1

Introduction and Motivation

1.1 Finding Words for Intuitions

At an intuitive level, machine learning is about designing algorithms that learn from data. The challenge is that the concepts and words are slippery, and a particular component of the machine learning system can be abstracted to different mathematical concepts. For example, the word "algorithm" is used in at least two different senses in the context of machine learning. In the first sense, we use the phrase "machine learning algorithm" to mean a system that makes predictions based on input data. We refer to these algorithms as *predictors*. In the second sense, we use the exact same phrase "machine learning algorithm" to mean a system that adapts some internal parameters of the predictor so that it performs well on future unseen input data. Here we refer to this adaptation as *training* a predictor.

predictors

training

The first part of this book describes the mathematical concepts needed to talk about the three main components of a machine learning system: data, models, and learning. We will briefly outline these components here, and we will revisit them again in Chapter 8 once we have the mathematical language under our belt. Adding to the challenge is the fact that the same English word could mean different mathematical concepts, and we can only work out the precise meaning via the context. We already remarked about the overloaded use of the word "algorithm", and the reader will be faced with other such phrases. We advise the reader to use the idea of "type checking" from computer science and apply it to machine learning concepts. Type checking allows the reader to sanity check whether the equation that they are considering contains inputs and outputs of the correct type, and whether they are mixing different types of objects.

While not all data is numerical it is often useful to consider data in a number format. In this book, we assume that the *data* has already been appropriately converted into a numerical representation suitable for reading into a computer program. In this book, we think of data as vectors. As another illustration of how subtle words are, there are three different ways to think about vectors: a vector as an array of numbers (a computer science view), a vector as an arrow with a direction and magnitude (a

data

data as vectors.

physics view), and a vector as an object that obeys addition and scaling (a mathematical view).

model

What is a *model*? Models are simplified versions of reality, which capture aspects of the real world that are relevant to the task. Users of the model need to understand what the model does not capture, and hence obtain an appreciation of the limitations of it. Applying models without knowing their limitations is like driving a vehicle without knowing whether it can turn left or not. Machine learning algorithms adapt to data, and therefore their behavior will change as it learns. Applying machine learning models without knowing their limitations is like sitting in a self-driving vehicle without knowing whether it has encountered enough left turns during its training phase. In this book, we use the word “model” to distinguish between two schools of thought about the construction of machine learning predictors: the probabilisitic view and the optimization view. The reader is referred to Domingos (2012) for a more general introduction to the five schools of machine learning.

learning

We now come to the crux of the matter, the *learning* component of machine learning. Assume we have a way to represent data as vectors and that we have an appropriate model. We are interested in training our model based on data so that it performs well on unseen data. Predicting well on data that we have already seen (training data) may only mean that we found a good way to memorize the data. However, this may not generalize well to unseen data, and in practical applications we often need to expose our machine learning system to situations that it has not encountered before. We use numerical methods to find good parameters that “fit” the model to data, and most training methods can be thought of as an approach analogous to climbing a hill to reach its peak. The peak of the hill corresponds to a maximization of some desired performance measure. The challenge is to design algorithms that learn from past data but generalizes well.

Let us summarize the main concepts of machine learning:

- We use domain knowledge to represent data as vectors.
- We choose an appropriate model, either using the probabilisitic or optimization view.
- We learn from past data by using numerical optimization methods with the aim that it performs well on unseen data.

1.2 Two Ways To Read This Book

We can consider two strategies for understanding the mathematics for machine learning:

- Building up the concepts from foundational to more advanced. This is often the preferred approach in more technical fields, such as mathematics. This strategy has the advantage that the reader at all times is able to rely on their previously learned definitions, and there are no murky hand-wavy arguments that the reader needs to take on faith. Unfortunately, for a practitioner many of the foundational concepts are not particularly interesting by themselves, and the lack of motivation means that most foundational definitions are quickly forgotten.

- Drilling down from practical needs to more basic requirements. This goal-driven approach has the advantage that the reader knows at all times why they need to work on a particular concept, and there is a clear path of required knowledge. The downside of this strategy is that the knowledge is built on shaky foundations, and the reader has to remember a set of words for which they do not have any way of understanding.

This book is split into two parts, where Part I lays mathematical foundations and Part II applies the concepts from Part I to a set of basic machine learning problems.

