COMS E6998-9: Algorithms for Massive Data (Fall'25)	Sep 29, 2025
Lecture 8: Compressed Sensing Extensions, NNS	
Instructor: *Alex Andoni*	Scribes: *Gillian Simpson*

1 Introduction

In the past few lectures, we have covered the fundamentals of compressed sensing. In this lecture, we consider faster compressed sensing techniques and their applications to machine learning.

2 Problem: Sparse Fourier Transform

Assume we are given measurements $y = Ax$, where $x \in \mathbb{R}^n$ is a signal of interest, and the goal is to recover x efficiently under the assumption that its Fourier transform $\hat{x} = Fx$ is k-sparse.

Let F be the discrete Fourier transform matrix, defined as:

$$F_{ij} = \frac{1}{\sqrt{n}} e^{-\frac{2\pi i}{n} ij}, \quad \text{where } n = 2^{\omega}$$

Theorem 1 (Candes, Tao '06)**.** *Let S be a subset of m indices, and define $A = (F^{-1})_S$, i.e., a submatrix of the inverse Fourier transform. Then A is $(O(k), 1/4)$-RIP, provided $|S| = O(k \log^4 n)$.*

Let $a \in \mathbb{R}^n$ be something of interest (for us to learn), and suppose that $\hat{a} = Fa$ is k-sparse (or close to it). Then we can think of $x = \hat{a}$ as the signal we want to recover. The measurement becomes:

$$y = Ax = (F^{-1}\hat{a})_S = a_S$$

The theorem guarantees that we can recover $x^* \approx x$ from y.

This implies that if the Fourier transform of a signal is k-sparse, then it can be approximately recovered from a small number of samples $|S|$ into a (i.e., never even touching most of entries of a). However, earlier approaches still required total time $> n$. The following result improves on this:

Note: The dependence $\log n$ on n typically arises due to noise. In the noiseless setting, the recovery may not require such dependence.

Theorem 2 (Hassanieh, Indyk, Katabi, Price '12)**.** *We can recover a k-sparse $\hat{a}$ from $m = O(k \log n \cdot \log(n/k))$ samples in $O(m)$ time.*

Here, recovery means computing an approximation $\hat{a}^*$ such that:

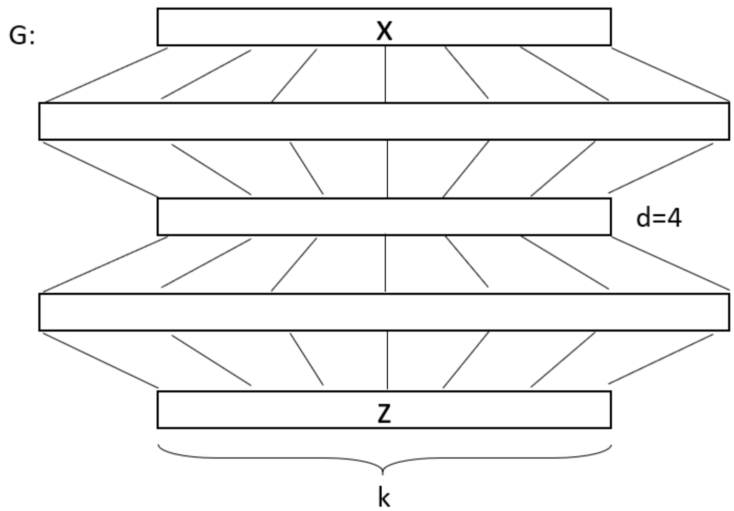
$$\|\hat{a}^* - \hat{a}\|_2 \leq O(1) \cdot \min_{\hat{a}' \text{ is } k\text{-sparse}} \|\hat{a}' - \hat{a}\|_2$$

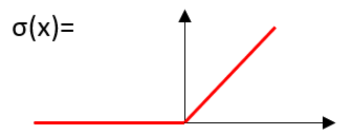
In earlier settings, the subset S of sampled rows was chosen uniformly at random. In this result, the sampling strategy is more structured.

2.1 Application to Machine Learning

Theorem 3 (Bora, Jalal, Price, Dimakis '17)**.** *Let $G : \mathbb{R}^k \to \mathbb{R}^n$ be a generative model represented by a depth-d neural network with $n^{O(1)}$ hidden units and ReLU activation function σ. Then, even if $x = G(z)$ is not k-sparse, it is possible to recover x approximately from compressed measurements.*

A visual representation of G and σ:





Here, G maps a low-dimensional vector z to a high-dimensional output x (e.g. an image). Suppose we observe $y = Ax + \eta$ with A as a random Gaussian matrix of size $m \times n$, where $m = O(kd \log n)$.

