5 (rank Null Space Property for Sum of Two Weighted Nuclear Norm Minimization)**.** Let $U, V \in \mathbb{R}^{n\times r}$ be the support of a rank - r matrix X. Let $\widetilde{U} \neq U, \widetilde{V} \neq V \in \mathbb{R}^{n\times r}$ with orthonormal columns being the support estimate. Let the principal angle between $\widetilde{U}, U$, and $\widetilde{V}, V$ be Θ_L and Θ_R and u_1, v_1 be the largest of them, respectively. Define B_L and B_R such that the columns of them form an orthonormal bases for $\mathbb{R}^n$ respectively s.t

$$\widetilde{U} = B_L \begin{bmatrix} \cos\Theta_L \\ \sin\Theta_L \\ 0_{(n-2r)\times r} \end{bmatrix}, \text{ and } \widetilde{V} = B_L \begin{bmatrix} \cos\Theta_R \\ \sin\Theta_R \\ 0_{(n-2r)\times r} \end{bmatrix}$$

Define the subspace

$$T := \{Z \in \mathbb{R}^{n\times n} Z = \mathcal{P}_U Z + Z\mathcal{P}_V - \mathcal{P}_U Z \mathcal{P}_V\}. \tag{2.36}$$

Let $\bar{(}Z) = B_L^* Z B_R$, define the subspace

$$\widetilde{T}_\perp := \left\{ Z \in \mathbb{R}^{n\times n} : Z = B_L \begin{bmatrix} 0_r & & \\ & \overline{Z}_{22} & \overline{Z}_{23} \\ & \overline{Z}_{32} & \overline{Z}_{33} \end{bmatrix} B_{R^*} \right\}. \tag{2.37}$$

Let $\mathscr{A}$ be a linear operator maps from $\mathbb{R}^{n\times n}$ to $\mathbb{R}^m$. $\mathscr{A}$ is said to satisfy the *Null Space Property for Sum of Two Nuclear Norm Minimization* relative to (U,V) and $(\widetilde{U},\widetilde{V})$ if any matrix $H \in \text{Null}(\mathscr{A})\backslash\{0\}$ it holds that

$$\begin{aligned} &\left(\sqrt{\frac{w^4\cos^2 u_1 + \sin^2 u_1}{w^2\cos^2 u_1 \sin^2 u_1}} + \sqrt{\frac{\lambda^2\cos^2 v_1 + \sin^2 v_1}{\lambda^2\cos^2 v_1 + \sin^2 v_1}}\right) \|\mathcal{P}_T(H)\|_* \\ &+ \left(\sqrt{\frac{2(1-w^2)\sin^2 u_1}{w^2\cos^2 u_1 + \sin^2 u_1}} + \sqrt{\frac{2(1-\lambda^2)\sin^2 v_1}{\lambda^2\cos^2 v_1 + \sin^2 v_1}}\right) \|\mathcal{P}_{\widetilde{T}_\perp}(H)\|_* + 2\|\mathcal{P}_T(H)\|_* \\ &< 2\|\mathcal{P}_{T_\perp}(H)\|_* \end{aligned} \tag{2.38}$$

Theorem 5. *Given a linear operator $\mathscr{A} : \mathbb{R}^{n\times n} \to \mathbb{R}^m$, a support (U,V), and a support estimate $(\widetilde{U},\widetilde{V})$, every matrix X supported on (U,V) is the unique solution of (2.6) if $\mathscr{A}$ satisfies the Null Space Property for Sum of Two Nuclear Norm Minimization with respect to (U,V) and $(\widetilde{U},\widetilde{V})$*

Proof. Let $Z = X + H$ be the optimizer of (2.25), where $H \in \mathrm{Null}(\mathscr{A})\backslash\{0\}$. Thus we have

$$\|Q_{\widetilde{U},w}(X+H)\|_* + \|(X+H)Q_{\widetilde{V},\lambda}\|_* \leq \|Q_{\widetilde{U},w}X\|_* + \|XQ_{\widetilde{V},\lambda}\|_* \tag{2.39}$$

the right hand side of (2.39) follows by

$$\begin{aligned}
&\|Q_{\widetilde{U},w}X\|_* + \|XQ_{\widetilde{V},\lambda}\|_* \\
=&\|B_L O_L L B_L^* X\|_* + \|XB_R R^* O_R^* B_R^*\|_* \\
=&\|B_L O_L L B_L^* X B_R\|_* + \|B_L^* X B_R R^* O_R^* B_R^*\|_* \\
=&\|L B_L^* X B_R\|_* + \|B_L^* X B_R R^*\|_*
\end{aligned} \tag{2.40}$$

And the left hand side of (2.39) gives us

$$\begin{aligned}
&\|Q_{\widetilde{U},w}(X+H)\|_* + \|(X+H)Q_{\widetilde{V},\lambda}\|_* \\
=&\|B_L O_L L B_L^* X + B_L O_L L B_L^* H\|_* + \|X B_R R^* O_R^* B_R^* + H B_R R^* O_R^* B_R^*\|_* \\
=&\|L B_L^* X B_R + L B_L^* H B_R\|_* + \|B_L^* X B_R R^* + B_L^* H B_R R^*\|_*.
\end{aligned} \tag{2.41}$$

$$B_L^* X B_R = \begin{bmatrix} U^* \\ U_\perp^* \end{bmatrix} U_X \Sigma_X V_X^* \begin{bmatrix} V & V_\perp \end{bmatrix} = \begin{bmatrix} U^* U_X \Sigma_X V_X^* V \ast & 0 \\ 0 & 0 \end{bmatrix} \tag{2.42}$$

Thus

$$L B_L^* X B_R = \begin{bmatrix} L_{11} & L_{12} & \\ & L_{22} & \\ & & I_{n-2r} \end{bmatrix} \begin{bmatrix} U^* U_X \Sigma_X V_X^* V & 0 \\ 0 & 0_{n-r} \end{bmatrix} = \begin{bmatrix} L_{11} U^* U_X \Sigma_X V_X^* V & 0 \\ 0 & 0_{n-r} \end{bmatrix} \tag{2.43}$$

Similarly,

$$B_L^* X B_R R^* = \begin{bmatrix} U^* U_X \Sigma_X V_X^* V R_{11} & 0 \\ 0 & 0_{n-r} \end{bmatrix} \tag{2.44}$$

Let

$$\overline{T} := \left\{ \overline{Z} \in \mathbb{R}^{n\times n} : \overline{Z} = \begin{bmatrix} \overline{Z}_{11} & \overline{Z}_{12} \\ \overline{Z}_{21} & 0 \end{bmatrix} \right\}$$

then for any $\overline{Z} = \begin{bmatrix} \overline{Z}_{11} & \overline{Z}_{12} \\ \overline{Z}_{21} & \overline{Z}_{22} \end{bmatrix}$, where $\overline{Z}_{11} \in \mathbb{R}^{r\times r}$ and $\overline{Z}_{22} \in \mathbb{R}^{(n-r)\times(n-r)}$ the orthogonal projection onto $\overline{T}$ and its complement are defined as $\mathcal{P}_{\overline{T}}(\overline{Z}) = \begin{bmatrix} \overline{Z}_{11} & \overline{Z}_{12} \\ \overline{Z}_{21} & 0 \end{bmatrix}$ and $\mathcal{P}_{\overline{T}_\perp}(\overline{Z}) = \begin{bmatrix} 0 & \\ & \overline{Z}_{22} \end{bmatrix}$. Denote $B_L^*HB_R$ as $\overline{H}$, decompose $\overline{H}$ by $B_L^*HB_R = \mathcal{P}_{\overline{T}}(\overline{H}) + \mathcal{P}_{\overline{T}_\perp}(\overline{H})$ Thus the last term in (2.41) become

$$\begin{aligned}
&\|LB_L^*XB_R + LB_L^*HB_R\|_* + \|B_L^*XB_RR^* + B_L^*HB_RR^*\|_* \\
=&\|LB_L^*XB_R + L\mathcal{P}_{\overline{T}}(\overline{H}) + L\mathcal{P}_{\overline{T}_\perp}(\overline{H})\|_* + \|B_L^*XB_RR^* + \mathcal{P}_{\overline{T}}(\overline{H})R^* + \mathcal{P}_{\overline{T}_\perp}(\overline{H})R^*\|_* \\
=&\|LB_L^*XB_R + L\mathcal{P}_{\overline{T}}(\overline{H}) + L\mathcal{P}_{\overline{T}_\perp}(\overline{H}) - \mathcal{P}_{\overline{T}_\perp}(\overline{H}) + \mathcal{P}_{\overline{T}_\perp}(\overline{H})\|_* \\
&+ \|B_L^*XB_RR^* + \mathcal{P}_{\overline{T}}(\overline{H})R^* + \mathcal{P}_{\overline{T}_\perp}(\overline{H})R^* - \mathcal{P}_{\overline{T}_\perp}(\overline{H}) + \mathcal{P}_{\overline{T}_\perp}(\overline{H})\|_* \\
=&\left\| LB_L^*XB_R + L\mathcal{P}_{\overline{T}}(\overline{H}) + L\begin{bmatrix} 0 & & \\ & \overline{H}_{22} & \overline{H}_{23} \\ & \overline{H}_{32} & \overline{H}_{33} \end{bmatrix} - \begin{bmatrix} 0 & & \\ & \overline{H}_{22} & \overline{H}_{23} \\ & \overline{H}_{32} & \overline{H}_{33} \end{bmatrix} + \mathcal{P}_{\overline{T}_\perp}(\overline{H}) \right\|_* \\
&+ \left\| B_L^*XB_RR^* + \mathcal{P}_{\overline{T}R^*}(\overline{H}) + \begin{bmatrix} 0 & & \\ & \overline{H}_{22} & \overline{H}_{23} \\ & \overline{H}_{32} & \overline{H}_{33} \end{bmatrix} R^* - \begin{bmatrix} 0 & & \\ & \overline{H}_{22} & \overline{H}_{23} \\ & \overline{H}_{32} & \overline{H}_{33} \end{bmatrix} + \mathcal{P}_{\overline{T}_\perp}(\overline{H}) \right\|_* \qquad (2.45) \\
:=&\|LB_L^*XB_R + L\mathcal{P}_{\overline{T}}(\overline{H}) + L\overline{H}_\perp - \overline{H}_\perp + \mathcal{P}_{\overline{T}_\perp}(\overline{H})\|_* \\
&+ \|B_L^*XB_RR^* + \mathcal{P}_{\overline{T}}(\overline{H})R^* + \overline{H}_\perp R^* - \overline{H}_\perp + \mathcal{P}_{\overline{T}_\perp}(\overline{H})\|_* \\
\geq&\|LB_L^*XB_R + \mathcal{P}_{\overline{T}_\perp}(\overline{H})\|_* - \|L\mathcal{P}_{\overline{T}}(\overline{H})\|_* - \|L\overline{H}_\perp - \overline{H}_\perp\|_* \\
&+ \|B_L^*XB_RR^* + \mathcal{P}_{\overline{T}_\perp}(\overline{H})\|_* - \|\mathcal{P}_{\overline{T}}(\overline{H})R^*\|_* - \|\overline{H}_\perp R^* - \overline{H}_\perp\|_*, \\
=&\|LB_L^*XB_R\|_* + \|\mathcal{P}_{\overline{T}_\perp}(\overline{H})\|_* - \|L\mathcal{P}_{\overline{T}}(\overline{H})\|_* - \|L\overline{H}_\perp - \overline{H}_\perp\|_* \\
&+ \|B_L^*XB_RR^*\|_* + \|\mathcal{P}_{\overline{T}_\perp}(\overline{H})\|_* - \|\mathcal{P}_{\overline{T}}(\overline{H})R^*\|_* - \|\overline{H}_\perp R^* - \overline{H}_\perp\|_*,
\end{aligned}$$

where the last inequality uses the triangle inequality. Above, we define

$$\overline{H}_\perp := \begin{bmatrix} 0 & & \\ & \overline{H}_{22} & \overline{H}_{23} \\ & \overline{H}_{32} & \overline{H}_{33} \end{bmatrix}$$

and the last equation is because

$$LB_L^* X B_R = \begin{bmatrix} L_{11} U^* U_X \Sigma_X V_X^* V & 0 \\ 0 & 0 \end{bmatrix} \text{ and } B_L^* X B_R R^* = \begin{bmatrix} U^* U_X \Sigma_X V_X^* V R_{11} & 0 \\ 0 & 0 \end{bmatrix}$$

as well as $\|A+B\|_* = \|A\|_* + \|B\|_*$ when $AB^* = A^* B = 0$ [RFP10] Thus combine (2.41) and (2.45), we have

$$\begin{aligned} & \|\mathcal{P}_{\overline{T}_\perp}(\overline{H})\|_* + \|\mathcal{P}_{\overline{T}_\perp}(\overline{H})\|_* \\ \leq & \|L\mathcal{P}_{\overline{T}}(\overline{H})\|_* + \|L\overline{H}_\perp - \overline{H}_\perp\|_* + \|\mathcal{P}_{\overline{T}}(\overline{H})R^*\|_* - \|\overline{H}_\perp R^* - \overline{H}_\perp\|_* \end{aligned} \tag{2.46}$$

Next, we simplify the terms above. First notice that

$$\begin{aligned} & \begin{bmatrix} 0 & & \\ & L_{22} & \\ & & I_{n-2r} \end{bmatrix} \mathcal{P}_{\overline{T}}(\overline{H}) \begin{bmatrix} 0 & & \\ & I_r & \\ & & I_{n-2r} \end{bmatrix} \\ = & \begin{bmatrix} 0 & & \\ & L_{22} & \\ & & I_{n-2r} \end{bmatrix} \begin{bmatrix} \overline{H}_{11} & \overline{H}_{12} & \overline{H}_{13} \\ \overline{H}_{21} & 0_r & \\ \overline{H}_{31} & & 0_{n-2r} \end{bmatrix} \begin{bmatrix} 0 & & \\ & I_r & \\ & & I_{n-2r} \end{bmatrix} = 0_n \end{aligned} \tag{2.47}$$

Then the first term in the right hand side is simplified as

$$
\begin{aligned}
&\|L\mathcal{P}_{\overline{T}}(\overline{H})\|_* \\
=&\left\| \begin{bmatrix} L_{11} & L_{12} & \\ & L_{22} & \\ & & I_{n-2r} \end{bmatrix} \mathcal{P}_{\overline{T}}(\overline{H}) I_n - \begin{bmatrix} 0 & & \\ & L_{22} & \\ & & I_{n-2r} \end{bmatrix} \mathcal{P}_{\overline{T}}(\overline{H}) \begin{bmatrix} 0 & & \\ & I_r & \\ & & I_{n-2r} \end{bmatrix} \right\|_* \\
=&\left\| \begin{bmatrix} L_{11} & L_{12} & \\ & 0_r & \\ & & 0_{n-2r} \end{bmatrix} \mathcal{P}_{\overline{T}}(\overline{H}) I_n + \begin{bmatrix} 0 & & \\ & L_{22} & \\ & & I_{n-2r} \end{bmatrix} \mathcal{P}_{\overline{T}}(\overline{H}) \begin{bmatrix} I_r & & \\ & 0_r & \\ & & 0_{n-2r} \end{bmatrix} \right\|_* \\
\leq&\|[L_{11}\ L_{12}]\|\|\mathcal{P}_{\overline{T}}(\overline{H})\|_* + \max\{\|L_{22}\|,1\}\|\mathcal{P}_{\overline{T}}(\overline{H})\|_* \\
\leq&\|[L_{11}\ L_{12}]\|\|\mathcal{P}_{\overline{T}}(\overline{H})\|_* + \|\mathcal{P}_{\overline{T}}(\overline{H})\|_*
\end{aligned}
\tag{2.48}
$$

The first inequality above used the fact that $\|AB\|_* \leq \|A\|\|B\|_*$ for all matrix $A,B \in \mathbb{R}^{n\times n}$. The second inequality is obtained by $\|L_{22}\| \leq \|L\| = \|Q_{\widetilde{U},w}\| = 1$, and the second equation used the polarization identity:

$$
AZC - BZD = (A-B)ZC + BZ(C-D) \tag{2.49}
$$

for matrices $A,B,C,D,Z \in \mathbb{R}^{n\times n}$.

$$
\|\mathcal{P}_{\overline{T}}(\overline{H})R^*\|_* \leq \quad \|[R_{11}\ R_{12}]\|\|\mathcal{P}_{\overline{T}}(\overline{H})\|_* + \|\mathcal{P}_{\overline{T}}(\overline{H})\|_* \tag{2.50}
$$

The second term on the right-hand size of (2.46) may also be bounded as:

$$\begin{aligned}
&\|L\overline{H}_\perp - \overline{H}_\perp\|_* \\
=&\left\| L\overline{H}_\perp I - \begin{bmatrix} L_{11} & & \\ & I_r & \\ & & I_{n-2r} \end{bmatrix} \overline{H}_\perp I \right\|_* \\
=&\left\| \begin{bmatrix} 0_r & L_{12} & \\ & L_{22} - I_r & \\ & & 0_{n-2r} \end{bmatrix} \overline{H}_\perp I - \begin{bmatrix} L_{11} & & \\ & I_r & \\ & & I_{n-2r} \end{bmatrix} \overline{H}_\perp 0_n \right\|_* \\
\leq&\|[L_{12}\; L_{22} - I_r]\| \|\overline{H}_\perp\|_*
\end{aligned} \tag{2.51}$$

The inequality uses $\|AB\|_* \leq \|A\| \|B\|_*$. The second equation uses (2.49). In the same way,

$$\|\overline{H}_\perp R^* - \overline{H}_\perp\|_* \leq \|[R_{12}\; R_{22} - I_r]\| \|\overline{H}_\perp\|_* \tag{2.52}$$

Then we have

$$\begin{aligned}
2\|\mathcal{P}_{\overline{T}_\perp}(\overline{H})\|_* \leq& \|L\mathcal{P}_{\overline{T}}(\overline{H})\|_* + \|L\overline{H}_\perp - \overline{H}_\perp\|_* + \|\mathcal{P}_{\overline{T}}(\overline{H})R^*\|_* - \|\overline{H}_\perp R^* - \overline{H}_\perp\|_* \\
\leq& \left(\|[L_{11}\; L_{12}]\| + \|[R_{11}\; R_{12}]\|\right) \|\mathcal{P}_{\overline{T}}(\overline{H})\|_* \\
&+ \left(\|[L_{12}\; L_{22} - I_r]\| + \|[R_{12}\; R_{22} - I_r]\|\right) \|\overline{H}_\perp\|_* + 2\|\mathcal{P}_{\overline{T}}(\overline{H})\|_* \\
\leq& \left(\sqrt{\frac{w^4 \cos^2 u_1 + \sin^2 u_1}{w^2 \cos^2 u_1 \sin^2 u_1}} + \sqrt{\frac{\lambda^2 \cos^2 v_1 + \sin^2 v_1}{\lambda^2 \cos^2 v_1 + \sin^2 v_1}} \right) \|\mathcal{P}_{\overline{T}}(\overline{H})\|_* \\
&+ \left(\sqrt{\frac{2(1-w^2)\sin^2 u_1}{w^2 \cos^2 u_1 + \sin^2 u_1}} + \sqrt{\frac{2(1-\lambda^2)\sin^2 v_1}{\lambda^2 \cos^2 v_1 + \sin^2 v_1}} \right) \|\overline{H}_\perp\|_* + 2\|\mathcal{P}_{\overline{T}}(\overline{H})\|_*
\end{aligned} \tag{2.53}$$

In the third inequality above, we applied Lemma 4 in [EYW18]. Since we know that

$$\|\mathcal{P}_T(H)\| = \|\mathcal{P}_{\overline{T}}(\bar{(}H)), \|\mathcal{P}_{T_\perp}(H)\| \quad = \|\mathcal{P}_{\overline{T}_\perp}(\bar{(}H))\|_*, \tag{2.54}$$

Since the linear subspace $\widetilde{T}_\perp \subset T_\perp \subset \mathbb{R}^{n\times n}$ as

$$\widetilde{T}_\perp := \left\{ Z \in \mathbb{R}^{n\times n} : Z = B_L \begin{bmatrix} 0_r & & \\ & \overline{Z}_{22} & \overline{Z}_{23} \\ & \overline{Z}_{32} & \overline{Z}_{33} \end{bmatrix} B_{R^*} \right\} \tag{2.55}$$

then we can write $\|\overline{H}_\perp\|_* = \|B_L \overline{H}_\perp B_R^*\|_* = \|\mathcal{P}_{\widetilde{T}_\perp}(H)\|_*$ since nuclear norm is invariant under orthonormal matrices. Thus we have

$$\begin{aligned} 2\|\mathcal{P}_{T_\perp}(H)\|_* \le & \left(\sqrt{\frac{w^4\cos^2 u_1 + \sin^2 u_1}{w^2\cos^2 u_1 \sin^2 u_1}} + \sqrt{\frac{\lambda^2\cos^2 v_1 + \sin^2 v_1}{\lambda^2\cos^2 v_1 + \sin^2 v_1}} \right) \|\mathcal{P}_{\overline{T}}(\overline{H})\|_* \\ & + \left(\sqrt{\frac{2(1-w^2)\sin^2 u_1}{w^2\cos^2 u_1 + \sin^2 u_1}} + \sqrt{\frac{2(1-\lambda^2)\sin^2 v_1}{\lambda^2\cos^2 v_1 + \sin^2 v_1}} \right) \|\mathcal{P}_{\widetilde{T}_\perp}(H)\|_* + 2\|\mathcal{P}_T(H)\|_* \end{aligned} \tag{2.56}$$

This implies that if every matrix $H \neq 0$ s.t. $\mathscr{A}(H) = 0$ satisfies

$$\begin{aligned} & \left(\sqrt{\frac{w^4\cos^2 u_1 + \sin^2 u_1}{w^2\cos^2 u_1 \sin^2 u_1}} + \sqrt{\frac{\lambda^2\cos^2 v_1 + \sin^2 v_1}{\lambda^2\cos^2 v_1 + \sin^2 v_1}} \right) \|\mathcal{P}_T(H)\|_* \\ & + \left(\sqrt{\frac{2(1-w^2)\sin^2 u_1}{w^2\cos^2 u_1 + \sin^2 u_1}} + \sqrt{\frac{2(1-\lambda^2)\sin^2 v_1}{\lambda^2\cos^2 v_1 + \sin^2 v_1}} \right) \|\mathcal{P}_{\widetilde{T}_\perp}(H)\|_* + 2\|\mathcal{P}_T(H)\|_* \\ & < 2\|\mathcal{P}_{T_\perp}(H)\|_* \end{aligned} \tag{2.57}$$

Then $Z = X$, i.e. the optimizer is the unique solution.