Part I is about Mathematics

We represent numerical data as vectors and represent a table of such data as a matrix. The study of vectors and matrices is called *linear algebra*, which we introduce in Chapter 2. The collection of vectors as a matrix is also described there. Given two vectors, representing two objects in the real world, we want to be able to make statements about their similarity. The idea is that vectors that are similar should be predicted to have similar outputs by our machine learning algorithm (our predictor). To formalize the idea of similarity between vectors, we need to introduce operations that take two vectors as input and return a numerical value representing their similarity. This construction of similarity and distances is called *analytic geometry* and is discussed in Chapter 3. In Chapter 4, we introduce some fundamental concepts about matrices and *matrix decomposition*. It turns out that operations on matrices are extremely useful in machine learning, and we use them for representing data as well as for modeling.

linear algebra

analytic geometry

matrix decomposition

We often consider data to be noisy observations of some true underlying signal, and hope that by applying machine learning we can identify the signal from the noise. This requires us to have a language for quantifying what noise means. We often would also like to have predictors that allow us to express some sort of uncertainty, e.g., to quantify the confidence we have about the value of the prediction for a particular test data point. Quantification of uncertainty is the realm of *probability theory* and is covered in Chapter 6. Instead of considering a predictor as a single function, we could consider predictors to be probabilistic models, i.e., models describing the distribution of possible functions.

probability theory

To apply hill-climbing approaches for training machine learning models, we need to formalize the concept of a gradient, which tells us the direction which to search for a solution. This idea of the direction to search

Table 1.1 The four pillars of machine learning

	Supervised	Unsupervised
Continuous	Regression (Chapter 9)	Dimensionality reduction (Chapter 10)
Categorical	Classification (Chapter 12)	Density estimation (Chapter 11)

calculus

optimization

is formalized by *calculus*, which we present in Chapter 5. How to use a sequence of these search directions to find the top of the hill is called *optimization*, which we introduce in Chapter 7.

It turns out that the mathematics for discrete categorical data is different from the mathematics for continuous real numbers. Most of machine learning assumes continuous variables, and except for Chapter 6 the other chapters in Part I of the book only discuss continuous variables. However, for many application domains, data is categorical in nature, and naturally there are machine learning problems that consider categorical variables. For example, we may wish to model sex (male/female). Since we assume that our data is numerical, we encode sex as the numbers -1 and 1 for male and female, respectively. However, it is worth keeping in mind when modeling that sex is a categorical variable, and the actual difference in value between the two numbers should not have any meaning in the model. This distinction between continuous and categorical variables gives rise to different machine learning approaches.

Part II is about Machine Learning

four pillars of machine learning

The second part of the book introduces *four pillars of machine learning* as listed in Table 1.1. The rows in the table distinguish between problems where the variable of interest is continuous or categorical. We illustrate how the mathematical concepts introduced in the first part of the book can be used to design machine learning algorithms. In Chapter 8, we restate the three components of machine learning (data, models and parameter estimation) in a mathematical fashion. In addition, we provide some guidelines for building experimental setups that guard against overly optimistic evaluations of machine learning systems. Recall that the goal is to build a predictor that performs well on future data.

supervised learning

label

The terms “supervised” and “unsupervised” (the columns in Table 1.1) learning refer to the question of whether or not we provide the learning algorithm with labels during training. An example use case of *supervised learning* is when we build a classifier to decide whether a tissue biopsy is cancerous. For training, we provide the machine learning algorithm with a set of images and a corresponding set of annotations by pathologists. This expert annotation is called a *label* in machine learning, and for many supervised learning tasks it is obtained at great cost or effort. After the classifier is trained, we show it an image from a new biopsy and hope that it can accurately predict whether the tissue is cancerous. An example use case of unsupervised learning (using the same cancer biopsy problem) is if we want to visualize the properties of the tissue around which we have found cancerous cells. We could choose two particular features of these images and plot them in a scatter plot. Alternatively we could use all the features and find a two dimensional representation that approximates all the features, and plot this instead. Since this type of machine learning task does not provide a label during training, it is called *unsupervised learning*. The second part of the book provides a brief overview of two fundamental supervised (*regression* and *classification*) and unsupervised (*dimensionality reduction* and *density estimation*) machine learning problems.

unsupervised learning

regression

classification

dimensionality reduction

density estimation

Of course there are more than two ways to read this book. Most readers learn using a combination of top-down and bottom-up approaches, sometimes building up basic mathematical skills before attempting more complex concepts, but also choosing topics based on applications of machine learning. Chapters in Part I mostly build upon the previous ones, but the reader is encouraged to skip to a chapter that covers a particular gap the the reader's knowledge and work backwards if necessary. Chapters Part II are loosely coupled and are intended to be read in any order. There are many pointers forward and backward between the two parts of the book to assist the reader in finding their way.

1.3 Exercises and Feedback

We provide some exercises in Part I, which can be done mostly by pen and paper. For Part II we provide programming tutorials (jupyter notebooks) to explore some properties of the machine learning algorithms we discuss in this book.

We appreciate that Cambridge University Press strongly supports our aim to democratize education and learning by making this book freely available for download at

`https://mml-book.com`

where you can also find the tutorials, errata and additional materials. You can also report mistakes and provide feedback using the URL above.