Foundations of Algorithms for Massive Datasets – 236779

Student Name	ID	E-mail
Muhammad Dahamshi	318396926	muhammad.dah@campus.technion.ac.il

Connection Between Deep Networks and Dimensionality Reduction/Sparse Recovery

1 Introduction

The article aims at solving the classical compressed sensing problem: reconstruct an **image** vectorx^* given a vector $y = Ax^* + \eta, A \in \mathbb{R}^{m \times n}, m \ll n$ of linear noisy measures of x^*. The article proposes to use generative models based on neural networks to circumvent the standard approaches based on sparsity. The idea is to estimate x^* as the best image $G(z)$ a generative model can generate given the measurement y.

2 Setup

More precisely, we are given a measurement matrix $A \in \mathbb{R}^{m \times n}$ and a generative model, which is a function $G: \mathbb{R}^k \to \mathbb{R}^n, k \ll n.$ G is determined by training a neural network on a set of image data, and associates an image $G(z) \in \mathbb{R}^n$ to any parameter z in the low-dimensional space $\mathbb{R}^k$.

We observe a noisy measurement $y = Ax^* + \eta$ of the true image x^*. The idea of this article is to propose as an estimate of x^*, the best image $G(z)$ that the model can generate, by minimizing the new criterion $\|AG(z) - y\|_2^2$ over $z \in \mathbb{R}^k$. In other words, this allows to approximate the closest possible point to x^* in the range of G.

The algorithm simply consists in minimizing the loss $\|AG(z) - y\|_2^2$ to find a terminal point $\hat{z}$ to the procedure. The estimate for x^* is then defined as $\hat{x} = G(\hat{z})$. This criterion is not convex, but a gradient descent method seems to give good empirical results. Indeed, for a small number of measurements, this method outperforms LASSO under the assumption that the measurement matrix satisfies some conditions called S-REC (Set-Restricted Eigenvalue Condition).

Before giving theoretical results to justify that approach, we first begin with a review of the generative models used in the paper.

2.1 Generative models

Suppose we are dealing with data points that belong to the *data space* $\mathbb{R}^n$ (also referred to as the *sample space*). A generative model $G: \mathbb{R}^k \to \mathbb{R}^n$ is a deterministic function that returns a point in the sample space given an input which belongs to the *latent space* $\mathbb{R}^k$ (also referred to as the *representation space*). Usually, $k \ll n$, so a model that successfully learnt to generate data points is expected to have created a compressed representation of the original data.

Let p_{data} denote the unknown data-generating distribution. The goal of generative modelling is to train a model that induces a distribution p_G on the data space such that p_G is similar to p_{data}. Probability distributions can be formally compared by using f-divergences or integral probability metrics.

Let θ be the parameter vector learnt by the model and assume that p_{data} and p_θ are absolutely continuous with respect to the same measure μ (for example Lebesgue measure on $\mathbb{R}^d$ or a counting measure), with densities q_{data} and q_θ. Let $f: \mathbb{R} \to \mathbb{R}$ a convex function with $f(1) = 0$.

The f-*divergence* between p_{data} and p_θ is defined as:

$$D_f(p_{\text{data}}||p_\theta) := \int f\left(\frac{q_{\text{data}}(x)}{q_\theta(x)}\right) q_\theta(x) d\mu(x)$$

Note that D_f may not be symmetric and the triangle inequality may not hold, hence the name divergence instead of distance. However, f-divergences satisfy the positivity property: $D_f(p_{\text{data}}||p_\theta) = 0 \Leftarrow p_{\text{data}} = p_\theta$. Popular instances include the Kullback-Leibler divergence ($f: t \mapsto t\text{log}t$), the Hellinger distance ($f: t \mapsto \frac{1}{2}(\sqrt{t} - 1)^2$) and the total variation distance ($f: t \mapsto \frac{1}{2}|t - 1|$).

Integral probability metrics (IPMs) are defined by

$$d_{\mathcal{F}}(p_{\text{data}}||p_\theta) := \sup_{f \in \mathcal{F}} E_{x \sim p_{\text{data}}}[f(x)] - E_{x \sim p_\theta}[f(x)]$$

where $\mathcal{F}$ is a class of functions from $\mathbb{R}^n$ to $\mathbb{R}$. Note that IPMs are easier to define since they not require density assumptions on the probability distributions. Furthermore, if $f \in \mathcal{F} - f \in \mathcal{F}$, then $d_{\mathcal{F}}$ is symmetric and satisfies the triangle inequality.

