Algorithms for solving inverse problems using generative models

by

Viraj Shah

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Electrical Engineering (Communications and Signal Processing)

Program of Study Committee:
Chinmay Hegde, Major Professor
Baskar Ganapathysubramanian
Soumik Sarkar

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this thesis. The Graduate College will ensure this thesis is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University

Ames, Iowa

2019

TABLE OF CONTENTS

Page

ACKNOWLEDGMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis. First and foremost, Dr. Chinmay Hegde for his guidance, patience and support throughout this research and the writing of this thesis. His insights and words of encouragement have often inspired me and renewed my hopes for completing my graduate education. I would also like to thank my committee members for their efforts and contributions to this work: Dr. Baskar Ganapathysubramanian and Dr. Soumik Sarkar.

ABSTRACT

The traditional approach of hand-crafting priors (such as sparsity) for solving inverse problems is slowly being replaced by the use of richer learned priors (such as those modeled by generative adversarial networks, or GANs). In this work, we study the algorithmic aspects of such a learning-based approach from a theoretical perspective. For certain generative network architectures, we establish a simple non-convex algorithmic approach that (a) theoretically enjoys linear convergence guarantees for certain linear and nonlinear inverse problems, and (b) empirically improves upon conventional techniques such as back-propagation. We support our claims with the experimental results for solving various inverse problems. We also propose an extension of our approach that can handle model mismatch (i.e., situations where the generative network prior is not exactly applicable.) Together, our contributions serve as building blocks towards a principled use of generative models in inverse problems with more complete algorithmic understanding.

CHAPTER 1. INTRODUCTION

1.1 Motivation

Inverse problems arise in a diverse range of application domains including computational imaging, optics, astrophysics, and seismic geo-exploration. In each of these applications, there is a target signal or image (or some other quantity of interest) to be obtained; a device (or some other physical process) records measurements of the target; and the goal is to reconstruct an estimate of the signal from the observations.

Let us suppose that $x^* \in \mathbb{R}^n$ denotes the signal of interest and $y = \mathcal{A}(x^*) \in \mathbb{R}^m$ denotes the observed measurements. The aim is to recover (an estimate of) the unknown signal x^* given y and $\mathcal{A}$. Based on the forward measurement operator $\mathcal{A}$, the inverse problem can be defined in two broad categories: linear, and nonlinear. Many important problems in signal and image processing can be modeled with a *linear* measurement operator $\mathcal{A}$, and thus called linear inverse problems. For example, compressive sensing, the classical problem of *super-resolution* or the problem of *image inpainting*. In case of nonlinear inverse problems, the operator $\mathcal{A}$ exhibits nonlinearity,e.g. phase retrieval and modulo recovery problems.

When $m < n$ the inverse problem is ill-posed, and some kind of prior (or regularizer) is necessary to obtain a meaningful solution. A common technique used to solve ill-posed inverse problems is to solve a constrained optimization problem:

$$\begin{aligned} \widehat{x} &= \arg\min \; F(x), \\ &\text{s. t.} \quad x \in \mathcal{S}, \end{aligned} \tag{1.1}$$

where F is an objective function that typically depends on y and $\mathcal{A}$, and $\mathcal{S} \subseteq \mathbb{R}^n$ captures some sort of *structure* that x^* is assumed to obey.

A very common modeling assumption, particularly in signal and image processing applications, is *sparsity*, wherein $\mathcal{S}$ is the set of sparse vectors in some (known) basis representation. The now-popular framework of *compressive sensing* studies the special case where the forward measurement operator $\mathcal{A}$ can be modeled as a linear operator that satisfies certain (restricted) stability properties; when this is the case, accurate estimation of x^* can be performed provided the signal x^* is sufficiently sparse [8].

Parallel to the development of algorithms that leverage sparsity priors, the last decade has witnessed analogous approaches for other families of structural constraints. These include structured sparsity [3, 24], union-of-subspaces [16], dictionary models [1, 17], total variation models [10], analytical transforms [39], among many others.

Lately, there has been renewed interest in prior models that are parametrically defined in terms of a *deep neural network*. We call these *generative network* models. Specifically, we define

$$\mathcal{S} = \{x \in \mathbb{R}^n \mid x = G(z),\ z \in \mathbb{R}^k\}$$

where z is a k-dimensional latent parameter vector and G is parameterized by the weights and biases of a d-layer neural network. One way to obtain such a model is to train a generative adversarial network (GAN) [19]). GANs have found remarkable applications in modeling image distributions [7, 11, 51, 52], and a well-trained GAN closely captures the notion of a signal (or image) being 'natural' [4], leading many to speculate that the range of such generative models can approximate a low-manifold containing naturally occurring images. Indeed, GAN-based neural network learning algorithms have been successfully employed to solve *linear* inverse problems such as image super-resolution and inpainting [31, 49]. However, most of these approaches are heuristic, and a general theoretical framework for analyzing the performance of such approaches is not available at the moment.

1.2 Contributions

Our goal in this work is to take some initial steps towards a principled use of GAN priors for inverse problems by i) proposing and analyzing the well known projected gradient descent (PGD)

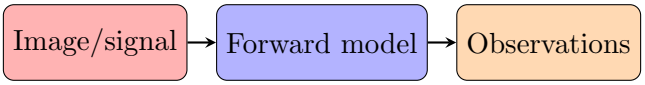


Figure 1.1 Block diagram for a generic inverse problem. The goal is to reconstruct (an estimate of) the data/signal given knowledge of the observations or measurements.

algorithm for solving (1.1) for both linear and nonlinear cases; ii) building a general theoretical framework for analyzing performance of such approaches from *algorithmic* standpoint.

Specifically, apart from providing algorithms to solve inverse problems using generative network models, we also wish to understand the *algorithmic* costs involved with such algorithms: how computationally challenging they are, whether they provably succeed, and how to make such models robust.

The starting point of our work is the recent, seminal paper by [5], who study the benefits of using generative models in the context of compressive sensing. In this work, the authors pose the estimated target as the solution to a non-convex optimization problem and establish upper bounds on the *statistical* complexity of obtaining a "good enough" solution. Specifically, they prove that if the generative network is a mapping $G : \mathbb{R}^k \to \mathbb{R}^n$ simulated by a d-layer neural network with width $\leq n$ and with activation functions obeying certain properties, then $m = O(kd \log n)$ random observations are sufficient to obtain a good enough reconstruction estimate. However, they do not explicitly discuss an *algorithm* to perform such non-convex optimization. Moreover, the authors do not study the *algorithmic* costs of solving the optimization problem, and standard results in non-convex optimization are sufficient to only obtain sublinear convergence rates.

In this work, we establish a PGD algorithm with linear convergence rates for the compressive sensing setup similar to [5], and demonstrate its empirical benefits. This constitutes **Contribution I** of this work. Further, we generalize this to a much wider range of *nonlinear* inverse problems. Using standard techniques, we propose a generic version of our PGD algorithm named $\varepsilon - PGD$ for solving (1.1), analyze its performance, and prove that it demonstrates linear convergence. We also

provide empirical results for solving couple of nonlinear inverse problems. This forms **Contribution II**.

A drawback of the work of [5] (and our contribution I) is the inability to deal with targets that are outside the range of the generative network model. This is not merely an artifact of their analysis; generative networks are rigid in the sense that once they are learned, they are incapable of reproducing any target outside their range. (This is in contrast with other popular parametric models such as sparsity models; these exhibit a "graceful decay" property in the sense that if the sparsity parameter s is large enough, such models capture all possible points in the target space.) This issue is addressed, and empirically resolved, in the recent work of [12] who propose a hybrid model combining both generative networks and sparsity. This leads to a non-convex optimization framework (called *SparseGen*) which the authors theoretically analyze to obtain analogous statistical complexity results. However, here too, the theoretical contribution is primarily statistical and the algorithmic aspects of their setup are not discussed.

We address this gap, and propose an alternative algorithm for this framework. We call it Myopic ε-PGD algorithm. It is novel, nonlinear extension of the previous works [22, 23]. Under (fairly) standard assumptions, this algorithm also can be shown to demonstrate linear convergence. This constitutes **Contribution III** of this work.

