I. Least Squares Optimization

Solving systems of equations using least squares

One of the most common situations where least squares problems arise is when we would like to "solve" systems of equations. If we have M equations with N unknowns, we can write this as

$$\begin{aligned} y_1 &= A_{1,1}x_1 + A_{1,2}x_2 + \cdots + A_{1,N}x_n \\ y_2 &= A_{2,1}x_1 + A_{2,2}x_2 + \cdots + A_{2,N}x_n \\ &\vdots \\ y_M &= A_{M,1}x_1 + A_{M,2}x_2 + \cdots + A_{M,N}x_n. \end{aligned}$$

Here, the y_m and the $A_{m,n}$ are known, and we wish to solve for $x_1, \dots, x_n$. We can write this much more compactly as

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \tag{1}$$

where $\boldsymbol{y} \in \mathbb{R}^M$, $\boldsymbol{x} \in \mathbb{R}^N$, and $\boldsymbol{A}$ is an $M \times N$ matrix.

Above I said we would like to "solve" this system – why the quotation marks? Well, in general, finding an $\boldsymbol{x}$ that satisfies (1) exactly may not be possible (when $M > N$). Alternatively, when $M \leq N$ there may be multiple solutions, and we must decide which one to choose.

In general, given $\boldsymbol{y}$, we want to find $\boldsymbol{x}$ in such a way that

1. when there is a unique solution, we return it;
2. when there is no solution, we return something reasonable;
3. when there are an infinite number of solutions, we choose one to return in a "smart" way.

A natural way to approach the problem that addresses the first two goals is to find an $\boldsymbol{x}$ such that the *residual*

$$\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}$$

is “small.” To translate this into something actionable, we need a mathematical notion of the “size” of the vector $\boldsymbol{r}$. This is exactly what a **norm** does for us. (See the technical details at the end of these notes for a quick overview of vector spaces and norms.) While there are many possible valid norms that we could use, the least squares approach involves trying to minimize $\|\boldsymbol{r}\|_2$, or equivalently, $\|\boldsymbol{r}\|_2^2$, where $\|\cdot\|_2$ is the standard Euclidean norm:

$$\|\boldsymbol{r}\|_2 = \sqrt{\sum_{m=1}^{M} r_m^2}.$$

Using the Euclidean norm (squared for convenience) to quantify our notion of the size of the residual, we obtain the least squares approach to solving the system in (1), that is, the optimization problem

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}} \ \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2. \tag{2}$$

We will soon describe how to solve this problem in general, but first we will consider a concrete example: *linear regression.*

Example: Regression

A fundamental problem in data science that we have already encountered is to estimate a function given point samples (that are possibly corrupted by noise). Recall that in this setting we observe pairs of points (x_m, y_m) for $m = 1, \ldots, M$, and want to find a function $f(x)$ such that

$$f(x_m) \approx y_m, \quad m = 1, \ldots, M.$$

Of course, the problem is not well-posed yet, since without any constraints on f, there are any number of functions for which $f(x_m) = y_m$ exactly. Thus, we typically specify a class that f belongs to. One way of doing this is by building f up out of a linear combination of some set of functions $\phi_n(\cdot)$:

$$f(x) = \sum_{n=1}^{N} \alpha_n \phi_n(x).$$

The functions ϕ_n could be polynomials, sinusoids, or anything else that might be appropriate given the application. We now fit our function by solving for the "best" coefficients $\alpha_1, \ldots, \alpha_N$. There is a classical complexity versus robustness trade-off in choosing the number N of functions that we are going to use to fit the data – generally speaking, letting N be large gives us a richer class of functions with more expressive power, but leads to a harder estimation problem requiring more data if we want our estimate to be accurate.

In the context of least squares, we will select the coefficients by optimizing the sum of the square of the difference of the observed value y_m and its prediction using the coefficients $\alpha_1, \ldots, \alpha_n$:

$$\left(y_m - \sum_{n=1}^{N} \alpha_n \phi_n(x_m) \right)^2.$$

To see how this is related to the least squares framework first described above, note that we can equivalently express our regression problem by putting it in matrix form. We form the $M \times N$ matrix $\boldsymbol{A}$ and the $N \times 1$ vector $\boldsymbol{\alpha}$:

$$\boldsymbol{A} = \begin{bmatrix} \phi_1(x_1) & \phi_2(x_1) & \cdots & \phi_N(x_1) \\ \phi_1(x_2) & \phi_2(x_2) & \cdots & \phi_N(x_2) \\ \vdots & & \ddots & \vdots \\ \phi_1(x_M) & \phi_2(x_M) & \cdots & \phi_N(x_M) \end{bmatrix} \qquad \boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix}$$

