Solving Inverse Problems using Generative Priors

Viraj Shah

Masters Thesis Defense

Committee:

Dr. Chinmay Hegde, Major Professor,
Dr. Baskar Ganapathysubramaniam,
Dr. Soumik Sarkar

February 9, 2021

Iowa State University

Parts of this work have been submitted to ICASSP 2018, ICASSP 2019.
An extended version to be submitted to IEEE TSP.

$$y = \mathcal{H}(x^*)$$

- Given x^*, determining y or $(\mathcal{H})$ is a Forward Modeling Problem;
- Given y, determining x^* is an **Inverse Modeling Problem**.

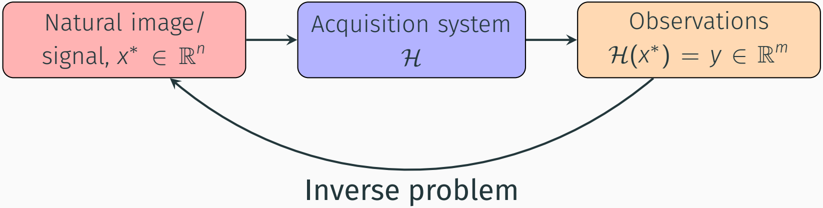
Example:

- Forces on a particle are x^*, trajectory of the particle is y.

Inverse problems are widely encountered in:

- Signal (or image) acquisition, optics, astrophysics, and seismic geo-exploration.

Inverse Problems in Signal Acquisition Systems



Goal is to recover x^*, given the $\mathcal{H}$ and y.

Challenge 1: High sample complexity:

- In general, large number of observations may needed for accurate solution;
- Requires $m \geq n$, implying over-determined system.

Obtaining large number of measurements is expensive

Use the fact that the natural signals are compressible!

It is observed empirically that the natural signals can be compressed.

- *I.e.*, A **stored** natural signal can be well approximated by sparse signal.
- Sparse signal: a signal with most of its components equal to zero.
- Sparse approximation is obtained by appropriate change of basis.
- Commonly used basis are: JPEG (DCT), or wavelet basis.

Original image (left), and its compressed version using DCT (right)

Why to take overdetermined samples in first place?

How to use the compression property during the signal acquisition itself?

Compressive Sensing

- Aim is to acquire the compressed version of a signal directly with $m \ll n$.
- Ill-posed problem, with infinite solutions in general.
- Achieved by exploiting the compressibility of the signal.

The natural signals obey some low-dimensional structure

If such structure is known, accurate reconstruction is possible with $m \ll n$

Formally, We aim to solve a constrained optimization problem:

$$\widehat{x} = \arg\min F(x^*), \quad \text{s. t.} \quad x^* \in \mathcal{S}, \tag{1}$$

where,

- where F is an objective function involving y and $\mathcal{H}$, e.g. $F(x) = \|y - \mathcal{H}(x)\|_2^2$
- The set $\mathcal{S} \subseteq \mathbb{R}^n$ captures the structure that x obeys.

Common choices for $\mathcal{S}$: Sparsity, structured sparsity, total variation.

Sparsity as a Prior

The prior $\mathcal{S} = \{x \in \mathbb{R}^n \mid \|x\|_0 \leq s\}$

- Most natural signals and images are sparse in some basis,
- It makes the sparsity the obvious choice as a prior.
- Under certain conditions, perfect recovery can be achieved.

Disadvantages:

- Poor discrimination capabilities: many noise-signals are sparse.
- Performs poorly when m is very low, as no learned information about the signal manifold.

These are sparse in DCT basis, but don't resemble to natural images!

The prior $\mathcal{S}$ should closely mimic the natural signals manifold.

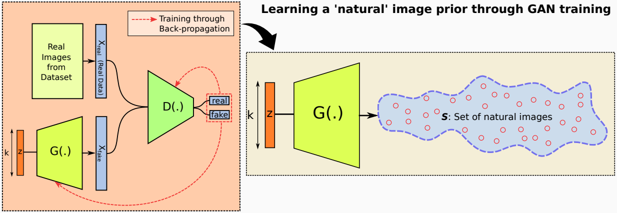
- Such $\mathcal{S}$ can be obtained by generative models trained on natural data.
- State-of-the-art deep generative models (e.g. GANs) can be employed.

Generative Adversarial Networks (GANs):

- Mimics the natural data distribution through adversarial training,
- Input is a random vector z, the output $G(z)$ resembles a natural image.

Generative Prior:

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



Problem Setup

We aim to recover $x^* \in \mathbb{R}^n$ given $\mathcal{H}$, and $\mathcal{H}(x) = y \in \mathbb{R}^m$.