In summary: we complement the work of [5] and [12] by providing PGD based algorithms, and algorithmic upper bounds for the corresponding problems that are studied in those works. Together, our contributions serve as further building blocks towards an algorithmic theory of generative models in inverse problems.

1.3 Techniques

At a high level, our algorithms are standard. The primary novelty is in their applications to generative network models, and some aspects of their theoretical analysis.

Suppose that $G : \mathbb{R}^k \to \mathbb{R}^n$ is the generative network model under consideration. The cornerstone of our analysis is the assumption of an ε-*approximate* (Euclidean) projection oracle onto the

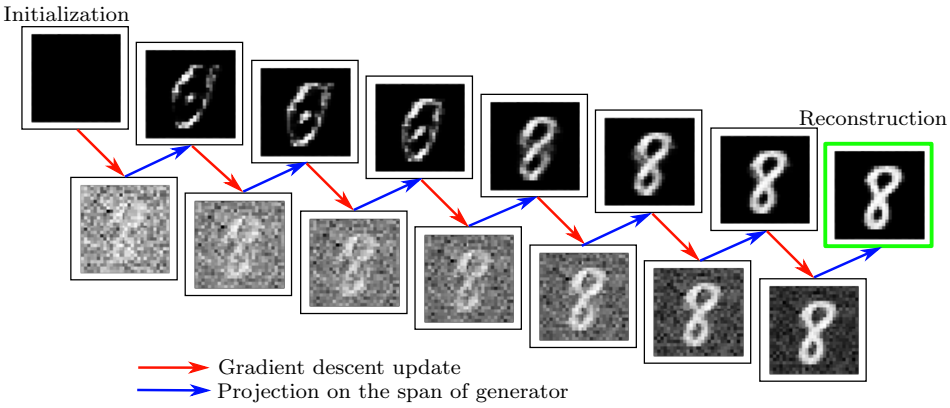


Figure 1.2 Illustration of our algorithm. Starting from a zero vector, we perform a gradient descent update step (red arrow) and projection step (blue arrow) alternatively to reach the final estimate.

range of G. In words, we pre-suppose the availability of a computational routine P_G that, given any vector $x \in \mathbb{R}^n$, can return a vector $x' \in \text{Range}(G)$ that approximately minimizes $\|x - x'\|_2^2$. The availability of this oracle, of course, depends on G and we comment on how to construct such oracles below in Section 5.

For a special case of linear inverse problems (and compressive sensing in particular), we assume such oracle to be simply a gradient descent routine minimizing the $\|x - x'\|_2^2$ over the latent variable z with $x' = G(z)$. Though this loss function is highly non-convex due to the presence of G, we find empirically that the gradient descent (implemented via back-propagation) works very well, and can be used as a projection oracle. Our procedure is depicted in Fig. 1.2. We choose a zero vector as our initial estimate (x_0), and in each iteration, we update our estimate by following the standard gradient descent update rule (red arrow in Fig. 1.2), followed by projection of the output onto the span of generator (G) (blue arrow in Fig. 1.2).

We support this specific PGD algorithm via a rigorous theoretical analysis. We show that the final estimate at the end of T iterations is an approximate reconstruction of the original signal x^*, with very small reconstruction error; moreover, under certain sufficiency conditions on the linear operator $\mathcal{A}$, PGD demonstrates linear convergence, meaning that $T = \log(1/\delta)$ is sufficient to achieve δ-accuracy. Further, we present a series of numerical results as validation of our approach.

We also provide a direct generalization of the above approach for nonlinear inverse problems. We call it $\varepsilon - PGD$. We analyze this generic algorithm to show a linear convergence by assuming that the objective function in (1.1) obeys the Restricted Strong Convexity/Smoothness assumptions [38]. With this assumption, proof of convergence follows from a straightforward modification of the proof given in [25]. Through our analysis, it is noticeable that the PGD algorithm for linear inverse problems is in fact a special case of the generalized $\varepsilon - PGD$.

The third algorithm (Myopic ε-PGD) is novel approach for handling model mismatch in the target. The main idea (following the lead from [12]) is to pose the target x^* as the superposition of two components: $x^* = G(z) + \nu$, where ν can be viewed as an "innovation" term that is s-sparse in some fixed, known basis B. The goal is now to recover both $G(z)$ and ν. This is reminiscent of the problem of source separation or signal demixing [33], and in our previous work [22, 44] we proposed greedy iterative algorithms for solving such demixing problems. We extend this work by proving a nonlinear extension, together with a new analysis, of the algorithm proposed in the work of [22].

CHAPTER 2. BACKGROUND AND RELATED WORK

2.1 Inverse problems

The study of solving inverse problems has a long history. As discussed above, the general approach to solve an ill-posed inverse problem of the form depicted in Fig. 1.1 is to assumes that the target signal/image obeys a *prior*. Older works mainly used hand-crafted signal priors to distinguish 'natural' signals from the infinite set of feasible solutions. The prior can be encoded in the form of either a constraint set (as in Eq. (1.1)) or an extra regularization penalty. Several works (including [14, 15, 48]) employ sparsity priors to solve *linear* inverse problems such as denoising, super-resolution and inpainting. In the works [17] and [1], sparse and redundant dictionaries are learned for image denoising, whereas in [9, 10, 41], total variation is used as a regularizer. Despite their successful practical and theoretical results, all such hand-designed priors often fail to restrict the solution space only to natural images, and it is easily possible to generate signals satisfying the prior but do not resemble natural data.

2.2 Neural network models

The last few years have witnessed the emergence of trained *neural networks* for solving such problems. The main idea is to eschew hand-crafting any priors, and instead *learn* an end-to-end mapping from the measurement space to the image space. This mapping is simulated via a deep neural network, whose weights are learned from a large dataset of input-output training examples [29]. The works [13, 26, 28, 34, 35, 47, 50] have used this approach to solve several types of (linear) inverse problems, and has met with considerable success. However, the major limitations are that a new network has to be trained for each new linear inverse problem; moreover, most of these methods lack concrete theoretical guarantees. An exception of this line of work is the powerful

framework of [40], which does *not* require retraining for each new problem; however, this too is not accompanied by theoretical analysis of statistical and computational costs.

2.3 Generative networks

A special class of neural networks that attempt to directly model the distribution of the input training samples are known as generative adversarial training networks, or GANs [19]. GANs have been shown to provide visually striking results [2, 4, 6, 53]. The use of GANs to solve linear inverse problems was advocated in [5]. Specifically, given (noisy) linear observations $y = Ax^* + e$ of a signal $x^* \in \mathbb{R}^n$, assuming that x^* belongs to the range of a generative network $G : \mathbb{R}^n \to \mathbb{R}^k$, this approach constructs the reconstructed estimate $\hat{x}$ as follows:

$$\hat{z} = \arg\min_{z \in \mathbb{R}^k} \|y - AG(z)\|_2^2, \quad \hat{x} = G(\hat{z})$$

If the observation matrix $A \in \mathbb{R}^{m \times n}$ comprises $m = O(kd \log n)$ i.i.d. Gaussian measurements, then together with regularity assumptions on the generative network, they prove that the solution $\hat{x}$ satisfies:

$$\|x^* - \hat{x}\|_2 \leq C\|e\|_2.$$

for some constant C that can be reliably upper-bounded. In particular, in the absence of noise the recovery of x^* is exact. However, there is no discussion of how computationally expensive this procedure is. Observe that the above minimization is highly non-convex (since for any reasonable neural network, G is a non-convex function) and possibly also non-smooth (if the neural network contains non-smooth activation functions, such as rectified linear units, or ReLUs). More recently, [46] improve upon the approach in [5] for solving more general nonlinear inverse problems (in particular, any inverse problem which has a computable derivative). Their approach involves simultaneously solving the inverse problem and training the network parameters; however, the treatment here is mostly empirical and a theoretical analysis is not provided.