Popular instances include the Wasserstein-1 distance($\mathcal{F} = \{1 - \text{Lipschitzfunctions}\}$), the total variation distance ($\mathcal{F} = \{\text{bounded functions with values in } [-1,1]\}$) , which is therefore at the intersection between IPMs and f-divergences) and the Maximum Mean Discrepancy ($\mathcal{F} = \{\text{unit ball of an RKHS}\}$).

The most well-known generative models based on neural networks are variational autoencoders and generative adversarial networks.

2.2 Variational AutoEncoders

Variational autoencoders (VAEs) aim at learning θ to minimize

$$\mathrm{KL}(p_{\text{data}}||p_\theta) = E_{x\sim p_{\text{data}}}(\log p_{\text{data}}(x)) - E_{x\sim p_{\text{data}}}(\log p_\theta(x))$$

Since $E_{x\sim p_{\text{data}}}(\log p_{\text{data}}(x))$ is independent of θ, minimizing $\mathrm{KL}(p_{\text{data}}||p_\theta)$ is equivalent to maximizing the likelihood $E_{x\sim p_{\text{data}}}(\log p_\theta(x))$. However computing $\log p_\theta(x)$ is not tractable, so a first step is to introduce a latent variable z and the joint distribution is decomposed as $p_\theta(x,z) = p_\theta(x|z)p(z)$ where $p(z)$ is a fixed prior and $p_\theta(x|z)$ is a simple distribution whose parameters are the output of a neural network. The second step is to define another distribution $q_\phi(z|x)$ which will also be modelled by a neural network. Next, note that

$$\begin{aligned}\log p_\theta(x) \quad &= \mathbb{E}_{z\sim q_\phi} \log\, p_\theta(x) = \mathbb{E}_{z\sim q_\phi}\left[\log p_\theta(x)\frac{q_\phi(z|x)}{q_\phi(z|x)}\right] \\ &= \mathbb{E}_{z\sim q_\phi}\left[\log\left[\frac{p_\theta(x|z)p(z)}{p_\theta(z|x)}\right]\frac{q_\phi(z|x)}{q_\phi(z|x)}\right] \\ &= \mathrm{KL}(q_\phi(z|x)||p_\theta(z|x)) - \mathrm{KL}(q_\phi(z|x)||p(z)) + \mathbb{E}_{z\sim q_\phi}(\log p_\theta(x|z)) \\ &\geq -\mathrm{KL}(q_\phi(z|x)||p(z)) + \mathbb{E}_{z\sim q_\phi}(\log p_\theta(x|z))\end{aligned}$$

This lower bound on $\log p_\theta(x)$ is computationally tractable and provides an objective function that we can maximize. It is known as the Evidence Lower-Bound (ELBO) in the literature. The VAE loss thus writes as

$$\mathcal{L}_{\text{VAE}} = \mathrm{KL}(q_\phi(z|x)||p(z)) - \mathbb{E}_{z\sim q_\phi}(\log p_\theta(x|z))$$

$q_\phi(z|x)$ can be seen as a *probabilistic encoder* from the data x to the latent space z, and $p_\theta(x|z)$ can be seen as a *probabilistic decoder* from the latent space z to the data x. Remember that the objective is to use gradient descent to learn the parameters θ and ϕ.

Let us give more details on the structure of a VAE applied to the MNIST dataset, as done in the paper. The encoder is assumed to follow $\mathcal{N}_k(\mu, \mathrm{diag}(\Sigma_{11}, \dots, \Sigma_{kk}))$ where the mean vector μ and the diagonal covariance matrix are outputs of the neural network corresponding to the encoder. Since the images are represented by black and white pixels, the decoder is assumed to follow a random vector with coordinates $\mathcal{B}(y_i)$ where the vector y is learnt by the neural network corresponding to the decoder. Lastly, the prior on z is assumed to be $\mathcal{N}_k(0, I_k)$. In this setting, we get

$$\mathrm{KL}(q_\phi(z|x)||p(z)) = \frac{1}{2}\sum_{i=1}^{k}(\exp\log\Sigma_{ii} - \log\Sigma_{ii} + \mu_i^2 - 1)$$

If we assume further that the individual pixels of the output of the decoder are independent, each one following $\mathcal{B}(y_i)$, we get

$$\log p_\theta(x|z) = \sum_{i=1}^{n} x_i \log y_i + (1 - x_i)\log(1 - y_i)$$

Writing down these expressions in a deep learning framework such as Tensorflow or Pytorch allows us to train a VAE on MNIST.

