Machine Learning
the Fundamentals

William J. Deuschle

Harvard College
Cambridge, Massachusetts
April, 2019

Contents

1 Machine Learning Introduction **1**

1.1 What is Machine Learning (not)? 1

1.2 The Machine Learning Framework 1

1.3 This Book's Notation 2

2 Linear Regression **4**

2.1 Introduction and Motivation 4

2.2 Technical 5

Chapter 1

Machine Learning Introduction

1.1 What is Machine Learning (not)?

There is a great deal of misunderstanding about what machine learning is, fueled by its recent success and subsequent media coverage. While its applications have been and will continue to be extraordinarily powerful under the right circumstances, it's important to gain some sense of where and why the tools presented in this book will be applicable. Broadly, machine learning is the application of statistical, mathematical, and numerical techniques to deriving some form of knowledge from data. This 'knowledge' may afford us some sort of summarization, visualization, grouping, or even predictive power over data sets.

With all that said, it seems most appropriate to start with a discussion of what machine learning is ***not***. First and foremost, it is not an end goal itself. It is not a solution problems that people have not yet been able to figure out on their own. It is not nor will it ever be a replacement for critical thought and methodical, procedural work in data science.

Indeed, machine learning is merely a loose collection of disciplines and tools. Where the lines begin that separate machine learning from statistics or mathematics or probability theory or any other handful of fields that it draws on are not clear. To that end, you should not treat this book so much as a synopsis of the basics of machine learning as you should a collection of tools that can be applied to a certain subset of problems.

From reading this book, you will hopefully come away with a sense of the strengths and weaknesses of the tools presented herein. Even more importantly, you will hopefully gain an understanding of how these tools are situated within the fabric of problems that machine learning purports to solve. To that end, it will be useful to develop a framework for thinking about the types of problems that we will be working on. That way, as we continue to add new tools and techniques to our repertoire, we will always have a clear view of the context in which we can expect to use them. This will not only create a nice categorization of the different practices in machine learning, it will also help motivate why these techniques exist in the first place.

1.2 The Machine Learning Framework

TODO: add cube images

Here for the first time we get to see the 'Machine Learning Cube'. The purpose of this cube is to describe the domain of problems we will encounter, and it will be a useful way to delineate the techniques we will apply to different types of problems. Understanding the different facets of the cube will aid you in understanding machine learning as a whole, and can even give you automatic

intuition about techniques that you have never encountered before. Let's describe the features of the cube.

Our cube has three axes. On the first axis we will put the domain of our data. The domain of our data can take on one of two forms: discrete or continuous. We can actually both describe both the domain of our inputs and outputs, and we can even have mixtures from each domain, but to keep it simple for now, we will just consider our data as belonging to only one domain. Discrete, or categorical data, is data that can only fall into one of n specific classes. For example: male or female, integer values between 1 and 10, or different states in the U.S. are all examples of categorical data.

The second axis of the cube is reserved for the statistical nature of the machine learning technique that we will apply to it. Specifically, it will fall into one of two broad categories: **probabilistic** or **non-probabilistic** techniques. Probabilistic techniques are those for which we incorporate our data using some form of statistical distribution or summary. In general, we are then able to discard some or all of our data once we have finished tuning our probabilistic model. In contrast, non-probabilistic techniques are those that use the data directly to perform some action. A very common and general example of this is comparing how close a new data point is to other points in your existing data set. Non-probabilistic techniques potentially make fewer assumptions, but they do require that you keep around some or all of your data. They are also potentially slower techniques at runtime because they may require touching all of the data in your dataset to perform some action. These are a very broad set of guidelines for the distinction between probabilistic and non-probabilistic techniques - you should expect to see some exceptions and even some techniques that fit into both of these categories to some degree. Having a sense for their general benefits and drawbacks is useful, and you will gain more intuition about the distinction as we begin to explore different machine learning techniques.

The third and final axis of the cube describes the type of training we will use. There are two major classes of machine learning techniques: those that use **supervised** training and those that use **unsupervised** training. In fact, these two classes of techniques are so important to describing the field of machine learning that we will divide this textbook into parts dedicated to techniques found within each of these categories. A supervised technique is one for which we get to observe a data set of both the inputs and the outputs ahead of time, to be used for training. For example, we might be given a data set about weather conditions and crop production over the years. Then, we could train a machine learning model that learns a relationship between the input data (weather) and output data (crop production). The implication here is that given new input data, we will be able to predict the unseen output data. An unsupervised technique is one for which we only get a data set of 'inputs' ahead of time. In fact, we don't even need to consider these as inputs anymore: we can just consider them to be a set of data points that we wish to describe somehow. Unsupervised techniques revolve around clustering or otherwise describing our data.

We will see examples of all of these as we progress throughout the book, and you will gain an intuition for where different types of data and techniques fall in our cube. Eventually, given just the information of a technique being for continuous data, operating non-probabilistically, in an unsupervised setting, will give you a great amount of detail for what type of method you are dealing with.

