**COS 511: Theoretical Machine Learning**

Lecturer: Rob Schapire
Scribe: Rob Schapire

Lecture #1
February 4, 2014

---

# 1 What is Machine Learning?

Machine learning studies computer algorithms for learning to do stuff. We might, for instance, be interested in learning to complete a task, or to make accurate predictions, or to behave intelligently. The learning that is being done is always based on some sort of observations or data, such as examples (the most common case in this course), direct experience, or instruction. So in general, machine learning is about learning to do better in the future based on what was experienced in the past.

The emphasis of machine learning is on *automatic* methods. In other words, the goal is to devise learning algorithms that do the learning automatically without human intervention or assistance. The machine learning paradigm can be viewed as "programming by example." Often we have a specific task in mind, such as spam filtering. But rather than program the computer to solve the task directly, in machine learning, we seek methods by which the computer will come up with its own program based on examples that we provide.

Machine learning is a core subarea of artificial intelligence. It is very unlikely that we will be able to build any kind of intelligent system capable of any of the facilities that we associate with intelligence, such as language or vision, without using learning to get there. These tasks are otherwise simply too difficult to solve. Further, we would not consider a system to be truly intelligent if it were incapable of learning since learning is at the core of intelligence.

Although a subarea of AI, machine learning also intersects broadly with other fields, especially statistics, but also mathematics, physics, theoretical computer science and more.

# 2 Examples of Machine Learning Problems

There are many examples of machine learning problems. Much of this course will focus on classification problems in which the goal is to categorize objects into a fixed set of categories. Here are several examples:

- optical character recognition: categorize images of handwritten characters by the letters represented
- face detection: find faces in images (or indicate if a face is present)
- spam filtering: identify email messages as spam or non-spam
- spoken language understanding: within the context of a limited domain, determine the meaning of something uttered by a speaker to the extent that it can be classified into one of a fixed set of categories
- medical diagnosis: diagnose a patient as a sufferer or non-sufferer of some disease

- classifying or predicting customer/user behavior: predict, for instance, which customers will respond to a particular promotion, or which users will click on a particular ad

- weather prediction: predict, for instance, whether or not it will rain tomorrow.

(In this last case, we most likely would actually be more interested in estimating the *probability* of rain tomorrow.)

Although much of what we will talk about will be about classification problems, there are other important learning problems. In classification, we want to categorize objects into fixed categories. In regression, on the other hand, we are trying to predict a real value. For instance, we may wish to predict *how much* it will rain tomorrow. Or, we might want to predict how much a house will sell for.

A richer learning scenario is one in which the goal is actually to behave intelligently, or to make intelligent decisions. For instance, a robot needs to learn to navigate through its environment without colliding with anything. To use machine learning to make money on the stock market, we might treat investment as a classification problem (will the stock go up or down) or a regression problem (how much will the stock go up), or, dispensing with these intermediate goals, we might want the computer to learn directly how to decide to make investments so as to maximize wealth. A final example is game playing where the goal is for the computer to learn to play well through experience.

## 3 Goals of Machine Learning Research

The primary goal of machine learning research is to develop general purpose algorithms of practical value. Such algorithms should be efficient. As usual, as computer scientists, we care about time and space efficiency. But in the context of learning, we also care a great deal about another precious resource, namely, the amount of data that is required by the learning algorithm.

Learning algorithms should also be as general purpose as possible. We are looking for algorithms that can be easily applied to a broad class of learning problems, such as those listed above.

Of primary importance, we want the result of learning to be a prediction rule that is as accurate as possible in the predictions that it makes.

Occasionally, we may also be interested in the interpretability of the prediction rules produced by learning. In other words, in some contexts (such as medical diagnosis), we want the computer to find prediction rules that are easily understandable by human experts.

As mentioned above, machine learning can be thought of as "programming by example." What is the advantage of machine learning over direct programming? First, the results of using machine learning are often more accurate than what can be created through direct programming. The reason is that machine learning algorithms are data driven, and are able to examine large amounts of data. On the other hand, a human expert is likely to be guided by imprecise impressions or perhaps an examination of only a relatively small number of examples.

Also, humans often have trouble expressing what they know, but have no difficulty labeling items. For instance, it is easy for all of us to label images of letters by the character represented, but we would have a great deal of trouble explaining how we do it in precise terms.