□

2.3.5 Minimum number of measurements given a prefect support estimate

Let $f(Z) = \|(w\mathcal{P}_{\widetilde{U}} + \mathcal{P}_{\widetilde{U}^\perp})Z\|_* + \|Z(\lambda\mathcal{P}_{\widetilde{V}} + \mathcal{P}_{\widetilde{V}^\perp})\|_*$. Suppose that we have the perfect estimate, i.e. $\mathcal{P}_{\widetilde{U}} = \mathcal{P}_U$, $\mathcal{P}_{\widetilde{V}} = \mathcal{P}_V$, and we set $w = \lambda = 0$ accordingly. Then X is the matrix that minimize the function f, because $f(X) = \|\mathcal{P}_{U^\perp}X\|_* + \|X\mathcal{P}_{V^\perp}\|_* = 0$. Let $Z = H + X$, where $H \in \text{Null}(\mathscr{A})$, then Z is feasible. Consider

$$\begin{aligned} f(Z) &= \|(w\mathcal{P}_{\widetilde{U}} + \mathcal{P}_{\widetilde{U}^\perp})(X+H)\|_* + \|(X+H)(\lambda\mathcal{P}_{\widetilde{V}} + \mathcal{P}_{\widetilde{V}^\perp})\|_* \\ &= \|\mathcal{P}_{U^\perp}(X+H)\|_* + \|(X+H)\mathcal{P}_{V^\perp}\|_* \\ &= \|\mathcal{P}_{U^\perp}H\|_* + \|H\mathcal{P}_{V^\perp}\|_*. \end{aligned}$$

Let Z be another minimizer of f, i.e. $f(Z) = 0$. Then H must satisfy:

$$\begin{cases} \mathcal{P}_{U^\perp}H = 0 \\ H\mathcal{P}_{V^\perp} = 0 \\ \mathscr{A}(H) = 0 \end{cases} \tag{2.58}$$

To find the condition of "X being the unique minimizer of (2.25), we need (2.58) to have the unique solution $H = 0$. Let $A = \begin{bmatrix} \overrightarrow{A_1}^T \\ \overrightarrow{A_2}^T \\ \cdots \\ \overrightarrow{A_m}^T \end{bmatrix}$ and rewrite (2.58) as

$$\begin{bmatrix} I \otimes \mathcal{P}_{U^\perp} \\ \mathcal{P}_{V^\perp} \otimes I \\ A \end{bmatrix} \overrightarrow{H} = \overrightarrow{0} \tag{2.59}$$

For this system to hold, H must satisfy

$$P_U H P_V = H,$$

i.e. H lives in the space spanned by $\{U, V\}$. Thus we can write H as

$$H = URV^*,$$

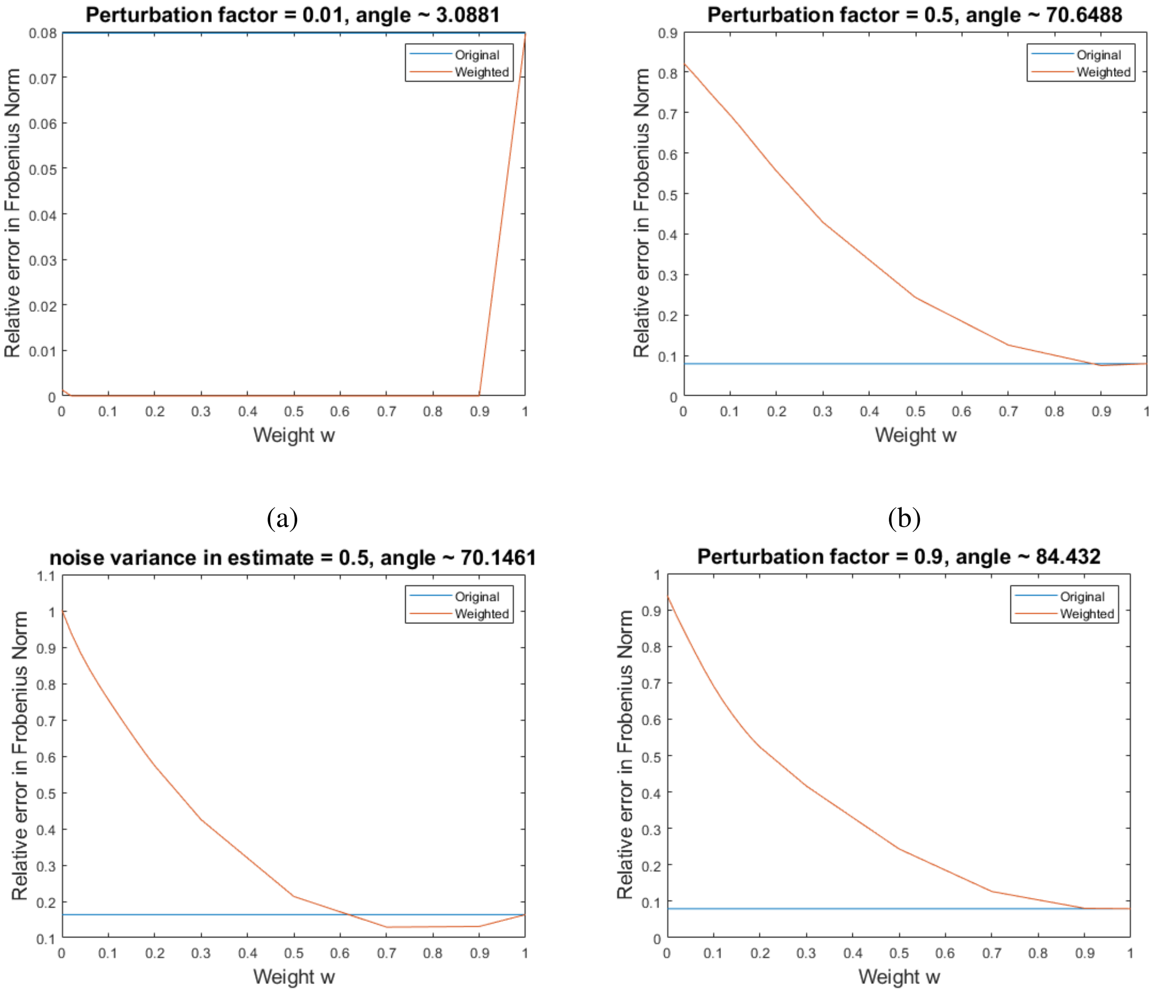
for some $R \in \mathbb{R}^{r^2}$, Thus to make $A\overrightarrow{H} = \overrightarrow{0}$, A must have exactly $m = r^2$ measurements.

2.4 Numerical experiments

In this section, we present numerical simulatiosn designed to evaluate the algorithms covered in this chapter. All simulations were performed using CVX[GB14].

First, we construct rank r matrix $X \in \mathcal{R}^{n_1 \times n_2}$ with $n = n_1 = n_2$, $X = U\Sigma V^*$, we draw U and V by orthogonalizing the columns of a standard random Gaussian matrix $G \in \mathbb{R}^{n \times r}$, and draw singular values of X from uniform distribution. We construct the prior information by adding a perturbation to the original U, V, i.e. $\widetilde{U} = U + \delta G$, where G is the random Gaussian matrix with i.i.d $\mathcal{N}(0,1)$ entries, δ is the perturbation factor. Thus the perturbation matrix has independent random Gaussian entries with mean zero and variance δ. $\widetilde{V}$ is obtained by doing the same kind of perturbation.

We sample without noise using random Gaussian measurement matrix, for various number of total measurements. Figure 2.1 shows the relative recovery error of *Weighted Nuclear Norm Minimization* (the first modification we studied) compared with standard nuclear norm minimization for $n = 25$, $r = 3$,$w = w = \lambda$, and with support estimate as $\widetilde{U} = U + \delta G$, $\widetilde{V} = V + \delta G$, where G is random Gaussian matrix with i.i.d. $\mathcal{N}(0,1)$ entries, and δ are chosen as follows: (a) $\delta = 0.01$, (b) $\delta = 0.1$, (c) $\delta = 0.5$, (d) $\delta = 0.9$.



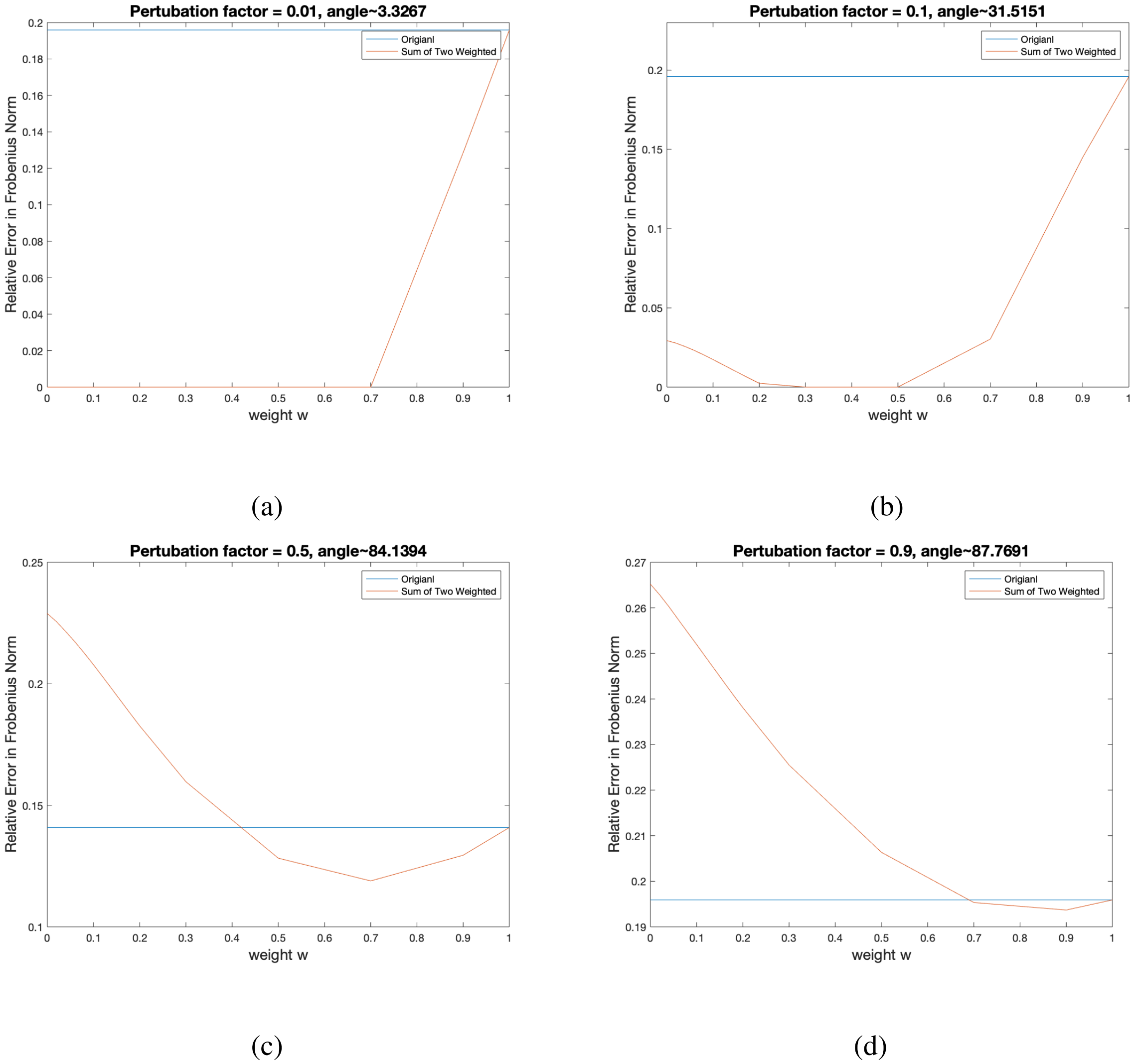
(a)

(b)

(c)

(d)

Figure 2.1: Relative recovery error of *Weighted Nuclear Norm Minimization* and original nuclear norm minimization δ, (a) $\delta = 0.01$, (b) $\delta = 0.1$, (c) $\delta = 0.5$, (d) $\delta = 0.9$



(a)

(b)

(c)

(d)

Figure 2.2: Relative recovery error of *Sum of Two Weighted Nuclear Norm Minimization* and the original nuclear norm minimization (a) $\delta = 0.01$, (b) $\delta = 0.1$, (c) $\delta = 0.5$, (d) $\delta = 0.9$

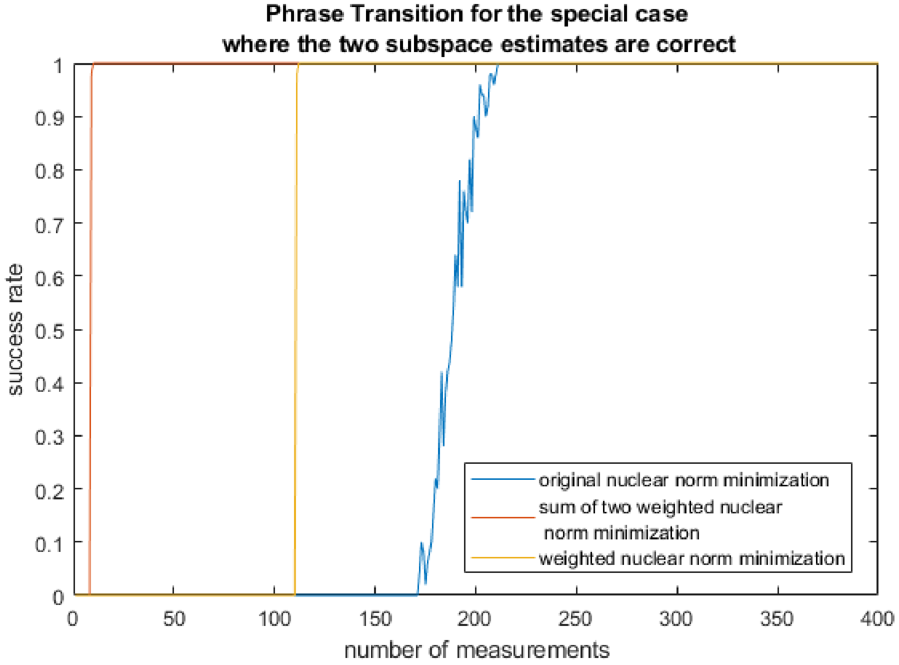


Figure 2.3: Phrase transition curves for the nuclear norm minimization and the two modified optimizations that we proposed, when the subspace estimate are perfectly correct

Figure 2.2 shows the relative recovery error associated with *Minimizing the Sum of Two Weighted Nuclear Norms* (the second modification we proposed) compared with the original nuclear norm minimization for $n = 20$, $r = 3$, $w = w = \lambda$, and with support estimate as $\widetilde{U} = U + \delta G$, $\widetilde{V} = V + \delta G$, where G is random Gaussian matrix with i.i.d. $\mathcal{N}(0,1)$ entries, and δ are choosing as follows: (a) $\delta = 0$, (b) $\delta = 0.1$, (c) $\delta = 0.5$, (d) $\delta = 0.9$.

Figure 2.3 shows the phrase transition curves for prefect prior information. In this case for the two modified optimizations, $w = \lambda = 0$. We can see that the *Sum of Two Weighted Nuclear Norm Minimization* outperforms the other two.

Figure 2.4 show the recovery of *Sum of Two Weighted Nuclear Norm Minimization* with $\widetilde{U} = U + 0.005G$ and $\widetilde{V} = V$, $\lambda = 0$ and for different w.

Figure 2.5 show the recovery of *Sum of Two Weighted Nuclear Norm Minimization* with $\widetilde{U} = U + 0.005G$ and $\widetilde{V} = V + 0.005G$, $\lambda = w$ for different w.

We can see from the above result that the alternative *Sum of Two Nuclear Norm Mini-*

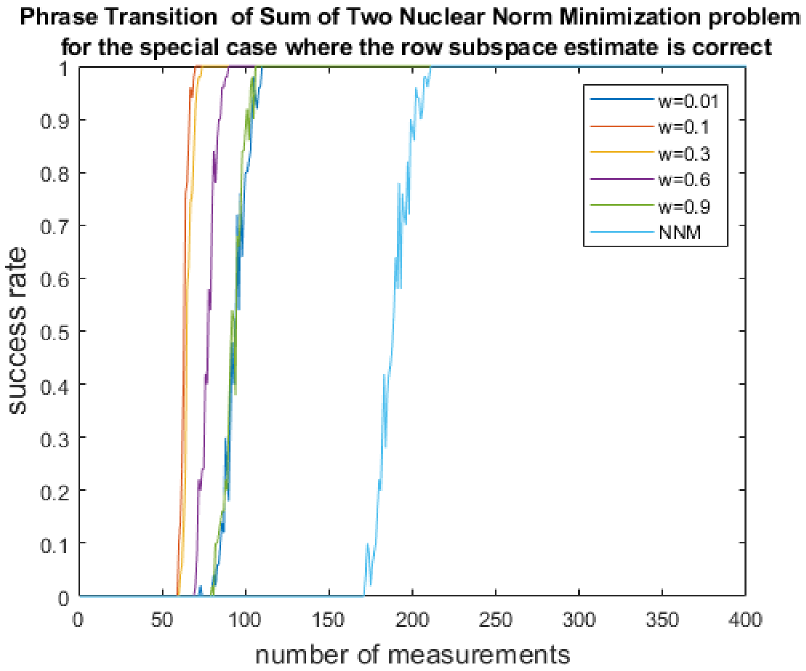


Figure 2.4: Phrase transition curves for the *Sum of Two Nuclear Norm Minimization* when one side of subspace estimate is correct with different choice of weight on the other side.

mization helps reduce the number of Gaussian measurements needed for exact recovery when providing proper prior information.

2.5 Conclusion and Future Work

In this chapter, we studied the problem of recovering a low-rank matrix X from compressed linear measurements

$$y = \mathscr{A}(X).$$

We first study the null space property of the original nuclear norm minimization by analyzing the sampling complexity of random Gaussian measurements. We proposed an alternative proof technique to show that when $m \gtrsim r(n_1 + n_2)$ the random Gaussian measurement satisfies the Null Space Property with high probability. Then we focused on the case when prior information about the support of the target matrix are provided. We proposed two optimization problems

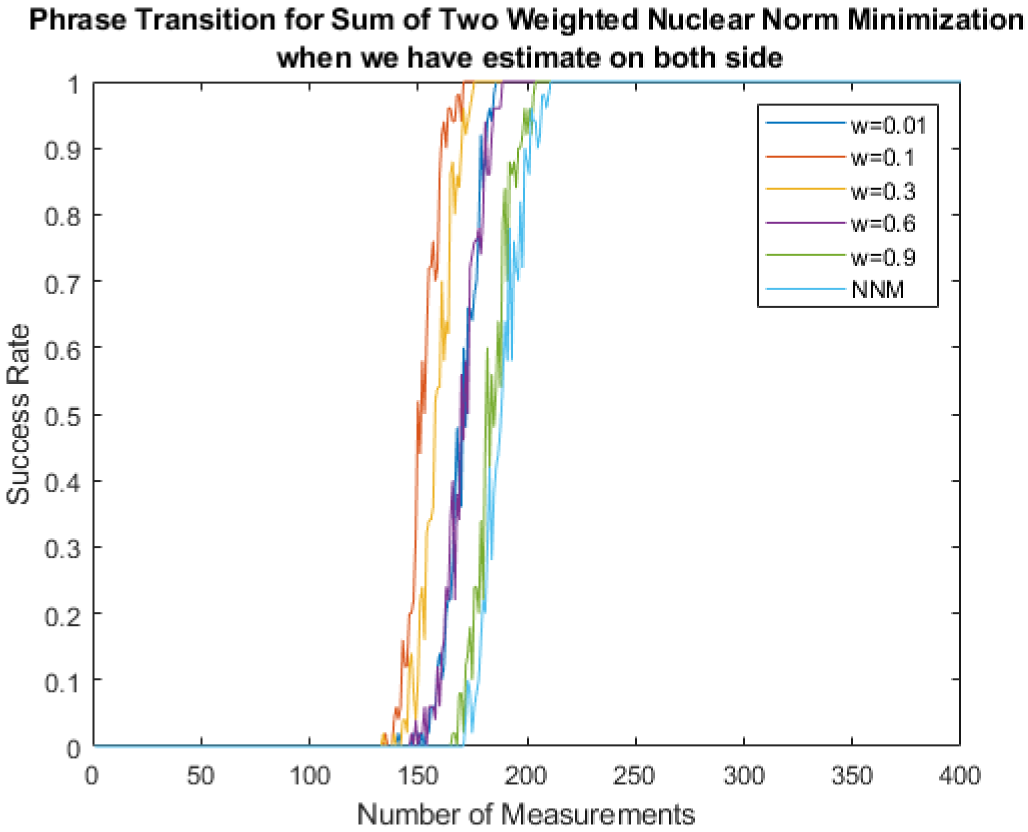


Figure 2.5: Phrase transition curves for the *Sum of Two Nuclear Norm Minimization* when both column and row subspace estimate has perturbations

i.e. the *Weighted Nuclear Norm Minimization* and the *Sum of Two Weighted Nuclear Norm Minimization*. We gave null space based conditions for exact recovery for both optimizations. Next, we analyze the sampling complex of the two optimizations when the support estimates are correct. We shows that when having correct estimates on both column and row space, the *Weighted Nuclear Norm Minimization* still needs $O(nr)$ measurements for exact recovery where as the *Sum of Two Weighted Nuclear Norm Minimization* only need $O(r^2)$ measurements. Finally, we present the numerical experiment to support our analysis. We leave the work of analyzing the number of Gaussian measurements needed to satisfy the corresponding NSP and the optimum weight choosing for different optimizations as future work.

2.6 Acknowledgements

Chapter 2, in full is currently being prepared for submission for publication of the material, co-authored with Saab, Rayan. The dissertation author was the primary investigator and author of this material.

Chapter 3

Learning Dictionaries with Fast Transforms

3.1 Introduction

The exponential growth of data has dramatically increased the interest in finding sparse signal representations in recent years. As we will discuss shortly, with sparse representation, we can not only store information more efficiently [Wal92] but also process the signals faster, and even collect measurements more efficiently [Don06]. One can benefit from using sparse representations to generate the signals [YWHM10], or use them in other applications [CNT11], [CYL13]. Applications that can benefit from the sparsity concepts include denoising [DJ94], [CD95], [SCD02], compression [MGBB00], compressive sensing [Don06], [GKSA11], [LDP07] and more.