Under similar statistical assumptions as [5], our previous work [43] provably establishes a linear convergence rate, provided that a projection oracle (on to the range of G) is available, but only

for the special case of compressive sensing. Our generalized result (Contribution II) extends this algorithm (and analysis) to more general nonlinear inverse problems.

More recently, [37] proposed a method that learns the a network-based projector for use in the PGD algorithm, making the projection step faster computationally. However, their theoretical result assumes the learned projector to be $\delta-$approximate, indicating that the effective training of the projector is crucial for the success of their method posing an additional challenge.

2.4 Model mismatch

A limitation of most generative network models is that they can only reproduce estimates that are within their range; adding more observations or tweaking algorithmic parameters are completely ineffective if a generative network model is presented with a target that is far away from the range of the model. To resolve this type of model mismatch, the authors of [12] propose to model the signal x^* as the superposition of two components: a "base" signal $u = G(z)$, and an "innovation" signal $v = B\nu$, where B is a known ortho-basis and ν is an l-sparse vector. In the context of compressive sensing, the authors of [12] solve a sparsity-regularized loss minimization problem:

$$(\hat{z}, \hat{v}) = \arg\min_{z,v} \left\| B^T v \right\|_1 + \lambda \|y - A(G(z) + v)\|_2^2.$$

and prove that the reconstructed estimate $\hat{x} = G(\hat{z}) + \hat{v}$ is close enough to x provided $m = O((k + l)d \log n)$ measurements are sufficient. However, as before, the algorithmic costs of solving the above problem are not discussed. Our third main result (Contribution III) proposes a new algorithm for dealing with model mismatches in generative network modeling, together with an analysis of its convergence and iteration complexity.

CHAPTER 3. MAIN ALGORITHMS AND ANALYSIS

Let us first establish some notational conventions. Below, $\|\cdot\|$ will denote the Euclidean norm unless explicitly specified. We use $O(\cdot)$-notation in several places in order to avoid duplication of constants.

We use $F(\cdot)$ to denote a (scalar) objective function.

3.1 Contribution I : PGD Algorithm for solving linear inverse problems using generative network

Let $\mathcal{S} \subseteq \mathbb{R}^n$ be the set of 'natural' images in data space with a vector $x^* \in \mathcal{S}$. We consider an ill-posed linear inverse problem (3.1) with the linear operator $\mathcal{A}(x) = Ax$, where A is a Gaussian random matrix. For simplicity, we do not consider the additive noise term.

$$y = Ax^*, \tag{3.1}$$

To solve for $\widehat{x}$ (estimate of x^*), we choose Euclidean measurement error as the loss function $F(\cdot)$ in Eqn. (1.1). Therefore, given y and A, we seek

$$\widehat{x} = \underset{x \in \mathcal{S}}{\arg\min} \|y - Ax\|^2. \tag{3.2}$$

3.1.1 Algorithm

Our algorithm is described in Alg. 1. We assume that our trained generator network (G) well approximates the high-dimensional probability distribution of the set $\mathcal{S}$. With this assumption, we limit our search for $\widehat{x}$ only to the range of the generator function ($G(z)$). The function G is assumed to be differentiable, and hence we use back-propagation for calculating the gradients of the loss functions involving G for gradient descent updates.

The optimization problem in Eqn. 3.2 is similar to a least squares estimation problem, and a typical approach to solve such problems is to use gradient descent. However, the candidate

Algorithm 1 PGD

Inputs: y, A, G, T, **Output:** $\widehat{x}$
$x_0 \leftarrow \mathbf{0}$
while $t < T$ **do**
    $w_t \leftarrow x_t + \eta A^T(y - Ax_t)$
    $x_{t+1} \leftarrow G\left(\arg\min_z \|w_t - G(z)\|\right)$
    $t \leftarrow t + 1$
$\widehat{x} \leftarrow x_T$

solutions obtained after each gradient descent update need not represent a 'natural' image and may not belong to set $\mathcal{S}$. We solve this limitation by projecting the candidate solution on the range of the generator function after each gradient descent update.

Thus, in each iteration of our proposed algorithm 1, two steps are performed in alternation: a gradient descent update step and a projection step. The first step is simply an application of a gradient descent update rule on the loss function $F(\cdot)$ with the learning rate η. In projection step, we minimize the projection loss by gradient descent updates with learning rate η_{in}:

$$P_G(w_t) \coloneqq G\left(\arg\min_z \|w_t - G(z)\|\right),$$

Though the projection loss function is highly non-convex due to the presence of G, we find empirically that the gradient descent (implemented via back-propagation) works very well. Thus, the gradient descent based minimization serves as a projection oracle in this case. In each of the T iterations, we run T_{in} gradient descent updates for calculating the projection. Therefore, $T \times T_{in}$ is the total number of gradient descent updates required in our approach.

3.1.2 Analysis

Drawing parallels with standard compressive sensing theory, in our case, we need to ensure that the difference vector of any two signals in the set $\mathcal{S}$ lies away from the nullspace of the matrix A. This condition is encoded via the S-REC (Set Restricted Eigenvalue Condition) defined in [5]. We slightly modify this condition and present it in the form of squared $l2$-norm :

Definition 1 *Let $\mathcal{S} \in \mathbb{R}^n$. A is $m \times n$ matrix. For parameters $\gamma > 0,\ \delta \geq 0$, matrix A is said to satisfy the S-REC$(\mathcal{S}, \gamma, \delta)$ if,*

$$\|A(x_1 - x_2)\|^2 \geq \gamma \|x_1 - x_2\|^2 - \delta,$$

for $\forall x_1, x_2 \in \mathcal{S}$.

Further, based on [18, 42], we propose the following theorem about the convergence of our algorithm:

Theorem 1 *Let $G : \mathbb{R}^k \to \mathbb{R}^n$ be a differentiable generator function with range $\mathcal{S}$. Let A be a random Gaussian matrix with $A_{i,j} \sim N(0, 1/m)$ such that it satisfies the S-REC$(\mathcal{S}, \gamma, \delta)$ with probability $1 - p$, and has $\|Av\| \leq \rho\|v\|$ for every $v \in \mathbb{R}^n$ with probability $1 - q$ with $\rho^2 \leq \gamma$. Then, for every vector $x^* \in \mathcal{S}$, the sequence (x_t) defined by the algorithm PGD [1] with $y = Ax^*$ converges to x^* with probability at least $1 - p - q$.*

Proof. $F(\cdot)$ is a squared error loss function as defined above. Then, we have:

$$\begin{aligned}
&F(x_{t+1}) - F(x_t) \\
&= \|Ax_{t+1}\|^2 - 2\langle y, Ax_{t+1}\rangle + 2\langle y, Ax_t\rangle - \|Ax_t\|^2, \\
&= \|Ax_{t+1} - Ax_t\|^2 + 2\langle x_t - x_{t+1}, A^T A(x^* - x_t)\rangle.
\end{aligned}$$

Substituting $y = Ax^*$ and rearranging yields,

$$2\langle x_t - x_{t+1}, A^T(y - Ax_t)\rangle = F(x_{t+1}) - F(x_t) - \|Ax_{t+1} - Ax_t\|^2. \tag{3.3}$$

Define:

$$w_t \coloneqq x_t + \eta A^T(y - Ax_t) = x_t + \eta A^T A(x^* - x_t)$$

Then, by definition of the projection operator P_G, the vector x_{t+1} is a better (or equally good) approximation to w as the true image x^*. Therefore, we have:

$$\|x_{t+1} - w_t\|^2 \leq \|x^* - w_t\|^2.$$

Substituting for w_t and expanding both sides, we get:

$$\|x_{t+1} - x_t\|^2 - 2\eta\langle x_{t+1} - x_t, A^T(y - Ax_t)\rangle \leq \|x^* - x_t\|^2 - 2\eta\langle x^* - x_t, A^T(y - Ax_t)\rangle.$$

Substituting $y = Ax^*$ and rearranging yields,

$$2\langle x_t - x_{t+1}, A^T(y - Ax_t)\rangle \leq \frac{1}{\eta}\|x^* - x_t\|^2 - \frac{1}{\eta}\|x_{t+1} - x_t\|^2 - 2F(x_t). \tag{3.4}$$