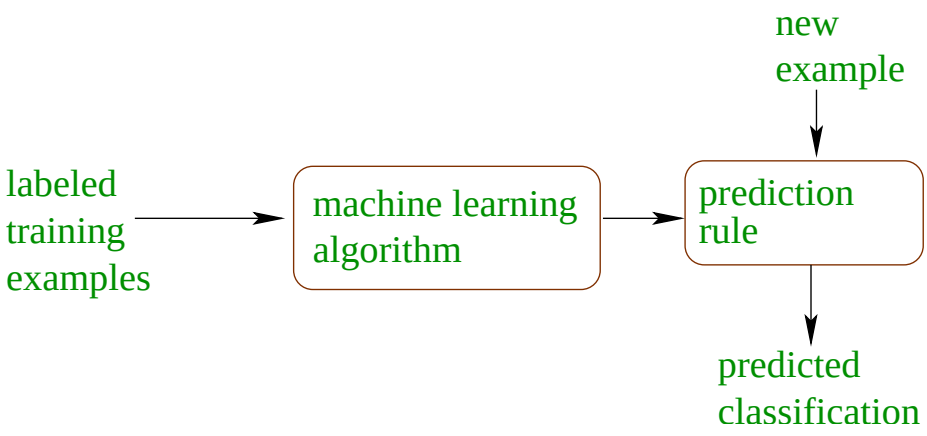


Figure 1: Diagram of a typical learning problem.

Another reason to study machine learning is the hope that it will provide insights into the general phenomenon of learning. Some of the questions that might be answered include:

- What are the intrinsic properties of a given learning problem that make it hard or easy to solve?
- How much do you need to know ahead of time about what is being learned in order to be able to learn it effectively?
- Why are "simpler" hypotheses better?

This course is focused on *theoretical* aspects of machine learning. Theoretical machine learning has much the same goals. We still are interested in designing machine learning algorithms, but we hope to analyze them mathematically to understand their efficiency. It is hoped that theoretical study will provide insights and intuitions, if not concrete algorithms, that will be helpful in designing practical algorithms. Through theory, we hope to understand the intrinsic difficulty of a given learning problem. And we also attempt to explain phenomena observed in actual experiments with learning algorithms.

This course emphasizes the study of mathematical models of machine learning, as well as the design and analysis of machine learning algorithms. Topics include:

- the number of random examples needed to learn;
- the theoretical understanding of practical algorithms, including boosting and support-vector machines;
- online learning from non-random examples (including portfolio selection);
- estimating a probability distribution from samples;
- game theory and its connection to learning.

The two most closely related computer science courses at Princeton are 402 (focusing on core areas of artificial intelligence) and 424 (focusing on techniques for the effective use of data, including machine learning, statistics and data mining). In comparison to 511 which focuses only on the theoretical side of machine learning, both of these offer a broader and more general introduction to machine learning — broader both in terms of the topics covered, and in terms of the balance between theory and applications.

| example | label |
|---|---|
| *train* | |
| ant | − |
| bat | + |
| dolphin | − |
| leopard | + |
| sea lion | − |
| zebra | + |
| shark | − |
| mouse | + |
| chicken | − |
| *test* | |
| tiger | |
| tuna | |
| platypus | |

Figure 2: A tiny learning problem.

# 4 A Typical Learning Problem

In a typical learning problem, such as optical character recognition, our goal is to create a system that can read handwritten characters. But rather than attacking the problem directly, we start by gathering a large collection of examples of images of characters which are labeled as to the letter or digit that they represent. We then feed these examples to a general purpose learning algorithm which in turn produces a prediction rule capable (we hope) of classifying new images. The entire process is depicted in Figure 1.

To get a feeling for learning, we looked first at the learning problem shown in Figure 2. Here, examples are labeled positive ("+") or negative ("−"). In this case, the pattern is that "land mammals" are positive, and others negative, although many other suggestions were made in class. Actually, the pattern is kind of hazy, since it might be, for instance, that the positive examples are animals that don't live in the ocean and don't lay eggs. Thus, test examples "tiger" and "tuna" are positive and negative, respectively, but we can only guess the correct label of "platypus" (an egg-laying mammal that spends much of its life in streams and rivers).

The second example is shown in Figure 3. Initially, examples were presented as animal names, and the problem seemed difficult. When the animal names are rewritten in terms of the position in the alphabet of each letter of each name, the pattern becomes more evident, namely, that an example is positive if and only if the third letter has an odd position in the alphabet (for instance, the third letter of "elephant" is "e" which is the fifth letter of the alphabet, so "elephant" is a positive example).