2.3 Generative Adversarial Networks

Generative adversarial networks (GANs) are made up of two separate networks: a *generator* G which induces a distribution p_θ on the sample space and a *discriminator* D which acts as a classifier. The discriminator is trained to differentiate between samples from the dataset and samples generated through p_θ. The loss of the standard GAN introduced in 2014 by Goodfellow writes as:

$$\min_G \max_D E_{x\sim p_{\text{data}}}(\log D(x)) + E_{x\sim p_\theta}(\log[1 - D(x)])$$

For a fixed generator, it is easy to show that the optimal discriminator is $D^*(x) = \frac{p_{\text{data}}(x)}{p_\theta(x)+p_{\text{data}}(x)}$.

For this D^*, the loss turns into

$$\min_G [2\ \text{JSD}(p_\theta \parallel p_{\text{data}}) - 2\log 2]$$

where JSD is the Jensen-Shannon divergence, a symmetric version of the Kullback-Leibler divergence. Thus training the generator to minimize the loss is equivalent to minimizing the JSD between p_θ and p_{data}.

GANs are easier to formulate compared to VAEs. However, they are also more difficult to train. Assuming that the discriminator has been trained to optimality, one is tempted to perform gradient steps on θ, however this does not work. In practice, as the discriminator gets better, the updates to the generator get worse. In his seminal paper Goodfellow indicates that when the generator is poor, the discriminator can reject samples with high confidence, thus the term $\mathbb{E}_{x\sim p_\theta}[-\log(1 - D(x))]$ is almost 0, so the gradient vanishes, hence the generator cannot improve. Besides, the loss only returns feedback on the quality of the outputs of the generator, regardless of their diversity (representativeness of the real data). A generator could thus return the exact same high-quality sample and get very low loss. This is usually referred to as *mode collapse*.

Over the years, researchers have come up with other loss that provide more training stability (Wasserstein-1 for example)

2.4 Applications in the paper

In the paper the authors do not put much emphasis on the architectures and the training of their generative models. They simply use common versions of VAEs and GANs, so we did the same in our experiments (see the notebook).

3 Theoretical Results

To ensure good reconstruction with high probability, a technical assumption on the measurement matrix A is needed, called the Set-Restricted Eigenvalue Condition.

Definition 3.1 *(Set-Restricted Eigenvalue Condition) Let* $S \in \mathbb{R}^n$*. For some parameters* $\gamma > 0$*,* $\delta \geq 0$ *a matrix* $A \in \mathbb{R}^{m \times n}$ *is said to satisfy the S-REC conditions* $S - REC(S, \gamma, \delta)$ *if* $\forall x_1, x_2 \in S,\ \|A(x_1 - x_2)\| \geq \|x_1 - x_2\| - \delta.$

This condition is close to the more classical REC condition, which is a sufficient condition for robust recovery by LASSO:

Definition 3.2 A *satisfies REC for a constant* $\gamma > 0$ *if for all approximately sparse vectors* $x, \|Ax\| \geq \gamma \|x\|.$

The main difference between both is that S-REC condition applies to vectors in a set S, and isn't restricted to sparse vectors. The idea is to apply this condition to $S = G(\mathbb{R}^k)$, the set of images which can be generated by a generative model.

For random Gaussian matrices, the S-REC conditions are ensured with high probability for large classes of generators: indeed, the article is to shows that in the case of d-layer neural-nets, taking $m = 0(kd\log(n))$ is sufficient to ensure the S-REC condition and to guarantee good reconstruction with high probability. In the following three lemmas, the matrix A will denote a Gaussian matrix generated according to $\forall i, j, A_{ij} \sim N(0, \frac{1}{m})$

The first lemma gives a sufficient condition to guarantee the S-REC in the case of a Lipschitz generative model:

Lemma 3.1 *Let* $G: \mathbb{R}^k \to \mathbb{R}^n$ *L-Lipschitz. Let* $\alpha, r > 0$ *and*

$$B^k(r) = \{z \in \mathbb{R}^k | \|z\| < r\}$$

If $m = \Omega(\frac{k}{\alpha^2} \log(\frac{Lr}{\delta}))$ and , then A satisfies $S - REC(G(B^k(r)), 1 - \alpha, \delta)$ wp$\geq 1 - e^{-\Omega(\alpha^2 m)}$

This lemma is well suited for our setting since it is easily shown that the function G is Lipschitz in the case of a Neural Network.