We now use 3.3 and 3.4 to obtain,

$$F(x_{t+1}) \leq \frac{1}{\eta}\|x^* - x_t\|^2 - F(x_t) - \left(\frac{1}{\eta}\|x_{t+1} - x_t\|^2 - \|Ax_{t+1} - Ax_t\|^2\right). \tag{3.5}$$

Now, from the S-REC, we know that,

$$\|A(x_1 - x_2)\|^2 \geq \gamma\|x_1 - x_2\|^2 - \delta.$$

As x^*, x_t and x_{t+1} are 'natural' vectors,

$$\frac{1}{\eta}\|x^* - x_t\|^2 \leq \frac{1}{\eta\gamma}\|y - Ax_t\|^2 + \frac{\delta}{\eta\gamma}. \tag{3.6}$$

Substituting 3.6 in 3.5,

$$F(x_{t+1}) \leq \left(\frac{1}{\eta\gamma} - 1\right)F(x_t) - \left(\frac{1}{\eta}\|x_{t+1} - x_t\|^2 - \|Ax_{t+1} - Ax_t\|^2\right) + \frac{\delta}{\eta\gamma}.$$

From our assumption that $\|Av\| \leq \rho\|v\|, \forall v \in \mathbb{R}^n$ with probability $1 - q$, we write:

$$\|Ax_{t+1} - Ax_t\|^2 \leq \rho^2\|x_{t+1} - x_t\|^2,$$

$$\|Ax_{t+1} - Ax_t\|^2 - \frac{1}{\eta}\|x_{t+1} - x_t\|^2 \leq \left(\rho^2 - \frac{1}{\eta}\right)\|x_{t+1} - x_t\|^2.$$

Let us choose learning rate(η) such that $\frac{1}{2\gamma} < \eta < \frac{1}{\gamma}$. We also have $\rho^2 \leq \gamma$. Combining both, we get $\rho^2 < \frac{1}{\eta}$, which makes the L.H.S. in the above equation negative. Therefore,

$$F(x_{t+1}) \leq \left(\frac{1}{\eta\gamma} - 1\right)F(x_t) + \frac{\delta}{\eta\gamma},$$

where δ is inversely proportional to the number of measurements m [5]. Provided sufficient number of measurements, δ is small enough and can be ignored. Also, $\frac{1}{2\gamma} < \eta < \frac{1}{\gamma}$ yields,

$$0 < \left(\frac{1}{\eta\gamma} - 1\right) < 1.$$

Hence,

$$F(x_{t+1}) \leq \alpha F(x_t);\ 0 < \alpha < 1, \tag{3.7}$$

with probability at least $1 - p - q$. ■

3.2 Contribution II: ε-PGD algorithm for solving nonlinear inverse problems using generative networks

We now present the generic version of PGD algorithm suitable for large class of nonlinear inverse problems by generalizing for the loss function $F(\cdot)$ and the projection oracle P_G.

We denote the $F(\cdot)$ to be a scalar function with continuous gradient, and assume that F has a continuous gradient $\nabla F = \left(\frac{\partial F}{\partial x_i}\right)_{i=1}^{n}$ which can be evaluated at any point $x \in \mathbb{R}^n$.

Recall that we wish to solve the problem:

$$\begin{aligned} \widehat{x} &= \arg\min\ F(x), \\ &\text{s. t.} \quad x \in \text{Range}(G), \end{aligned} \tag{3.8}$$

where G is a generative network. To do so, we now employ a generalized version of *projected gradient descent* algorithm using the ε-approximate projection oracle for G. The algorithm is described in Alg. 2.

We define the ε-approximate projection oracle P_G as,

Definition 2 (Approximate projection) *A function $P_G : \mathbb{R}^n \to Range(G)$ is an ε-approximate projection oracle if for all $x \in \mathbb{R}^n$, $P_G(x)$ obeys:*

$$\|x - P_G(x)\|_2^2 \leq \min_{z \in \mathbb{R}^k} \|x - G(z)\|_2^2 + \varepsilon.$$

We will assume that for any given generative network G of interest, such a function P_G exists and is computationally tractable[1]*. Here, $\varepsilon > 0$ is a parameter that is known* a priori.

[1] *This may be a very strong assumption, but at the moment we do not know how to relax this. Indeed, the computational complexity of our proposed algorithms are directly proportional to the complexity of such a projection oracle.*

In contrast to our previous analysis, here we introduce more general restriction conditions on the $F(\cdot)$:

Definition 3 (Restricted Strong Convexity/Smoothness) *Assume that F satisfies $\forall x, y \in S$:*

$$\frac{\alpha}{2}\|x-y\|_2^2 \leq F(y) - F(x) - \langle \nabla F(x), y - x\rangle \leq \frac{\beta}{2}\|x-y\|_2^2.$$

for positive constants α, β.

This assumption is by now standard; see [25, 38] for in-depth discussions. This means that the objective function is strongly convex / strongly smooth along certain directions in the parameter space (in particular, those restricted to the set S of interest). The parameter $\alpha > 0$ is called the restricted strong convexity (RSC) constant, while the parameter $\beta > 0$ is called the restricted strong smoothness (RSS) constant. Clearly, $\beta \geq \alpha$. In fact, throughout in this work, we assume that $1 \leq \frac{\beta}{\alpha} < 2$, which is a fairly stringent assumption but again, one that we do not know at the moment how to relax.

Definition 4 (Incoherence) *A basis B and Range(G) are called μ-incoherent if for all $u, u' \in$ Range(G) and all $v, v' \in$ Span(B), we have:*

$$|\langle u - u', v - v'\rangle| \leq \mu \big\|u - u'\big\|_2 \big\|v - v'\big\|_2.$$

for some parameter $0 < \mu < 1$.

Remark 1 *In addition to the above, we will make the following assumptions in order to aid the analysis. Below, γ and Δ are positive constants.*

- $\|\nabla F(x^*)\|_2 \leq \gamma$ *(gradient at the minimizer is small).*
- *diam*(*Range*(G)) $= \Delta$ *(range of G is compact).*
- $\gamma\Delta \leq O(\varepsilon)$.

We obtain the following theoretical result:

Algorithm 2 ε-PGD

Inputs: y, T, ∇; **Output:** $\widehat{x}$
$x_0 \leftarrow \mathbf{0}$
while $t < T$ **do**
    $w_t \leftarrow x_t - \eta \nabla F(x_t)$
    $x_{t+1} \leftarrow P_G(w_t)$
    $t \leftarrow t + 1$
$\widehat{x} \leftarrow x_T$

Theorem 2 *If F satisfies RSC/RSS over Range(G) with constants α and β, then ε-PGD (Alg. 2) convergences linearly up to a ball of radius $O(\gamma\Delta) \approx O(\varepsilon)$.*

$$F(x_{t+1}) - F(x^*) \leq \left(\frac{\beta}{\alpha} - 1\right)(F(x_t) - F(x^*)) + O(\varepsilon)\,.$$

Proof. The proof is a minor modification of that in [25]. For simplicity we will assume that $\|\cdot\|$ refers to the Euclidean norm. Let us suppose that the step size $\eta = \frac{1}{\beta}$. Define

$$w_t = x_t - \eta \nabla F(x_t).$$

By invoking RSS, we get:

$$\begin{aligned}
&F(x_{t+1}) - F(x_t) \\
&\leq \langle \nabla F(x_t), x_{t+1} - x_t \rangle + \frac{\beta}{2}\|x_{t+1} - x_t\|^2 \\
&= \frac{1}{\eta}\langle x_t - w_t, x_{t+1} - x_t \rangle + \frac{\beta}{2}\|x_{t+1} - x_t\|^2 \\
&= \frac{\beta}{2}\left(\|x_{t+1} - x_t\|^2 + 2\langle x_t - w_t, x_{t+1} - x_t\rangle + \|x_t - w_t\|^2\right) \\
&\quad - \frac{\beta}{2}\|x_t - w_t\|^2 \\
&= \frac{\beta}{2}\left(\|x_{t+1} - w_t\|^2 - \|x_t - w_t\|^2\right),
\end{aligned}$$

where the last few steps are consequences of straightforward algebraic manipulation.