There are some things to notice about this experiment. First of all, more data is better. For instance, we were unsure whether platypus should be a positive or negative example. However, if we had more training data, including, for instance, live-bearing non-mammalian animals (like a scorpion), or other mammals that live in rivers rather than the ocean, we might have been able to better define what the unknown pattern is. Also, many of the other competing suggestions for the first example problem could have been quickly eliminated with more data.

| example | | | | | | | | | | label |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | *train* | | | | | | |
| aardvark | → | 1 | 1 | 18 | 4 | 22 | 1 | 18 | 11 | − |
| cow | → | 3 | 15 | 23 | | | | | | + |
| giraffe | → | 7 | 9 | 18 | 1 | 6 | 6 | 5 | | − |
| termite | → | 20 | 5 | 18 | 13 | 9 | 20 | 5 | | − |
| oyster | → | 15 | 25 | 19 | 20 | 5 | 18 | | | + |
| dove | → | 4 | 15 | 22 | 5 | | | | | − |
| spider | → | 19 | 16 | 9 | 4 | 5 | 18 | | | + |
| dog | → | 4 | 15 | 7 | | | | | | + |
| elephant | → | 5 | 12 | 5 | 16 | 8 | 1 | 14 | 20 | + |
| | | | | *test* | | | | | | |
| rabbit | → | 18 | 1 | 2 | 2 | 9 | 20 | | | |
| frog | → | 6 | 18 | 15 | 7 | | | | | |
| kangaroo | → | 11 | 1 | 14 | 7 | 1 | 18 | 15 | 15 | |

Figure 3: A second toy learning problem. Examples were intially presented as animal names as in the left column, but later rewritten with the corresponding number of each letter of each animal name, as shown to the right of each name.

Second, it seems very natural to try to find a pattern that is consistent with the data, i.e., a rule that "explains" all of the observed training data. In practice, of course, it might not be possible to find a rule that is perfectly consistent with all of the data, but we still might find it desirable to find a rule that makes as few mistakes as possible.

Thirdly, it seems natural to seek a rule that is not only consistent, but also as simple as possible. For instance, no one was satisfied with a rule that says, on the first problem, that an animal is positive if and only if it begins with a "b" or an "l" or a "z" or an "m" (which was not far from one of the suggestions for this problem). The notion of "simplicity" is tied up with our prior beliefs about what kind of rule we are expecting. When the point of view on the second problem was changed by altering the representation, the kind of rules that seemed simple and natural also changed.

So in sum, there are three conditions that must be met for learning to succeed. First, we need enough data. Second, we need to find a rule that makes a low number of mistakes on the training data. And third, we need that rule to be as simple as possible. Note that the last two requirements are typically in conflict with one another: we sometimes can only find a rule that makes a low number of mistakes by choosing a rule that is more complex, and conversely, choosing a simple rule can sometimes come at the cost of allowing more mistakes on the training data. Finding the right balance is perhaps the most central problem of machine learning.

The notion that simple rules should be preferred is often referred to as "Occam's razor."

## 5 Learning Models

To study machine learning mathematically, we need to formally define the learning problem. This precise definition is called a *learning model*. A learning model should be rich enough to capture important aspects of real learning problems, but simple enough to study the

problem mathematically. As with any mathematical model, simplifying assumptions are unavoidable.

A learning model should answer several questions:

- What is being learned?
- How is the data being generated? In other words, where does it come from?
- How is the data presented to the learner? For instance, does the learner see all the data at once, or only one example at a time?
- What is the goal of learning in this model?

## 6 Definitions

Before getting to our first learning model, we will need some definitions. An *example* (sometimes also called an *instance*) is the object that is being classified. For instance, in OCR, the images are the examples.

Usually, an example is described by a set of *attributes*, also known as *features* or *variables* or *dimensions*. For instance, in medical diagnosis, a patient might be described by attributes such as gender, age, weight, blood pressure, body temperature, etc.

The *label* or *class* is the category that we are trying to predict. For instance, in OCR, the labels are the possible letters or digits being represented. During training, the learning algorithm is supplied with *labeled examples*, while during testing, only *unlabeled examples* are provided.

To make things as simple as possible, we will often assume that only two labels are possible that we might as well call 0 and 1. We also will make the simplifying assumption that there is a mapping from examples to labels. This mapping is called a *concept*. Thus, a concept is a function of the form $c : X \to \{0, 1\}$ where $X$ is the space of all possible examples called the *domain* or *instance space*. A collection of concepts is called a *concept class*. We will often assume that the examples have been labeled by an unknown concept from a *known* concept class.

## 7 The Consistency Model

Our first learning model, called the consistency model, is rather unrealistic, but it is intuitive, and it is a good place to start. Many of the ideas that come up will also be of value later. The model captures the intuitive idea that the prediction rule that is derived from a set of examples should be consistent with their observed labelings.

We say that a concept class $\mathcal{C}$ is learnable in the consistency model if there is an algorithm $A$ which, when given any set of labeled examples $(x_1, y_1), \ldots, (x_m, y_m)$, where $x_i \in X$ and $y_i \in \{0, 1\}$, finds a concept $c \in \mathcal{C}$ that is consistent with the examples (so that $c(x_i) = y_i$ for all $i$), or says (correctly) that there is no such concept. Moreover, we are especially interested in finding efficient algorithms in this model.

Next time, we will look at examples of concept classes that are or are not learnable in this model.