Data Sparse Matrix Computation
Lecture 19 Sparse Recovery (Intro and Application)

Scribers: Louise Lee, Qiantong Xu, Lifan Wu

October 24, 2017

1 Introduction

In this section of the class, we will be dealing with the last big topic, sparse recovery. In the next few lessons, we will talk about algorithms to solve the problems that are raised in this lecture, compressed sensing, and low rank sparsity decompositions.

2 The problem

Given $A \in \mathbb{R}^{m \times n}$, $m < n$ or even $m \ll n$, and some $b \in \mathbb{R}^n$, we want to solve $Ax = b$. There are two possible cases:

1. There is no solution, for example if $b \notin range(A)$, or A is singular
2. There are an infinite number of solutions, for example if $b \in range(A)$, A full row-rank.

We are not interested in the first case, and will assume that A is full (row)-rank from now onwards unless explicitly stated.

Two examples of when this comes up are compressed sensing, and redundant representations.

2.1 Compressed sensing

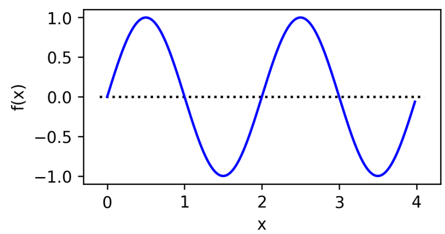
Think of x as an image (or object), A is an alteration of the image (or measurement operator), and b as the image taken (or measurements taken). We can only "view the object indirectly", since $m < n$ means less measurements can be taken than degrees of freedom in representing the image.

For example, A can be a blurring and downsampling operator. In this setting, many original images x can generate the same measurements b. A big question then arises as to how to pick the "proper" x from all possible images. We will answer this question in the next major section.

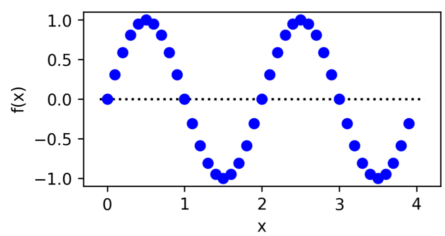
2.2 Redundant representations

Another useful way to think about sparse recovery is "redundant representations". b is something we care about, but want another representation for it.

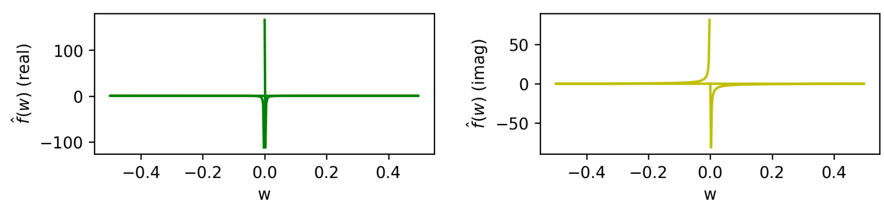
For example, b can be a 1-dimensional signal that looks like:



which we then sample to get:



Therefore, b is a time sampled signal. It may be more natural to look at Fb, the fourier transform of b, because it will be come much more sparse.