When measuring signals, noise is often measured together with the underlying true signals. The goal of denoising is to separate the true signals from the noise. Due to the fact that lots of signals of interest are sparse under certain basis (e.g. images are sparse under wavelet basis) while noise is not (e.g. white noise), we can remove the noise by approximating the measured signals using sparse representation. Wavelet methods and shift-invariant variations that exploit over-complete representation are some of the the best algorithms for this task [DJ94], [CD95], [SCD02].

Applications such as data storage, information backup, signal processing, transmission and communication require large volumes of memory, operations and computational resources if the signals are large. Data compression can significantly reduce the size of a file, so that the time and memory cost for processing it can be decreased accordingly. One of the most commonly used compression algorithms, JPEG2000 coding system, is successful due the fact that natural images are sparse under the wavelet basis [MGBB00].

Compressed sensing allows acquiring and reconstructing signals more efficiently by solving an undetermined linear system when we have structural informations about the target signals [Don06], [GKSA11], [LDP07]. This decreases the number of measurements required and therefore reduces the storage and transmission resources needed. Specialized equipment is usually involved like “Rice single pixel camera” [DDT$^+$08], and the machines used in sparse MRI [LDP07]. In these examples the signal can be compressed in a more efficient way at the time of sampling.

3.1.1 The Dictionary

We assume that the signal $y \in \mathbb{C}^N$ can be described as $y = Dx$, where $D \in \mathbb{C}^{N \times n}$ is a linear mapping called a dictionary, $x \in \mathbb{C}^n$ is the representation of the signal y under the dictionary D, and is assumed to be sparse. The columns of the dictionary D are often called atoms [DH01] [AEB06]. A dictionary that leads to sparse representations can either be chosen as a pre-specified set of functions or designed by learning to fit a given set of signal examples through learning algorithms.

When choosing a dictionary that fits a particular application, one can consider pre-constructed dictionaries, such as undecimated wavelets [SFM07], contourlets [DV05], curvelets

[SCD02], etc. Many of these are specifically designed for images, especially when dealing with cartoon-like image content, given that they are piecewise smooth and with smooth boundaries [Ela10]. The choice of using a pre-specified transform matrix is appealing as it is simple. It also often leads to fast algorithms for the computation of the sparse representation and original signal recovery. For example, Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) algorithms [Nus81] compute the discrete Fourier transform (DFT) and its inverse for a n-dimensional signal with $O(n \log n)$ complexity instead of $O(n^2)$. When choosing the predefined dictionaries, tight frames that are easy to invert are usually preferred. In applications the performance of these dictionaries rely on how well they can sparingly represent the target signals.

While pre-constructed (model driven) dictionaries like the Fast Fourier Transform (FFT) can lead to fast transforms, they can be limited in their ability to sparsify the signals they are designed to handle. Furthermore, most of those dictionaries are restricted to signals or images of a certain type, and cannot be used for a new and arbitrary family of signals of interest. To handle these potential short-comings, we turn to a learning-based (i.e., data driven) approach for obtaining dictionaries. This route is to design a dictionary from data based on learning technique. Indeed, it has been shown that using a learned dictionary from training data rather than fixed frames like Fourier or Wavelets basis derives better results in many practical applications such as face recognition [HDRL16], [ZL10], image de-noising [DLZS11], [EA06], image super-resolution [YWL+12], [ZZX+12] and image segmentation [ZZM12]. On the other hand, dictionaries learned from data usually do not exhibit the kind of structure that yields a fast transform. Thus, applying these dictionaries to a vector will usually require $O(n^2)$ operations, which when n is large can be prohibitive, especially if the dictionary needs to also be applied to a large data-set.

In this chapter, we will introduce an approach to dictionary learning which combines the computational efficiency of certain model-based sparsifying transforms – such as the Discrete Fourier Transform, with the advantages of data-driven dictionaries. That is, we use data to learn dictionaries that admit a fast algorithm for the corresponding linear transform. Our algorithm is based on alternating minimization, where we minimize a sequence of convex functions, and finds $n \times n$ dictionaries that can be applied in $O(n \log n)$ time.

3.1.2 Related Work

Dictionary learning is a branch of signal processing and machine learning. Its goal is to find the dictionary which makes a class of signal admit sparse representations. Given the input dataset of p signals, $Y = [y_1, y_2, \cdots, y_p]$, $y_i \in \mathbb{C}^N$, we wish to find the dictionary $D = [d_1, d_2, \cdots, d_n] \in \mathbb{C}^{N \times n}$ and a representation $X = [x_1, x_2, \cdots, x_p]$, $x_i \in \mathbb{C}^n$, such that $\|Y - DX\|_F$ is minimized and such that x_i's are sparse for each i.This leads, for example, to the optimization problem:

$$\underset{D \in \mathbb{C}^{N \times n}, x_i \in \mathbb{C}^n}{minimize} \sum_{i=1}^{p} \|y_i - Dx_i\|_{\ell_2} + \lambda \|x_i\|_{\ell_1}. \tag{3.1}$$

The ℓ_1 norm, defined as the sum of absolute values of all elements of a vector, is the convex envelop of the "ℓ_0 norm" (the number of non-zero entries in a vector) and has been shown to lead to sparse results [DE03]. One may also add constraints on columns of D so that they have unit ℓ_2 norm to ensure that the values of d_i does not going arbitrary large which leads to the values of x_i being arbitrary small but not zero [MPS$^+$09].

The above optimization problem is convex with respect to either the dictionary D or the sparse representation x while fixing the other one of the two. Thus, it is common in dictionary learning to solve the problem using iterative methods that alternate between sparse coding of the signals based on the current dictionary and an update process for the dictionary atoms so as to better fit the data [AEB06].

Sparse coding algorithms

Sparse coding is the process of recovering the representation coefficients x based on the given signal y and the current dictionary D. The task is to solve the optimization problem

$$\underset{x}{minimize}\|x\|_0 \text{ subject to } y = Dx \tag{3.2}$$

or

$$\underset{x}{minimize}\|x\|_0 \text{ subject to } \|Dx - y\|_2 \leq \varepsilon, \tag{3.3}$$

As we discussed in Chapter 2, looking for sparsest representation by minimizing ℓ_0 norm from linear measurement is NP-hard. Instead, we consider the algorithms that look for the sparse approximation of y. To find x in (3.2) we noticed that the unknown x is composed by two parts to be found: the support which indicates the location of non-zero coefficients and the coefficients on the support. Thus there are two different ideas to find the solution. One way to solve x is to find the support first and then simply use Least-Square to calculate the values on the support. This leads to family of greedy algorithms. The simplest greedy algorithm is the Orthogonal Matching Pursuit (OMP) algorithm [PRK93], [DMA97]. It is an iterative method that selects at each step the column which is most correlated with the current residuals [CW11]. The algorithm is simple and easy to implement.

An alternative route of solving x is to ignore the support, try to smooth the ℓ_0 penalty function and solve the problem by optimizing a continuous function instead. The common used algorithm for this perspective is called Basic Pursuit (BP) [CDS01], [DE03]. It uses ℓ_1 norm as convex relaxation of ℓ_0 norm. Theoretical analysis on those pursuit algorithms show that those algorithm can successfully solve x exactly given x is sparse or approximately sparse [DH01] [EB02], [TG07], [Fuc04]. Further results about stable recovery in sparse approximations is established in [DET05], [Tro04].

Training of the dictionaries

The method of optimal directions (MOD) [EAH99] is presented by Engan et al. in 1999. Here the whole dictionary learning algorithm alternates between sparse coding stage using Orthogonal Matching Pursuit (OMP) and the dictionary updating stage by applying Moore-Penrose pseudo-inverse. Fixing the sparse coding for each signal, define the error by $e_i = y_i - Dx_i$. Then the sum of square errors over all the signals is

$$\|E\|_F^2 = \|[e_1, e_2, \cdots, e_p]\|_F^2 = \|Y - DX\|_F^2$$

Since at the nth iteration, the estimate $X^{(n)}$ of X is fixed, we can seek D by minimizing the error above:

$$\begin{aligned} D^{(n+1)} &= \underset{D}{argmin} \|Y - DX^{(n)}\|_F^2 \\ &= YX^{(n)^T}(X^{(n)}X^{(n)^T})^{-1} \\ &= YX^{(n)^\dagger}. \end{aligned}$$

One then renormalize $D^{(n+1)}$ to fit the unit norm constrain.

K-SVD is another algorithm proposed by Aharon et al.[AEB06], it also alternates sparse coding and dictionary updating. The algorithm is flexible and works in conjunction with any pursuit algorithm [ZL10]. In K-SVD, the atoms in D are handled sequentially. Only one column of D is updated each time. In l-th step, the algorithm keeps all the columns fixed except the l-th one, d_l. It iteratively update the l-th column of D by minimizing

$$\|Y - DX\|_F^2 = \left\| Y - \sum_{j=1}^{p} d_j x_j^T \right\|_F^2 = \left\| \left(Y - \sum_{j \neq l} d_j x_j^T \right) - d_l x_l^T \right\|_F^2 = \|E_l - d_l x_l^T\|_F^2, \tag{3.4}$$

where x_l^T denotes the l-th row of X. In last equation of (3.4) we denoted

$$E_l = Y - \sum_{j \neq l} d_j x_j^T, \tag{3.5}$$

The multiplication DX has been decomposed to the sum of rank-1 matrices. Among those, $l-1$ terms are assumed fixed, and the l-th is assumed to be known. The matrix E_l stands for the error for all the examples when the l-th atom is removed. Then it is natural to use singular value decomposition(SVD) to find the best rank-1 approximation of E_l which give us the d_l and x_l^T that minimize the equation (3.4). However, the new solution of vector X_l^T is very likely to be non-sparse as we desired, because the sparsity constraint is not enforced.

```
Algorithm 1: The K-SVD Algorithm for Dictionary Learning
Input: $Y = [y_1, y_2, \cdots, y_p]$
Output: dictionary D
Initialization: Set the dictionary matrix $D$ to have unit $\ell_2$-norm columns, i = 1
while not converge do
    Step 1: For each given example $y_1$, employing the classicial sparse representation
      with $\ell_0$-norm regularization to solve problem
      $\underset{X}{argmin}\|Y - DX\|_F^2$ s.t. $\|x_i\|_0 \leq k, i = 1, 2, \cdots, p$ for further estimating $X^i$, set
      $l = 1$
    while $l \neq k$ do
        Step 2: Compute the overall representation residual $E_l = Y - \sum_{j \neq l} d_j x_j^T$
        Step 3: Extract the column items of $E_l$ which corresponds to the nonzero
          elements of $x_l^T$ and obtain $E_l^R$.
        Step 4: SVD decomposes $E_l^R$ into $E_l^R = U \Lambda V^T$
        Step 5: Update $d_l$ to the first column of $U$ and update corresponding
          coefficients in $x_l^T$ by $\Lambda(1,1)$ times the first column of $V$
        Step 6: $l = l + 1$
```

In order to maintain the sparsity of x_l^T, only the non-zero elements of x_l^T should be preserved. Define ω_l as the group of indices that pointing to examples $\{y_i\}_{i=1}^N$ that use the atom d_l,

$$\omega_l = \{i | 1 \leq i \leq N, x_l^T(i) \neq 0\}.$$

Define Ω_l as matrix of size $N \times |\omega_l|$ with ones on the $(\omega_l(i), i)$ entries and zeros elsewhere. Let $E_l^R = E_l \Omega_l$, and the previous optimization can be rewritten as

$$\|E_l \Omega_l - d_l x_l^T \Omega_l\|_F^2 = \|E_l^R - d_l x_l^R\|_F^2$$

Then, SVD decomposes E_l^R into $E_l^R = U \Lambda V^T$, and then updates dictionary d_l. The algorithm of K-SVD algorithm is summarized to Algorithm 1, and more information can be found in the literature [AEB06]. It may be worth mentioning that we can also use stochastic gradient descent method to update the dictionary [AE08], we first update the dictionary D by stochastic gradient descent and then project the solution into the constrain set

$$\{D = [d_1, \cdots, d_n] \in \mathbb{R}^{N \times n} : \|d_i\| \leq 1, \forall i = 1, 2, \cdots, n\}. \tag{3.6}$$

3.1.3 Content

The rest of this chapter is structured as follows: In section 3.2, we describe our algorithm in details. The numerical experiment involving real image data are given in section 3.3. We conclude and discuss future possible research direction in section 3.4.

3.2 Methodology

3.2.1 DFT and the Fast Fourier Transform (FFT) Algorithms

For a input signal $x = [x_0, \cdots, x_{N-1}] \in \mathbb{C}^N$, the Discrete Fourier Transform (DFT) transforms it into another vector $X = [X_0, X_1, \cdots, X_{N-1}]$, by put the vector into exponential basis:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn}. \tag{3.7}$$

Let

$$F_N = \begin{bmatrix} \omega_N^{0\cdot 0} & \omega_N^{0\cdot 1} & \omega_N^{0\cdot 2} & \cdots & \omega_N^{0\cdot(N-1)} \\ \omega_N^{1\cdot 0} & \omega_N^{1\cdot 1} & \omega_N^{1\cdot 2} & \cdots & \omega_N^{1\cdot(N-1)} \\ \omega_N^{2\cdot 0} & \omega_N^{2\cdot 1} & \omega_N^{2\cdot 2} & \cdots & \omega_N^{2\cdot(N-1)} \\ \vdots & & \vdots & & \vdots \\ \omega_N^{(N-1)\cdot 0} & \omega_N^{(N-1)\cdot 1} & \omega_N^{(N-1)\cdot 2} & \cdots & \omega_N^{(N-1)\cdot(N-1)} \end{bmatrix}, \tag{3.8}$$

where $\omega_N = e^{-\frac{2\pi i}{N}}$ is a primitive N-th root of unity. Then the Discrete Fourier Transform can be written as matrix multiplication by the DFT matrix $F_N \in \mathbb{C}^{N\times N}$:

$$X = F_N x. \tag{3.9}$$

For the Cooley-Tukey Fast Fourier Transform algorithm, we first compute the DFTs of the even indexed inputs $x_{even} = [x_0, x_2, \cdots, x_{N-2}]$ and the odd indexed inputs $x_{odd} = [x_1, x_3, \cdots, x_{N-1}$, and then combines those two results to produce the DFT of the whole sequence.

$$X_k = \sum_{m=0}^{N/2-1} x_{2m} e^{-\frac{2\pi i}{N/2}mk} + e^{-\frac{2\pi i}{N}k} \sum_{i=0}^{N/2-1} x_{2m+1} e^{-\frac{2\pi i}{N/2}mk}$$

$$X_{k+N/2} = \sum_{m=0}^{N/2-1} x_{2m} e^{-\frac{2\pi i}{N/2}mk} - e^{-\frac{2\pi i}{N}k} \sum_{i=0}^{N/2-1} x_{2m+1} e^{-\frac{2\pi i}{N/2}mk}$$

In matrix form:

$$F_N x = \begin{bmatrix} F_{N/2}x_{even} + \Omega_{N/2}F_{N/2}x_{odd} \\ F_{N/2}x_{even} - \Omega_{N/2}F_{N/2}x_{odd} \end{bmatrix} \tag{3.10}$$

$$= \begin{bmatrix} I_{N/2} & \Omega_{N/2} \\ I_{N/2} & -\Omega_{N/2} \end{bmatrix} \begin{bmatrix} F_{N/2} & 0 \\ 0 & F_{N/2} \end{bmatrix} P_N x \tag{3.11}$$

Where $\Omega_{N/2}$ is the diagonal matrix with entries $1, \omega_N^1, \omega_N^2, \cdots, \omega_N^{N/2-1}$, and P_N is the permutation matrix that splits and sorts the even and odd indices. This idea can then be performed recursively, let

$$B_N := \begin{bmatrix} I_{N/2} & \Omega_{N/2} \\ I_{N/2} & -\Omega_{N/2} \end{bmatrix} \tag{3.12}$$

then,

$$F_N = B_N \begin{bmatrix} B_{N/2} & 0 \\ 0 & B_{N/2} \end{bmatrix} \begin{bmatrix} F_{N/4} & 0 & 0 & 0 \\ 0 & F_{N/4} & 0 & 0 \\ 0 & 0 & F_{N/4} & 0 \\ 0 & 0 & 0 & F_{N/4} \end{bmatrix} \begin{bmatrix} P_{N/2} & 0 \\ 0 & P_{N/2} \end{bmatrix} P_N$$

$$= \cdots$$

$$= \left(B_N \begin{bmatrix} B_{N/2} & 0 \\ 0 & B_{N/2} \end{bmatrix} \cdots \begin{bmatrix} B_2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & B_2 \end{bmatrix} \right) \left(\begin{bmatrix} P_2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & P_2 \end{bmatrix} \cdots \begin{bmatrix} P_{N/2} & 0 \\ 0 & P_{N/2} \end{bmatrix} P_N \right).$$

We call $B_{N/2^k}$, $k = 0, \cdots, log_2(N) - 1$ butterfly factors, each of them is a 2×2 block matrix. In the rest of this chapter, we define butterfly factors $B_{N/2^k}$ as 2×2 block matrices of size $(N/2^k) \times (N/2^k)$, where each of its block is a diagonal matrix of size $(N/2^{k+1}) \times (N/2^{k+1})$. i.e.

$$B_{N/2^k} = \left(\begin{array}{cc|cc} \times & 0 & \times & 0 \\ & \ddots & & \ddots \\ 0 & \times & 0 & \times \\ \hline \times & 0 & \times & 0 \\ & \ddots & & \ddots \\ 0 & \times & 0 & \times \end{array}\right), \quad \begin{array}{l} \text{where each block is of size } (N/2^{k+1}) \times (N/2^{k+1}), \\ \text{and each `}\times\text{' can be arbitrary number} \end{array} \tag{3.13}$$

3.2.2 Learning Dictionaries that Admit Fast Transforms

Inspired by the FFT algorithm, we propose an algorithm to learn a sparse representation and fast transform given a matrix of data points.

We assume that we have p signals $y_1, \cdots, y_p \in \mathbb{C}^N$ that can be sparse represented under some dictionary. Thus for each y_i we have

$$y_i = Dx_i,$$

where $x_i \in \mathbb{C}^N$ is the sparse representation of y_i. In matrix form, concatenating all p signals, we have

$$Y = DX$$

where $Y \in \mathbb{C}^{N \times p}$, $D \in \mathbb{C}^{N \times N}$, $X \in \mathbb{C}^{N \times p}$, i.e. each column of Y represents a given N - dimensional signal and each column of X represents the corresponding sparse representation of the signal. We will use our algorithm to find D and X such that X is sparse, and D can be applied with complexity $O(n \log n)$.

The idea is to solve the following optimization problem to obtain a sparse X, and a dictionary D that has a factorization for D such that the recovery of each y can be calculated in a lower complexity:

$$\underset{D,X}{\text{minimize}} \|Y - DX\|_F.$$

Now let

$$D = \left(B_N \begin{bmatrix} B_{N/2} & 0 \\ 0 & B_{N/2} \end{bmatrix} \cdots \begin{bmatrix} B_2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & B_2 \end{bmatrix} \right) \left(\begin{bmatrix} P_2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & P_2 \end{bmatrix} \cdots \begin{bmatrix} P_{N/2} & 0 \\ 0 & P_{N/2} \end{bmatrix} P_N \right)$$

We will solve the below optimization instead:

$$\begin{aligned} \underset{B_{N/2^k}, P_{N/2^k}, X}{\text{minimize}} \quad & \left\| Y - \left(B_N \begin{bmatrix} B_{N/2} & 0 \\ 0 & B_{N/2} \end{bmatrix} \cdots \begin{bmatrix} B_2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & B_2 \end{bmatrix} \right) \left(\begin{bmatrix} P_2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & P_2 \end{bmatrix} \cdots \begin{bmatrix} P_{N/2} & 0 \\ 0 & P_{N/2} \end{bmatrix} P_N \right) X \right\|_F \\ \text{subject to} \quad & B_{N/2^k} \text{ are butterfly factors, for } k = 0, 1, 2, \cdots, \log_2(N) - 1 \\ & P_{N/2^k} \text{ are permutation matrices, for } k = 0, 1, 2, \cdots, \log_2(N) - 1 \\ & X \text{ is sparse.} \end{aligned} \tag{3.14}$$

This problem is highly non-convex; however, if we only focus on one variable (i.e., one unknown matrix) and fix all the others, the problem becomes convex in that variable. Thus as in standard dictionary learning, we can approach it by solving a sequence of convex optimization problems. In each iteration, we will alternatively solve one of the unknown matrices from X, P_N, $P_{N/2}$, $P_{N/4}$ $\cdots$, P_2, B_2, B_4,$\cdots$,$B_{N/2}$, B_N and fix all the others.

In order to accelerate the optimization we exploit the structure in our problem. Denoting I_n as the identity matrix with size $n \times n$, we can rewrite the objective function as:

$$\begin{aligned} & \left\| Y - \left(B_N \begin{bmatrix} B_{N/2} & 0 \\ 0 & B_{N/2} \end{bmatrix} \cdots \begin{bmatrix} B_2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & B_2 \end{bmatrix} \right) \left(\begin{bmatrix} P_2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & P_2 \end{bmatrix} \cdots \begin{bmatrix} P_{N/2} & 0 \\ 0 & P_{N/2} \end{bmatrix} P_N \right) X \right\|_F \\ & = \left\| Y - \left((I_1 \otimes B_N) \left(I_2 \otimes B_{N/2} \right) \cdots \left(I_{N/2} \otimes B_2 \right) \right) \left(\left(I_{N/2} \otimes P_2 \right) \cdots \left(I_2 \otimes P_{N/2} \right) (I_1 \otimes P_N) \right) X \right\|_F . \end{aligned} \tag{3.15}$$

Here $\otimes$ represents the Kronecker product, i.e. if A is a $m \times n$ matrix, and B is a $p \times q$ matrix, $A \otimes B$ is a $pm \times qn$ matrix:

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \cdots \\ a_{m1}B & \cdots & a_{nm}B \end{bmatrix}$$

Next we introduce the details of how to solve each unknown in our algorithm.