- We assume $m \ll n$.
- $\mathcal{H}$ is parameterized by matrix $A_{m\times n}$.

$\mathcal{H}(\cdot)$ is linear $\implies$ Linear inverse problem

Imaging problem of the form $y = Ax^*$:

- **Compressive sensing**, where $m \ll n$, and A is i.i.d. Gaussian.
- **Image inpainting**, where rows of A contains blocks of zeros.
- **Image super-resolution**, with A being downsampling operator.

$\mathcal{H}(\cdot)$ is nonlinear $\implies$ Nonlinear inverse problem

For e.g.:

- **Sigmoid recovery**; with $\mathcal{H}(x) = sigmoid(Ax) + \ noise$
- **Phase retrieval**, with $\mathcal{H}(x) = |Ax| + \ noise$

Employing Generative Priors: Prior Work (CSGM [1])

Key assumption:

- The range of well-trained G provides good approximation of set of natural images.

CSGM (Bora et al.):

- Obtain a well-trained Generator $G : \mathbb{R}^k \to \mathbb{R}^n$.
- Construct the estimate $\hat{x}$ as follows:

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

- Solve for $\hat{z}$, to obtain $\hat{x} = G(\hat{z})$.

Limitations:

- No discussion about an algorithm to perform non-convex optimization of Eqn. (2).
- Instead, they use gradient descent directly on Eqn. (2)
- Study of the algorithmic costs of solving the optimization is not provided.

Our Contributions

In this work,

- We propose a PGD algorithm for solving linear inverse problems.
- We provide proof of linear convergence of our algorithm.
- We extend the algorithm to a much wider range of nonlinear problems.
- We present empirical results supporting our claims.
- We also extend our approach to handle model mismatch.

Key assumptions:

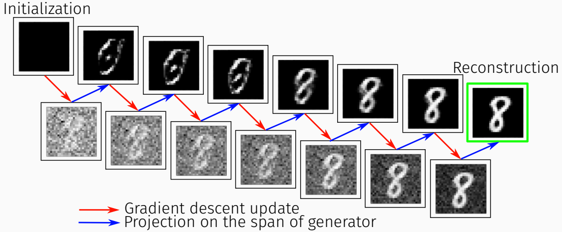
- Availability of a well-trained Generator G.
- Availability of a projection oracle onto G (P_G);
- Given any vector $x \in \mathbb{R}^n$, $x' = P_G(x) \in \text{Range}(G)$ that minimizes $\|x - x'\|_2^2$.

For linear case of compressive sensing, we set A as i.i.d. Gaussian, F as Euclidean norm. We seek,

$$\widehat{x} = F(x) = \underset{x \in G(z)}{\arg\min} \|y - Ax\|_2^2 \tag{3}$$

PGDGAN: **Projected Gradient Descent using GAN priors**

1. **Initialization:** Initialize x^0 with zero vector.
2. **Estimation:** For $t = 1, 2, ..., T$:
    - Gradient descent update.
    - Projection on the span of generator (G).



PGD Algorithm

Step 1: Gradient Descent Update

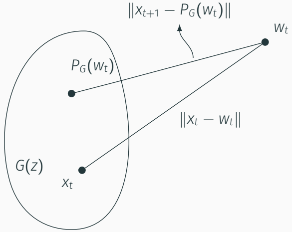
application of a gradient descent update rule on the loss function $F(\cdot)$ with the learning rate η.

$$w_t \leftarrow x_t + \eta A^T(y - Ax_t)$$

Step 2: Projection

$$x_{t+1} = P_G\left(w_t\right) := G\left(\arg\min_z \|w_t - G(z)\|\right),$$

We use gradient descent (implemented via back-propagation) as a projection oracle, with learning rate η_{in}.



- In each of the T iterations, we run T_{in} gradient descent updates for calculating the projection.

$T \times T_{in}$ is the total number of gradient descent updates on G

Linear Convergence of PGDGAN Algorithm

Theorem (Guarantee: linear convergence)

Under certain conditions on A and m, the sequence (x_t) defined by the PGDGAN algorithm with converges to x^* with high probability.