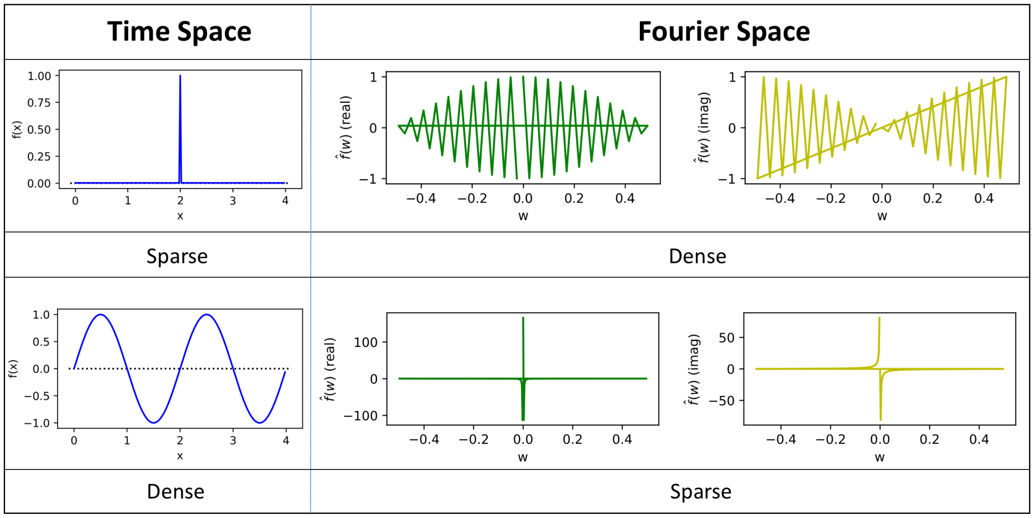
By looking at b in Fourier space, it can give insights about the signal as well as allow us to store it in a compressed way. We can increase our modeling power by choosing to representing the signal in different ways rather than just in time space or Fourier space.

However, a problem arises if we do not know if b or Fb is a better presentation. The following diagram shows two signals: the first row shows a signal that is sparse in time space but dense in Fourier space. The second row shows a signal that is dense in time space but sparse in Fourier space. In general, the sparsity of a signal in time space and Fourier space is always opposite to each other.

Therefore, it is not easy to decide which representation is better without looking at the signal. To address this problem, consider the system

$$b = \begin{bmatrix} I & F \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

That is, $b = x_1 + Fx_2$, where x_1 is the component of the signal represented in time space and Fx_2 is the part of the signal represented in Fourier space. A natural way to choose a "best" representation of b is to try to find a sparse x that satisfies the equation. That is, we add a constraint to maximise the number of zeroes in these overcomplete set of signals, and solve for this constraint on x.



More generally, given some descriptors A, where the rows are sample data and the columns are features, we want to use as few of them is possible. Therefore, the key to solving these two types of problems is to determine what a *best* x is, subject to the constraint that $Ax = b$.

3 A *better* x

We define $J(x)$ that evaluates the quality of a given solution x, where smaller $J(x)$ is better. Then, we want to solve

$$\min_x J(x)$$
$$\text{s.t. } Ax = b$$

Examples of possible J(x) that can be chosen include

1. $J(x) = \|x\|_2^2$, which is a common choice that will not be used in this class. There is a unique solution for this system: $x = A^\dagger b$, where $A^\dagger$ is the pseudoinverse of A.
2. $J(x) = \|Bx\|_2^2$ where B is positive definite. There is a unique solution:
$$x = (B^T B)^{-1} A^T (A(B^T B)^{-1} A^T)^{-1} b.$$

From now on, a good choice of x is one that is sparse. Recently, we want to solve

$$\min_x \|x\|_0$$
$$\text{s.t. } Ax = b$$

where $\|x\|_0$ is the number of non zeros in x. Note that $\|x\|_0$ is not a true norm. There no longer exists a closed form expression of the solution. This becomes a combinatorial problem that is hard to solve, and we will therefore not try to attack this directly.

This class will talk about several strategies to solve this problem. First, we consider what an alternative formulation of this problem might be without using $\|x\|_0$, since it is not a norm and difficult to work with.

4 $\ell 1$ norm

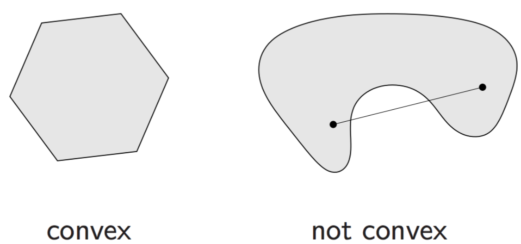
We want to find the function J such that

$$\begin{aligned} \min \quad & J(x) \\ \text{s.t.} \quad & Ax = b \end{aligned}$$

is easy to solve, and has a sparse solution.