Optimizing for the butterfly factors

First, we introduce how we solve for $B_{N/2^k}$. In this step, we are aiming to solve the following optimization problem for each $k = 1, 2, \cdots, \log_2(N) - 1$:

$$\begin{aligned} \widehat{B_{N/2^k}} = \underset{B_{N/2^k}}{\arg\min} \| Y - & \left((I_1 \otimes B_N) \left(I_2 \otimes B_{N/2} \right) \cdots \left(I_{N/2} \otimes B_2 \right) \right) \\ & \left(\left(I_{N/2} \otimes P_2 \right) \cdots \left(I_2 \otimes P_{N/2} \right) (I_1 \otimes P_N) \right) X \big\|_F \end{aligned} \tag{3.16}$$

The matrix $I_{2^k} \otimes B_{N/2^k} = \begin{bmatrix} B_{N/2^k} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & B_{N/2^k} \end{bmatrix}$ in the optimization have fixed structure, thus can be easily optimized. For each k, i.e. for each unknown $B_{N/2^k}$, we denote the fixed part "to its left" as

$$L = \left((I_1 \otimes B_N) \cdots \left(I_{2^{k-1}} \otimes B_{N/2^{k-1}} \right) \right)$$

and the fixed part "to its right" as

$$R = \left(\left(I_{2^{k+1}} \otimes B_{N/2^{k+1}} \right) \cdots \left(I_{N/2} \otimes B_2 \right) \right) \left(\left(I_{N/2} \otimes P_2 \right) \cdots \left(I_2 \otimes P_{N/2} \right) (I_1 \otimes P_N) \right) X.$$

Then, our optimization problem is

$$\begin{aligned} &\underset{B_{N/2^k}}{\text{minimize}} \|Y - L(I_{2^k} \otimes B_{N/2^k})R\|_F \\ &\text{subject to } B_{N/2^k} \text{ is a butterfly factor.} \end{aligned}$$

In turn, this can be written as a least square problem by vectorizing $B_{N/2^k}$.

Notice that

$$\begin{aligned} L(I_{2^k} \otimes B_{N/2^k})R &= L \begin{bmatrix} B_{N/2^k} & & & \\ & B_{N/2^k} & & \\ & & \ddots & \\ & & & B_{N/2^k} \end{bmatrix} R \\ &= \sum_{i=0}^{2^k-1} L_{:,i(N/2^k)+1:(i+1)(N/2^k)} B_{N/2^k} R_{i(N/2^k)+1:(i+1)(N/2^k),:} \end{aligned}$$

For a matrix $m \times n$ matrix A, $\overrightarrow{A}$ denotes the vectorization of A, i.e. the $mn \times 1$ column vector obtained by stacking the columns of the matrix A on top of one another, since we know that $\left(B^T \otimes A\right)\overrightarrow{X} = \overrightarrow{AXB}$ ([HHJ94]), we have

$$\begin{aligned} \overrightarrow{L(I_{2^k} \otimes B_{N/2^k})R} &= \sum_{i=0}^{2^k-1} \left(\left(R^T_{i(N/2^k)+1:(i+1)(N/2^k),:} \otimes L_{:,i(N/2^k)+1:(i+1)(N/2^k)} \right) \overrightarrow{B_{N/2^k}} \right) \\ &= \left(\sum_{i=0}^{2^k-1} \left(R^T_{i(N/2^k)+1:(i+1)(N/2^k),:} \otimes L_{:,i(N/2^k)+1:(i+1)(N/2^k)} \right) \right) \overrightarrow{B_{N/2^k}}. \end{aligned}$$

Our optimization can be written as

$$\overrightarrow{B_{N/2^k}}^* = \underset{\overrightarrow{B_{N/2^k}}}{\arg\min} \left\| \overrightarrow{Y} - \left(\sum_{i=0}^{2^k-1} \left(R^T_{i(N/2^k)+1:(i+1)(N/2^k),:} \otimes L_{:,i(N/2^k)+1:(i+1)(N/2^k)} \right) \right) \overrightarrow{B_{N/2^k}} \right\|_2$$
$$\text{subject to } B_{N/2^k} \text{ is a butterfly factor.} \tag{3.17}$$

Notice that the butterfly factors are defined as 2 by 2 blocks with diagonal matrix as its block (3.13). the sparsity pattern of $B_{N/2^k}$ is fixed, we only need to solve for the non-zero entries. In other words, we can only solve the system with $2 \times 2^k = 2^{k+1}$ non-zero unknowns instead of all N^2, which can easily be done. In the end, we transfer the optimal vector $\overrightarrow{B_{N/2^k}}^*$ back to matrix form according to the sparse pattern of $B_{N/2^k}$ to get $\widehat{B_{N/2^k}}$.

Optimizing for permutations – convex relaxation

In this section, we introduce our method for solving $P_{N/2^k}$ from

$$\widehat{P_{N/2^k}} = \underset{P_{N/2^k}}{\arg\min} \| Y - \left((I_1 \otimes B_N) \left(I_2 \otimes B_{N/2} \right) \cdots \left(I_{N/2} \otimes B_2 \right) \right)$$
$$\left(\left(I_{N/2} \otimes P_2 \right) \cdots \left(I_2 \otimes P_{N/2} \right) (I_1 \otimes P_N) \right) X \|_F \,.$$

Unlike the butterfly factors, the permutation factors that we want to optimize form a discrete set. In this case, for the k-th unknown factor $P_{N/2^k}$, we need to choose the best permutation that minimizes the objective function from among the set of $\frac{N}{2^k}!$ permutation matrices. Of course a brute-force works but it is too slow. Instead we provide a convex relaxation to solve for the permutation matrix.

A nonnegative matrix A of size $n \times n$, $A = [a]_{ij}$ is called a doubly stochastic matrices if $\sum_{i=1}^{n} a_{ij} = 1$ and $\sum_{j=1}^{n} a_{ij} = 1$. The set of doubly stochastic matrices of size $n \times n$ is the convex hull of all permutation matrices of size $n \times n$ ([Bru06]). Thus, for the permutation matrix, we can use a convex relaxation:

$$\begin{aligned}\underset{P_{N/2^k}}{\text{minimize}} \quad & \left\| Y - L(I_{2^k} \otimes P_{N/2^k})R \right\|_F \\ \text{subject to} \quad & \mathbb{1}^T P_{N/2^k} = \mathbb{1}^T \\ & \mathbb{1}^T P_{N/2^k}^T = \mathbb{1}^T \\ & \mathbf{0} \le P_{N/2^k} \le \mathbf{1},\end{aligned}$$

where $\mathbb{1}$ denotes the vector of all ones, and $\mathbf{0}, \mathbf{1}$ denotes of matrix of all zeros and all ones respectively. Using the same trick we used for butterfly factors, we can also rewrite this optimization problem into:

$$\begin{aligned}\overrightarrow{P_{N/2^k}}^* = \underset{\overrightarrow{P_{N/2^k}}}{\arg\min} \quad & \left\| \overrightarrow{Y} - \left(\sum_{i=0}^{2^k-1} \left(R^T_{i(N/2^k)+1:(i+1)(N/2^k),:} \otimes L_{:,i(N/2^k)+1:(i+1)(N/2^k)} \right) \right) \overrightarrow{P_{N/2^k}} \right\|_2 \\ \text{subject to} \quad & \mathbb{1}^T P_{N/2^k} = \mathbb{1}^T \\ & \mathbb{1}^T P_{N/2^k}^T = \mathbb{1}^T \\ & \mathbf{0} \le P_{N/2^k} \le \mathbf{1}.\end{aligned} \tag{3.18}$$

We can then solve this problem as a linear constraint least-square problem, and then reshape $\overrightarrow{P_{N/2^k}}^*$ to matrix $\widehat{P_{N/2^k}}$. Of course, since this is only a convex relaxation, there will be instances when the optimal $\overrightarrow{P_{N/2^k}}^*$ is not the vectorization of a permutation matrix. In these cases, we round the solution $\widehat{P_{N/2^k}}$ by making the largest element for each column to be 1 and the rest of the elements to be 0.

Solving for X

When solving X, we use ℓ_1 norm regularization for each x_i, to promote sparsity. For each iteration, we first calculate $\widehat{D}$, our current estimate of the dictionary using the updated B's and

P's, then solve

$$x_i^* = \arg\min_{x_i} \|x_i\|_{\ell_1} \text{ subject to } \|\widehat{D}x_i - y_i\|_2 \le \sigma_i. \tag{3.19}$$

for $i = 1, 2, 3, \cdots, p$ with some hyper-parameters σ_i.

After solving for each x_i, we concatenate them into $\widehat{X}$.

Let $\widehat{D}^{(l)}$ and $\widehat{X}^{(l)}$ denote the updated dictionary and the sparse representations in l-th iteration. We define the error after each iteration as

$$E^{(l)} = \|Y - \widehat{D}^{(l)}\widehat{X}^{(l)}\|_F \tag{3.20}$$

In l-th iteration, we choose $\sigma_i = \alpha^l \|y_i\|_2$, $0 < \alpha < 1$. Here α can be chosen to control the trade off between the speed of error decreasing and the sparsity of $\widehat{X}$, as the closer α is to 1, the slower the error $E^{(l)}$ decrease and the more sparse result we can get.

The whole algorithm can be described as below:

Algorithm 2: Dictionary Learning with fast transform

```
Input: $Y$
Output: $B_{N/2^k}, P_{N/2^k}, X$ for $k = 0, 1, 2, \cdots, \log_2(N) - 1$
Random initial $B_{N/2^k}, P_{N/2^k}$ for $k = 0, 1, 2, \cdots, \log_2(N) - 1$
Optimizing over $X$ by solving (3.19) for each $i$ and concatenating the results.
while error E defined in (3.20) is not small enough do
    for $k = 0, 1, 2, \cdots, \log_2(N) - 1$ do
        Fix $X$, all $B_{N/2^k}$'s and $P_{N/2^{log_2 N-1}}, \cdots, P_{N/2^{k+1}}, P_{N/2^{k-1}}, \cdots, P_{N/2^0}$. Optimizing
          over $P_{N/2^k}$ by solving (3.18) and then reshape the result into permutation
          matrix.
    for $k = \log_2(N) - 1, \cdots, 2, 1, 0$ do
        Fix $X$, all $P_{N/2^k}$'s and $B_{N/2^{log_2 N-1}}, \cdots, B_{N/2^{k+1}}, B_{N/2^{k-1}}, \cdots, B_{N/2^0}$.
          Optimizing over $B_{N/2^k}$ by solving (3.17) and then transfer the result into
          butterfly factor.
        Normalize $B_{N/2^k}$ into matrix with unit $\ell_2$ norm columns.
    Optimizing over $X$ by solving (3.19) for each $i$ and concatenating the results.
```

3.3 Numerical Experiment

3.3.1 Datasets

In this section we illustrate our algorithm – on a stylized example – by applying it to real image data obtained from the CIFAR-10 dataset [KNH]. The CIFAR-10 dataset consists of 60000 32x32 color images. We uniformly sample 1000 of them as training data for our algorithm. We merge the RGB channels into one channel by forming a weighted sum of the R, G, and B components using the “rgb2gray” function in matlab[”Ma] , and normalize the image so the pixels' range is in $[0,1]$.

3.3.2 Implementation details

The natural images are approximately sparse under the 2D Discrete Fourier Transform (DFT) [Lim90], i.e. for each image Y_i, we have

$$X_i = \text{2D-FFT}(Y_i) = DY_iD^T \iff \overrightarrow{X_i} = (D \otimes D)\overrightarrow{Y_i},$$

where D is the transformation matrix of Discrete Fourier Transform, and X_i is the sparse representation matrix of the image Y_i under the 2D-DFT. Our task is to learn, from the images, a sparsifying transform that hopefully competes with the DFT in terms of the sparsity of the resulting coefficients. Note that the 2-D DFT is a separable transformation, i.e., it can be implemented by first applying the 1-D DFT to the columns of Y and then applying it again to the rows of the result. We will also assume that the sparsifying transform that we learn is separable, however we allow the factors to be different. Thus we will approximately minimize the objective $\sum_i \|Y_i - D_1^{-1} X_i D_2^{T^{-1}}\|_F^2$.

Since the inverse of the Discrete Fourier Transform has the same Fast Transform structure as the DFT itself, but with the opposite sign in the exponent and a $1/N$ factor[AH05],[Wik], any FFT algorithm can easily be adapted for it. We use the same factorization that we introduced before.

Denote $B_{(N/2^k);1}$ and $P_{(N/2^k);1}$ as the butterfly factors and the permutation matrices for D_1^{-1}, we write

$$D_1^{-1} = \left((I_1 \otimes B_{N;1}) \cdots \left(I_{N/2} \otimes B_{2;1}\right)\right)\left(\left(I_{N/2} \otimes P_{2;1}\right) \cdots (I_1 \otimes P_{N;1})\right), \tag{3.21}$$

The factorization for D_2^{-1} has the same form , but with possibly different values for the butterfly factors and permutation matrices, which we denote as $B_{(N/2^k);2}$ and $P_{(N/2^k);2}$,

$$D_2^{-1} = \left((I_1 \otimes B_{N;2}) \cdots \left(I_{N/2} \otimes B_{2;2}\right)\right) \left(\left(I_{N/2} \otimes P_{2;2}\right) \cdots (I_1 \otimes P_{N;2})\right). \tag{3.22}$$

Note that for any nonsingular matrix $(D^T)^{-1} = (D^{-1})^T$, we have

$$\begin{aligned} Y_i = &\left((I_1 \otimes B_{N;1}) \cdots \left(I_{N/2} \otimes B_{2;1}\right)\right) \left(\left(I_{N/2} \otimes P_{2;1}\right) \cdots (I_1 \otimes P_{N;1})\right) \cdot X_i \\ &\cdot \left(\left((I_1 \otimes B_{N;2}) \cdots \left(I_{N/2} \otimes B_{2;2}\right)\right) \left(\left(I_{N/2} \otimes P_{2;2}\right) \cdots (I_1 \otimes P_{N;2})\right)\right)^T. \end{aligned}$$

When solving $B_{(N/2^k);1}$ and $P_{(N/2^k);1}$, D_2^{-1} (i.e. $B_{(N/2^k);2}$ and $P_{(N/2^k);2}$ for each k) and X_i are treated as known matrices. We can first calculate

$$X_i \cdot \left(\left((I_1 \otimes B_{N;2}) \cdots \left(I_{N/2} \otimes B_{2;2}\right)\right) \left(\left(I_{N/2} \otimes P_{2;2}\right) \cdots (I_1 \otimes P_{N;2})\right)\right)^T, \tag{3.23}$$

and denote it as $Z_{i;2}$ for i-th training example (since it is calculated by fixing D_2^{-1}). Then each image Y_i can be written as:

$$Y_i = \left((I_1 \otimes B_{N;1}) \cdots \left(I_{N/2} \otimes B_{2;1}\right)\right) \left(\left(I_{N/2} \otimes P_{2;1}\right) \cdots (I_1 \otimes P_{N;1})\right) \cdot Z_{i;2}.$$

Concatenate Y_i's and $Z_{i;2}$'s by stacking the matrices on right of one another to get Y_{concat} and $Z_{concat;2}$ respectively. Now we have the following problem with the same form of the highly non-convex optimization (3.14) that we introduced at the beginning of Section 3.2.2:

$$
\begin{aligned}
\underset{X,B_{(N/2^k);1},P_{(N/2^k);1}}{minimize} \quad & \|Y_{concat} - \big((I_1 \otimes B_{N;1}) \cdots \big(I_{N/2} \otimes B_{2;1}\big)\big) \\
& \qquad \big(\big(I_{N/2} \otimes P_{2;1}\big) \cdots (I_1 \otimes P_{N;1})\big) \cdot Z_{concat;2}\|_F, \qquad (3.24) \\
\text{subject to} \quad & P_{(N/2^k);1} \text{ are permutation matrices, for } k = 0,1,2,\cdots,\log_2(N)-1 \\
& B_{(N/2^k);1} \text{ are butterfly factors, for } k = 0,1,2,\cdots,\log_2(N)-1.
\end{aligned}
$$

For each unknown $B_{N/2^k;1}$, we denote the fixed part "to its left" as

$$L = \Big((I_1 \otimes B_{N;1}) \cdots \Big(I_{2^{k-1}} \otimes B_{(N/2^{k-1});1}\Big)\Big)$$

and the fixed part "to its right" as

$$R = \Big(\Big(I_{2^{k+1}} \otimes B_{(N/2^{k+1});1}\Big) \cdots \big(I_{N/2} \otimes B_{2;1}\big)\Big)\,\big(\big(I_{N/2} \otimes P_{2;1}\big) \cdots (I_1 \otimes P_{N;1})\big)\, Z_{concat;2}.$$

Let

$$A = \left(\sum_{i=0}^{2^k-1} \Big(R^T_{i(N/2^k)+1:(i+1)(N/2^k),:} \otimes L_{:,i(N/2^k)+1:(i+1)(N/2^k)} \Big) \right),$$

we solve the following optimization

$$
\begin{aligned}
\overrightarrow{B_{(N/2^k);1}}^* = & \underset{\overrightarrow{B_{(N/2^k);1}}}{\arg\min} \left\| \overrightarrow{Y_{concat}} - A\overrightarrow{B_{(N/2^k);1}} \right\|_2 \qquad (3.25) \\
& \text{subject to } B_{N/2^k} \text{ is a butterfly factor.}
\end{aligned}
$$

restricted on non-zero elements of $\overrightarrow{B_{(N/2^k);1}}$, and then transfer the optimal result back to matrix.

For each unknown $P_{N/2^k;1}$, we denote the fixed part "to its left" as

$$L = ((I_1 \otimes B_{N;1}) \cdots (I_{2^{k-1}} \otimes B_{2;1}))\,\Big(\big(I_{N/2} \otimes P_{2;1}\big) \cdots \Big(I_1 \otimes P_{(N/2^{k-1});1}\Big)\Big)$$

and the fixed part “to its right” as

$$R = \left(\left(I_{(N/2^{k+1});1} \otimes P_{2;1}\right) \cdots \left(I_1 \otimes P_{N;1}\right)\right) Z_{concat;2},$$

and let

$$A = \left(\sum_{i=0}^{2^k-1} \left(R^T_{i(N/2^k)+1:(i+1)(N/2^k),:} \otimes L_{:,i(N/2^k)+1:(i+1)(N/2^k)} \right) \right)$$

then solve the following convex relaxation:

$$\begin{aligned} \overrightarrow{P_{(N/2^k);1}}^{*} = \underset{\overrightarrow{P_{(N/2^k);1}}}{\arg\min} \quad & \left\| \overrightarrow{Y_{concat}} - A\overrightarrow{P_{(N/2^k);1}} \right\|_2 \\ \text{subject to} \quad & \mathbb{1}^T P_{(N/2^k);1} = \mathbb{1}^T \\ & \mathbb{1}^T P^T_{(N/2^k);1} = \mathbb{1}^T \\ & \mathbf{0} \le P_{(N/2^k);1} \le \mathbf{1}. \end{aligned} \tag{3.26}$$

After we update all $B_{(N/2^k);1}$’s and $P_{(N/2^k);1}$’s, we then fix them and calculate

$$\left(\left(I_1 \otimes B_{N;1}\right) \cdots \left(I_{N/2} \otimes B_{2;1}\right)\right) \left(\left(I_{N/2} \otimes P_{2;1}\right) \cdots \left(I_1 \otimes P_{N;1}\right)\right) \cdot X_i, \tag{3.27}$$

for each X_i, and denote it as $Z_{i;1}$. For each example, we have

$$Y_i = Z_{i;1} \cdot \left(\left(\left(I_1 \otimes B_{N;2}\right) \cdots \left(I_{N/2} \otimes B_{2;2}\right)\right) \left(\left(I_{N/2} \otimes P_{2;2}\right) \cdots \left(I_1 \otimes P_{N;2}\right)\right)\right)^T. \tag{3.28}$$

Transpose the system

$$Y_i^T = \left(\left(\left(I_1 \otimes B_{N;2}\right) \cdots \left(I_{N/2} \otimes B_{2;2}\right)\right) \left(\left(I_{N/2} \otimes P_{2;2}\right) \cdots \left(I_1 \otimes P_{N;2}\right)\right)\right) \cdot Z^T_{i;1}. \tag{3.29}$$

and concatenate Y_i^T and $Z^T_{i;1}$, the optimization can again be rewritten into the same form as (3.14). $B_{(N/2^k);2}$’s and $P_{(N/2^k);2}$’s can then be solved sequentially.

After solving all the $B_{(N/2^k);1}$'s, $P_{(N/2^k);1}$'s, $B_{(N/2^k);2}$'s and $P_{(N/2^k);2}$'s, we can calculate D_1^{-1} and D_2^{-1} respectively. Then X can be updated by solving

$$\overrightarrow{X_i}^* = \arg\min_{\overrightarrow{X_i}} \left\|\overrightarrow{X_i}\right\|_{\ell_1} \text{ subject to } \left\|\left(D_2^{-1} \otimes D_1^{-1}\right)\overrightarrow{X_i} - \overrightarrow{y_i}\right\|_2 \leq \sigma_i. \tag{3.30}$$

The algorithm for image data can be find in (Alg 3). In our experiment, we use "lsqlin" function [Mat] built in MATLAB to solve the convex relaxation of permutation problem (3.18). And the spgl1 [vdBF19] to solve X from (3.19), with $\sigma_i = \alpha^l \|y_i\|_2$, $\alpha = 0.99$.

3.3.3 Numerical Sparsity

We use ℓ_0 norm of x, number of nonzero elements in x, to represent the sparsity of a signal. Although the ℓ_0 norm plays significant role in many aspect of theoretical signal processing analysis, practically it has clear disadvantage when the signal is approximately sparse instead of strictly sparse i.e. when there are many coefficients close to 0 but not exactly equal to 0. In this case the function $\|x\|_0$ is no longer a very meaningful description of the number of significant coefficients (the estimate of the sparsity) of the signal. Instead, we use numerical sparsity, the ratio between square of the ℓ_1 and ℓ_2 norm of the signal, to estimate the sparsity of a given signal x:

$$s(x) = \frac{\|x\|_1^2}{\|x\|_2^2}. \tag{3.31}$$

$s(x)$ always satisfies $1 \leq s(x) \leq \|x\|_0$, and it is a sharp lower bound of $\|x\|_0$ for any nonzero x [Lop13]. For example, if a vector $x \in \mathbb{R}^N$ has only s large coefficients and $N - s$ small coefficients, $\|x\|_{\ell_0} = N$ but $s(x)$ is close to s, which reflects the real "sparsity" i.e. number of significant coefficients.