Lemma 3.2 *Let* $G: \mathbb{R}^k \to \mathbb{R}^n$ *be a neural network with* d *layers, where each layer is a linear transformation followed by a pointwise non-linearity. Suppose there are at most* c *nodes per layer, and the non-linearities are piecewise linear with at most two pieces.*
Let $m = O\left(\frac{1}{\alpha^2} kdlog(c)\right)$ *for some* $\alpha < 1$*. Then* A *satisfies* $S - REC(G(\mathbb{R}^k), 1 - \alpha, 0)$ *wp* $\geq 1 - e^{-\Omega(\alpha^2 m)}$

Lemma 3.3 *Let A be drawn from a distribution that satisfies the $S-REC(S,\gamma,\delta)$ with probability $\geq 1-p$ and such that for every fixed $x \in \mathbb{R}^n$, $\|Ax\| \leq 2\|x\|$, with probability $1-p$. For any $x^* \in \mathbb{R}^n$ and noise η let $y = Ax^* + \eta$. Let $\hat{x}$ approximately minimizing $\|y - Ax\|$ over $x \in S$, ie:*

$$\|y - A\hat{x}\| \leq \min_{x \in S} \|y - Ax\| + \epsilon$$

Then, $\|\hat{x} - x^*\| \leq (\frac{4}{\gamma} + 1)\min_{x \in S}\|x - x^*\| + \frac{1}{\gamma}(2\|\eta\| + \epsilon + \delta)$ *with probability 1-2p*

These three lemmas allow to prove the following two theorems, which are this article's key contributions.

Theorem 3.1 *Let $G: \mathbb{R}^k \to \mathbb{R}^n$ be a generative model from a d-layer neural network using ReLU activations. Let $A \in \mathbb{R}^{m \times n}$ be a random Gaussian matrix for $m = O(kdlog(n)$, scaled so that $\forall i, j, A_{ij} \sim N\left(0, \frac{1}{m}\right)$. For any $x^* \in IR^n$ and any observation $y = Ax^* + \eta$, let $\hat{z}$ minimize $\|y - AG(z)\|_2$ to within additive ϵ of the optimum. Then with $1 - e^{-\Omega(m)}$ probability,*

$$\|G(\hat{z}) - x^*\| \leq 6 \min_{z^* \in IR^k} \|G(z^*) - x^*\|_2 + 3\|\eta\|_2 + 2\epsilon$$

Theorem 3.2 *$G: \mathbb{R}^k \to \mathbb{R}^n$ be an L-Lipschitz function. Let $A \in \mathbb{R}^{m \times n}$ be a random Gaussian matrix for $m = O(klog\frac{Lr}{\delta})$, scaled so that $\forall i, j, A_{ij} \sim N\left(0, \frac{1}{m}\right)$. For any $x^* \in \mathbb{R}^n$ and any observation $y = Ax^* + \eta$, let $\hat{z}$ minimize $\|y - AG(z)\|_2$ to within additive ϵ of the optimum over vectors with $\|\hat{z}\|_2 \leq r$. Then with probability $1 - e^{-\Omega(m)}$*

$$\|G(\hat{z}) - x^*\| \leq 6 \min_{z^* \in IR^k, \|z^*\|_2 \leq r} \|G(z^*) - x^*\|_2 + 3\|\eta\|_2 + 2\epsilon + 2\delta$$

4 Experiments

The authors use two image datasets (MNIST and CelebA) and two kinds of generative models: VAEs and GANs. The results of the article are stated for $\hat{z} = arg\ min\|y - AG(z)\|_2$, but in their experiments, they suddenly use $\hat{z} = arg\ min\|y - AG(z)\|_2^2 + \lambda\|z\|^2$, which arguably weakens the theoretical guarantees they provide at the beginning.

They apply their method to both data sets and compare their performances to the baseline LASSO method, by comparing the criterion $\|x^* - \hat{x}\|^2$ for each method.