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

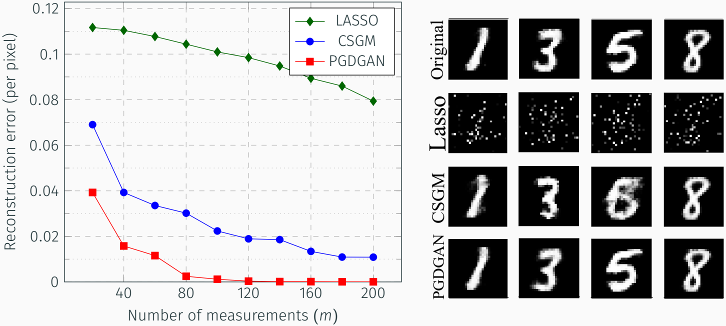
Proved using:

- The difference of any two signals in $\mathcal{S}$ lies away from nullspace of A. (Set Restricted Eigenvalue Condition).
- Spectral norm of A is upper-bounded by $\sqrt{\gamma}$.
- $P_G(\cdot)$ is a orthogonal projection operator.
- The learning rate obeys: $\frac{1}{2\gamma} < \eta < \frac{1}{\gamma}$

Experimental Setup: Compressive Sensing

- We provide results on two different datasets using two different GAN architectures.
- The results are compared with the CSGM [1], and Lasso-DCT [5].
- **MNIST Dataset:**
  - We construct a simple GAN with both G and D are fully-connected neural networks with one hidden layer.
  - G is constructed as: $20 - 200 - 784$; D is constructed as $784 - 128 - 1$.
  - Dimensions of the input z is $k = 20$.
  - Test images are chosen from the range of the G to get rid of representation error.
  - $T = 15$ and $T_{in} = 200$. Thus, the total number of update steps is fixed to 3000.
  - For comparison, we use the reconstruction error $= \|\widehat{x} - x^*\|^2$.
  - We reconstruct the images with $m = 100$ measurements.

Experimental Results: Compressive Sensing



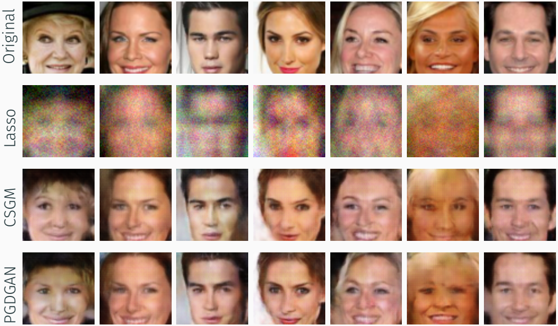
(a) (b)

PGDGAN is able to explore the space outside the range of G

Helps in mitigating the effects of local minima

Doesn't require random restarts

Experimental Results: Compressive Sensing for CelebA



Reconstruction results on CelebA:

- We use DCGAN with both G and D are CNNs with 4 hidden layers each.
- Dimensions of the input z is $k = 100$.
- Test images are kept unseen during training.
- Total number of updates is set to 1000, with $T = 10$ and $T_{in} = 100$.
- We reconstruct the images with $m = 1000$ measurements.

Extension to Nonlinear Inverse Problems

We extend the above algorithm for nonlinear inverse problems:

- We generalize the loss function to be $F(\cdot)$ and the projection oracle to P_G.
- Assume that the F has a continuous gradient $\nabla F = \left(\frac{\partial F}{\partial x_i}\right)_{i=1}^{n}$.
- We define the ε-approximate projection oracle P_G as,

Approximate projection

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

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

$\epsilon-$PGD Algorithm:

- Initialization: $x_0 \leftarrow \mathbf{0}$
- Gradient update step: $w_t \leftarrow x_t - \eta \nabla F(x_t)$
- Projection step: $x_{t+1} \leftarrow P_G(w_t)$

Theorem (Linear Convergence of $\epsilon-$PGD)

Under certain conditions on F, ϵ-PGD algorithm convergences linearly up to a ball of radius $O(\gamma\Delta) \approx O(\varepsilon)$.

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

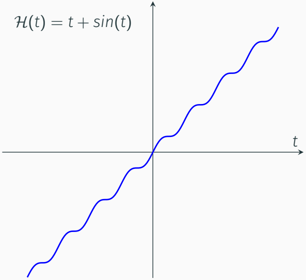
The analysis for linear problem is a special case of the above theorem.

Proved using:

- F follows Restricted Strong Convexity/Smoothness conditions with constants α, β.
- Gradient at the minimizer is small: $\|\nabla F(x^*)\|_2 \leq \gamma$
- Range of G is compact: $\mathrm{diam}(\mathrm{Range}(G)) = \Delta$.
- $\gamma\Delta \leq O(\varepsilon)$.
- $1 \leq \frac{\beta}{\alpha} < 2$

We provide empirical results for two nonlinear inverse problems.

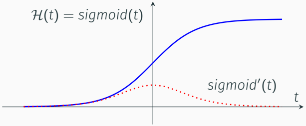
1. **Sinusoidal model**, with
$\mathcal{H}(x) = Ax + sin(Ax)$.
   - We use l_2-loss as F.