Algorithm 3: Dictionary Learning with fast transform for images

```
Input: images $Y_i$, $i = 1, 2, \cdots, p$
Output: $B_{(N/2^k);1}$, $P_{(N/2^k);1}$, $B_{(N/2^k);2}$, $P_{(N/2^k);2}$ and $X$ for $k = 0, 1, 2, \cdots, \log_2(N) - 1$
Random initial $B_{(N/2^k);1}$, $P_{(N/2^k);1}$, $B_{(N/2^k);2}$, $P_{(N/2^k);2}$ for $k = 0, 1, 2, \cdots, \log_2(N) - 1$
Find the sparse representation $X_i$ by solving (3.30) for each $i$
while error is not small enough do
    Calculate $D_2^{-1}$ using $B_{(N/2^k);2}$ and $P_{(N/2^k);2}$ according to equation (3.22)
    Calculate $Z_{i;2} = X_i D_2^{-1^T}$ for each $i$
    Concatenate $Y_i$ into $Y_{concat}$, $Z_{i;2}$ into $Z_{concat;2}$ by stacking the matrices on the right
      of one another.
    for $k = 0, 1, 2, \cdots, \log_2(N) - 1$ do
        Fix all $B_{(N/2^k);1}$'s and $P_{(N/2^{log_2 N-1});1}, \cdots, P_{(N/2^{k+1});1}, P_{(N/2^{k-1});1}, \cdots, P_{(N/2^0);1}$.
          Optimizing over $P_{(N/2^k);1}$ by solving (3.26) and then reshape the result into
          permutation matrix.
    for $k = \log_2(N) - 1, \cdots, 2, 1, 0$ do
        Fix all $P_{(N/2^k);1}$'s and $B_{(N/2^{log_2 N-1});1}, \cdots, B_{(N/2^{k+1});1}, B_{(N/2^{k-1});1}, \cdots, B_{(N/2^0);1}$.
          Optimizing over $B_{(N/2^k);1}$ by solving (3.25) and then transfer the result into
          butterfly factor.
        Normalize $B_{(N/2^k);1}$ into matrix with unit $\ell_2$ norm columns.
    Calculated $D_1 - 1$ using $B_{(N/2^k);1}$, $P_{(N/2^k);1}$ according to equation (3.21)
    Calculate $Z_{i;1} = D_2^{-1} X_i$ for each $i$
    Concatenate $Y_i^T$ into $Y_{concat}$, $Z_{i;2}^T$ into $Z_{concat;1}$ by stacking the matrices on right of
      one another.
    for $k = 0, 1, 2, \cdots, \log_2(N) - 1$ do
        Fix all $B_{(N/2^k);2}$'s and $P_{(N/2^{log_2 N-1});2}, \cdots, P_{(N/2^{k+1});2}, P_{(N/2^{k-1});2}, \cdots, P_{(N/2^0);2}$.
          Optimizing over $P_{(N/2^k);2}$ by solving the convex relaxation similar as (3.26)
          and then reshape the result into permutation matrix.
    for $k = \log_2(N) - 1, \cdots, 2, 1, 0$ do
        Fix all $P_{(N/2^k);2}$'s and $B_{(N/2^{log_2 N-1});2}, \cdots, B_{(N/2^{k+1});2}, B_{(N/2^{k-1});2}, \cdots, B_{(N/2^0);2}$.
          Optimizing over $B_{(N/2^k);2}$ by solving the optimization similar as (3.25) and
          then transfer the result into butterfly factor.
        Normalize $B_{(N/2^k);1}$ into matrix with unit $\ell_2$ norm columns.
    Calcuate $D_2^{-1}$ using $B_{(N/2^k);2}$, $P_{(N/2^k);2}$
    Find the sparse representation $X_i$ by solving (3.30) for each $i$.
```

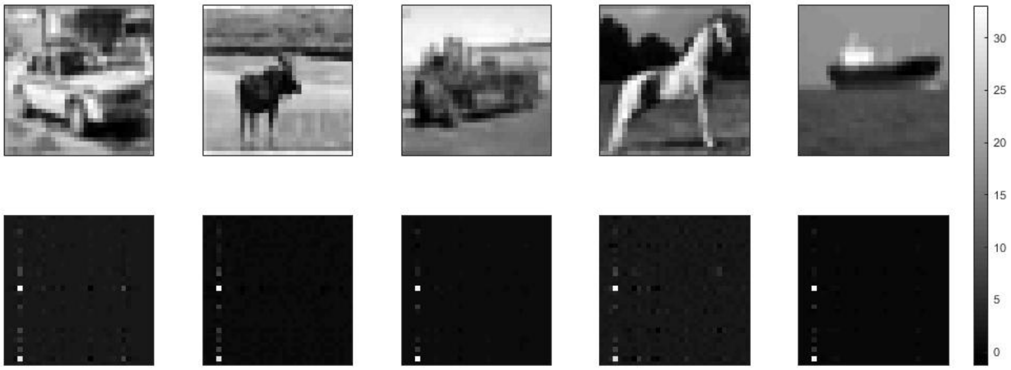


Figure 3.1: Examples of approximately sparse representation

3.3.4 Results

Denote the learned approximately sparse representation for each training image as $\widehat{X_i}$, the learned butterfly factors as $\widehat{B_{(N/2^k);1}}$, $\widehat{B_{(N/2^k);2}}$ and the learned permutation matrix as $\widehat{P_{(N/2^k);1}}$, $\widehat{P_{(N/2^k);2}}$. To recover the image from $\widehat{X_i}$, We simply do the calculation

$$\begin{aligned}\widehat{Y_i} = &\left(I_1 \otimes \widehat{B_{N;1}}\right)\left(I_2 \otimes \widehat{B_{(N/2);1}}\right)\cdots\left(I_{N/2} \otimes \widehat{B_{2;1}}\right)\left(I_{N/2} \otimes \widehat{P_{2;1}}\right)\cdots\left(I_2 \otimes \widehat{P_{(N/2);1}}\right)\left(I_1 \otimes \widehat{P_{N;1}}\right)\cdot\widehat{X_i}\\ &\cdot\left(\left(I_1 \otimes \widehat{B_{N;2}}\right)\left(I_2 \otimes \widehat{B_{(N/2);2}}\right)\cdots\left(I_{N/2} \otimes \widehat{B_{2;2}}\right)\left(I_{N/2} \otimes \widehat{P_{2;2}}\right)\cdots\left(I_2 \otimes \widehat{P_{(N/2);2}}\right)\left(I_1 \otimes \widehat{P_{N;2}}\right)\right)^T\end{aligned}$$

In our result, the average numerical sparsity of the learned representation over the training data is 30.5143. We post-process the learned sparse representations $\widehat{X_i}$ by keeping the largest 120 (approximately 4 times numerical sparsity for each image) coefficients of $\widehat{X_i}$ to make the representation sparse, and denote the post-processed sparse representation as $\widehat{X_{i;post}}$.

The experiments shows that our algorithm can successfully find a fast factorization of the dictionary as well as the "sparse" representation of the images under the dictionary. Figure 3.1 shows several examples of the approximated sparse representation of our training data. We can see from the plot that, in the whole 1024 coefficients of the images, only several of them are significant large in the representations that we found.

Figure 3.2 shows examples of the recovery from sparse representation of the training

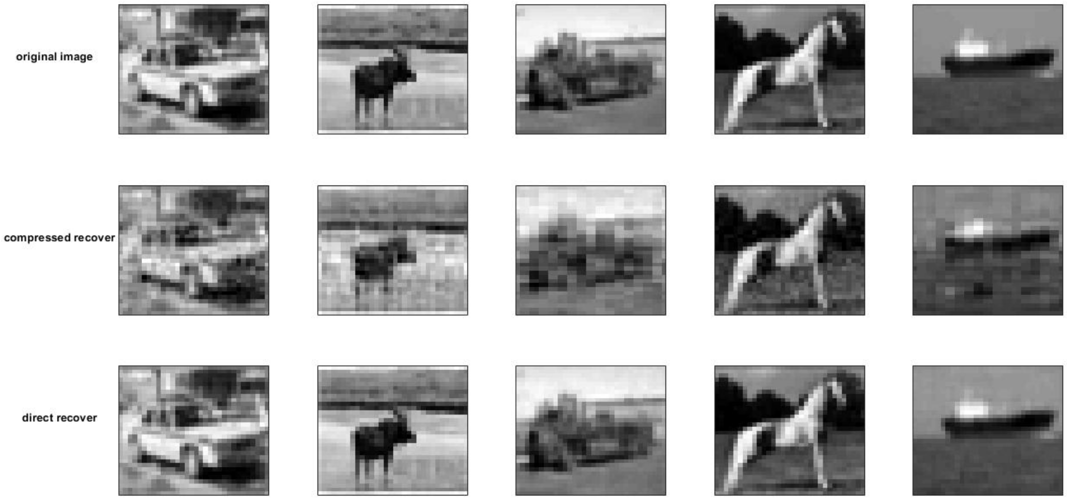


Figure 3.2: Recovered images of examples from training images

data. For each example, we plot the original image, the image recovered from the post-processed representation $\widehat{X}_{i;post}$ and the recovered image directly from the learned representation $\widehat{X}_i$. For each column, the 1st row prints the original image Y_i, the 2nd row shows the recovery image from the post-processed sparse representation $\widehat{X}_{i;post}$, and the last row shows the recovery image from the learned representation $\widehat{X}_i$.

Furthermore, to test if our learned dictionary can be applied to unknown examples to sparsify them, we tried our learned dictionary on test images (randomly selected from CIFIR10 dataset excepting the 1000 training images). We apply

$$\widehat{X}_{i;test} = \left(\left(I_1 \otimes \widehat{B_{N;1}}\right)\cdots\left(I_{N/2}\otimes\widehat{B_{2;1}}\right)\left(I_{N/2}\otimes\widehat{P_{2;1}}\right)\cdots\left(I_2\otimes\widehat{P_{(N/2);1}}\right)\left(I_1\otimes\widehat{P_{N;1}}\right)\right)^{-1}\cdot Y_{i;test}$$
$$\cdot\left(\left(\left(I_1\otimes\widehat{B_{N;2}}\right)\cdots\left(I_{N/2}\otimes\widehat{B_{2;2}}\right)\left(I_{N/2}\otimes\widehat{P_{2;2}}\right)\cdots\left(I_2\otimes\widehat{P_{(N/2);2}}\right)\left(I_1\otimes\widehat{P_{N;2}}\right)\right)^{-1}\right)^T$$

on each testing image $Y_{i;test}$ to get the approximately sparse representation $\widehat{X}_{i;test}$. We also post-process $\widehat{X}_{i;test}$ in the same way of trained representation to get $\widehat{X}_{i;test;post}$. Figure 3.3 shows the result of applying the dictionary on testing data. Figure 3.3 shows recovered images of examples

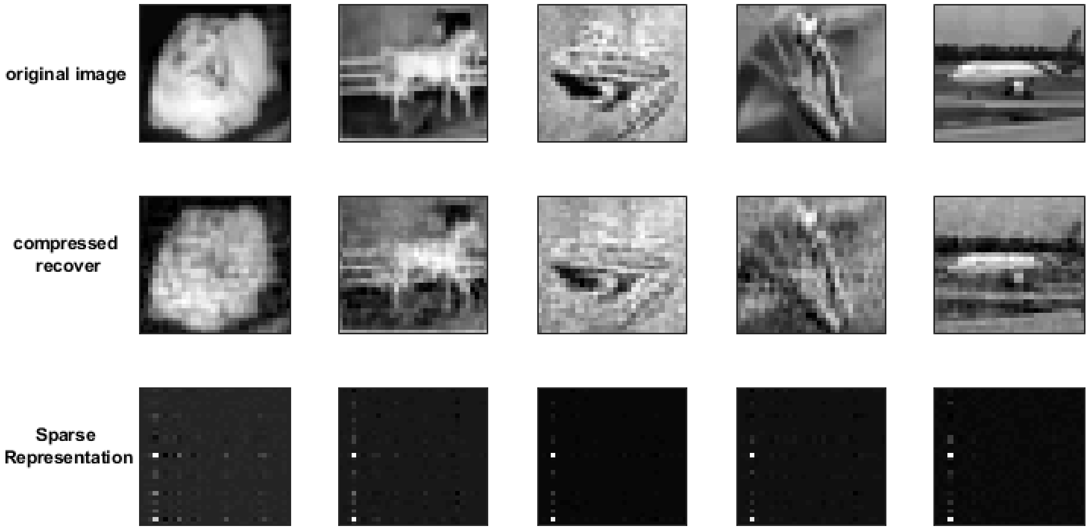


Figure 3.3: Recovered images of examples from testing images

from testing data. For each column, the 1st row prints the original image $Y_{i;test}$, the 2nd row shows the recovery image from the post-processed compressed sparse representation $\widehat{X}_{i;test;post}$, and the last row shows the corresponding sparse representation by directly applying D_1 and D_2 that we learned to the testing data. The average numerical sparsity over the 100 random selected testing data is 34.5797, the dictionary generalized well to unknown examples.

We compared the numerical sparsity of the trained sparse representation $\widehat{X}_i$ from our algorithm with it of the 2-D DFT and 2-D DCT transformation. Figure 3.4 and Figure 3.5 shows the comparison of sparsity. For each dots represents an training example, its x coordinate represents the numerical sparsity of its 2-D DFT or 2-D DCT coefficients, respectly, where as the y coordinates represents the numerical sparsity of its learned sparse representation from our algorithm. Since for both polots, the “dots” are mostly below that line $y = x$, it means that our results is better. Our learned dictionary outperforms the coefficients of 2D-DFT transform and is slightly better than 2D-DCT result.

Figure 3.6 shows the learned dictionaries obtained from our algorithm. Combined with

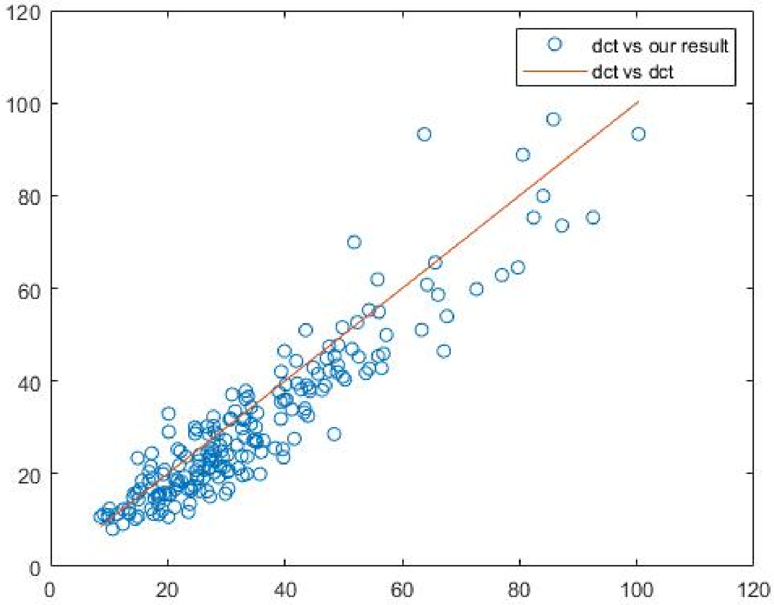


Figure 3.4: Comparison of sparsities of our result with it of 2D-DCT

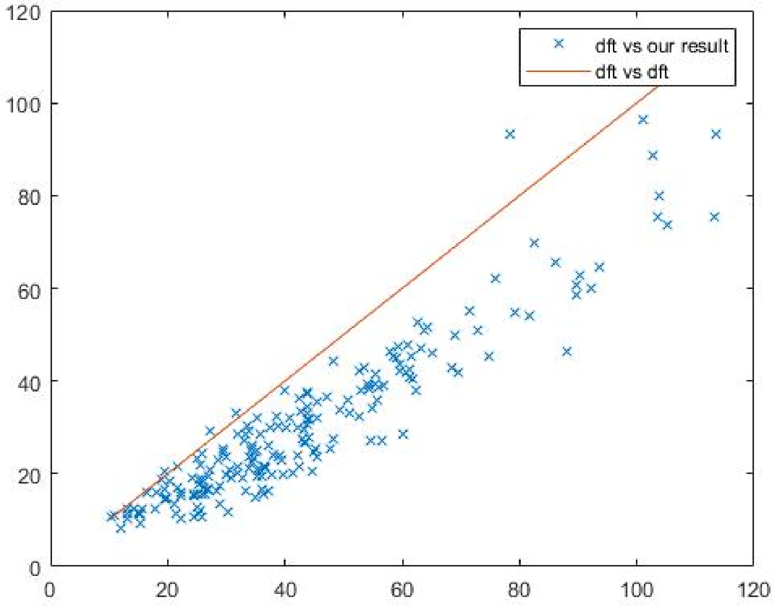


Figure 3.5: Comparison of sparsities of our result with it of 2D-DFT

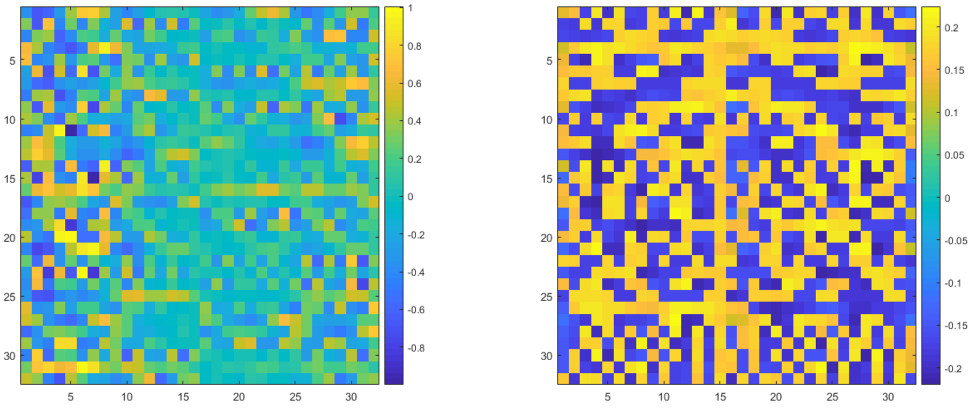


Figure 3.6: $\widehat{D_1}$ and $\widehat{D_2}$

figure 3.7, which shows the columns of $\widehat{D_2}$, we can see that the some columns of our learned dictionary is acting somewhat like sin/cosine basis with different amplitude and period, some of them are acting like combinations of wavelet basis, and the dictionaries has some clear pattern especially D_2 who looks like a DCT matrix.

3.4 Conclusion

We proposed an algorithm which can learn the fast factorization of a linear dictionary as well as the approximate sparse representation from the given training data. In our proposed algorithm, we learned a $N \times N$ linear transformation matrix use $O(N)$ degree of freedom. Besides, the output of the algorithm forms a factorization of the dictionary which can obtain fast recovery by $O(N \log N)$ calculations instead of $O(N^2)$. We leave it to future work to show that the sequence of optimization problems leads to a stationary point of the original highly non-convex optimization.

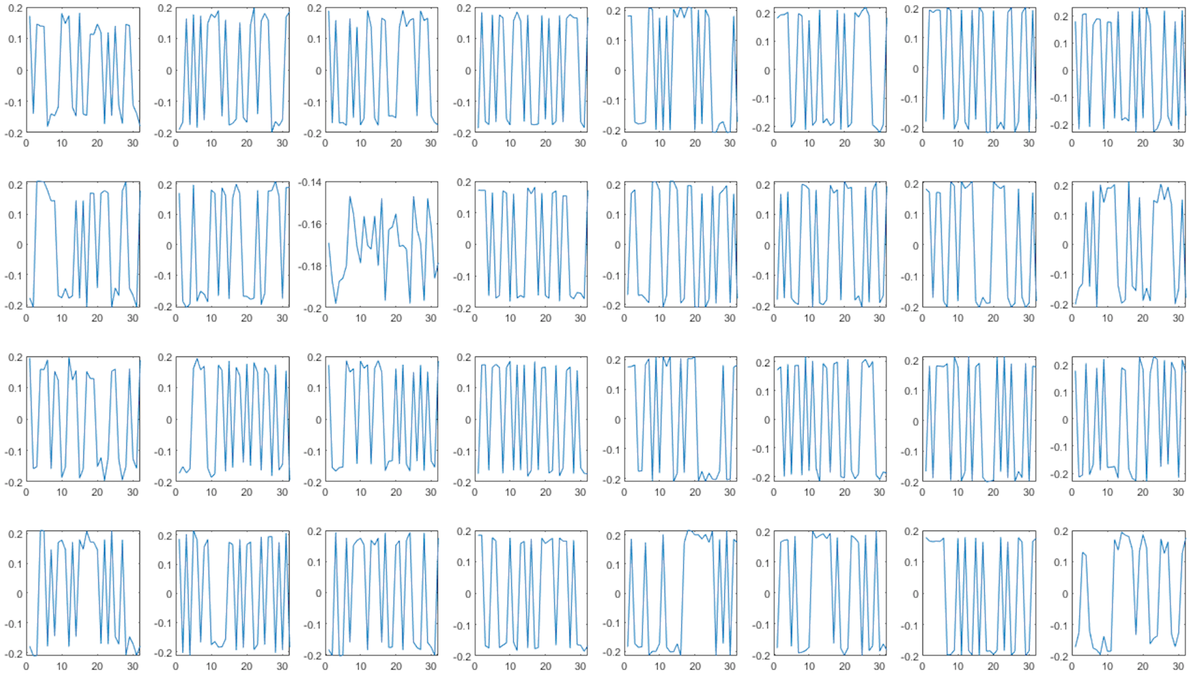


Figure 3.7: Columns of $\widehat{D_2}$ (only show every 4 of them)

3.5 Acknowledgements

Chapter 3, in full is currently being prepared for submission for publication of the material, co-authored with Saab, Rayan. The dissertation author was the primary investigator and author of this material.

Chapter 4

Conditional Generative Latent Optimization

4.1 Introduction

We introduced traditional dictionary and representation learning in previous chapters. Dictionary learning shows promising results in reconstruction since the basis is learned adaptively from a data-set of training signals that are representative of the class of signals of interest. The dictionaries we introduced above are linear transformations which can be interpreted as single layer neural networks (also known as shallow neural networks). We learn the dictionary and the sparse representation by assuming that

$$x = Dz$$

But in some applications, it is not enough to just have the linear dictionaries. For example, in image generation, even though we know that natural images are approximately sparse under wavelet basis, when you feed a random sparse vector to a wavelet basis, it is not necessary that you get a “natural image”. Thus we may assume that images have more complex structure. In recent years, deep learning based techniques have replaced shallow networks for representation learning.

Nevertheless, the fundamental idea is the same. One assumes that the signals encountered in some application follow a similar distribution, therefore the model learned from training data can be applied to unknown examples. In this chapter, motivated by a specific application, we will propose a deep generative model using representation learning technique.

4.1.1 Generative model

Generative models learn a target distribution that generates a class of data. For example, [RMC15] uses a generative model to learn the distribution of facial images and use the learned model to draw new samples to generate fake facial images. Generative Adversarial Nets (GANs) [GPAM$^+$14] is one of the major breakthroughs in the area of generative modeling in the machine learning community.

