Mathematical Foundations of Data Science

Georgia Tech ECE 4803 Fall 2020; Notes by M. Davenport. Last updated 22:29, August 21, 2020

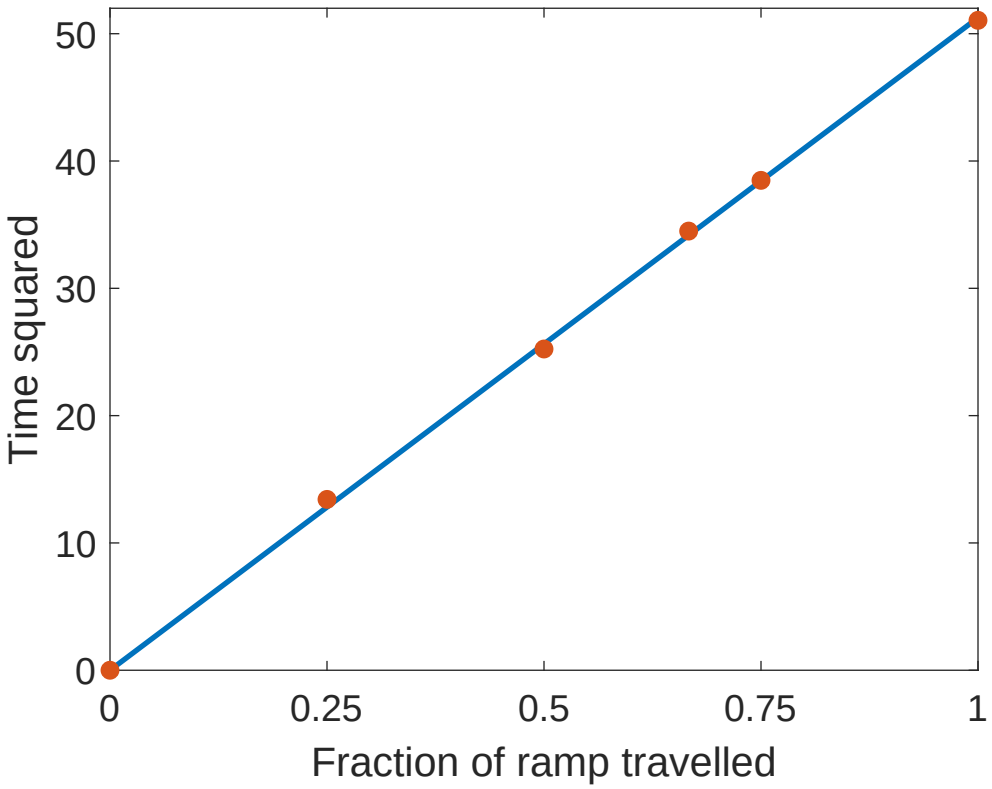
The field of **data science** revolves around a growing body of tools whose focus is the extraction of info *information* from *data*. In this course we will explore the mathematical foundations of this field.

Modern data science is built on two fundamental pillars: **probability** and **optimization**. Probability arises naturally when dealing with data because it provides us with principled ways to model and reason about uncertainty. This is critical, because no matter how well we understand a problem – no matter how good of a mathematical model we develop – when we have to deal with real-world data, our models can almost never predict this data *exactly*. This could be because of modelling error, measurement error, or both. In any case, probability provides the mathematical framework we need to handle such errors in a careful manner. However, probability on its own only gives us good ways to model our data. When it comes time to extract some kind of information from our data, we will see that most often, this can be posed as an *optimization problem*. We will soon be more formal about what this means, but a useful informal definition is something along the lines of "a problem where we wish to select the best element from some set of possibilities."

Let's give a concrete example of this that demonstrates a possible use of optimization in a real problem that also illustrates that data science has been influencing the world for perhaps longer than you might first think. A classic result in physics due to Galileo Galilei is that an object in free-fall experiences uniform acceleration in time. Specifically, the notion of uniform acceleration means that the change in speed should be linearly proportional to the amount of time that has passed, and as a consequence that the distance an object falls should be proportional to the square of the amount of time that has passed. In 1638, Galileo argued in his book *Dialogues Concerning Two New Sciences* that this *ought* to be the case on philosophical

grounds (by appealing to Aristotle). But Galileo went further: he argued that one can experimentally verify that bodies in nature really do in fact experience uniform acceleration!