$\boldsymbol{A}$ maps a set of coefficients $\boldsymbol{\alpha} \in \mathbb{R}^N$ to a set of M predictions for the vector of observations $\boldsymbol{y} \in \mathbb{R}^M$. Finding the $\boldsymbol{\alpha}$ that minimizes the squared error is now reduced to the standard least squares problem:[1]

$$\underset{\boldsymbol{\alpha} \in \mathbb{R}^N}{\text{minimize}} \ \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\alpha}\|_2^2.$$

We will soon discuss the solution for this problem in general, but first let's look at an important special case: suppose that $\phi_1(x) = x$, and $\phi_2(x) = 1$. This corresponds to **linear regression**, so that our prediction will be of the form $f(x) = \alpha_1 x + \alpha_2$. In this case, we can write our optimization problem as

$$\underset{\alpha_1, \alpha_2 \in \mathbb{R}}{\text{minimize}} \ \sum_{m=1}^{M} (y_m - \alpha_1 x_m - \alpha_2)^2. \tag{3}$$

You hopefully recall from calculus that the minimum of a quadratic function will occur when the derivative is zero. Since (3) is quadratic with respect to both α_1 and α_2, we can find the minimum by taking

[1] This is exactly the same problem as before, but we have had to substitute $\boldsymbol{\alpha}$ for $\boldsymbol{x}$ to avoid confusion since $x_1, \ldots, x_M$ represent our sample locations.

partial derivatives with respect to both variables, setting these equal to zero, and solving for the minimizing α_1 and α_2.

Towards this end, let

$$g(\alpha_1, \alpha_2) = \sum_{m=1}^{M} (y_m - \alpha_1 x_m - \alpha_2)^2$$

and note that

$$\begin{aligned} \frac{\partial}{\partial \alpha_1} g(\alpha_1, \alpha_2) &= -2 \sum_{m=1}^{M} x_m (y_m - \alpha_1 x_m - \alpha_2) \\ \frac{\partial}{\partial \alpha_2} g(\alpha_1, \alpha_2) &= -2 \sum_{m=1}^{M} (y_m - \alpha_1 x_m - \alpha_2). \end{aligned}$$

Setting these both equal to zero and rearranging yields

$$\begin{aligned} \sum_{m=1}^{M} x_m y_m &= \alpha_1 \sum_{m=1}^{M} x_m^2 + \alpha_2 \sum_{m=1}^{M} x_m \\ \sum_{m=1}^{M} y_m &= \alpha_1 \sum_{m=1}^{M} x_m + M \alpha_2. \end{aligned}$$

We can write this as a 2 × 2 system of equations in matrix form as

$$\begin{bmatrix} \sum_{m=1}^{M} x_m^2 & \sum_{m=1}^{M} x_m \\ \sum_{m=1}^{M} x_m & M \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} \sum_{m=1}^{M} x_m y_m \\ \sum_{m=1}^{M} y_m \end{bmatrix}.$$

Thus, we can obtain the solution to our optimization problem, which we will denote by $(\widehat{\alpha}_1, \widehat{\alpha}_2)$, by simply inverting this system, i.e., computing

$$\begin{bmatrix} \widehat{\alpha}_1 \\ \widehat{\alpha}_2 \end{bmatrix} = \begin{bmatrix} \sum_{m=1}^M x_m^2 & \sum_{m=1}^M x_m \\ \sum_{m=1}^M x_m & M \end{bmatrix}^{-1} \begin{bmatrix} \sum_{m=1}^M x_m y_m \\ \sum_{m=1}^M y_m \end{bmatrix}.$$

We can express the solution to this system in closed form by explicitly computing the inverse. Using the notation

$$\bar{x} = \frac{1}{M} \sum_{m=1}^M x_m \qquad \bar{y} = \frac{1}{M} \sum_{m=1}^M y_m,$$

the solution to this system reduces to

$$\begin{bmatrix} \widehat{\alpha}_1 \\ \widehat{\alpha}_2 \end{bmatrix} = \frac{1}{\sum_{m=1}^M x_m^2 - M\bar{x}^2} \begin{bmatrix} \sum_{m=1}^M x_m y_m - M\bar{x}\bar{y} \\ \bar{y}\sum_{m=1}^M x_m^2 - \bar{x}\sum_{m=1}^M x_m y_m \end{bmatrix}.$$

Next time we will generalize this approach and show how to solve the general least squares problem of (2).