In generative adversarial nets, two neural networks are learned, the generator G and the discriminator D. The generator G can be modeled as a differentiable function that takes random input z from a latent space Z following a distribution $p_z(z)$ and outputs data x that should follow (after training) the targeted probability distribution $p_{data}(x)$:

$$G : Z \to \mathbb{R}^n,$$

where Z is the latent space and n is the dimensionality of the data space. Its adversary, the discriminator network D, is a simple classifier neural network that takes an input data x that can be a “real” one drawn from the data set or a “fake” one whose density is induced by $p_z(z)$ going through the generator G, and that returns a probability $D(x)$ indicating the likelihood of x being “real” data:

$$D : \mathbb{R}^n \to [0,1]. \tag{4.1}$$

Learning the generative adversarial networks model can be regarded as a zero-sum game in which the generator and adversary networks must compete against each other. We use $V(D,G)$ to model

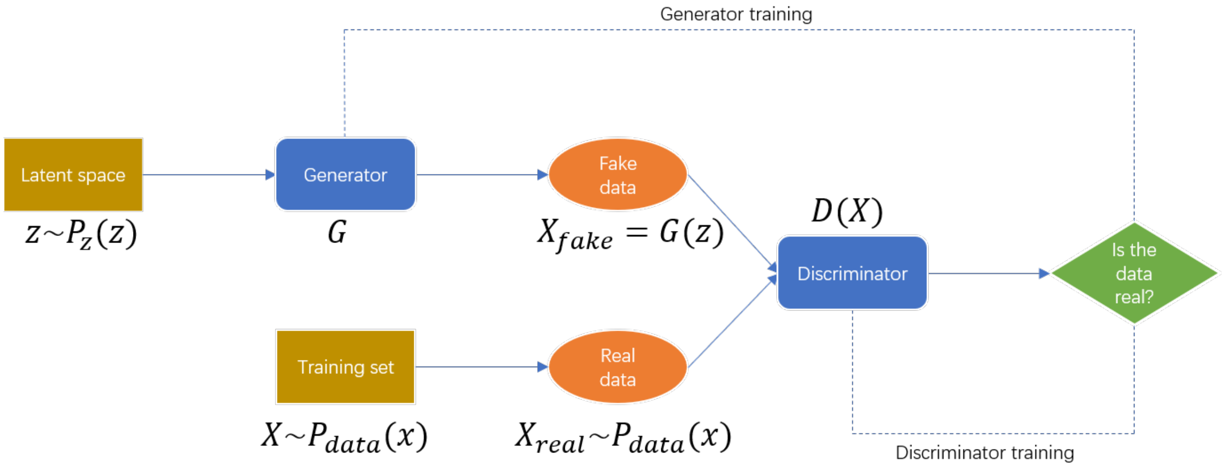


Figure 4.1: Generative Adversarial Nets architecture

the probability of assigning the correct label to both real training examples and samples generated from G:

$$V(D,G) = \mathbb{E}_{x \sim p_{data}(x)}[\log(D(x))] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

The goal of the discriminator is to detect fake generated data, so the discriminative neural network is trained to maximize the probability. Conversely, the goal of the generator is to fool the discriminator, so the generative neural network is trained to minimize $V(D,G)$. So that at convergence,

$$G^* = \arg\min_G \max_D V(D,G)$$

Ideally, the generated samples are indistinguishable from the real data, and the discriminator outputs $\frac{1}{2}$ everywhere. The discriminator may then be discarded.

Generative adversarial nets can be extended to a conditional model if both the generator and discriminator are conditioned on some extra information y. y could be any kind of auxiliary information, such as class labels or data from other modalities. The model is then trained to sample from a conditional distribution $p(x \mid y)$ rather than simply sampling from a marginal distribution $p(x)$. [MO14] proposes conditional generative adversarial nets by combining the prior input noise $z \sim p_z(z)$ with the condition y into a joint latent representation, and then inputing

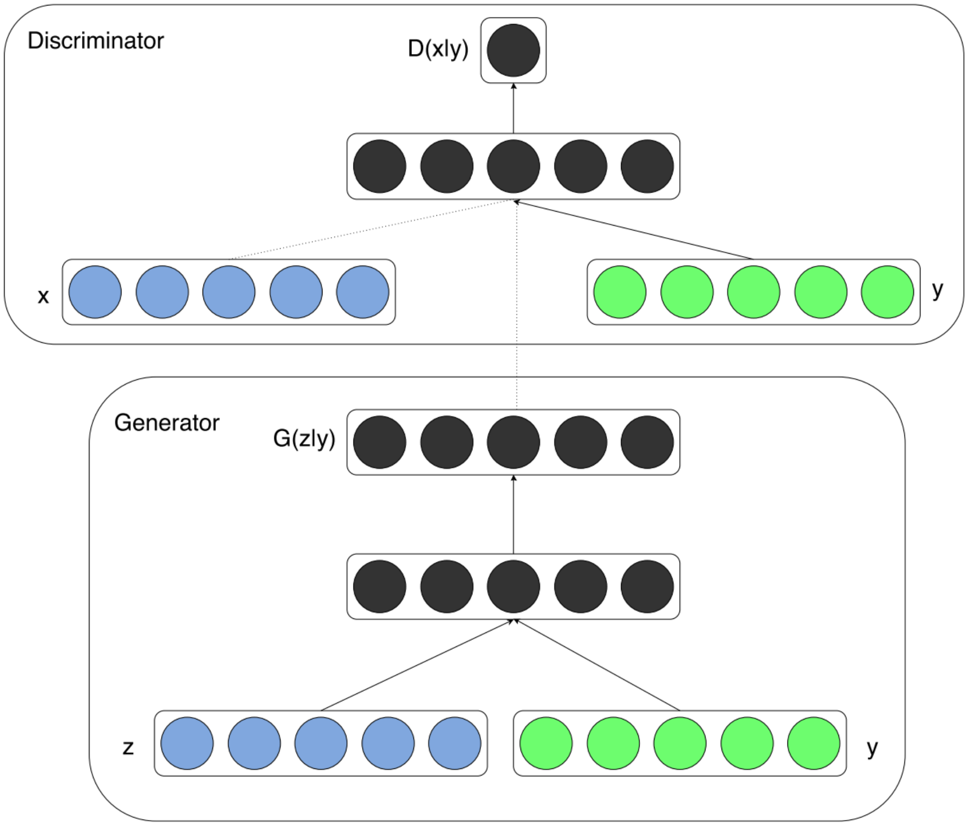


Figure 4.2: Conditional adversarial net [MO14]

y together with the data to the discriminator and do the following optimization:

$$\min_{G}\max_{D} V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log(D(x\,|\,y))] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z\,|\,y)))]$$

Fig 4.2 illustrates the structure of a simple conditional adversarial net.

While the generative adversarial net learns the distribution of data by applying the adversarial game between the generator and the discriminator, Generative Latent Optimization (GLO) [BJLPS18] maps one learnable noise vector to each of the data in the training set by minimizing a simple reconstruction loss:

$$\min_{\omega,z} \ell(x, G_\omega(z)),$$

where ω represents the parameters of the generator G and $\ell(x,x')$ is some deterministic loss function. Thus the model can also be interpreted as a representation learning model, where the latent representation for each training sample and the generator network are learned adaptively from the training data.

4.1.2 Contributions

In this chapter, we will propose a deep conditional generative neural network structure inspired by dictionary learning and Generative Latent Optimization (GLO), and focused on a specific application that arises in the computer gaming industry. Our work is motivated by the fact that practitioners in the gaming industry often encounter the issue that building layouts on most available maps are largely missing especially in remote areas. Our goal is to train a conditional model that can take simple inputs conditions such as the locations of roads, natural areas (e.g., vegetation, bodies of water) and generate the possible placement of buildings around them under various styles. We built our model based on Generative Latent Optimization (GLO) [BJLPS18], and extend the framework to a conditional version, which we call conditional GLO (cGLO). cGLO framework allows us to not only learn a generator network, but also learn a latent representation of each building layout sample hinting a style. Note that style refers to details like building size and density that could differentiate between neighborhoods, e.g. residential area and commercial areas. The trained latent representation can then be used as style reference of a specific building layout example in generating new examples with arbitrary input conditions.

We provide three main contributions:

- We propose conditional GLO (cGLO) to control the generation on user-specified maps while training latent representations of various neighborhood styles.
- We enhance the generator in cGLO by adversarial training and crossing latent vectors, in order to learn more realistic and generic neighborhood styles, and decouple the latent variable from associated conditions.

- We formulate the problem of building placements in the scope of image synthesis and format the map data in individual channels. In this way, the dataset can also be used for other tasks such as road generation.

The rest of the paper will be organized as follows: section 4.3 introduced the technical details of our generative model. Implementation details and the experiment results are presented in section 4.4, section 4.5 gives the conclusion and summaries the future works.

4.2 Related Work

4.2.1 Generating Building Placements

Modeling a virtual city is important for a number of applications such as mapping, urban planning, video games [KM07], etc.

Procedural techniques [STBB14] are widely used to create urban space with a few steps, including generating road network [CEW$^+$08], placing buildings and other objects [VKW$^+$12], and creating geometries for single objects [MWZ$^+$14]. In [PM01], Parish and Muller proposed a method to generate extensive street layouts and buildings using L-systems. Kelly and McCabe proposed an interactive system named Citygen [KM07]. They generate building layouts by calculating all the enclosed areas between secondary roads and then subdividing them into lots. The buildings are then placed within the lots and the relevant materials are applied to the generated geometry. However, such procedural methods rely on manually designed grammar and rules that require substantial expertise. Instead, we propose a data-driven method that can automatically generate building placements according to given input road placements, natural obstructions, as well as a specific style example.

4.2.2 Deep Dictionary Learning and Latent Representation Learning

Wu, Rosca, and Lillicrap propose “deep dictionary learning” (DDL) which combines deep learning with a dictionary learning task using a greedy algorithm [TMSV16]. Tariyal, Aggarwal and Majumdar shows that greedy DDL outperforms deep belief network (DBN) and stacked autoencoder (SAE) based techniques for hyperspectral image classification[TAM16]. It proceeds by learning a single layer of the dictionary in each stage where the coefficients from the previous layer act as inputs to the subsequent layer as a greedy algorithm. Singhal and Majumdar propose an alternative solution to DDL whereby all the layers of dictionaries are solved simultaneously [SM18]

Latent Representation provides higher data qualities, as it is trained by mathematical models to reduce the data dimension or filter out the noise. These data representation plays an important role in the results in detection or classification tasks [WYG$^+$08] [CNT11]. Probabilistic graphical models (PGMs) [KF09] learn the latent representation z of the input data x by formulating the joint distributions $p(z,x)$. Autoencoders (AE) [Ng11], [LCLL14] are neural networks that aims to copy the inputs to the outputs. They work by compressing the input into a latent-space representation, and then reconstructing the output from this representation. Autoencoders can be combined with different constraints and embed vectors in various applications.

The Generative model in Generative adversarial nets(GANs) [GPAM$^+$14] learns to map points in the latent space to generated images. However, training GANs requires carefully balancing updates to Discriminator and Generator and is sensitive to both architecture and algorithm choices [SGZ$^+$16]. Inspired by compressed sensing [Don06] [CT06], Wu et al. [WRL19] and Bojanowski et al.[BJLPS18] introduce latent optimization for GANs to improve the stability of training. [WDB$^+$19] provides theoretical analysis from the perspectives of differentiable games and stochastic approximation for [WRL19].

4.2.3 Deep Generative Models

Generative adversarial nets [GPAM$^+$14] provide a novel way to generate images of high quality, and the conditional extension [MO14] learns the relationship between a pair of images so that users have more control over the input. The non-adversarial version, GLO [BJLPS18], was proposed to embed training images in a latent space which is optimized with the generator simultaneously. While these techniques have great success in image synthesis, the potential has also been investigated in computer graphics [LW16] [YM16]. The first attempt in urban modeling using GAN was road networks synthesis [HWWK17]. To fill the gap of building placements, we propose a conditional extension of GLO to learn the intrinsic style of building distribution from single images. In addition, we leverage adversarial training from GAN to learn a better latent representation of an urban neighborhood, that encodes the style of placing buildings towards surrounding environments. The technique that is most relative to our work is ArtGAN [TCAT17] which trains cGAN while feeding labels into the generator. Different from ArtGAN, we train a latent vector for each image and back propagate errors in both generator and discriminator. To the best of our knowledge, there is no method generating buildings and placements following an example style. Therefore, we evaluate our method through comparison with ArtGAN, and also demonstrate the effect of adversarial training in section 4.4.

4.3 Conditional Generative Latent Optimization

In this section we describe our framework in detail. Our data set are acquired from google map api [Goo], we preprocessed the data into 6 channels, each of which is a binary image, representing highways, arterial, local roads, waterways, vegetation and buildings. The first 5 channels serve as input condition, and the building channel as target. Fig 4.3 shows an example of our data. We aim to train the generator to learn the conditional distribution of building layout channel condition on the informations given by the highways, arterial, local roads,

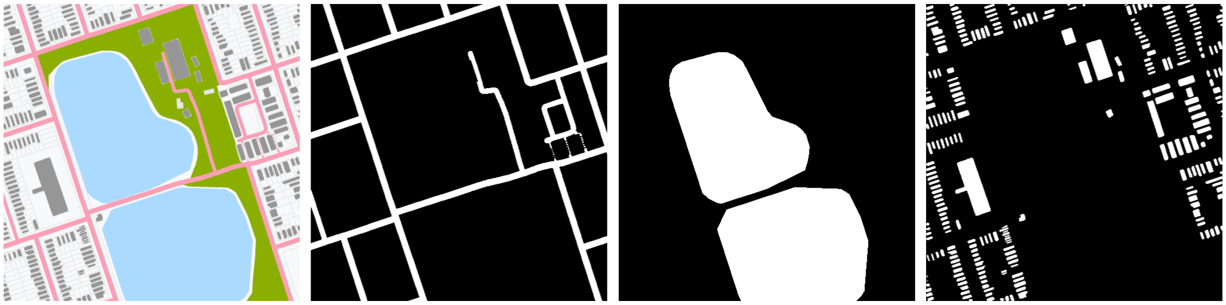


Figure 4.3: Data format. The left image shows a sample neighborhood. The other three images are example channels after preprocessing, which are local roads, waterways and buildings from left to right.

waterways, vegetation channel. To enhance the learning, we combine the cGLO framework with an adversarial training [GPAM$^+$14], where we use two discriminators to differentiate between real and fake samples, along with trainable latent vectors. We detailed our network structure in Figure 4.4.

In order to simplify the notations, we use C to denote input channels, i.e., local roads, arterial, highways, vegetation and waterways. We use z to refer to the latent representations, and X the scattering of buildings.

4.3.1 Generator

First, we introduce our generator, Conditional Generative Latent Optimization (cGLO). In our case, the generator is a neural network that takes in the condition channels and the latent representation, and try to generate building layouts. We denote the generator network by G_ω, where $\omega \in \Omega$ represents the parameters of the generator to be learned.

Conditional GLO

Our training data consists of a set of samples $(X_i \,|\, C_i)$, $i = 1, \cdots, N$, where $X_i \in \mathbb{R}^{n\times n}$ is the i-th sample of $n \times n$ building placement and $C_i \in \mathbb{R}^{5\times n\times n}$ is the associated input condition, each of it contains 5 $n \times n$ channels In order to apply cGLO, we assign a latent vector to each

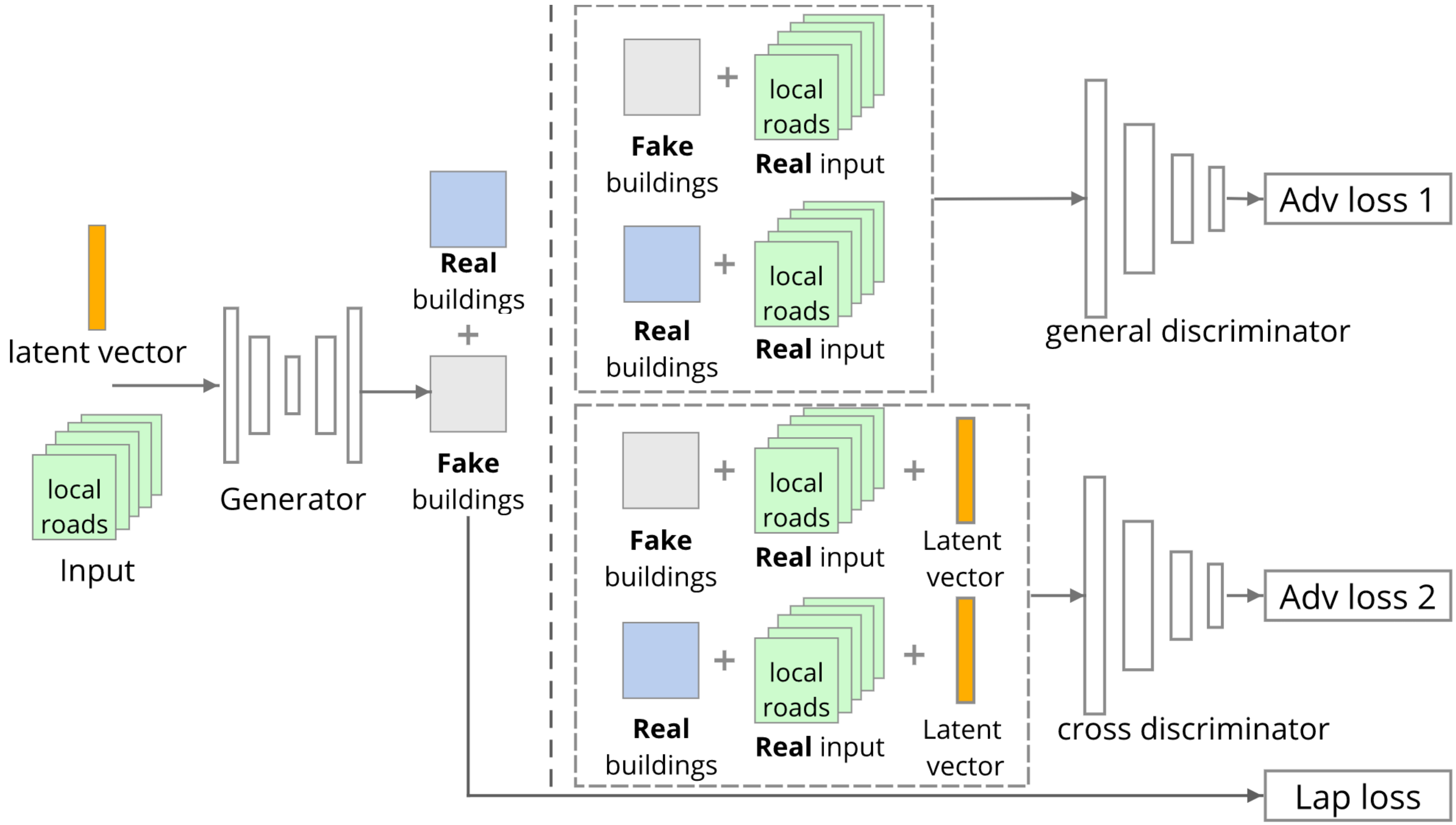


Figure 4.4: Overview of the network structure. The input condition contains 5 channels: highway, arterial, local roads, waterways and vegetation.

sample, resulting in a set of triplets $(X_i, z_i \,|\, C_i)$, $i = 1, \cdots, N$. The objective of cGLO is to optimize a reconstruction loss

$$\min_{\omega, z_1, \cdots, z_N} \ell_g(\omega, z_1, \cdots, z_N) \tag{4.2}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \ell(G_\omega(C_i \,|\, z_i), X_i \,|\, C_i) \tag{4.3}$$

over the network parameters for the generator as well as the latent representations.

The generator architecture G_ω is inspired by the U-net [RFB15], which is mainly used in image segmentation. U-net is comprised of two sub-networks, the contracting sub-network and the expansive sub-network. While U-net looks similar to the popular encoder-decoder architectures, the main difference is the use of skip connections between corresponding layers in the contracting and expansive sub-networks [RFB15]. Fig 4.5 shows the detail design of our generator.

Latent Space and style transfer

Our proposed architecture differs from the other popular uses of the U-net architecture in that we also concatenate a unique trainable latent vector into each input. In cGLO we jointly optimize the input latent vectors $\{z_1, \cdots, z_N\}$ and the model parameters ω. In this way, our trained generative model and the latent representation can both be more adaptive to the training data. For each training sample, our model can offer a latent representation optimized for the generating task. Thus we can use each trained latent vector z_i as a style embedding, together with the input conditions, to perform style transfer at inference.

4.3.2 Discriminators

To help the generator learn the details of the ground truth, we add two discriminators in our training.

