Deep Compressed sensing

Guillo Clément Meunier Dimitri

ENS Paris Saclay, MVA Master

March 23, 2020

Classic Compressed Sensing

Goal

- recover $x \in \mathbb{R}^n$ with m measurements, $m << n$
- Formalization: $y = Ax$

Hypothesis

- x lives in a high dimensional space and has to be sparse (with sparsity s) or *approximatly* sparse
- requires a minimal number of measures $s\log(en/s)$ if A is Gaussian random matrix

Problem

Optimization problem: For given A and y, solve,

$$\underset{x \in \mathbb{R}^n : Ax = y}{\arg\min} \ \|x\|_1 \qquad \text{Basis Pursuit}$$

Noisy formulation:

$$\underset{x \in \mathbb{R}^n : \|Ax - y\|_2^2 \leq \delta}{\arg\min} \ \|x\|_1 \qquad \text{Denoising Basis Pursuit}$$

Equivalent to the LASSO,

$$\underset{x \in \mathbb{R}^n}{\arg\min} \frac{1}{2} \|Ax - y\|_2^2 + \lambda \|x\|_1$$

Solution, constraints

- Convex optimization allows to solve this problems
  (Linear programming, simplex methods, interior point methods)
- Can be relaxed with approximate sparsity and change-of-basis (wavelets, Fourier)
- The sparsity hypothesis is still very strong
- Use of Generative models in order to relax the hypothesis

Classic Compressed Sensing

Compressed Sensing with Generative Models

Deep Compressed Sensing

Experiments

Generative Models

- let $G : z \longmapsto \hat{x}$ be a Generative model, $z \in \mathbb{R}^k$ and $\hat{x} \in \mathbb{R}^n$
- z belongs to the latent space such that $k << n$
- we want to use the generative power of G to reconstruct x
- If $z \sim p_z$, G is trained such that $p_{G(z)} \approx p_{data}$
- Two main types of generative models: Generative Adversarial Networks (GAN) and Variational Auto Encoders (VAE)

Pre-trained Generative model

- new problem:

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

- use of a pre-trained generative model G
- optimization in the latent space in order to find $\hat{z}$, highly non convex
- use of deep learning libraries to back-propagate through the generative model and perform gradient descent

guarantees, limits

- (Ashish Bora et al): with only $m = O(kdlog(n))$ measures, $G(\hat{z})$ can be sufficiently closed to x
- we do not need the sparsity constraint anymore, k replaces the sparsity s here
- we need the signal to lie in the range of G
- performance limited by the generative power of G and the dimension of the latent space
- trade-off between the generative power of G and the required number of measurements
- Alternative $(\hat{z}, \hat{v}) \in \underset{z \in \mathbb{R}^k, v \in \mathbb{R}^n}{\arg\min} \|A(G(z) + v) - y\|_2^2 + \|v\|_1$, unstable

Classic Compressed Sensing

Compressed Sensing with Generative Models

Deep Compressed Sensing

Experiments

Learning Generative models

- We use a pretrain generative model in the previous slides
- We want now learn the generative model during the reconstruction process
- Deep compressed sensing algorithm: $G \longrightarrow G_\theta$ is optimized such that $\left(A\left(G_\theta\left(\hat{z}_i\right)\right) - y_i\right)^2$ is small for all observations i.
- need to avoid the degenerate mapping of G into the null space of A, pseudo-RIP condition enforced by minimizing an empirical loss $\frac{1}{N_s} \sum_{j=1}^{N_s} \left[\left(\left\|A\left(x_1^j - x_2^j\right)\right\|_2 - \left\|x_1^j - x_2^j\right\|_2\right)^2\right]$

Algorithm 1

Algorithm 1: DCS Algorithm

Data: Input minibatchs of data $\{x_i\}_{i=1}^N$

initialize A, G_θ, learning rates α_z, α_θ initialization;

for $i = 1$ *to* N **do**

    $y_i = A(x_i)$;

    $\hat{z}_i \sim p_z(z)$;

    for *for* $t = 1$ *to* T **do**

        Optimize $\hat{z}_i = \hat{z}_i - \alpha_z \frac{\partial}{\partial z} \mathbb{E}(y_i, \hat{z}_i)$

$\mathcal{L}_G = \frac{1}{N} \sum_{i=1}^N \mathbb{E}_\theta(y_i, \hat{z}_i)$

Sample $(x_1^j, x_2^j)_{j=1}^{N_s}$ with (x_1^j, x_2^j) coming from the data or the Generator G_θ

$\mathcal{L}_A = \frac{1}{N_s} \sum_{j=1}^{N_s} \left[\left(\left\| A(x_1^j - x_2^j) \right\|_2 - \left\| x_1^j - x_2^j \right\|_2 \right)^2 \right]$

$\theta = \theta - \alpha_\theta \frac{\partial}{\partial \theta} (\mathcal{L}_G + \mathcal{L}_A)$

- requires an inner loop for each observation, very costly
- The second algorithm aims to learn the measurement function during the training,
- now, $A \longrightarrow A_\phi(.)$ with A a **function** represented by another neural network.