2. **Sigmoid model**, with
$\mathcal{H}(x) = sigmoid(Ax) = \frac{1}{1+\exp(-Ax)}$.
   - We use a loss function specified as:

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

where, $\Theta(\cdot)$ is integral of $\mathcal{H}(\cdot)$, and a_i represents the rows of the measurement matrix A.

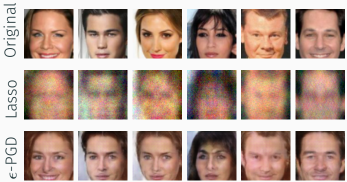


The gradient of the loss:

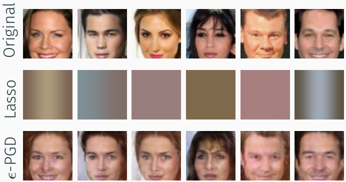
$$\nabla F(t) = \frac{1}{m}A^T(sigmoid(At) - y).$$

Experimental Results: Nonlinear Inverse Problems

- **We perform the experiments on CelebA Dataset:**
  - We use DCGAN with both G and D are CNNs with 4 hidden layers each.
  - Dimensions of the input z is $k = 100$.
  - Test images are kept unseen during training.
  - Total number of updates is set to 1000, with $T = 10$ and $T_{in} = 100$.
  - We reconstruct the images with $m = 1000$ measurements.



(a) Sinusoidal model;



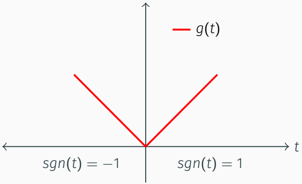
(b) Sigmoid model.

Extension to Phase Retrieval Problem

We also extend the algorithm for phase retrieval problem:

1. Phase retrieval,

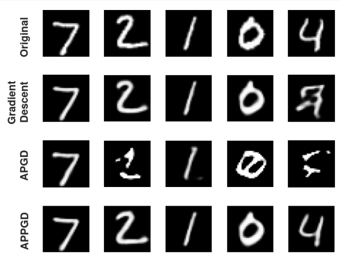
$$\widehat{x} = \arg\min_{x} \|y - |Ax|\|^2$$
$$\text{s.t.} \quad x = G(z),$$



$g(t) = \mathrm{abs}(t)$

Alternating Phase Projected Gradient Descent:

Inputs: y, A, G, T, **Output:** $\widehat{x}$
Choose an initial point $x_0 \in \mathbb{R}^n$
for t = 1,...T **do**
  $p_{t-1} \leftarrow sgn(Ax_{t-1})$
  $w_{t-1} \leftarrow x_{t-1} + \eta A^T(y \odot p_{t-1} - Ax_{t-1})$
  $x_t \leftarrow \mathcal{P}_G(w_{t-1}) = G\left(\arg\min_z \|w_{t-1} - G(z)\|\right)$
end for
$\widehat{x} \leftarrow x_T$



A Note on Representation Error

Three sources of this error:

- Representation error: the image being sensed is not in the range of G,
- Measurement error: measurements do not contain all the information,
- Optimization error: The optimization procedure did not find the best x.

Representation error is dominant term.

Superior GAN models can reduce representation errors.

(a) Glow model [3],

(b) Progressive-GAN model [2].

Summary and future work

Summary:

- Novel PGD-based algorithms with theoretical guarantees
- Works for variety of linear and nonlinear inverse problems
- Extension to phase retrieval problem

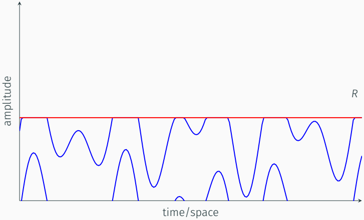
Future directions:

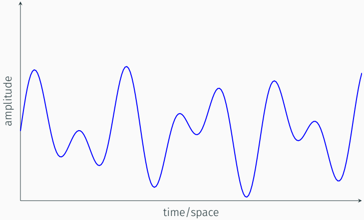
- Extend to modulo recovery and other nonlinear problems
- Employing state-of-the-art generative priors such as deep image priors

Questions?