Linear Algebra Review I: Vector spaces and norms

> *Linear algebra has become as basic and as applicable as calculus, and fortunately it is easier.*
>
> – Gilbert Strang

Linear algebra is the branch of mathematics that deals with solving systems of equations, with matrices and vectors being the key objects of study. But what exactly is a vector? Two intuitive ways of thinking about a vector might come to mind. First, the kind of vector we encounter in solving a system of equations is simply a list of numbers. However, the other place you have likely encountered this idea is in Euclidean geometry or physics, where a vector typically refers to a (directed) line segment between two points. The fact that we can use the same word for both of these concepts is chiefly due to the revolutionary idea of René Descartes that we can describe geometry via their coordinates, i.e., a list of numbers (a vector).

Descartes initiated what might be called the "*algebraization*" of geometry: if we can describe geometry in terms of vectors, then we can reduce geometric problems to ones of algebra. Beginning in the 19th century, mathematics began trending in a different direction, leading to a "*geomertrization*" of algebra: extending the equivalence between geometry and vectors, we can apply geometric concepts to lists of numbers. It is now common to apply geometric notions such as length, distance, and angle from three-dimensional space to vectors that live in much higher-dimensional (even infinite-dimensional) spaces.

In order to do this, mathematicians needed to form a more abstract and precise definition of what we mean by a vector and the kind of sets of vectors where such geometric notions make sense. The basic building block here is the **vector space**. We will not worry too much about defining this in all of its abstract glory, but informally, a **vector space** $\mathcal{S}$ is a set of elements, called *vectors*, that has rules for adding vectors and multiplying them by scalars.[2]

These rules mostly just capture familiar properties that we would expect of addition (e.g., it is commutative and associative) and multiplication (e.g., distributive and associative). The most salient requirement is that the set $\mathcal{S}$ of vectors must be *closed* under vector addition and scalar multiplication, which simply means that adding two vectors (or multiplying a vector by a scalar) will produce another vector in $\mathcal{S}$. This requirement is called *linearity*, since it implies that we can take arbitrary linear combinations of vectors without producing nonsense, and as a result vector spaces are also often known as **linear vector spaces** or **linear spaces**.

The simplest example of a vector space, and the most important one for this course, is $\mathbb{R}^N$, i.e., the set of vectors consisting of lists of N real numbers, with the usual notions of vector (element-wise) addition and scalar multiplication (with real-valued scalars). To see the value of this more abstract definition, note that the following are also valid vector spaces:

- The set of infinite-length sequences.
- The set of polynomials of degree p.
- The set of continuous functions on the real line.
- The set of functions bandlimited to Ω.

[2] Most commonly, a scalar simply refers to a real or complex number.

An important detail to keep in mind is that not all sets of vectors actually qualify as a vector space – typically because the set fails to be *closed*. For example, the set of vectors in $\mathbb{R}^2$ that live within the unit circle is *not* a vector space. To see why not, think about whether all linear combinations of such vectors will also live within the unit circle.

While the definition of a vector space described above generalizes some of our intuition from Euclidean space, we gain much more by also defining a **norm** together with our vector space. A norm allows us to talk about the *length* of a vector or the *distance* between two vectors.[3]

Definition. A **norm** $\|\cdot\|$ on a vector space $\mathcal{S}$ is a mapping

$$\|\cdot\| \ : \ \mathcal{S} \to \mathbb{R}$$

with the following properties for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{S}$:

1. $\|\boldsymbol{x}\| \geq 0, \ \text{and} \quad \|\boldsymbol{x}\| = 0 \ \Leftrightarrow \ \boldsymbol{x} = \mathbf{0}$.
2. $\|\boldsymbol{x} + \boldsymbol{y}\| \ \leq \ \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$ (triangle inequality)
3. $\|a\boldsymbol{x}\| = |a| \cdot \|\boldsymbol{x}\|$ for any scalar a (homogeneity)

Other related definitions:

- The **length** of $\boldsymbol{x} \in \mathcal{S}$ is simply $\|\boldsymbol{x}\|$.
- The **distance** between $\boldsymbol{x}$ and $\boldsymbol{y}$ is $\|\boldsymbol{x} - \boldsymbol{y}\|$.
- A vector space in which we have defined a norm is called a **normed vector space**.