Now, since x_{t+1} is an ε-approximate projection of w_t onto Range(G) and $x^* \in$ Range(G), we have:

$$\|x_{t+1} - w_t\|^2 \leq \|x^* - w_t\|^2 + \varepsilon.$$

Therefore, we get:

$$\begin{aligned}
&F(x_{t+1}) - F(x_t) \\
&\leq \frac{\beta}{2}\left(\|x^* - w_t\|^2 - \|x_t - w_t\|^2\right) + \frac{\beta\varepsilon}{2} \\
&= \frac{\beta}{2}\left(\|x^* - x_t + \eta\nabla F(x_t)\|^2 - \|\eta\nabla F(x_t)\|^2\right) + \frac{\beta\varepsilon}{2} \\
&= \frac{\beta}{2}\left(\|x^* - x_t\|^2 + 2\eta\langle x^* - x_t, \nabla F(x_t)\rangle\right) + \frac{\beta\varepsilon}{2} \\
&= \frac{\beta}{2}\|x^* - x_t\|^2 + \langle x^* - x_t, \nabla F(x_t)\rangle + \frac{\beta\varepsilon}{2}.
\end{aligned}$$

However, due to RSC, we have:

$$\begin{aligned}
\frac{\alpha}{2}\|x^* - x_t\|^2 &\leq F(x^*) - F(x_t) - \langle x^* - x_t, \nabla F(x_t)\rangle, \\
\langle x^* - x_t, \nabla F(x_t)\rangle &\leq F(x^*) - F(x_t) - \frac{\alpha}{2}\|x^* - x_t\|^2.
\end{aligned}$$

Therefore,

$$\begin{aligned}
&F(x_{t+1}) - F(x_t) \\
&\leq \frac{\beta - \alpha}{2}\|x^* - x_t\|^2 + F(x^*) - F(x_t) + \frac{\beta\varepsilon}{2} \\
&\leq \frac{\beta - \alpha}{2} \cdot \frac{2}{\alpha}\left(F(x_t) - F(x^*) - \langle x_t - x^*, \nabla F(x^*)\rangle\right) \\
&\quad + F(x^*) - F(x_t) + \frac{\beta\varepsilon}{2} \\
&\leq \left(2 - \frac{\beta}{\alpha}\right)\left(F(x^*) - F(x_t)\right) + \frac{\beta - \alpha}{\alpha}\gamma\Delta + \frac{\beta\varepsilon}{2},
\end{aligned}$$

where the last inequality follows from Cauchy-Schwartz and the assumptions on $\|\nabla F(x^*)\|$ and the diameter of Range(G). Further, by assumption, $\gamma\Delta \leq O(\varepsilon)$. Rearranging terms, we get:

$$F(x_{t+1}) - F(x^*) \leq \left(\frac{\beta}{\alpha} - 1\right)\left(F(x_t) - F(x^*)\right) + C\varepsilon.$$

for some constant $C > 0$. ■

This theorem asserts that the distance between the objective function at any iteration to the optimum *decreases by a constant factor* in every iteration. (The decay factor is $\frac{\beta}{\alpha} - 1$, which by assumption is a number between 0 and 1). Therefore, we immediately obtain linear convergence of ε-PGD up to a ball of radius $O(\varepsilon)$:

Corollary 1 *After* $T = O(\log \frac{F(x_0)-F(x^*)}{\varepsilon})$ *iterations,* $F(x_T) \leq F(x^*) + O(\varepsilon)$.

Therefore, the overall running time can be bounded as follows:

$$\text{Runtime} \leq (T_{\varepsilon-\text{PROJ}} + T_{\nabla}) \times \log(1/\varepsilon).$$

It is noticeable that the analysis of PGD algorithm for linear problem is a special case of the generalized analysis given by Theorem 2. That is because once we set the $F(\cdot)$ as defined in Eq. (3.2), the RSC for $F(\cdot)$ can be obtained through the S-REC condition from Eq. (3.3). Similarly, we can use the upper bound of the spectral norm for the Gaussian matrix A to obtain RSS for $F(\cdot)$.

In Sec. 4, we provide empirical results for solving nonlinear inverse problems using Alg. 2. Specifically, we consider two nonlinear forward models: a sinusoidal model with $\mathcal{A}(x^*) = Ax^* + sin(Ax^*)$; and a sigmoid model with $\mathcal{A}(x^*) = sigmoid(Ax^*) = \frac{1}{1+\exp(-Ax^*)}$. While we use l_2-loss as a loss function in the case of sinusoidal model, for sigmoid nonlinearity, we use a loss function specified as:

$$F(x) = \frac{1}{m}\sum_{i=1}^{m}\left(\Theta(a_i^T x) - y_i a_i^T x\right),$$

where, $\Theta(\cdot)$ is integral of $\mathcal{A}(\cdot)$, and a_i represents the rows of the measurement matrix A. The gradient of the loss can be calculated in closed form:

$$\nabla F(x^*) = \frac{1}{m}A^T(sigmoid(Ax) - y). \tag{3.9}$$

Such choice of the loss function is inspired by GLM and SIM estimation in the statistics literature [36]. [45] also advocates the usage of such loss function.

3.3 Contribution III: Addressing signal model mismatch

We now generalize the ε-PGD algorithm to handle situations involving signal model mismatch. Assume that the target signal can be decomposed as:

$$x^* = G(z) + v,$$

Algorithm 3 MYOPIC ε-PGD

Inputs: y, T, ∇; **Output:** $\widehat{x}$
$x_0, u_0, v_0 \leftarrow \mathbf{0}$
while $t < T$ **do**
    $u_{t+1} = P_G(u_t - \eta \nabla_x F(x_t))$
    $v_{t+1} = \text{Thresh}_{B,l}(v_t - \eta \nabla_x F(x_t))$
    $x_{t+1} = u_{t+1} + \nu_{t+1}$
    $t \leftarrow t + 1$
$\widehat{x} \leftarrow x_T$

where $\left\|B^T v\right\|_0 \leq l \ll n$ for some ortho-basis B.

For this model, we attempt to solve a (slightly) different optimization problem:

$$\widehat{x} = \arg\min \; F(x), \tag{3.10}$$

$$\text{s. t.} \quad x = G(z) + v, ,$$

$$\left\|B^T v\right\|_0 \leq l. \tag{3.11}$$

We propose a new algorithm to solve this problem that we call *Myoptic ε-PGD*. This algorithm is given in Alg. 3[2].

Theorem 3 *Let $\oplus$ denote the Minkowski sum. If F satisfies RSC/RSS over Range$(G) \oplus$ Span(B) with constants α and β, and if we assume μ-incoherence between B and Range(G), we have:*

$$F(x_{t+1}) - F(x^*) \leq \left(\frac{2 - \frac{\beta}{\alpha}\frac{1-2.5\mu}{1-\mu}}{1 - \frac{\beta}{2\alpha}\frac{\mu}{1-\mu}} \right) (F(x_t) - F(x^*)) + O(\varepsilon)\,.$$

Proof. We will generalize the proof technique of [22]. We first define some auxiliary variables that help us with the proof. Let:

$$w_t = x_t - \eta \nabla F(x_t),$$

$$w_t^u = u_t - \eta \nabla F(x_t),$$

$$w_t^v = v_t - \eta \nabla F(x_t).$$

[2]The algorithm is a variant of block-coordinate descent, except that the block updates share the same gradient term.

and let $x^* = u^* + v^*$ be the minimizer that we seek. As above, by invoking RSS and with some algebra, we obtain:

$$F(x_{t+1}) - F(x^*) \leq \frac{\beta}{2}\left(\|x_{t+1} - w_t\|^2 - \|x_t - w_t\|^2\right), \tag{3.12}$$