Modulo Recovery

Modulo Recovery →

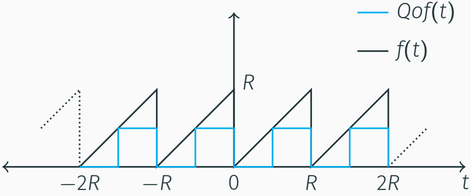


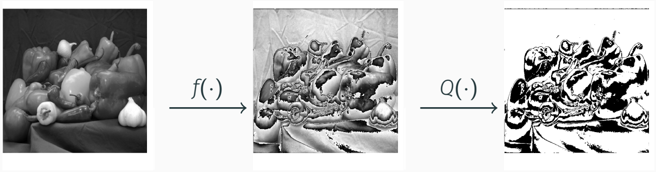


Modulo Recovery →

Reconstruction from Periodic Non-linearities [4], Spring 2017

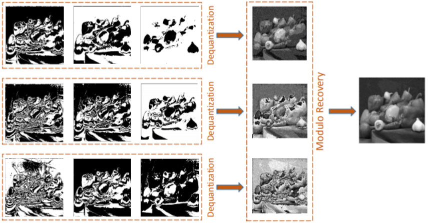
V. Shah and M.Soltani and C. Hegde,
Reconstruction from Periodic Nonlinearities, with Applications to HDR Imaging,
Asilomar Conference on Signals, Systems, and Computers, November 2017.





Reconstruction from Periodic Non-linearities [4]

Our Algorithm: RQM-Recovery from Quantized Modulo images



Stagewise approach

- **Dequantization**: Recover u from $y = Q(Cu)$ using HM algorithm.
- **Modulo recovery**: Recover z from $u = f(Dx)$ using MF-Sparse [2] or Multi-shot UHDR [1].
- **Sparse recovery**: Recover x from $z = Bx$ using any stable sparse recovery algorithm (e.g. CoSaMP).

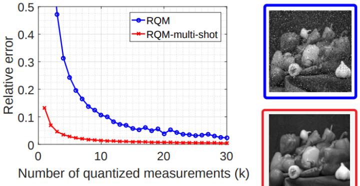
Algorithm:RQM

Inputs: $y, D, B, C, k, k', \Omega, s$
Output: $\hat{x}$
Stage 1: Harmonic dequantization
$\hat{u} \leftarrow \text{HMDEQUANTIZATION}(y, C, k)$
Stage 2: Modulo recovery[2]
$\theta \leftarrow \exp(i\hat{u})$
for $l = 1 : q$ **do**
$t \leftarrow D(l : q : (k'-1)q + l, l)$
$\phi \leftarrow \theta(l : q : (k'-1)q + l)$
$\hat{z}_l = \text{argmax}_{\omega \in \Omega} |\langle y, \psi_\omega \rangle|$
end for
$\hat{z} \leftarrow [\hat{z}_1, \hat{z}_2 \ldots, \hat{z}_q]^T$
Stage 3: Sparse recovery
$\hat{x} \leftarrow \text{COSAMP}(\hat{z}, B, s)$

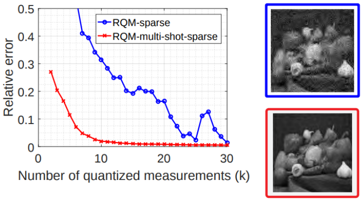
Experimental Results

Recovery from Quantized Modulo images

without sparsity



with sparsity



A. Bora, A. Jalal, E. Price, and A. Dimakis.
Compressed sensing using generative models.
Proc. Int. Conf. Machine Learning, 2017.

T. Karras, T. Aila, S. Laine, and J. Lehtinen.
Progressive growing of gans for improved quality, stability, and variation.
arXiv preprint arXiv:1710.10196, 2017.

D. P. Kingma and P. Dhariwal.
Glow: Generative flow with invertible 1x1 convolutions.
In *Advances in Neural Information Processing Systems*, pages 10215–10224, 2018.

V. Shah, M. Soltani, and C. Hegde.
Reconstruction from periodic nonlinearities, with applications to HDR imaging.
In *Proc. Asilomar Conf. Signals, Systems, and Computers*, Nov. 2017.

R. Tibshirani.
Regression shrinkage and selection via the lasso.
Journal of the Royal Statistical Society: Series B (Methodological), 58(1):267–288, 1996.

Appendix

Linear Convergence of PGDGAN Algorithm

The difference vector of any two signals in the set $\mathcal{S}$ should lie away from the nullspace of the matrix A.

S-REC (Set Restricted Eigenvalue Condition)

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

Theorem (Guarantee: linear convergence)

Let G be a generator with range $\mathcal{S}$. A is satisfying 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 $1 - q$. $\rho^2 \leq \gamma$.
Then, for every $x^* \in \mathcal{S}$, the sequence (x_t) defined by the PGDGAN algorithm converges to x^* with probability at least $1 - p - q$.

$\epsilon-$ PGD: Theoretical Results

We introduce more general restriction conditions on the $F(\cdot)$:

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.

Theorem (Linear Convergence of $\epsilon-$PGD)

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

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

The analysis for linear problem is a special case of the above theorem.