Please see the notebook for our attempt at reproducing their results.

4.1 Comparison on MNIST

The method developed in this article, applied to MNIST, gives significantly better results than LASSO using significantly fewer measures (25 versus 400 for LASSO). However, the method is limited since its output is constrained to be in the range of the generator. Therefore it quickly saturates, and additional measurements give no additional improvement, whereas the LASSO's performance isn't limited and keeps increasing, eventually outperforming this algorithm.

Indeed, the performance of this method is necessarily bounded below by the minimal distance of x^* to its closest point on the range of G, since the algorithm's output must be of the form $G(z)$. Therefore it induces a large systematic error. The authors suggest to use a better generative model, able to fit a wider range of images, to remedy this drawback.

4.2 Comparison on CelebA

For 500 measurements, the result is quite convincing: the method obtains good results whereas the LASSO's output is quite blurry. The same saturation phenomenon occurs for 5000 measurements.

4.3 Sources of error

The sources of errors seem threefold.

- Representation error: the actual image is far from $G(\mathbb{R}^k)$
- Measurement error: a finite number of measurements do not contain all the information about the unknown image
- Optimization error: the optimum found is a local optimum far from the global one.

The authors suggest that the dominant term is the representation error and that the residual two errors are actually very small, which would be extremely advantageous for the method they propose. Indeed, it would mean that training more accurate generative models would allow to considerably reduce the systematic error, leading to performances much better than those based on sparsity, with much fewer measurements. To assess the contribution of each source, they propose two experiments:

- A first one where the representation error is zero, to estimate the sum of the other two errors. FOr this one, they choose an image x^* which is of the form $x^* = G(z)$, and find excellent results for the reconstruction, even with a very small number of measurements.
- A second one, they apply their method on both data sets, with $A = Id$, meaning that no measurement error is introduced by A. Then, they compare the performance on this particular model to the performance on the mode general model $y = Ax^* + \eta$ and they find that the errors are quite similar. This suggests that the representation error is the major component of the total error.

5 Proofs

In this section we only provide proofs for Lemma 1 and Lemma 3.

5.1 Proof of lemma 1

Lemma 5.1 *If a matrix* A *satisfies* $S - REC(S, \gamma, \delta)$ *then* $\forall x_1, x_2 \in S$ *such that* $\|Ax_1 - y\| \leq \epsilon_1$ *and* $\|Ax_2 - y\| \leq \epsilon_2$*, we have* $\|x_1 - x_2\| \leq \frac{\epsilon_1 + \epsilon_2 + \delta}{\gamma}$

$Proof$: $\|x_1 - x_2\| \leq \frac{1}{\gamma}(\|Ax_1 - Ax_2\| + \delta) \leq \frac{1}{\gamma}(\|Ax_1 - y\| + \|Ax_2 - y\| + \delta) \leq \frac{\epsilon_1 + \epsilon_2 + \delta}{\gamma}$

Lemma 5.2 *Let* $G: \mathbb{R}^k \to \mathbb{R}^n$ *be L-Lipschitz,* M *a* $\frac{\delta}{L}$*-net on* $B^k(r)$ *such that* $|M| \leq klog(\frac{4Lr}{\delta})$*. Let* $A \in \mathbb{R}^{m \times n}$ *such that* $\forall (i,j), A_{ij} \sim N\left(0, \frac{1}{m}\right)$*. If* $m = O\left(klog\left(\frac{Lr}{\delta}\right)\right)$ *then for any* $x \in S$*, if* $x' = arg\, min_{\tilde{x} \in G(M)} \|x - \tilde{x}\|$*, we have* $\|A(x - x')\| = O(\delta)$ *with probability* $1 - e^{-\Omega(m)}$.

$Proof$: Since $\frac{\|Ax\|^2}{\|x\|^2}$ is subgamma$\left(\frac{1}{\sqrt{m}}, \frac{1}{m}\right)$, then for $f > 0$,

$$\epsilon \geq 2 + \frac{4}{m}\log\frac{2}{f} \geq max\left(\sqrt{\frac{2}{m}\log\frac{2}{f}}, \frac{2}{m}\log\frac{2}{f}\right)$$

which yields $P(\|Ax\| \geq (1 + \epsilon)\|x\|) \leq f$

Let $M = M_0 \subset \ldots \subset M_t$ be a chain of epsilon nets of $B^k(r)$ such that M_i is a δ_i/L-net and $\delta_i = \frac{\delta_0}{2^i}$, $\delta_0 = \delta$. We know that there exist nets such that $\log|M_i| \leq k\log(\frac{4Lr}{\delta_i}) \leq ik + k\log(\frac{4Lr}{\delta_0})$.