Define the recovery as:

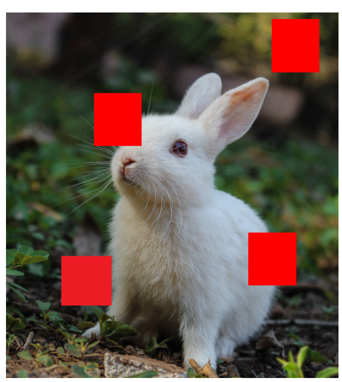
$$\hat{z} = \arg\min_z \|y - AG(z)\|_2$$

Then the recovered image $G(\hat{z})$ satisfies:

$$\|G(\hat{z}) - G(z)\|_2 \leq 6 \cdot \min_{z'} \|G(z') - G(z)\|_2 + 3\|\eta\|_2 + 2\varepsilon$$

This provides a robust approximation to the original signal/image $x = G(z)$.

Example 4 (Image In-Painting)**.** *Suppose parts of an image are missing (shown in red):*



Let G be a generative network for images. Suppose A selects the unmasked (non-red) pixel locations, so $y = AG(z)$. A is no longer Gaussian, so we cannot directly apply the previous theorem. However, it motivates us to apply the above algorithm (heuristically) to recover $\hat{z}$ as follows:

$$\hat{z} = \arg\min_{z'} \|y - AG(z')\|_2$$

The reconstructed image is then $G(\hat{z})$.

3 Problem: Nearest Neighbor Search (a.k.a. Vector Search)

Given a dataset $P \subseteq \mathbb{R}^d$ of n points and a query $q \in \mathbb{R}^d$, the goal is to return:

$$p^* = \arg\min_{p \in P} \|q - p\|_2$$

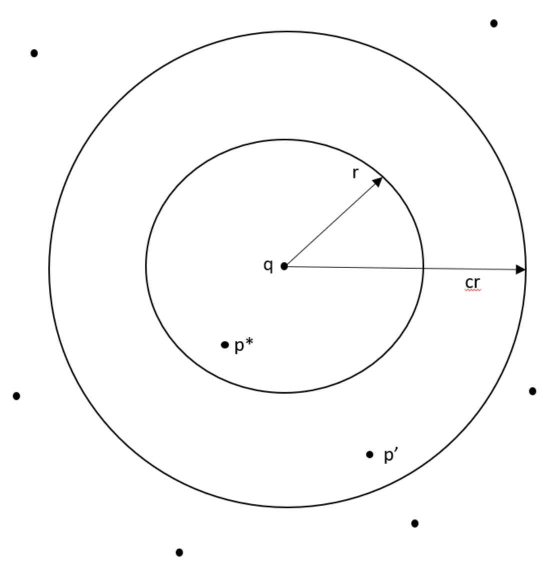
- **Naive approach:** $O(nd)$ query time
- **Goal:** Sublinear query time $\ll n$ and close to $O(n)$ space

Theorem 5. *It is impossible to achieve query time $n^{1-\varepsilon}$, space and preprocessing time n^2 unless the Strong Exponential-Time Hypothesis is false.*

Note: Strong Exponential-Time Hypothesis is a stronger version of $P \neq NP$, which asserts that certain problems cannot be solved in even exponential time.

We now consider an approximate variant. Fix approximation factor $c > 1$ and threshold radius $r > 0$. Preprocess P such that given query q:

- If $\exists p^* \in P$ with $\|q - p^*\|_2 \leq r$, return p' such that $\|q - p'\|_2 \leq c \cdot r$
- If no such p^* exists, the algorithm may return nothing.



Interpretation:

- Inside the smaller circle: definite yes (must return a near neighbor)
- Between the two circles: maybe
- Outside the larger circle: definite no

Often, the approximate algorithm can be interpreted as filtering algorithms. In partiuclar, most algorithms discussed can be modified such that we return a list $L \subseteq P$ such that:

- If $\|q - p\|_2 \leq r$, then $\Pr[p \in L] \geq 0.9$
- If $p \in L$, then $\|q - p\|_2 \leq cr$

Thus, we have a probabilistic guarantee:

$$\Pr[\text{return } p'] \geq 0.9$$

Theorem 6. *The c-Approximate Nearest Neighbor Search can be reduced to the c-Approximate Near Neighbor Search with only $O(\log n)$ overhead in space and time.*