Definition 4.1 (Convex Set)**.** A set Ω is convex if $\forall x_1, x_2 \in \Omega$ and $t \in [0,1]$, we have $x = tx_1 + (1-t)x_2 \in \Omega$.

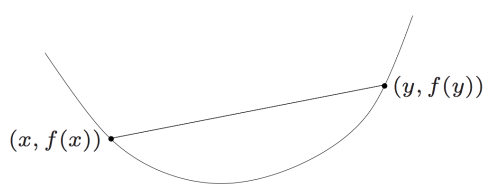
Visually speaking, if the line segment connecting any two points in a set still remains in the set, then the set is convex. Considering the following example. The one on the right hand side is not convex because the line segment is not fully contained in the set.



Definition 4.2 (Convex Function)**.** The function $J : \Omega \to \mathbb{R}$ is convex if $\forall x_1, x_2 \in \Omega$ and $t \in [0,1]$, $x = tx_1 + (1-t)x_2$ satisfies

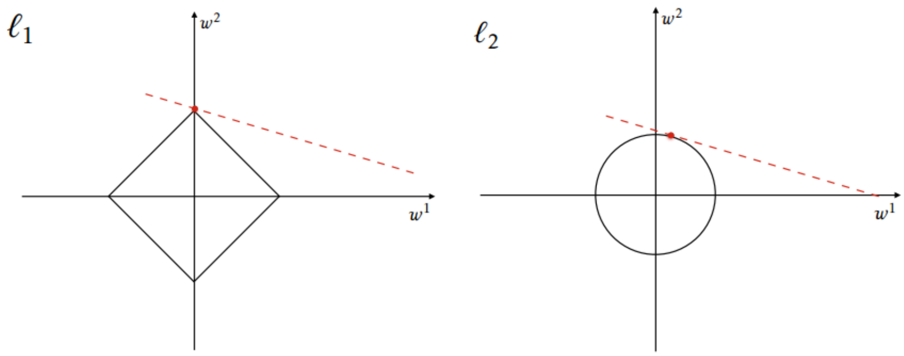
$$J(x) \le tJ(x_1) + (1-t)J(x_2)$$

This says that the function graph between any two points lies below the line connecting themselves. See the following graph as an illustration.



If J is strictly convex, then a unique solution exists to the minimization problem. $\|\cdot\|_2$ is such an example. Unfortunately, the sparsity is not ensured when we use $\ell 2$ norm. Alternatively, we may choose $J(x) = \|x\|_1$. Though $\|\cdot\|_1$ is not strictly convex, and hence the problem $\min \; J(x)$ s.t. $Ax = b$ may have more than one solution, the uniqueness fails only under some special circumstances. We will discuss it later. But minimizing $\ell 1$ norm has an appealing feature that the solution will be sparse. If we consider all x with unit $\|\cdot\|_2$, it turns out that the sparsest such x are the ones with smallest $\ell 1$ norm. Conversely, for those x's with unit $\ell 2$ norm, the least sparse one corresponds to the one with largest $\|\cdot\|_1$.

Now we want to see in detail why $\|\cdot\|_1$ is a desired objective function to work with. Recall that the set of solution to $Ax = b$ forms an affine hyperplane. Graphically, if our objective is to minimize $\|\cdot\|_2$, we draw a circle and keep enlarging its radius until it first hits the line (the graph on the right). The point is indeed the optimal solution. But it is dense unless the line is vertical or horizontal.



Let's go back to $\|\cdot\|_1$. Likewise, the optimal solutions occurs when the diamond hits the line $Ax = b$. The optimal solution points in the direction of the canonical basis vector, and thus sparse (consider the graph on the left as an example). We see that in 2-dimensional case, a unique solution exists unless the line $Ax = b$ lies $45°$, which corresponds to the matrix A with same entries (eg. $A = [1\ 1]$). Generally speaking, as long as the entries are not equal, the $\|\cdot\|_1$ will have a solution that is both sparse and unique.

Remark 4.1. We don't consider any q-norm with $q < 1$, because $\|\cdot\|_q$ is not convex for $q < 1$, and thus hard to solve.