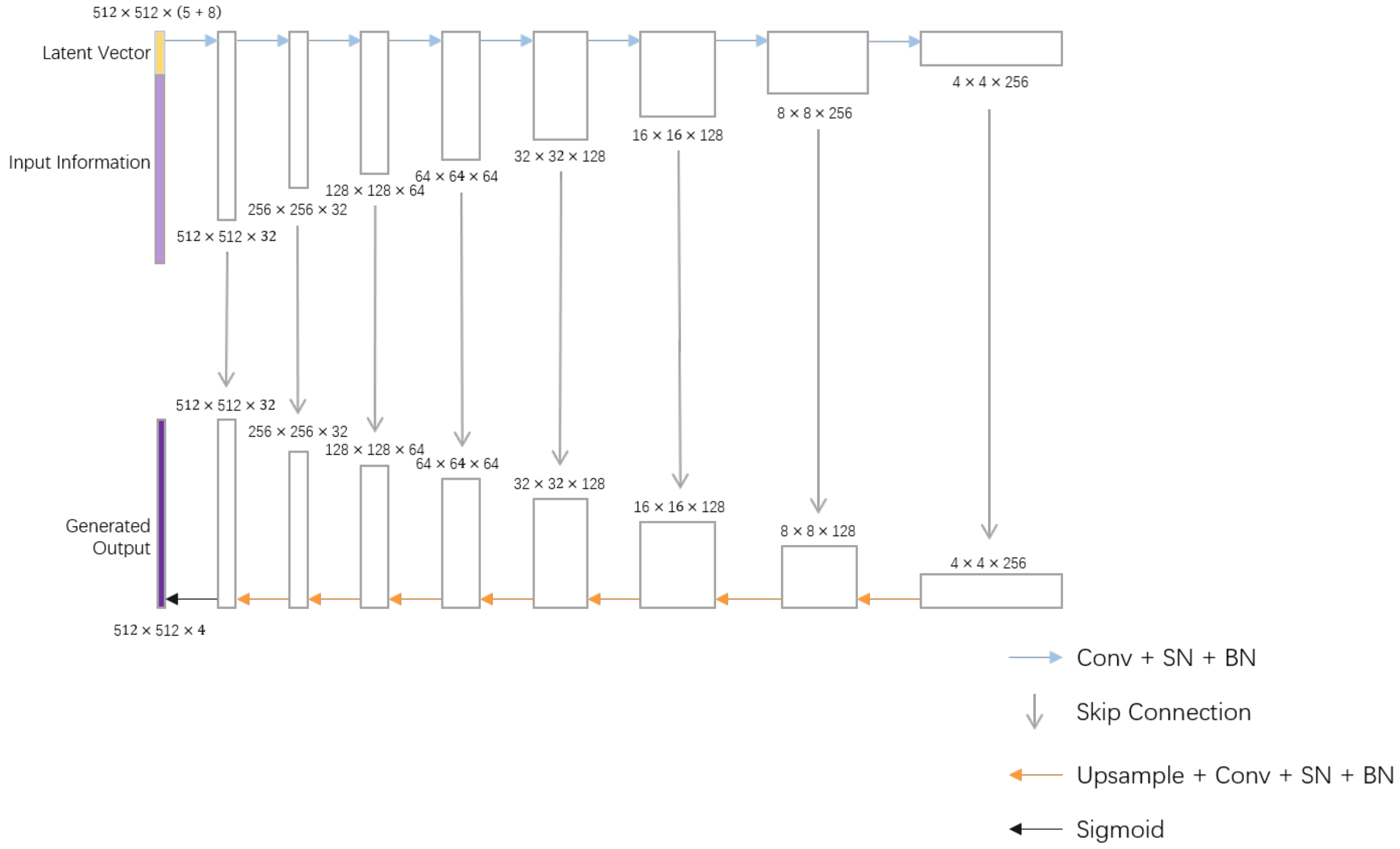


Figure 4.5: The structure of the generator

General Discriminator

The general discriminator network takes input channels and the building layouts either coming from the ground truth training data or generated by the generator, and produces a soft score in $[0,1]$ indicating the probability that the building scattering is faked from the input conditions. Hence, all the true buildings should get low scores while the fake buildings get high scores. We first feed the input channels and the latent vector into our cGLO to generate fake building scatterings. Then we train the general discriminator using those fake building scatterings together with the corresponding real building scatterings. We use multiple convolutional layers with Spectral Normalization [MKKY18] and batch normalization [IS15] followed by down-sampling to obtain a vectorized representation of the input tuple. In the end, we use a fully connected layer with sigmoid activation to produce the predicted probability.

Cross Discriminator

The cross discriminator differ from the general one by taking the latent vector z as input as well. This discriminator could help the model to enhance the style embedding with the latent vector z. We want to let the latent vectors to learn the style information from the training sample, i.e. the number of buildings, the average size, and the density with respect to surrounding environment. However, we observed that the latent vectors can overfit too much road information during training, that the scattering of buildings exhibit the shapes of roads from the reference style. To deal with this situation, we proposed a special procedure to our training in both generator and discriminator. For each iteration in the training, we randomly decided whether we are going to feed the generator with matched input channels i and the latent vector i or mismatched input channels i and latent vector j. By doing this, we can update both z_i and z_j to produce better results for input condition i, which reduce the overfitting for each z_j.

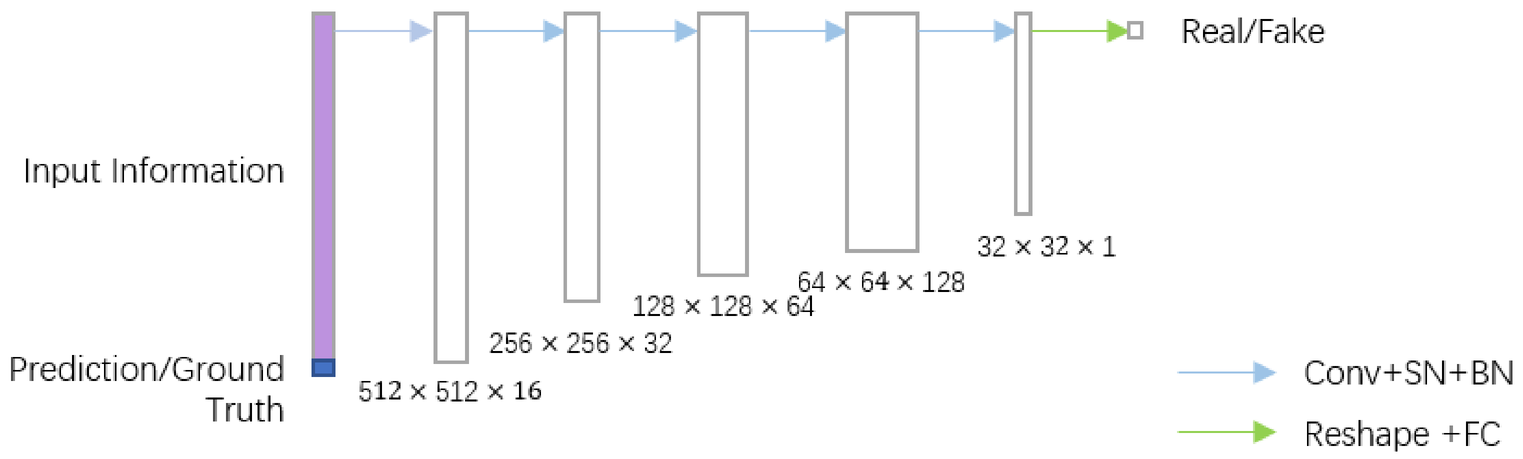


Figure 4.6: The structure of the general discriminator.

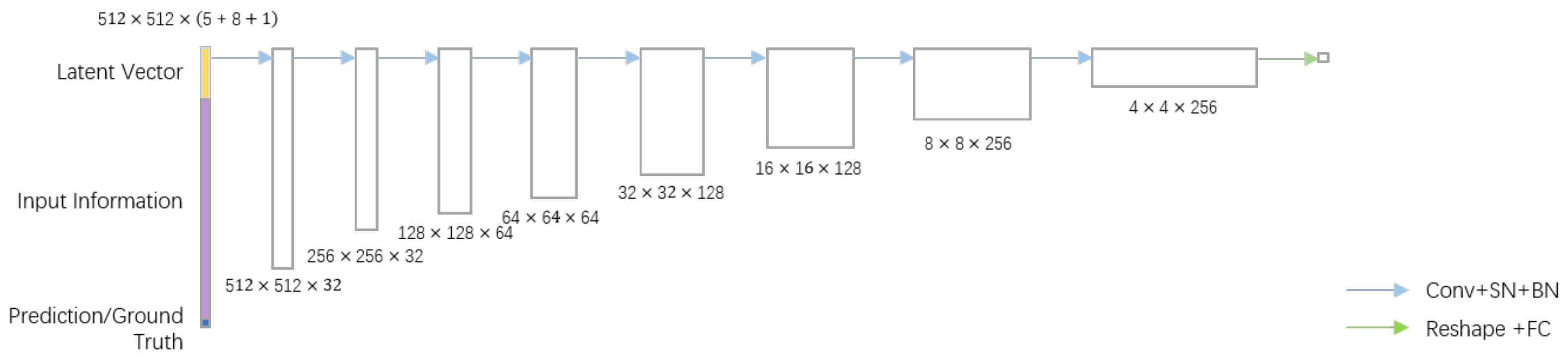


Figure 4.7: The structure of the cross discriminator

4.3.3 Training Loss

The overall training losses of our generative model is comprised by several different components.

The first component is the loss that we use to increase the fidelity of our synthetic building levels, we use a distance function to compare the difference between real and fake scattering of buildings. Although the squared-loss function $\ell_2(x,x') = \|x - x'\|_2^2$ is a simple choice, it leads to blurry (average) reconstructions of natural images. Instead, we use Laplacian pyramid Lap_1 loss:

$$\text{Lap}_1(x,x') = \sum_j 2^{2j} \left| L_j(x) - L^j(x') \right|_1, \tag{4.4}$$

where $L_j(x)$ is the j-th level of the Laplacian pyramid representation of x [LO06]. The Lap_1 loss weights the details at fine scales more heavily, so that the model can learn to fit details in the target.

The second and third components are the adversarial losses for the two discriminators.

$$L_{adv_1}(G, D_{general}) = \sum_i [\log D_{general}(C_i, X_{ireal})] + \sum_i [\log(1 - D_{general}(C_i, X_{ifake}))], \tag{4.5}$$

and

$$L_{adv_2}(G, D_{cross}, z) = \sum_i [\log D_{cross}(z_i, C_i, X_{ireal})] + \sum_i [\log(1 - D_{cross}(z_i, C_i, X_{ifake}))]. \tag{4.6}$$

One can view the adversarial loses as the negative of the cross entropy between the scores of discriminator and the true real/fake labels. Thus, it is natural for the discriminators to adversarially increase the adversarial losses, whereas the generator wants to decrease them. It is important to note that the $X_{fake} = G_\omega(C,z)$ in the second terms of the two losses both depend on the Generator G_ω.

4.4 Implementations and Results

4.4.1 Datasets

The training data are collected from Google Maps API. We queried three cities, i.e. San Francisco, Los Angeles and London as these cities contain rich information on road networks and buildings. We trained a model for each city. We observed that all the models exhibit similar performance since the neighborhoods from metropolitan areas share similar patterns at the level of building scattering. For example, a residential area has dense buildings, while suburban areas are sparse. We take Los Angeles area (LA) as example to present our experiment results. We query 6 channels of data for each sample/neighborhood, i.e., highway, arterial, local roads, vegetation area, waterways and buildings. Each channel is a 512×512 binary image. The building channel serves as our ground truth and all the other 5 channels as input channels. There are in total 5732 training samples in LA data set. In our experiments, instead of using [0, 1] for the building channel, we set the labels with 0.05 (non-building) and 0.95 (building), so that it is easier for the network to generate target values. The reason is that generating 0 and 1 as outputs of sigmoid requires the input to go to $-\infty$ and $+\infty$ respectively.

4.4.2 Implementation details

The latent vectors z and weights ω are updated by Stochastic Gradient Descent (SGD). The loss that we use to train z and ω is

$$L_{G_\omega} = Lap_1(X_{real}, G_\omega(C,z)) + \lambda_1 L_{adv_1} + \lambda_2 L_{adv_2}, \tag{4.7}$$

where λ_1 and λ_2 are non-negative hyper-parameters that balance the influence of different components of the loss function. The gradient of the loss function with respect to z and ω can be obtained by back propagation through the loss [BJPD17]. Our model is trained up to 200 epochs

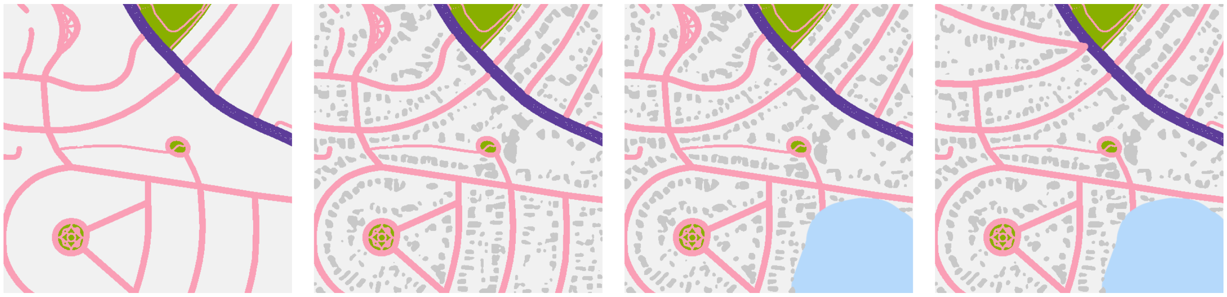


Figure 4.8: An example of building generations under iterative design

with learning rate 0.5 on generator, 0.00001 on discriminator, and 800 on Z. The dimension of our latent vector is 8192 for we found that lower dimensions could not capture enough style information. We train our model alternatively with 5 updates of Z and 1 update of the generator's weight for each iteration, and 1 update of discriminator's weight every 3 iterations.

4.4.3 Experiments of building layout generation

Our trained model has been used to generate buildings to fill the gaps on a game map where buildings are missing. Users can pick a trained example/style from training data and input either a real world map or self designed map to generate buildings in 2 seconds for a 512×512 meters map. We choose commercial and residential neighborhood as examples since these two have distinct styles, i.e., residential area has dense and small size buildings, and commercial area has larger buildings. Our method is robust to local changes of maps, so that artists can freely tune their design of maps and regenerate buildings in real time. Figure 4.8 show an example that starting from a map having a road network and vegetation area, we can iteratively edit the input and our model can adapt building placements to the changes. The left image is the source input map and the second one shows the synthesized building placements. A lake is then added in the third image and the roads are changed in the fourth image, while building placements are adjusted automatically. The advantage of our method is that users do not need to tune any parameters and the generation is fully automatic.

4.4.4 Comparison with state-of-the-art methods

We evaluate our method by comparing with ArtGAN [TCAT17] which is the closest technique to ours. As mentioned earlier, we propose cGLO with two discriminators on top of that. We will show how adversarial training helps cGLO. Given two example inputs, Fig. 4.9 demonstrates the generated buildings corresponding to the three methods under two different styles, i.e., residential and commercial. To train ArtGAN accordingly, we assign the two style classes to the training data. When we retrieved map data we also acquired the information that if a building is commercial or not, thus we can determine the category for each sample by checking if the area of commercial buildings exceeds half of the total building area. The result shows that ArtGAN averages the style and generates fuzzy building areas. It could not distinguish one style from the other, either. One observation is that some neighborhoods exhibit multiple styles, for example, a residential area is next to commercial or industrial districts. Therefore, it is unrealistic to label accurate semantic styles to each sample which leads to poor performance for ArtGAN. However, by learning a latent representation for each sample, cGLO can generate buildings with clear boundaries of the buildings adjacent to each other. Additionally, the generated buildings from our method exhibit significant variability of size, aspect ratios and irregularity. While large buildings dominate in the example commercial style, cGLO failed to capture such features (see Fig. 4.9). However, with adversarial training, cGLO is able to generate more buildings that resemble the given style.

We use the metric proposed in [VKW$^+$12] to measure how much the distribution of building sizes deviates from the expected style. We use the three models (ArtGAN, cGLO and cGLO+Adv) to generate the building layout with the same input condition channels. We transfer the style from the two references used in Fig. 4.9. Fig. 4.10 plots the probability density function of each output sample for comparison. To quantitatively measure the difference, we calculated the Kullback–Leibler divergence for measuring the distance between the distribution of each result sample and the reference style. We can see that the proposed cGLO outperforms both styles while

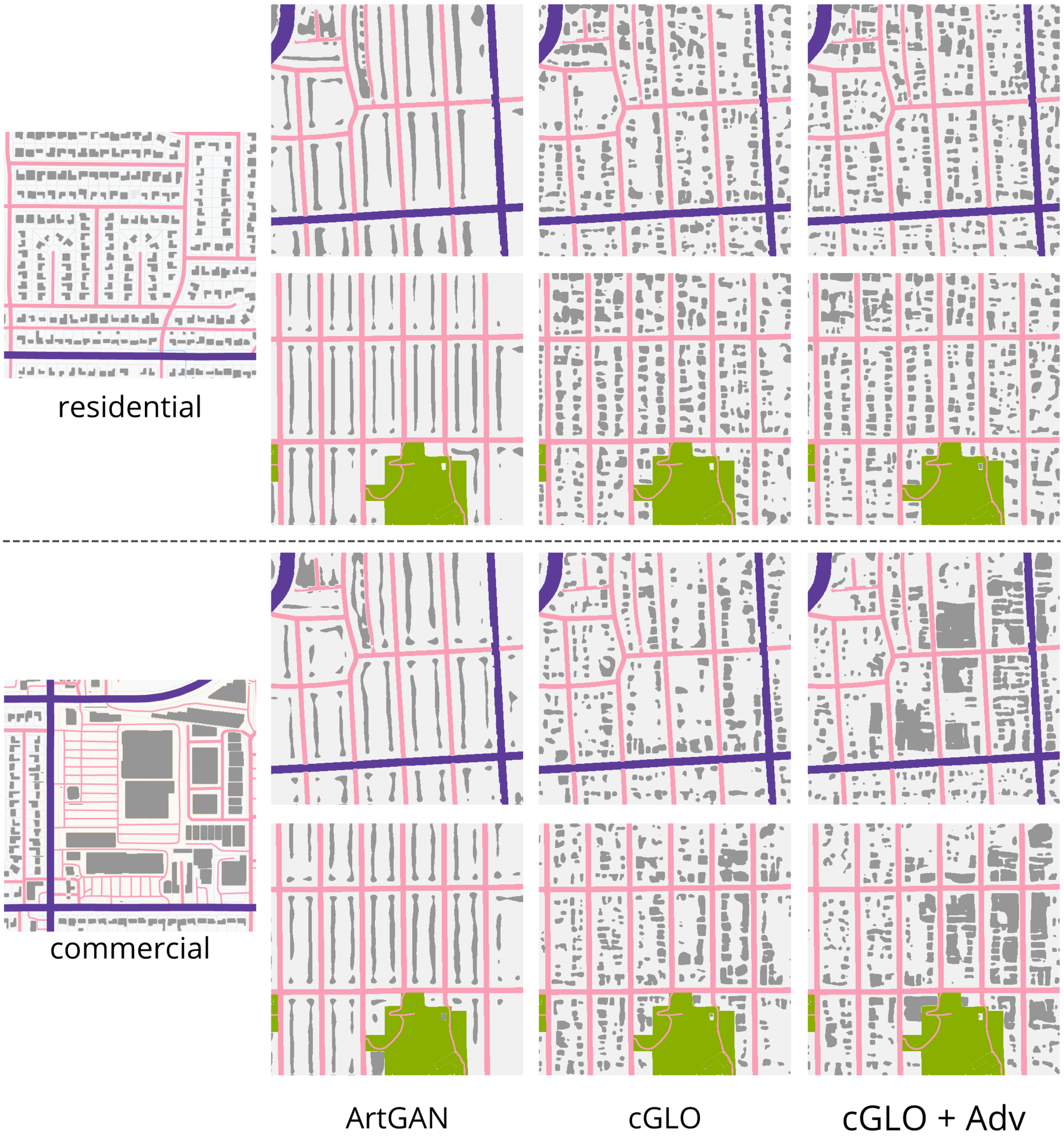
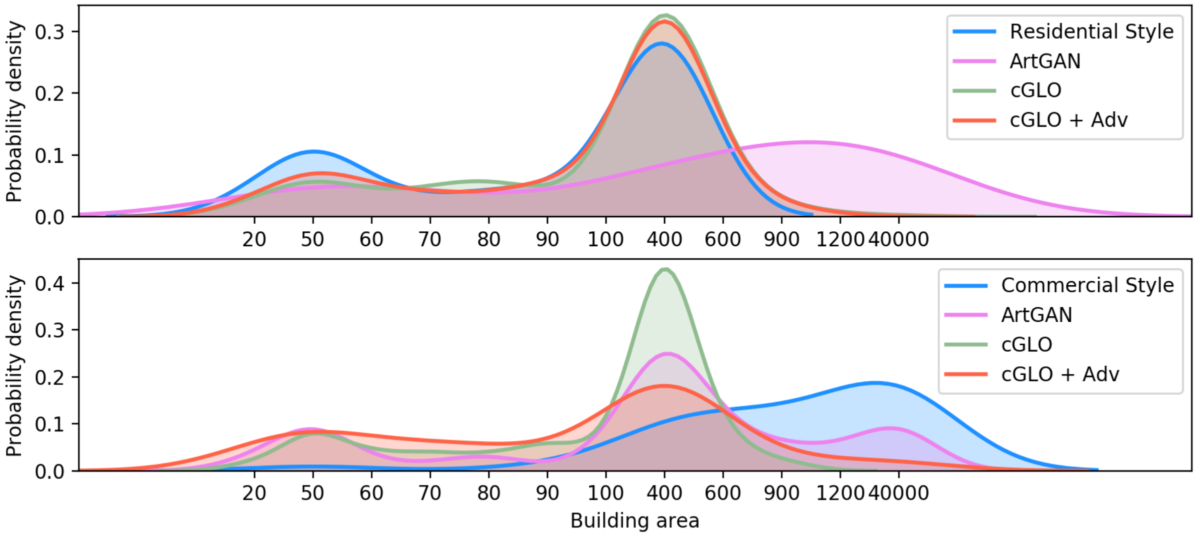


Figure 4.9: The comparion of style transfer using ArtGAN [TCAT17], cGLO and cGLO + Adv we proposed.



KL Divergence	residential	commercial
ArtGAN	0.04691	0.01215
CGLO	0.00304	0.00885
CGLO + Adv	0.00526	0.00791

Figure 4.10: We compare the distributions graph of single building area generated by CCGAN [TCAT17], and our proposed methods, CGLO and CGLO + Adv.

adversarial training enhances the commercial style.

4.5 Conclusion and Future Work

We presented a novel solution to synthesize high quality of building placements using conditional generative latent optimization together with adversarial training. The capability of the proposed method is demonstrated in various examples. The inference is nearly in realtime, thus it can assist designers to iterate their designs of virtual cities quickly. A limitation of the work is that we did not consider the exact orientations of buildings to the streets, which can be incorporated in future work. A simple solution is to assume buildings are aligned to the nearest roads. We believe the work can be extended to other applications, for example, it would be interesting to learn the land use of each building which is important for real world urban planning.

4.6 Acknowledgement

Chapter 4, in full, is a version of the paper "Building Placements in Urban Modeling Using Conditional Generative Latent Optimization." The paper is co-authered with Han Liu, Yiwei Zhao, Maziar Sanjabi, Mohsen Sardari, Harold Chaput, Navid Aghdaie, Kazi Zaman. The manuscript has been submitted for publication. The dissertation author was the primary investigator and author of this paper.

Bibliography

[ABMD92] Marc Antonini, Michel Barlaud, Pierre Mathieu, and Ingrid Daubechies. Image coding using wavelet transform. *IEEE Transactions on image processing*, 1(2):205–220, 1992.

[AE08] Michal Aharon and Michael Elad. Sparse and redundant modeling of image content using an image-signature-dictionary. *SIAM Journal on Imaging Sciences*, 1(3):228–247, 2008.

[AEB06] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.

[AH05] Kendall Atkinson and Weimin Han. *Theoretical numerical analysis*, volume 39. Springer, 2005.