However, by definition,

$$x_{t+1} = u_{t+1} + v_{t+1},$$
$$x_t = u_t + v_t.$$

Therefore,

$$\begin{aligned}
&\|x_{t+1} - w_t\|^2 \\
&= \|u_{t+1} - (u_t - \eta\nabla F(x_t)) + v_{t+1} - (v_t - \eta\nabla F(x_t)) + \eta\nabla F(x_t)\|^2 \\
&= \|u_{t+1} - (u_t - \eta\nabla F(x_t))\|^2 + \|\eta\nabla F(x_t)\|^2 + \|v_{t+1} - (v_t - \eta\nabla F(x_t))\|^2 \\
&\quad + 2\langle u_{t+1} - (u_t - \eta\nabla F(x_t)), \eta\nabla F(x_t)\rangle + 2\langle v_{t+1} - (v_t - \eta\nabla F(x_t)), \eta\nabla F(x_t)\rangle \\
&\quad + 2\langle u_{t+1} - (u_t - \eta\nabla F(x_t)), v_{t+1} - (v_t - \eta\nabla F(x_t))\rangle.
\end{aligned}$$

But u_{t+1} is an ε-projection of w_t^u and u^* is in the range of G, we have:

$$\|u_{t+1} - w_t^u\|^2 \leq \|u^* - w_t^u\|^2 + \varepsilon.$$

Similarly, since v_{t+1} is an l-sparse thresholded version of w_t^v, we have:

$$\|v_{t+1} - w_t^v\|^2 \leq \|v^* - w_t^v\|^2.$$

Plugging in these two upper bounds, we get:

$$\begin{aligned}
&\|x_{t+1} - w_t\|^2 \\
&\leq \|u^* - (u_t - \eta\nabla F(x_t))\|^2 + \varepsilon + \|\eta\nabla F(x_t)\|^2 + \|v^* - (v_t - \eta\nabla F(x_t))\|^2 \\
&\quad + 2\langle u_{t+1} - (u_t - \eta\nabla F(x_t)), \eta\nabla F(x_t)\rangle + 2\langle v_{t+1} - (v_t - \eta\nabla F(x_t)), \eta\nabla F(x_t)\rangle \\
&\quad + 2\langle u_{t+1} - (u_t - \eta\nabla F(x_t)), v_{t+1} - (v_t - \eta\nabla F(x_t))\rangle.
\end{aligned}$$

Expanding squares and cancelling (several) terms, the right hand side of the above inequality can be simplified to obtain:

$$\begin{aligned}\|x_{t+1} - w_t\|^2 &\leq \|u^* + v^* - w_t\|^2 + \varepsilon + 2\langle u_{t+1} - u_t, v_{t+1} - v_t\rangle - 2\langle u^* - u_t, v^* - v_t\rangle \\ &= \|x^* - w_t\|^2 + \varepsilon + 2\langle u_{t+1} - u_t, v_{t+1} - v_t\rangle - 2\langle u^* - u_t, v^* - v_t\rangle.\end{aligned}$$

Plugging this into (3.12), we get:

$$\begin{aligned}F(x_{t+1}) - F(x^*) \leq \underbrace{\frac{\beta}{2}\left(\|x^* - w_t\|^2 - \|x_t - w_t\|^2\right)}_{\mathbb{T}_1} \\ + \underbrace{\beta\left(\langle u_{t+1} - u_t, v_{t+1} - v_t\rangle - 2\langle u^* - u_t, v^* - v_t\rangle\right)}_{\mathbb{T}_2} + \frac{\beta\varepsilon}{2}.\end{aligned}$$

We already know how to bound the first term $\mathbb{T}_1$, using an identical argument as in the proof of Theorem 2. We get:

$$\mathbb{T}_1 \leq \left(2 - \frac{\beta}{\alpha}\right)(F(x^*) - F(x_t)) + \frac{\beta - \alpha}{\alpha}\gamma\Delta.$$

The second term $\mathbb{T}_2$ can be bounded as follows. First, observe that,

$$\begin{aligned}|\langle u_{t+1} - u_t, v_{t+1}\rangle| &\leq \mu\|u_{t+1} - u_t\|\|v_{t+1} - v_t\| \\ &\leq \frac{\mu}{2}\left(\|u_{t+1} - u_t\|^2 + \|v_{t+1} - v_t\|^2\right) \\ &\leq \frac{\mu}{2}\left(\|u_{t+1} + v_{t+1} - u_t - v_t\|^2\right) + \mu|\langle u_{t+1} - u_t, v_{t+1} - v_t\rangle|.\end{aligned}$$

This gives us the following inequalities:

$$\begin{aligned}|\langle u_{t+1} - u_t, v_{t+1}\rangle| &\leq \frac{\mu}{2(1-\mu)}\|x_{t+1} - x_t\|^2 \\ &= \frac{\mu}{2(1-\mu)}\Big(\|x_{t+1} - x^*\|^2 + \|x_t - x^*\|^2 + 2|\langle x_{t+1} - x^*, x_t - x^*\rangle|\Big) \\ &\leq \frac{\mu}{1-\mu}\left(\|x_{t+1} - x^*\|^2 + \|x_t - x^*\|^2\right).\end{aligned}$$

Similarly,

$$\begin{aligned}|\langle u^* - u_t, v^* - v_t\rangle| &\leq \mu\|u^* - u_t\|\|v^* - v_t\| \\ &\leq \frac{\mu}{2}\left(\|u^* - u_t\|^2 + \|v^* - v_t\|\right) \\ &= \frac{\mu}{2}\left(\|u^* + v^* - u_t - v_t\|^2\right) + \mu|\langle u^* - u_t, v^* - v_t\rangle|,\end{aligned}$$

which gives:

$$|\langle u^* - u_t, v^* - v_t \rangle| \leq \frac{\mu}{2(1-\mu)} \|x^* - x_t\|^2.$$

Combining, we get:

$$\mathbb{T}_2 \leq \frac{\beta\mu}{2(1-\mu)} \left(3\|x^* - x_t\|^2 + \|x^* - x_{t+1}\|^2 \right).$$

Moreover, by invoking RSC and Cauchy-Schwartz (similar to the proof of Theorem 2), we have:

$$\|x^* - x_t\|^2 \leq \frac{1}{\alpha} \left(F(x_t) - F(x^*) \right) + O(\varepsilon),$$
$$\|x^* - x_{t+1}\|^2 \leq \frac{1}{\alpha} \left(F(x_{t+1}) - F(x^*) \right) + O(\varepsilon).$$

Therefore we obtain the upper bound on $\mathbb{T}_2$:

$$\mathbb{T}_2 \leq \frac{3\beta\mu}{2\alpha(1-\mu)} \left(F(x_t) - F(x^*) \right) + \frac{\beta\mu}{2\alpha(1-\mu)} \left(F(x_{t+1}) - F(x^*) \right) + C'\varepsilon.$$

Plugging in the upper bounds on $\mathbb{T}_1$ and $\mathbb{T}_2$ and re-arranging terms, we get:

$$\left(1 - \frac{\beta\mu}{2\alpha(1-\mu)} \right) \left(F(x_{t+1}) - F(x^*) \right) \leq \left(2 - \frac{\beta}{\alpha} + \frac{3\beta\mu}{2\alpha(1-\mu)} \right) \left(F(x_t) - F(x^*) \right) + C'\varepsilon,$$

which leads to the desired result. ■

CHAPTER 4. MODELS AND EXPERIMENTS

In this chapter, we describe our experimental setup and report the performance of our algorithms.