1.3 This Book's Notation

Note: this is an ongoing thing that I will be updating as I write new chapters

The machine learning community uses a number of different conventions, and learning to de-

cipher the different versions of those conventions is important to understanding work done in the greater community. For this book, we will try to stick to a standard notation that we will define here. In addition to mathematical and statistical notation, we will also describe the conventions used in this book for explaining and breaking up different concepts.

Mathematical and Statistical Notation

While we will always describe the dimensionality of our variables in specific cases, we will often describe generic variables while explaining different techniques. Boldface variables ($\mathbf{x}$) represent vectors, capital boldface characters (X) represent matrices, and standard typeface variables (x) describe scalars.

Statistical distributions will most often be described in terms of their probability distribution function (PDF), e.g. $Y \sim \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$. Alternatively, in the case of a well known probability distribution, we will describe those in terms of their standard summary, e.g. $Y \sim \mathcal{N}(\mu, \sigma^2)$

Textbook Specific Notation

We have also introduced a few conventions to make consumption of this material easier to understand. We have boxes dedicated to definitions, explaining techniques in the context of the ML Framework Cube, and for explaining common misconceptions:

Definition 1.3.1 (Definition Explanation): You will find definitions in these gray boxes.

ML Framework Cube: ML Framework Cube

You will find ML Framework Cube explanations in these blue wrapped boxes.

⋆ You will find explanations for commonly misunderstood concepts in these red wrapped boxes.

Chapter 2

Linear Regression

A major component of machine learning, the one that most people associate with ML, is dedicated to making predictions under uncertain conditions. Given some input data, we would like to produce a target output. In this chapter, we're going to focus on the case where our prediction is a continuous, real number. This type of problem is known as **regression**.

2.1 Introduction and Motivation

We can imagine many situations where regression is useful:

1. Predicting a person's height given the height of their parents.
2. Predicting the probability that someone pays back a loan given their credit history.
3. Predicting what time a package will arrive given current weather and traffic conditions.

We're specifically going to focus on **linear regression**, which means that our goal is to find some linear combination of the $x_1, ..., x_D$ values that predict our target y.

Definition 2.1.1 (Linear Regression): Suppose we have an input $\mathbf{x} \in \mathbb{R}^D$ and a continuous target $y \in \mathbb{R}$. Linear regression determines weights $w_i \in \mathbb{R}$ that balance the values of x_i to produce y:

$$y = w_0 + w_1 x_1 + ... + w_D x_D \tag{2.1}$$

⋆ Notice w_0 in the expression above, which doesn't have a corresponding x_0 value. This is known as the *bias* term. If you consider the definition of a line $y = mx + b$, the bias term is comparable to the intercept b. It can account for a general trend in our data, such as if all of our target y values are greater than 50.

All of these follow a similar formula: a data input $\mathbf{x}$ gets transformed into prediction y. For example, consider 10 year old Sam. She is curious about how tall she will be when she grows up. She has a data set of parents' heights and the final heights of their children. The inputs $\mathbf{x}$ are:

$$x_1 = \text{height of mother (cm)}$$
$$x_2 = \text{height of father (cm)}$$

Using linear regression, she determines the weights $\mathbf{w}$ to be:

$$\mathbf{w} = [34, 0.39, 0.33]$$

Sam's mother is 165 cm tall and her father is 185 cm tall. Using the results of the linear regression solution, Sam solves for her expected height:

$$\text{Sam's height} = 34 + 0.39(165) + 0.33(185) = \mathbf{159.4\ cm}$$

ML Framework Cube: Linear Regression

Let's inspect the categories linear regression falls into for our ML framework cube. First, as we've already stated, linear regression deals with a **continuous** input and output domain. Second, our goal is to make predictions on future data points, and to construct something capable of making those predictions we first need a labeled data set of inputs and outputs. This makes linear regression is a **supervised** technique. Third and finally, linear regression is a special case when it comes to being **probabilistic or non-probabilistic**. Depending on our interpretation, it can be either one! We will explain how this works later in the chapter.

Domain	***Training***	***Probabilistic***
Continuous	Supervised	Yes / No

2.2 Technical

The most basic form of linear regression is a simple weighted combination of the input variables $\mathbf{x}$, which you will often see written as:

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + ... + w_D x_D \tag{2.2}$$

TODO: add something about the bias trick here

Let's think about what this means. Your algorithm is provided with a collection of data points x_i, and it weights those values using w_i to produce an output value y. For example, in the example of Sam's height above, we weighted the height of her mother with the factor 0.39, the height of her father as 0.33, and had a bias term (shift) of 34. Then the natural question arises: how did we find the appropriate values for $\mathbf{w}$?

That is the entire goal of linear regression, and the remaining focus of this chapter.