[3] A fancy mathematical way to say this is that a norm adds a layer of *topological structure* on top of the algebraic structure defining a vector space.

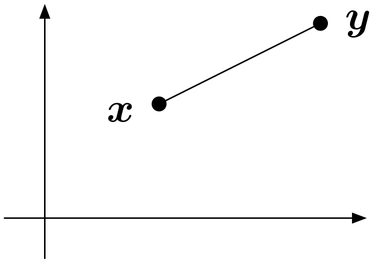
Examples:

1. $\mathcal{S} = \mathbb{R}^N$,

$$\|\boldsymbol{x}\|_2 = \left(\sum_{n=1}^{N} |x_n|^2\right)^{1/2}$$

This is called the "ℓ_2 norm", or "standard Euclidean norm"

In $\mathbb{R}^2$:



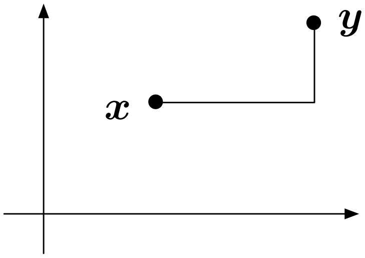
$$\|\boldsymbol{x}-\boldsymbol{y}\|_2 = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

2. $\mathcal{S} = \mathbb{R}^N$

$$\|\boldsymbol{x}\|_1 = \sum_{n=1}^{N} |x_n|$$

This is the "ℓ_1 norm" or "taxicab norm" or "Manhattan norm"

In $\mathbb{R}^2$:



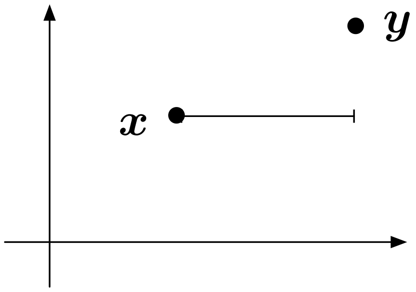
$$\|\boldsymbol{x} - \boldsymbol{y}\|_1 = |x_1 - y_1| + |x_2 - y_2|$$

3. $\mathcal{S} = \mathbb{R}^N$

$$\|\boldsymbol{x}\|_\infty = \max_{n=1,\ldots,N} |x_n|$$

This is the "ℓ_∞ norm" or "Chebyshev norm"

In $\mathbb{R}^2$:



$$\|\boldsymbol{x}-\boldsymbol{y}\|_\infty = \max\left(|x_1 - y_1|,\ |x_2 - y_2|\right)$$

4. $\mathcal{S} = \mathbb{R}^N$

$$\|\boldsymbol{x}\|_p = \left(\sum_{n=1}^{N} |x_n|^p\right)^{1/p} \qquad \text{for some } 1 \leq p < \infty$$

This is the "ℓ_p norm".

5. The same definitions extend straightforwardly to infinite sequences:
$\mathcal{S}$ = sequences (discrete-time signals) $x[n]$ indexed by the integers $n \in \mathbb{Z}$

$$\|x[n]\|_p = \left(\sum_{n=-\infty}^{\infty} |x[n]|^p\right)^{1/p}$$

It is easy to verify that the set of all discrete-time signals that have $\|\boldsymbol{x}\|_p < \infty$ is a normed vector space; we call this space ℓ_p.

6. $\mathcal{S}$ = continuous-time signals on the real line

$$\|x(t)\|_2 = \left(\int_{-\infty}^{\infty} |x(t)|^2 \ \mathrm{d}t \right)^{1/2}$$

This is called the L_2 norm.[4] In engineering, we often refer to $\|x(t)\|_2^2$ as the **energy** in the signal.

Similarly,

$$\|x(t)\|_p = \left(\int_{-\infty}^{\infty} |x(t)|^p \ \mathrm{d}t \right)^{1/p}$$

and

$$\|x(t)\|_\infty = \sup_{t \in \mathbb{R}} |x(t)|, \quad \text{where sup = ``least upper bound''}$$

Note that we are also using $\|\cdot\|_p$ for the discrete version of these norms, but I do not expect this will cause any confusion.

The set of continuous-time signals that have finite L_p norm are a normed vector space; we call this space $L_p(\mathbb{R})$.

[4] The L is for Lebesgue, the mathematician who formalized the modern theory of integration in the early 1900s.