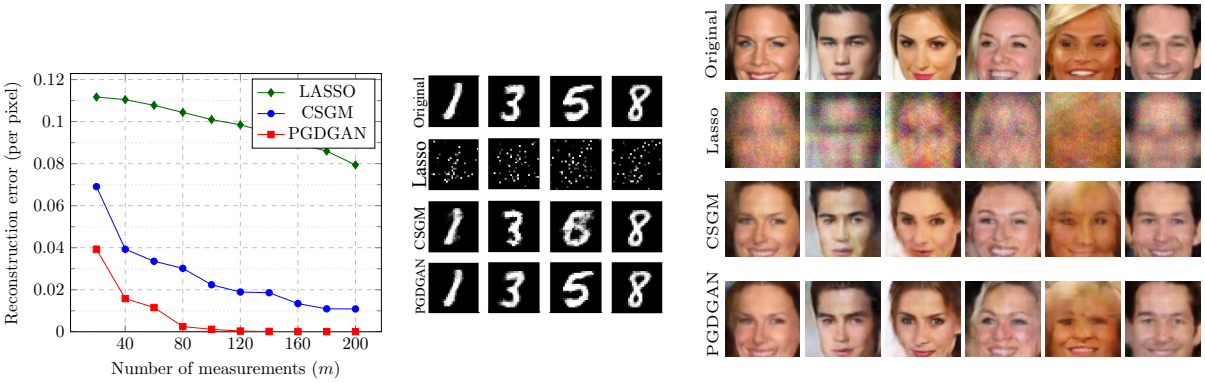
4.1 Solving linear inverse problems using PGD algorithm

We use two different GAN architectures and two different datasets in our experiments to show that our approach [1] can work with variety of GAN architectures and datasets.

In our experiments, we choose the entries of the matrix A independently from a Gaussian distribution with zero mean and $1/m$ standard deviation. We ignore the presence of noise; however, our experiments can be replicated with additive Gaussian noise. We use a gradient descent optimizer keeping the total number of update steps ($T \times T_{in}$) fixed for both algorithms and doesn't allow random restarts.

In the first experiment, we use a very simple GAN model trained on the MNIST dataset, which is collection of $60,000$ handwritten digit images, each of size 28×28 [30]. In our GAN, both the generator and the discriminator are fully-connected neural networks with only one hidden layer. The generator consists of 20 input neurons, 200 hidden-layer neurons and 784 output neurons, while the discriminator consists of 784 input neurons, 128 hidden layer neurons and 1 output neuron. The size of the latent space is set to $k = 20$, i.e., the input to our generator is a standard normal vector $z \in R^{20}$. We train the GAN using the method described in [19]. We use the Adam optimizer [27] with learning rate 0.001 and mini-batch size 128 for the training.

We test the MNIST GAN with 10 images taken from the span of generator to get rid of the representation error, and provide both quantitative and qualitative results. For PGD-GAN, because of the zero initialization, a high learning rate is required to get a meaningful output before passing it to the projection step. Therefore, we choose $\eta \geq 0.5$. The parameter η_{in} is set to 0.01 with $T = 15$

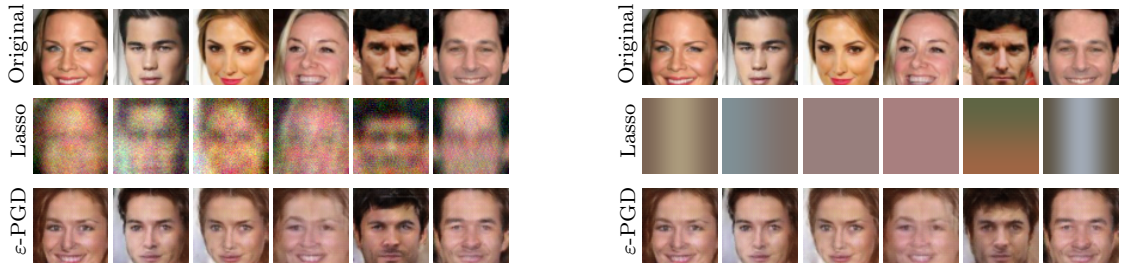


(a) (b) (c)

Figure 4.1 (a) Comparison of our algorithm (Alg. 1) with CSGM [5] and Lasso on MNIST; (b) Reconstruction results with $m = 100$ measurements; (c) Reconstruction results on celebA dataset with $m = 1000$ measurements.

and $T_{in} = 200$. Thus, the total number of update steps is fixed to 3000. Similarly, the algorithm of [5] is tested with 3000 updates and $\eta = 0.01$. For comparison, we use the reconstruction error $= \|\widehat{x} - x^*\|^2$. In Fig. 4.1(a), we show the reconstruction error comparisons for increasing values of number of measurements. We observe that our algorithm performs better than the other two methods. Also, as the input images are chosen from the span of the generator itself, it is possible to get close to zero error with only 100 measurements. Fig. 4.1(b) depicts reconstruction results for selected MNIST images.

The second set of our experiments are performed on a Deep Convolutional GAN (DCGAN) trained on the celebA dataset, which contains more than $200,000$ face images of celebrities [32]. We use a pre-trained DCGAN model, which was made available by [5]. Thus, the details of the model and training are the same as described in [5]. The dimension of latent space for DCGAN is $k = 100$. We report the results on a held out test dataset, unseen by the GAN at the time of training. Total number of updates is set to 1000, with $T = 10$ and $T_{in} = 100$. Learning rates for PGD-GAN are set as $\eta = 0.5$ and $\eta_{in} = 0.1$. The algorithm of [5] is run with $\eta = 0.1$ and 1000 update steps. Image reconstruction results from $m = 1000$ measurements with our algorithm are



(a) (b)

Figure 4.2 Comparison of our algorithm (Alg. 2) with Lasso for non-linear forward models (a) with $\mathcal{A}(x^*) = Ax^* + sin(Ax^*)$; (b) with $\mathcal{A}(x^*) = sigmoid(Ax^*)$. Reconstruction results are on celebA dataset with $m = 1000$ measurements.

displayed in Fig. 4.1(c). We observe that our algorithm produces better reconstructions compared to the other baselines.

4.2 Experiments for non-linear inverse problems

We extend our experiments to nonlinear models to depict the performance of our algorithm as described in Sec. 3.2. We present image reconstructions from the measurements obtained using two non-linear forward models: sinusoidal model with $\mathcal{A}(x^*) = Ax^* + sin(Ax^*)$, and sigmoid model with $\mathcal{A}(x^*) = sigmoid(Ax^*)$. Similar to linear case, these experiments are performed using a DCGAN trained on celebA dataset.

For sinusoidal case, we use l_2-loss as the loss function, therefore the reconstruction algorithm uses the gradient updates similar to the linear case. However, as explained in Sec. 3.2, a different loss function is used for sigmoid case. The gradient updates for the reconstruction from the sigmoid measurements are calculated using the Eqn. 3.9. The learning rate (η) for the gradient descent updates are tuned accordingly in both the cases. We set $m = 1000$, $T_{in} = 100$, and $T = 20$ for our experiments. For comparison, we calculate the reconstructions using the Lasso with DCT. It is evident that our algorithm produces superior reconstructions as depicted in Fig. 4.2.

CHAPTER 5. CONCLUSION

We provide some concluding remarks and potential directions for future work.

Our contributions in this work are primarily theoretical, and we also have explored the practical benefits of our approach in the context of linear inverse problems such as compressive sensing. Our algorithms proposed in this work are generic, and can be used to solve a variety of *nonlinear* inverse problems.

The main algorithmic message of our work is to show that solving a variety of nonlinear inverse problems using a generative network model can be reduced to performing a set of ε-projections onto the range of the network model. This can be challenging in general; for most interesting generative networks, this itself is a nonconvex problem, and potentially hard. However, recent work by [20, 21] have studied special cases where this type of projection can be tractable; in particular, for certain neural networks satisfying certain randomness conditions, one can solve the projection problem using a variation of gradient descent (which is more or less what all approaches employ in practice). Studying the landscape of such projection problems is an interesting direction of future research.

We make several assumptions to enable our analysis. Some of them (for example, restricted strong convexity/smoothness; incoherence) are standard analysis tools and are common in the high-dimensional statistics and compressive sensing literature. However, in order to be applicable, they need to be verified for specific problems. A broader characterization of problems that *do* satisfy these assumptions will be of great interest.

BIBLIOGRAPHY

[1] Aharon, M., Elad, M., and Bruckstein, A. (2006). *rmk*-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Processing*, 54(11):4311–4322.

[2] Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In *Proc. Int. Conf. Machine Learning*.

[3] Baraniuk, R., Cevher, V., Duarte, M., and Hegde, C. (2010). Model-based compressive sensing. *IEEE Trans. Inform. Theory*, 56(4):1982–2001.