Galileo provided a remarkably clear description of how this was done. He constructed an inclined ramp with marks indicating the points $\frac{1}{4}$, $\frac{1}{2}$, $\frac{2}{3}$, $\frac{3}{4}$, and all the way down the ramp. He then repeatedly rolled a ball down the ramp, recording the amount of time required for the ball to reach each point (as determined using a water clock that measured the volume of water dripping out of a small spout). Below I illustrate the results from a contemporary recreation of this experiment.[1]



[1] S. Straulino. "Reconstruction of Galileo Galilei's experiment: The inclined plane." *Physics Education*, 43(3) 316, 2008.

We can conclude from visual inspection of the results that the data clearly support Galileo's claim. However, note that there is not a *perfect* agreement between the data and the linear fit to the data that I have also included in the figure. This might raise many questions, including: How did I actually decide on the slope of this linear fit? Is there any sense in which one could determine the "best" fit? While not the approach taken by Galileo, this leads us directly to an illustration of the role played by optimization in data science.

In particular, suppose we observe pairs of points (x_m, y_m) for $m = 1, \ldots, M$, and want to find a function $f(x)$ of the form $f(x) = \alpha x$ such that

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

To pose this as an optimization problem, we must quantify what we mean by "$\approx$". There are many choices here, but a particulary common one is to measure our "approximation error" using the square of the difference between the observed value y_m and its prediction using $f(x_m)$, averaged over all the observations. Mathematically, we can write this as

$$\frac{1}{M} \sum_{m=1}^{M} (y_m - f(x_m))^2,$$

or in the case where $f(x)$ is of the form $f(x) = \alpha x$,

$$\frac{1}{M} \sum_{m=1}^{M} (y_m - \alpha x_m)^2. \tag{1}$$

We are finally ready to pose this as an optimization problem: if we would like to obtain the "best" linear fit to our data in the sense of squared error, we need to choose α to minimize (1). We can write

this as

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

You might already see how one might go about solving this, but for now it is enough to note that there are efficient ways to solve problems of this form, and this is precisely what I did in the figure above. While a simple example, much of modern data science is just an elaboration on solutions to this basic problem.

Mathematical optimization

We have just encountered our first optimization problem of the course. Optimization problems arise any time we have a collection of elements and wish to select the "best" one (according to some criterion). The process of casting a real world problem as being one of mathematical optimization consists of three main components

1. a set of variables, often called **decision variables**, that we have control over;
2. an **objective function** that maps the decision variables to some quality that we want to maximize (goodness of fit, profit, etc.) or some cost that we want to minimize (error, loss, etc.); and
3. a **constraint set** that dictates restrictions on the decision variables imposed by physical limitations, budgets on resources, design requirements, etc.

In its most general form, we can express such an optimization problem mathematically as

$$\underset{\boldsymbol{x}}{\text{minimize}} \ f(\boldsymbol{x}) \ \text{ subject to } \ \boldsymbol{x} \in \mathcal{X}, \tag{3}$$

where $f : \mathcal{X} \to \mathbb{R}$ is our objective function and $\mathcal{X}$ is our constraint set. Compare this with the problem described above in (2).

In order to solve this optimization problem, we must find an $\widehat{\boldsymbol{x}} \in \mathcal{X}$ such that

$$f(\widehat{\boldsymbol{x}}) \leq f(\boldsymbol{x}) \text{ for all } \boldsymbol{x} \in \mathcal{X}. \tag{4}$$

We call an $\widehat{\boldsymbol{x}}$ satisfying (4) a **minimizer** of f in $\mathcal{X}$, and a **solution** to the optimization problem (3).

By convention, we will focus only on *minimization* problems, noting that $\widehat{\boldsymbol{x}}$ *maximizes* f in $\mathcal{X}$ if and only if $\widehat{\boldsymbol{x}}$ minimizes $-f$ in $\mathcal{X}$ — thus any maximization problem can be easily turned into an equivalent minimization problem.

There are a number of fundamental questions that arise when considering an optimization problem of the form (3). Our primary interest will be in developing efficient procedures for computing a/the solution to (3). However, we will also need to address more fundamental questions along the way, such as when we can guarantee that a solution even exists, and if so, when we can expect it to be unique. We will begin by exploring these questions in the context of a concrete problem that is ubiquitous in data science: least squares optimization.