Let $N_i = G(M_i)$. G is L-Lipschitz, so N_i is a δ_i-net for $S = G(B^k(r))$, with $|N_i| =$

$|M_i|$. For $1 \leq i \leq l-1$ we define $T_i := \{x_{i+1} - x_i | x_{i+1} \in N_{i+1}, x_i \in N_i\}$

Thus $|T_i| \leq |N_{i+1}||N_i|$ hence $\log|T_i| \leq \log|N_{i+1}| + \log|N_i| \leq (2i+1)k + 2k\log(\frac{4LR}{\delta_0}) \leq 3ik + 2k\log(\frac{4Lr}{\delta_0})$.

Moreover, we assume $m = 3k\log(\frac{4Lr}{\delta_0}, log(f_i) = -(m+4ik)$ and $\epsilon_i = 2 + \frac{4}{m}\log\frac{2}{f_i} = 2 + \frac{4}{m}\log 2 + 4 + \frac{16ik}{m} = 0(1) + \frac{16ik}{m}$.

Then we have $\forall i \in \{1, \ldots, l-1\}, \forall t \in T_i, P\left(\|At > (1+\epsilon_i)\|t\|\|\right) \leq f_i$, which yields $P\left(\|At \leq (1+\epsilon_i)\|t\|\|\forall i, \forall t \in T_i\right) \geq 1 - \sum_{i=0}^{t-1} |T_i|f_i$.

Now,

$$\log(|T_i|f_i) = \log|T_i| + \log|f_i| \leq -k\log(\frac{4Lr}{\delta_0}) - ik = -m/3 - ik$$

Hence $\sum_{i=0}^{t-1} |T_i|f_i \leq e^{-m/3}\sum_{i=0}^{t-1} e^{-ik} \leq e^{-m/3}(\frac{1}{1-e^{-1}}) \leq 2e^{-m/3}$

Let's define $x_f = x - x_l$, then $x - x_0 = \sum_{i=0}^{l-1} (x_{i+1} - x_i) + x_f$, with $x_i \in N_i$

Since each $x_{i+1} - x_i \in T_i$we have with probability at least $1 - 2e^{-m/3}$:

$$\sum_{i=0}^{l-1} \|A(x_{i+1} - x_i)\| = \sum_{i=0}^{l-1} (1+\epsilon_i)\|(x_{i+1} - x_i)\| \leq \sum_{i=0}^{l-1} (1+\epsilon_i)\delta_i = \delta_0 \sum_{i=0}^{l-1} \frac{1}{2^i}(O(1) + \frac{16ik}{m}) = O(\delta_0) + \delta_0 \frac{16k}{m}\sum_{i=0}^{l-1} \frac{i}{2^i} + O(\delta_0)$$

Now, $\|x_f\| = \|x - x_l\| \leq d_l = \frac{\delta_0}{2^l}$, and $\|x_{i+1} - x_i\| \leq \delta_i$ due to the properties of epsilon-nets. We know that $\|A\| \leq 2 + \sqrt{n/m}$ with probability at least $1 - 2e^{-m/2}$. By setting $l = \log(n)$, we get that $\|A\|\|x_f\| \leq (2 + \sqrt{n/m})\frac{\delta_0}{2^i} = O(\delta_0)$ with probability $\geq 1 - 2e^{-m/2}$.

Combining these two results, and noting that it is possible to choose $x' = x_0$, we get that with probability $1 - e^{-\Omega(m)}$,

$\|A(x - x')\| = \|A(x - x_0)\| \leq \sum_{i=0}^{l-1} \|A(x_{i+1} - x_i)\| + \|Ax_f\| = O(\delta_0) + \|A\|\|x_f\| = O(\delta)$

which ends the proof.

Lemma 5.3 *Let $G: IR^k \to IR^n$, L-Lipschitz. If $m = \Omega(\frac{k}{\alpha^2}log\frac{Lr}{\delta}))$, then A satisfies the $S-REC(B^k(r)$,$1-\alpha, \delta$ with $1 - e^{-\Omega(\alpha^2 m)}$ probability.*

Proof. We construct a $\frac{\delta}{L}$-net N on $B^k(r)$. THere exists a net such that $log|N| \leq k\log\frac{4Lr}{\delta}$.