[4] Berthelot, D., Schumm, T., and Metz, L. (2017). Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*.

[5] Bora, A., Jalal, A., Price, E., and Dimakis, A. (2017). Compressed sensing using generative models. *Proc. Int. Conf. Machine Learning*.

[6] Brock, A., Donahue, J., and Simonyan, K. (2018). Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*.

[7] Brock, A., Lim, T., Ritchie, J., and Weston, N. (2016). Neural photo editing with introspective adversarial networks. *arXiv preprint arXiv:1609.07093*.

[8] Candès, E. et al. (2006). Compressive sampling. In *Proc. of the intl. congress of math.*, volume 3, pages 1433–1452. Madrid, Spain.

[9] Chambolle, A. (2004). An algorithm for total variation minimization and applications. *Jour. of Math. imaging and vision*, 20(1):89–97.

[10] Chan, T., Shen, J., and Zhou, H. (2006). Total variation wavelet inpainting. *Jour. of Math. imaging and Vision*, 25(1):107–125.

[11] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Proc. Adv. in Neural Processing Systems (NIPS)*, pages 2172–2180.

[12] Dhar, M., Grover, A., and Ermon, S. (2018). Modeling sparse deviations for compressed sensing using generative models. In *Proc. Int. Conf. Machine Learning*.

[13] Dong, C., Loy, C., He, K., and Tang, X. (2016). Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Machine Intell.*, 38(2):295–307.

[14] Dong, W., Zhang, L., Shi, G., and Wu, X. (2011). Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE Trans. Image Processing*, 20(7):1838–1857.

[15] Donoho, D. (1995). De-noising by soft-thresholding. *IEEE Trans. Inform. Theory*, 41(3):613–627.

[16] Duarte, M., Hegde, C., Cevher, V., and Baraniuk, R. (2009). Recovery of compressible signals from unions of subspaces. In *Proc. IEEE Conf. Inform. Science and Systems (CISS)*.

[17] Elad, M. and Aharon, M. (2006). Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Processing*, 15(12):3736–3745.

[18] Foucart, S. and Rauhut, H. (2013). *A mathematical introduction to compressive sensing*, volume 1. Springer.

[19] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Proc. Adv. in Neural Processing Systems (NIPS)*, pages 2672–2680.

[20] Hand, P. and Voroninski, V. (2017). Global guarantees for enforcing deep generative priors by empirical risk. *COLT*.

[21] Heckel, R., Huang, W., Hand, P., and Voroninski, V. (2018). Deep denoising: Rate-optimal recovery of structured signals with a deep prior. *arXiv preprint arXiv:1805.08855*.

[22] Hegde, C. and Baraniuk, R. (2012a). Signal recovery on incoherent manifolds. *IEEE Trans. Inform. Theory*, 58(12):7204–7214.

[23] Hegde, C. and Baraniuk, R. (2012b). SPIN: Iterative signal recovery on incoherent manifolds. In *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*.

[24] Hegde, C., Indyk, P., and Schmidt, L. (2015). Fast algorithms for structured sparsity. *Bulletin of the EATCS*, 1(117):197–228.

[25] Jain, P. and Kar, P. (2017). Non-convex optimization for machine learning. *Foundations and Trends in Machine Learning*, 10(3-4):142–336.

[26] Kim, J., Kwon Lee, J., and Mu Lee, K. (2016). Accurate image super-resolution using very deep convolutional networks. In *Proc. IEEE Conf. Comp. Vision and Pattern Recog. (CVPR)*, pages 1646–1654.

[27] Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[28] Kulkarni, K., Lohit, S., Turaga, P., Kerviche, R., and Ashok, A. (2016). Reconnet: Non-iterative reconstruction of images from compressively sensed measurements. In *Proc. IEEE Conf. Comp. Vision and Pattern Recog. (CVPR)*, pages 449–458.

[29] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

[30] LeCun, Y., Bottou, L. o., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11):2278–2324.

[31] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. *Proc. IEEE Conf. Comp. Vision and Pattern Recog. (CVPR)*, pages 105–114.

[32] Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proc. of the IEEE Intl. Conf. on Comp. Vision*, pages 3730–3738.

[33] McCoy, M. and Tropp, J. (2014). Sharp recovery bounds for convex demixing, with applications. *Foundations of Comp. Math.*, 14(3):503–567.

[34] Mousavi, A. and Baraniuk, R. (2017). Learning to invert: Signal recovery via deep convolutional networks. *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing (ICASSP)*.

[35] Mousavi, A., Patel, A., and Baraniuk, R. (2015). A deep learning approach to structured signal recovery. In *Proc. Allerton Conf. Communication, Control, and Computing*, pages 1336–1343.

[36] Negahban, S., Yu, B., Wainwright, M., and Ravikumar, P. (2009). A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. In *Proc. Adv. in Neural Processing Systems (NIPS)*, pages 1348–1356.

[37] Raj, A., Li, Y., and Bresler, Y. (2019). Gan-based projector for faster recovery in compressed sensing with convergence guarantees. *arXiv preprint arXiv:1902.09698*.

[38] Raskutti, G., Wainwright, M. J., and Yu, B. (2010). Restricted eigenvalue properties for correlated gaussian designs. *J. Machine Learning Research*, 11(Aug):2241–2259.

[39] Ravishankar, S. and Bresler, Y. (2013). Learning sparsifying transforms. *IEEE Trans. Signal Processing*, 61(5):1072–1086.

[40] Rick Chang, J., Li, C., Poczos, B., Vijaya Kumar, B., and Sankaranarayanan, A. (2017). One network to solve them all–solving linear inverse problems using deep projection models. In *Proc. IEEE Conf. Comp. Vision and Pattern Recog. (CVPR)*, pages 5888–5897.

[41] Rudin, L., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268.

[42] Shah, P. and Chandrasekaran, V. (2011). Iterative projections for signal identification on manifolds: Global recovery guarantees. In *Proc. Allerton Conf. Communication, Control, and Computing*, pages 760–767.

[43] Shah, V. and Hegde, C. (2018). Solving linear inverse problems using gan priors: An algorithm with provable guarantees. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing (ICASSP)*.

[44] Soltani, M. and Hegde, C. (2017a). Fast algorithms for demixing signals from nonlinear observations. *IEEE Trans. Signal Processing*, 65(16):4209–4222.

[45] Soltani, M. and Hegde, C. (2017b). Fast algorithms for demixing sparse signals from nonlinear observations. *IEEE Transactions on Signal Processing*, 65(16):4209–4222.

[46] Van Veen, D., Jalal, A., Price, E., Vishwanath, S., and Dimakis, A. (2018). Compressed sensing with deep image prior and learned regularization. *arXiv preprint arXiv:1806.06438*.

[47] Xu, L., Ren, J., Liu, C., and Jia, J. (2014). Deep convolutional neural network for image deconvolution. In *Proc. Adv. in Neural Processing Systems (NIPS)*, pages 1790–1798.

[48] Xu, Z. and Sun, J. (2010). Image inpainting by patch propagation using patch sparsity. *IEEE Trans. Image Processing*, 19(5):1153–1165.

[49] Yeh, R., Chen, C., Lim, T., Hasegawa-Johnson, M., and Do, M. (2016). Semantic image inpainting with perceptual and contextual losses. *arXiv preprint arXiv:1607.07539*.

[50] Yeh, R., Chen, C., Lim, T.-Y., Schwing, A., Hasegawa-Johnson, M., and Do, M. (2017). Semantic image inpainting with deep generative models. In *Proc. IEEE Conf. Comp. Vision and Pattern Recog. (CVPR)*, volume 2, page 4.

[51] Zhao, J., Mathieu, M., and LeCun, Y. (2016). Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*.

[52] Zhu, J., Krähenbühl, P., Shechtman, E., and Efros, A. (2016). Generative visual manipulation on the natural image manifold. In *Proc. European Conf. Comp. Vision (ECCV)*.

[53] Zhu, J.-Y., Park, T., Isola, P., and Efros, A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. IEEE Conf. Comp. Vision and Pattern Recog. (CVPR)*.