Algorithm 2

Algorithm 2: DCS Algorithm

Data: Input minibatchs of data $\{x_i\}_{i=1}^{N}$

initialize A_ϕ, G_θ, learning rates α_z, α_θ,α_ϕ initialization;

for $i = 1$ *to* N **do**

$\quad y_i = A_\phi(x_i)$;

$\quad \hat{z}_i \sim p_z(z)$;

$\quad$ **for** *for* $t = 1$ *to* T **do**

$\quad\quad$ Optimize $\hat{z}_i = \hat{z}_i - \alpha_z \frac{\partial}{\partial z} \mathbb{E}(y_i, \hat{z}_i)$

$\mathcal{L}_G = \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_\theta(y_i, \hat{z}_i)$

Sample $(x_1^j, x_2^j)_{j=1}^{N_s}$ with (x_1^j, x_2^j) coming from the data or the Generator G_θ

$\mathcal{L}_A = \frac{1}{N_s} \sum_{j=1}^{N_s} \left[\left(\left\| A_\phi(x_1^j - x_2^j) \right\|_2 - \left\| x_1^j - x_2^j \right\|_2 \right)^2 \right]$

$\theta = \theta - \alpha_\theta \frac{\partial}{\partial \theta} \mathcal{L}_G$

$\phi = \phi - \alpha_\phi \frac{\partial}{\partial \phi} \mathcal{L}_A$

- closed to Generative Adversarial Network frame work with G the Generator and D the discriminator
- The obtained results during our experiments are not relevant, G does not train correctly, need a lot of computation time to study the algorithm behaviour
- Our experiments focus on the first methods

Classic Compressed Sensing

Compressed Sensing with Generative Models

Deep Compressed Sensing

Experiments

Experiments description

- work on the data set Fashion Mnist more complex than the classical Mnist
- Comparison between Basis Pursuit results, LASSO and Compressed sensing with generative models
- We use VAE generative models with different possible architectures here

Example of images from Mnist

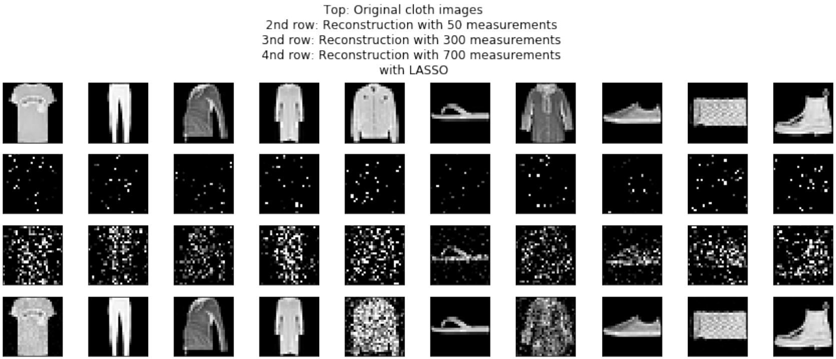
Training Image Example: Fashion MNIST (10 classes)

Reconstruction error by experiment

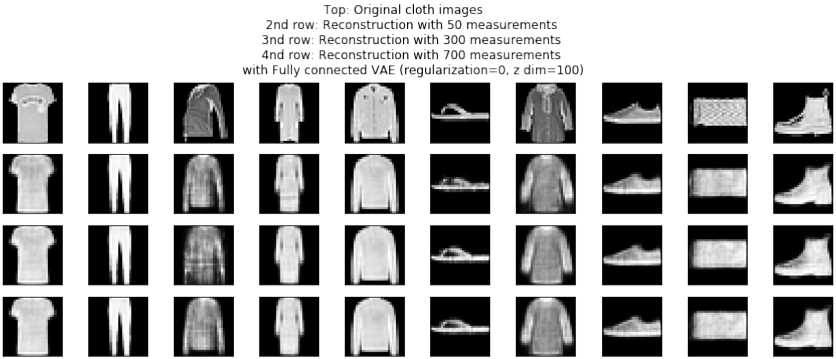
Reconstruction error obtained by experiment:

	m=50	m=300	m=700
Basis Pursuit	0.197316	0.141270	0.010251
LASSO	0.197908	0.136915	0.010131
VAE reg=0.1 (fc)	0.023475	0.051690	0.075682
VAE reg=0 (fc)	0.009879	0.007643	0.007189
VAE reg=0.1 (cnn)	0.025307	0.051965	0.076860
VAE reg=0, (cnn)	0.025307	0.051965	0.076860
Sparse-VAE (fc)	0.010684	0.007541	0.007273
Sparse-VAE (cnn)	0.043342	0.009443	0.008696

Qualitative results LASSO

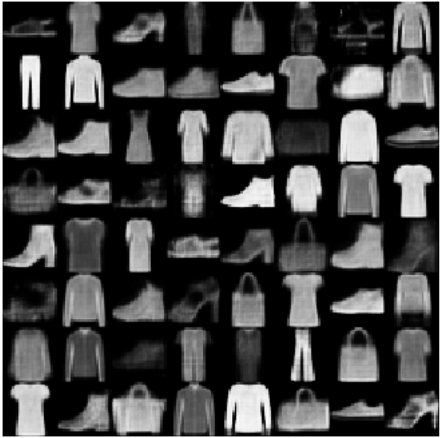


Qualitative results VAE



Generation with VAE

Training Image Example: Fashion MNIST (10 classes)



Observations

- VAE is better with a small number of measurements
- The Basis pursuit and LASSO needs a lot of measurements in order to perform correctly
- generative model capacity limits the performances even if the number of measurements increases

Thank You

for Listening.