Since N is a $\frac{\delta}{L}$-cover of $B^k(r)$, due to the L-Lipschitz property of G, we get that $G(N)$ is a δ-cover of $G(B^k(r))$. Let T denote the pairwise differences between the elements in $G(N)$, i.e.

$$T = \{G(z_1) - G(z_2) | z_1, z_2 \in N\}$$

Then $|T| \leq |N|^2 \Rightarrow log|T| \leq 2\log|N| \leq 2k\log\frac{4Lr}{\delta}$.

For any $z, z' \in B^k, \exists z_1, z_2 \in N; G(z_1), G(z_2)$ are δ -close to $G(z)$ and $G(z')$ respectively. Thus :

$$\|G(z) - G(z')\| \leq \|G(z) - G(z_1)\| + \|G(z_1) - G(z_2)\| + \|G(z_2) - G(z')\| \leq \|G(z_1) - G(z_2)\| + 2\delta$$

$$\|AG(z_1) - AG(z_2)\| \leq \|AG(z_1) - AG(z)\| + \|AG(z) - AG(z')\| + \|AG(z') - AG(z_2)\|$$

Now by the previous lemma, with probability $1 - e^{-\Omega(m)}$, $\|AG(z_1) - AG(z)\| = O(\delta)$ and $\|AG(z') - AG(z_2)\| = O(\delta)$, thus $\|AG(z_1) - AG(z_2)\| = \|AG(z) - AG(z')\| + O(\delta)$.

By the Johnson-Lindenstrauss Lemma, for a fixed $x \in \mathbb{R}^n, P(\| Ax \|^2 < (1-\alpha) \| x \|^2) < e^{-\alpha^2 m}$. Therefore, we can union bound over all vectors in T to get

$$P(\|Ax\|^2 \geq (1-\alpha)\|x\|^2 \forall x \in T) \geq 1 - e^{-\Omega(\alpha^2 m)}$$

Since $\alpha < 1$ and $z_1, z_2 \in N, G(z_1) - G(z_2) \in T$, we have

$$(1-\alpha)\|G(z_1) - G(z_2)\| \leq \sqrt{1-\alpha}\|G(z_1) - G(z_2)\| \leq \|AG(z_1) - AG(z_2)\|$$

Combining the three results above we get that with probability $1 - e^{-\Omega(\alpha^2 m)}$

$$(1-\alpha)\|G(z) - G(z')\| \leq (1-\alpha)\|G(z_1) - G(z_2)\| + 0(\delta) \leq \|AG(z_1) - AG(z_2)\| + 0(\delta) \leq \|AG(z) - AG(z')\| + 0(\delta)$$

Thus A satisfies $S - REC(S, 1-\alpha, \delta)$ with probability $1 - e^{-\Omega(\alpha^2 m)}$.

5.2 Proof to lemma 3

Let $\bar{x} = argmin_{x \in S}\|x - x^*\|$. Then we have by Lemma 5.1 and the hypothesis on $\hat{x}$ that

$$\|\bar{x} - \hat{x}\| \leq \frac{\|A\bar{x} - y\| + \|A\hat{x} - y\| + \delta}{\gamma} \leq \frac{2\|A\bar{x} - y\| + \epsilon + \delta}{\gamma} \ \ leq \frac{2\|A(\bar{x} - x^*)\| + 2\|\eta\|\epsilon + \delta}{\gamma}$$

as long as A satisfies the S-REC, which happens with probability $1 - p$. Now, since x and x^* are independent of A, by assumption we also have $\|A(\bar{x} - x^*)\| \leq 2\|\bar{x} - x^*\|$ with probability $1 - p$. Therefore

$$\|\hat{x} - x^*\| \leq \|\bar{x} - x^*\| + \frac{4\|\hat{x} - x^* 2\|\eta\|\epsilon + \delta\|}{\gamma}$$

which ends the proof.

6 References

[1] 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 of Proceedings of Machine Learning Research, pages 537–546, 2017

[2] Dhar, M.; Grover, A.; and Ermon, S. 2018. Modeling sparse deviations for compressed sensing using generative models. arXiv preprint arXiv:1807.01442.