[AKM$^+$14] Aleksandr Aravkin, Rajiv Kumar, Hassan Mansour, Ben Recht, and Felix J Herrmann. Fast methods for denoising matrix completion formulations, with applications to robust seismic data interpolation. *SIAM Journal on Scientific Computing*, 36(5):S237–S266, 2014.

[AS14] Joel Andersson and Jan-Olov Strömberg. On the theorem of uniform recovery of random sampling matrices. *IEEE Transactions on Information Theory*, 60(3):1700–1710, 2014.

[BB86] Ronald Newbold Bracewell and Ronald N Bracewell. *The Fourier transform and its applications*, volume 31999. McGraw-Hill New York, 1986.

[BDDW08] Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, 2008.

[BHSN06] Waheed Bajwa, Jarvis Haupt, Akbar Sayeed, and Robert Nowak. Compressive wireless sensing. In *Proceedings of the 5th international conference on Information processing in sensor networks*, pages 134–142. ACM, 2006.

[BJLPS18] Piotr Bojanowski, Armand Joulin, David Lopez-Pas, and Arthur Szlam. Optimizing the latent space of generative networks. In *International Conference on Machine Learning*, pages 599–608, 2018.

[BJPD17] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis. Compressed sensing using generative models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 537–546. JMLR. org, 2017.

[Bru06] Richard A Brualdi. *Combinatorial matrix classes*, volume 13. Cambridge University Press, 2006.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[BW16] Bubacarr Bah and Rachel Ward. The sample complexity of weighted sparse approximation. *IEEE Transactions on Signal Processing*, 64(12):3145–3155, 2016.

[Can08] Emmanuel J Candès. The restricted isometry property and its implications for compressed sensing. *Comptes rendus mathematique*, 346(9-10):589–592, 2008.

[CD95] Ronald R Coifman and David L Donoho. Translation-invariant de-noising. In *Wavelets and statistics*, pages 125–150. Springer, 1995.

[CDD09] Albert Cohen, Wolfgang Dahmen, and Ronald DeVore. Compressed sensing and best k-term approximation. *Journal of the American mathematical society*, 22(1):211–231, 2009.

[CDH18] Kai-Yang Chiang, Inderjit S Dhillon, and Cho-Jui Hsieh. Using side information to reliably learn low-rank matrices from missing and corrupted observations. *The Journal of Machine Learning Research*, 19(1):3005–3039, 2018.

[CDS01] Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159, 2001.

[CEW$^+$08] Guoning Chen, Gregory Esch, Peter Wonka, Pascal Müller, and Eugene Zhang. Interactive procedural street modeling. In *ACM SIGGRAPH 2008 papers*, pages 1–10. 2008.

[CNT11] Yi Chen, Nasser M Nasrabadi, and Trac D Tran. Hyperspectral image classification using dictionary-based sparse representation. *IEEE transactions on geoscience and remote sensing*, 49(10):3973–3985, 2011.

[Cod92] Mac A Cody. The fast wavelet transform: Beyond fourier transforms. Number 4, pages 16–28. Citeseer, 1992.

[CP10] Emmanuel J Candès and Yaniv Plan. Tight oracle bounds for low-rank matrix recovery from a mininal number of noisy random measurements. *arXiv preprint arXiv:1001.0339*, 2010.

[CP11] Emmanuel J Candès and Yaniv Plan. Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements. *IEEE Transactions on Information Theory*, 57(4):2342–2359, 2011.

[CR07] Emmanuel J Candès and Justin Romberg. Sparsity and incoherence in compressive sampling. *Inverse problems*, 23(3):969, 2007.

[CR09] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717, 2009.

[CRT06a] Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509, 2006.

[CRT06b] Emmanuel J Candès, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(8):1207–1223, 2006.

[CT05] Emmanuel J Candès and Terence Tao. Decoding by linear programming. *arXiv preprint math/0502327*, 2005.

[CT06] Emmanuel J Candès and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE transactions on information theory*, 52(12):5406–5425, 2006.

[CT10] Emmanuel J Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

[CTL08] Guang-Hong Chen, Jie Tang, and Shuai Leng. Prior image constrained compressed sensing (piccs): a method to accurately reconstruct dynamic ct images from highly undersampled projection data sets. *Medical physics*, 35(2):660–663, 2008.

[CW11] T Tony Cai and Lie Wang. Orthogonal matching pursuit for sparse signal recovery with noise. *IEEE Transactions on Information theory*, 57(7):4680–4688, 2011.

[CWX09] T Tony Cai, Lie Wang, and Guangwu Xu. Shifting inequality and recovery of sparse signals. *IEEE Transactions on Signal Processing*, 58(3):1300–1308, 2009.

[CWX10] T Tony Cai, Lie Wang, and Guangwu Xu. New bounds for restricted isometry constants. *IEEE Transactions on Information Theory*, 56(9):4388–4394, 2010.

[CYL13] Yang Cong, Junsong Yuan, and Ji Liu. Abnormal event detection in crowded scenes using sparse representation. *Pattern Recognition*, 46(7):1851–1864, 2013.

[CZ13] T Tony Cai and Anru Zhang. Sharp rip bound for sparse signal and low-rank matrix recovery. *Applied and Computational Harmonic Analysis*, 35(1):74–93, 2013.

[DAH18] Sajad Daei, Arash Amini, and Farzan Haddadi. Optimal weighted low-rank matrix recovery with subspace prior information. *arXiv preprint arXiv:1809.10356*, 2018.

[DD00] David L Donoho and Mark R Duncan. Digital curvelet transform: strategy, implementation, and experiments. In *Wavelet applications VII*, volume 4056, pages 12–30. International Society for Optics and Photonics, 2000.

[DDT$^+$08] Marco F Duarte, Mark A Davenport, Dharmpal Takhar, Jason N Laska, Ting Sun, Kevin F Kelly, and Richard G Baraniuk. Single-pixel imaging via compressive sampling. *IEEE signal processing magazine*, 25(2):83–91, 2008.

[DE03] David L Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1-minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003.

[DET05] David L Donoho, Michael Elad, and Vladimir N Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on information theory*, 52(1):6–18, 2005.

[DF10] Krishnamurthy Dvijotham and Maryam Fazel. A nullspace analysis of the nuclear norm heuristic for rank minimization. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3586–3589. IEEE, 2010.

[DH01] David L Donoho and Xiaoming Huo. Uncertainty principles and ideal atomic decomposition. *IEEE transactions on information theory*, 47(7):2845–2862, 2001.

[DJ94] David L Donoho and Iain M Johnstone. Ideal denoising in an orthonormal basis chosen from a library of bases. *Comptes rendus de l'Académie des sciences. Série I, Mathématique*, 319(12):1317–1322, 1994.

[DLZS11] Weisheng Dong, Xin Li, Lei Zhang, and Guangming Shi. Sparsity-based image denoising via dictionary learning and structural clustering. In *CVPR 2011*, pages 457–464. IEEE, 2011.

[DMA97] Geoff Davis, Stephane Mallat, and Marco Avellaneda. Adaptive greedy approximations. *Constructive approximation*, 13(1):57–98, 1997.

[Don06] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[DR16] Mark A Davenport and Justin Romberg. An overview of low-rank matrix recovery from incomplete observations. *IEEE Journal of Selected Topics in Signal Processing*, 10(4):608–622, 2016.

[DV05] Minh N Do and Martin Vetterli. The contourlet transform: an efficient directional multiresolution image representation. *IEEE Transactions on image processing*, 14(12):2091–2106, 2005.

[EA06] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006.

[EAH99] Kjersti Engan, Sven Ole Aase, and J Hakon Husoy. Method of optimal directions for frame design. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, volume 5, pages 2443–2446. IEEE, 1999.

[EB02] Michael Elad and Alfred M Bruckstein. A generalized uncertainty principle and sparse representation in pairs of bases. *IEEE Transactions on Information Theory*, 48(9):2558–2567, 2002.

[Ela10] Michael Elad. *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer Science & Business Media, 2010.

[EYW18] Armin Eftekhari, Dehui Yang, and Michael B Wakin. Weighted matrix completion and recovery with prior subspace information. *IEEE Transactions on Information Theory*, 64(6):4044–4071, 2018.

[Faz02] Maryam Fazel. Matrix rank minimization with applications. *PhD thesis, Stanford University*, 2002.

[FL09] Simon Foucart and Ming-Jun Lai. Sparsest solutions of underdetermined linear systems via ℓ_q-minimization for $0 < q \leq 1$. *Applied and Computational Harmonic Analysis*, 26(3):395–407, 2009.

[FMSY11] Michael P Friedlander, Hassan Mansour, Rayan Saab, and Özgür Yilmaz. Recovering compressively sampled signals using partial support information. *IEEE Transactions on Information Theory*, 58(2):1122–1134, 2011.

[Fou10] Simon Foucart. A note on guaranteed sparse recovery via ℓ_1-minimization. *Applied and Computational Harmonic Analysis*, 29(1):97–103, 2010.

[FR13] Simon Foucart and Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*. Birkhäuser Basel, 2013.

[Fuc04] J-J Fuchs. On sparse representations in arbitrary redundant bases. *IEEE transactions on Information theory*, 50(6):1341–1344, 2004.

[GB14] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, March 2014.

[GKSA11] TS Gunawan, OO Khalifa, AA Shafie, and E Ambikairajah. Speech compression using compressive sensing on a multicore system. In *2011 4th International Conference on Mechatronics (ICOM)*, pages 1–4. IEEE, 2011.

[Goo] Google. Google maps api. `https://developers.google.com/maps/documentation`.

[Gor88] Yehoram Gordon. On milman's inequality and random subspaces which escape through a mesh in $\mathbb{R}^n$. In *Geometric Aspects of Functional Analysis*, pages 84–106. Springer, 1988.

[GPAM$^+$14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[Gro11] David Gross. Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory*, 57(3):1548–1566, 2011.

[HDRL16] Ke-Kun Huang, Dao-Qing Dai, Chuan-Xian Ren, and Zhao-Rong Lai. Learning kernel extended dictionary for face recognition. *IEEE transactions on neural networks and learning systems*, 28(5):1082–1094, 2016.

[HHJ94] Roger A Horn, Roger A Horn, and Charles R Johnson. *Topics in matrix analysis*. Cambridge university press, 1994.

[HWWK17] Stefan Hartmann, Michael Weinmann, Raoul Wessel, and Reinhard Klein. Streetgan: Towards road network synthesis with generative adversarial networks. 2017.

[IS15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[KF09] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[KKRT16] Maryia Kabanava, Richard Kueng, Holger Rauhut, and Ulrich Terstiege. Stable low-rank matrix recovery via null space properties. *Information and Inference: A Journal of the IMA*, 5(4):405–441, 2016.

[KM07] George Kelly and Hugh McCabe. Citygen: An interactive system for procedural city generation. In *Fifth International Conference on Game Design and Technology*, pages 8–16, 2007.

[KNH] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

[KOH12] Anilesh K Krishnaswamy, Samet Oymak, and Babak Hassibi. A simpler approach to weighted ℓ_1 minimization. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3621–3624. IEEE, 2012.

[Kor09] Yehuda Koren. The bellkor solution to the netflix grand prize. *Netflix prize documentation*, 81(2009):1–10, 2009.

[KRT17] Richard Kueng, Holger Rauhut, and Ulrich Terstiege. Low rank matrix recovery from rank one measurements. *Applied and Computational Harmonic Analysis*, 42(1):88–116, 2017.

[KXAH09] M Amin Khajehnejad, Weiyu Xu, A Salman Avestimehr, and Babak Hassibi. Weighted ℓ_1 minimization for sparse recovery with prior information. In *2009 IEEE international symposium on information theory*, pages 483–487. IEEE, 2009.

[KXAH11] M Amin Khajehnejad, Weiyu Xu, A Salman Avestimehr, and Babak Hassibi. Analyzing weighted ℓ_1 minimization for sparse recovery with nonuniform sparse models. *IEEE Transactions on Signal Processing*, 59(5):1985–2001, 2011.

[LCLL14] Cheng-Yuan Liou, Wei-Chen Cheng, Jiun-Wei Liou, and Daw-Ran Liou. Autoencoder for words. *Neurocomputing*, 139:84–96, 2014.

[LDP07] Michael Lustig, David Donoho, and John M Pauly. Sparse mri: The application of compressed sensing for rapid mr imaging. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007.

[LDSP08] Michael Lustig, David L Donoho, Juan M Santos, and John M Pauly. Compressed sensing mri. *IEEE signal processing magazine*, 25(2):72, 2008.

[LHDJ11] Sajan Goud Lingala, Yue Hu, Edward DiBella, and Mathews Jacob. Accelerated dynamic mri exploiting sparsity and low-rank structure: kt slr. *IEEE transactions on medical imaging*, 30(5):1042–1054, 2011.

[Lim90] Jae S Lim. *Two-dimensional signal and image processing*. ph, 1990.

[LO06] Haibin Ling and Kazunori Okada. Diffusion distance for histogram comparison. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 246–253. IEEE, 2006.

[Lop13] Miles Lopes. Estimating unknown sparsity in compressed sensing. In *International Conference on Machine Learning*, pages 217–225, 2013.

[LV10] Wei Lu and Namrata Vaswani. Exact reconstruction conditions and error bounds for regularized modified basis pursuit (reg-modified-bp). In *2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers*, pages 763–767. IEEE, 2010.

[LW16] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European conference on computer vision*, pages 702–716. Springer, 2016.

["Ma] "MathWorks". rgb2gray function matlab documentation. `mathworks.com/help/matlab/ref/rgb2gray.html`.

[Mat] MathWorks. lsqlin. `https://www.mathworks.com/help/optim/ug/lsqlin.html`.

[MDR17] João FC Mota, Nikos Deligiannis, and Miguel RD Rodrigues. Compressed sensing with prior information: Strategies, geometry, and bounds. *IEEE Transactions on Information Theory*, 63(7):4472–4496, 2017.

[MGBB00] Michael W Marcellin, Michael J Gormish, Ali Bilgin, and Martin P Boliek. An overview of jpeg-2000. In *Proceedings DCC 2000. Data Compression Conference*, pages 523–541. IEEE, 2000.

[MKKY18] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[ML11] Qun Mo and Song Li. New bounds on the restricted isometry constant δ2k. *Applied and Computational Harmonic Analysis*, 31(3):460–468, 2011.

[MO14] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[MOW12] Pat Muldowney, Krzysztof Ostaszewski, and Wojciech Wojdowski. The darth vader rule. *Tatra Mountains Mathematical Publications*, 52(1):53–63, 2012.

[MP15] Sidhant Misra and Pablo A Parrilo. Weighted ℓ_1-minimization for generalized non-uniform sparse model. 2015.

[MPS$^+$09] Julien Mairal, Jean Ponce, Guillermo Sapiro, Andrew Zisserman, and Francis R Bach. Supervised dictionary learning. In *Advances in neural information processing systems*, pages 1033–1040, 2009.

[MPTJ08] Shahar Mendelson, Alain Pajor, and Nicole Tomczak-Jaegermann. Uniform uncertainty principle for bernoulli and subgaussian ensembles. *Constructive Approximation*, 28(3):277–289, 2008.

[MS17] Hassan Mansour and Rayan Saab. Recovery analysis for weighted ℓ_1-minimization using the null space property. *Applied and Computational Harmonic Analysis*, 43(1):23–38, 2017.

[MWZ$^+$14] Niloy J Mitra, Michael Wand, Hao Zhang, Daniel Cohen-Or, Vladimir Kim, and Qi-Xing Huang. Structure-aware shape processing. In *ACM SIGGRAPH 2014 Courses*, pages 1–21. 2014.

[MY11] Hassan Mansour and Özgür Yilmaz. Weighted-l1 minimization with multiple weighting sets. In *Wavelets and Sparsity XIV*, volume 8138, page 813809. International Society for Optics and Photonics, 2011.

[Ng11] Andrew Ng. Sparse autoencoder. *CS294A Lecture notes*, 2011.

[NSW17] Deanna Needell, Rayan Saab, and Tina Woolf. Weighted-minimization for sparse recovery under arbitrary prior information. *Information and Inference: A Journal of the IMA*, 6(3):284–309, 2017.

[Nus81] Henri J Nussbaumer. The fast fourier transform. In *Fast Fourier Transform and Convolution Algorithms*, pages 80–111. Springer, 1981.

[OMFH11] Samet Oymak, Karthik Mohan, Maryam Fazel, and Babak Hassibi. A simplified approach to recovery conditions for low rank matrices. *arXiv preprint arXiv:1103.1178*, 2011.

[PM01] Yoav IH Parish and Pascal Müller. Procedural modeling of cities. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 301–308. ACM, 2001.

[PRK93] Yagyensh Chandra Pati, Ramin Rezaiifar, and Perinkulam Sambamurthy Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*, pages 40–44. IEEE, 1993.

[Rau10] Holger Rauhut. Compressive sensing and structured random matrices. *Theoretical foundations and numerical methods for sparse recovery*, 9:1–92, 2010.

[Rec11] Benjamin Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12(Dec):3413–3430, 2011.

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[RFP10] Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

[RMC15] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[RV10] Mark Rudelson and Roman Vershynin. Non-asymptotic theory of random matrices: extreme singular values. In *Proceedings of the International Congress of Mathematicians 2010 (ICM 2010) (In 4 Volumes) Vol. I: Plenary Lectures and Ceremonies Vols. II–IV: Invited Lectures*, pages 1576–1602. World Scientific, 2010.

[RYRD15] Nikhil Rao, Hsiang-Fu Yu, Pradeep K Ravikumar, and Inderjit S Dhillon. Collaborative filtering with graph information: Consistency and scalable methods. In *Advances in neural information processing systems*, pages 2107–2115, 2015.

[SCD02] Jean-Luc Starck, Emmanuel J Candès, and David L Donoho. The curvelet transform for image denoising. *IEEE Transactions on image processing*, 11(6):670–684, 2002.

[SED12] Jonathan Scarlett, Jamie S Evans, and Subhrakanti Dey. Compressed sensing with prior information: Information-theoretic limits and practical decoders. *IEEE Transactions on Signal Processing*, 61(2):427–439, 2012.

[SFM07] Jean-Luc Starck, Jalal Fadili, and Fionn Murtagh. The undecimated wavelet decomposition and its reconstruction. *IEEE transactions on image processing*, 16(2):297–309, 2007.

[SGZ$^+$16] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[SM18] Vanika Singhal and Angshul Majumdar. Majorization minimization technique for optimally solving deep dictionary learning. *Neural Processing Letters*, 47(3):799–814, 2018.

[STBB14] Ruben M Smelik, Tim Tutenel, Rafael Bidarra, and Bedrich Benes. A survey on procedural modelling for virtual worlds. In *Computer Graphics Forum*, volume 33, pages 31–50. Wiley Online Library, 2014.

[TAM16] Snigdha Tariyal, Hemant Aggarwal, and Angshul Majumdar. Greedy deep dictionary learning for hyperspectral image classification. In *2016 8th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–4. IEEE, 2016.

[TCAT17] Wei Ren Tan, Chee Seng Chan, Hernán E Aguirre, and Kiyoshi Tanaka. Artgan: Artwork synthesis with conditional categorical gans. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3760–3764. IEEE, 2017.

[TG07] Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory*, 53(12):4655–4666, 2007.

[TMSV16] Snigdha Tariyal, Angshul Majumdar, Richa Singh, and Mayank Vatsa. Deep dictionary learning. *IEEE Access*, 4:10096–10109, 2016.

[Tro04] Joel A Tropp. Just relax: Convex programming methods for subset selection and sparse approximation. *ICES report*, 404, 2004.

[vdBF19] E. van den Berg and M. P. Friedlander. SPGL1: A solver for large-scale sparse reconstruction, December 2019. https://friedlander.io/spgl1.

[Ver18] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[VKW$^+$12] Carlos A Vanegas, Tom Kelly, Basil Weber, Jan Halatsch, Daniel G Aliaga, and Pascal Müller. Procedural generation of parcels in urban modeling. In *Computer graphics forum*, volume 31, pages 681–690. Wiley Online Library, 2012.

[VL10] Namrata Vaswani and Wei Lu. Modified-cs: Modifying compressive sensing for problems with partially known support. *IEEE Transactions on Signal Processing*, 58(9):4595–4607, 2010.

[Wal92] Gregory K Wallace. The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv, 1992.

[WDB$^+$19] Yan Wu, Jeff Donahue, David Balduzzi, Karen Simonyan, and Timothy Lillicrap. Logan: Latent optimisation for generative adversarial networks. *arXiv preprint arXiv:1912.00953*, 2019.

[WEBB15] Lior Weizman, Yonina C Eldar, and Dafna Ben Bashat. Compressed sensing for longitudinal mri: an adaptive-weighted approach. *Medical physics*, 42(9):5195–5208, 2015.

[Wik] Wikipedia. Fast fourier transform. `https://en.wikipedia.org/wiki/Fast_Fourier_transform`.

[WLD$^+$06] Michael B Wakin, Jason N Laska, Marco F Duarte, Dror Baron, Shriram Sarvotham, Dharmpal Takhar, Kevin F Kelly, and Richard G Baraniuk. An architecture for compressive imaging. In *2006 International Conference on Image Processing*, pages 1273–1276. IEEE, 2006.

[WRL19] Yan Wu, Mihaela Rosca, and Timothy Lillicrap. Deep compressed sensing. *arXiv preprint arXiv:1905.06723*, 2019.

[WYG$^+$08] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):210–227, 2008.

[XJZ13] Miao Xu, Rong Jin, and Zhi-Hua Zhou. Speedup matrix completion with side information: Application to multi-label learning. In *Advances in neural information processing systems*, pages 2301–2309, 2013.

[YM16] M Ersin Yumer and Niloy J Mitra. Learning semantic deformation flows with 3d convolutional networks. In *European Conference on Computer Vision*, pages 294–311. Springer, 2016.

[YWHM10] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010.

[YWL$^+$12] Jianchao Yang, Zhaowen Wang, Zhe Lin, Scott Cohen, and Thomas Huang. Coupled dictionary training for image super-resolution. *IEEE transactions on image processing*, 21(8):3467–3478, 2012.

[ZL10] Qiang Zhang and Baoxin Li. Discriminative k-svd for dictionary learning in face recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2691–2698. IEEE, 2010.

[ZZM12] Shaoting Zhang, Yiqiang Zhan, and Dimitris N Metaxas. Deformable segmentation via sparse representation and dictionary learning. *Medical Image Analysis*, 16(7):1385–1396, 2012.

[ZZX$^+$12] Jian Zhang, Chen Zhao, Ruiqin Xiong, Siwei Ma, and Debin Zhao. Image super-resolution via dual-dictionary learning and sparse representation. In *2012 IEEE International Symposium on Circuits and Systems*, pages 1688–1691. IEEE, 2012.