1 Introduction to Machine Learning

Machine learning is a set of tools that, broadly speaking, allow us to "teach" computers how to perform tasks by providing examples of how they should be done. For example, suppose we wish to write a program to distinguish between valid email messages and unwanted spam. We could try to write a set of simple rules, for example, flagging messages that contain certain features (such as the word "viagra" or obviously-fake headers). However, writing rules to accurately distinguish which text is valid can actually be quite difficult to do well, resulting either in many missed spam messages, or, worse, many lost emails. Worse, the spammers will actively adjust the way they send spam in order to trick these strategies (e.g., writing "vi@gr@"). Writing effective rules — and keeping them up-to-date — quickly becomes an insurmountable task. Fortunately, machine learning has provided a solution. Modern spam filters are "learned" from examples: we provide the learning algorithm with example emails which we have manually labeled as "ham" (valid email) or "spam" (unwanted email), and the algorithms learn to distinguish between them automatically.

Machine learning is a diverse and exciting field, and there are multiple ways of defining it:

1. **The Artifical Intelligence View.** Learning is central to human knowledge and intelligence, and, likewise, it is also essential for building intelligent machines. Years of effort in AI has shown that trying to build intelligent computers by programming all the rules cannot be done; automatic learning is crucial. For example, we humans are not born with the ability to understand language — we learn it — and it makes sense to try to have computers learn language instead of trying to program it all it.

2. **The Software Engineering View.** Machine learning allows us to program computers by example, which can be easier than writing code the traditional way. As applications and large-scale systems become more and more complex, the use of learning tools to fine-tune or customize (personalize) system performance will be more and more common.

3. **The Stats View.** Machine learning is the marriage of computer science and statistics, the application of computational techniques to statistical problems. While statistics has focused historically on trying to maximize what one can infer from relatively small amounts of data (since experiments are often costly to perform), modern methods are concerned with learning models from large amounts of complex data, for which sophisticated models and fast algorithms are needed.

Often, machine learning methods are broken into two phases:

1. **Training:** A model is learned from a collection of **training data**.

2. **Application:** The model is used to make decisions about some new **test data**.

For example, in the spam filtering case, the training data constitutes email messages labeled as ham or spam, and each new email message that we receive (and wish to classify) is test data. However, there are other ways in which machine learning is used as well.

Machine learning is a large field, and learning problems abound, often in very different forms, discrete and continuous, with various degrees of labeled supervisory information. Some of the main types of machine learning are

1. **Supervised Learning,** in which the training data is labeled with the correct answers, e.g., "spam" or "ham." The two most common types of supervised learning are **classification** (where the outputs are discrete labels, as in spam filtering) and **regression** (where the outputs are real-valued).

2. **Unsupervised learning,** in which we are given a collection of unlabeled data, which we wish to analyze and discover patterns within. The two most important examples are **dimension reduction** and **clustering**.

3. **Reinforcement learning,** in which an agent (e.g., a robot or controller) seeks to learn the optimal actions to take based on the state of the world, and hence the consequences of past actions.

There are many other types of machine learning as well, most of which we do not have time to cover in this introductory course. Some key examples include

1. **Semi-supervised learning**, in which only a subset of the training data is labeled;

2. **Active learning**, in which obtaining data is expensive, and so an algorithm must determine which training data to acquire;

3. **Transfer/Meta/Few-Shot Learning**, in which models learned from tasks with large training sets, or many tasks with moderate amounts of training data, can be used to help constrain learning on related problems that have relatively small datasets;

4. **Structured Prediction**, in which we learn models for complex structured objects, such as graphs or images, with many dimensions and complex dependencies among the predicted variables;

5. **Time-series forecasting**, such as in financial markets;

6. **Anomaly detection** such as used for fault-detection in factories and in surveillance;

7. **Deep Learning** concerns the design and application of multi-layer artificial neural networks. They've enjoyed great success in a wide range of supervised learning tasks, including speech and image recognition.

1.1 A Simple Problem

Figure 1 shows a 1D regression problem. The goal is to fit a 1D curve to a few points. Which curve is best to fit these points? There are infinitely many curves that fit the data, and, because the data might be noisy, we might not even want to fit the data precisely. Hence, machine learning requires that we make certain choices:

1. How do we parameterize the model we fit? For the example in Figure 1, how do we parameterize the curve; should we try to explain the data with a linear function, a quadratic, or a sinusoidal curve?

2. What criteria (e.g., objective function) do we use to judge the quality of the fit? For example, when fitting a curve to noisy data, it is common to measure the quality of the fit in terms of the squared error between the data we are given and the fitted curve. When minimizing the squared error, the resulting fit is usually called a least-squares estimate.

3. Some types of models and some model parameters can be very expensive to optimize well. How long are we willing to wait for a solution, or can we use approximations (or hand-tuning) instead?

4. Ideally we want to find a model that will provide useful predictions in future situations. That is, although we might learn a model from *training data*, we ultimately care about how well it works on future *test data*. When a model fits training data well, but performs poorly on test data, we say that the model has *overfit* the training data; i.e., the model has fit properties of the input that are not particularly relevant to the task at hand (e.g., Figures 1 (top row and bottom left)). Such properties are refered to as *noise*. When this happens we say that the model does not *generalize* well to the test data. Rather it produces predictions on the test data that are much less accurate than you might have hoped for given the fit to the training data.

Machine learning provides a wide selection of options by which to answer these questions, along with the vast experience of the community as to which methods tend to be successful on a particular class of data-set. Some more advanced methods provide ways of automating some of these choices, such as automatically selecting between alternative models, and there is some beautiful theory that assists in gaining a deeper understanding of learning. In practice, there is no single “silver bullet” for all learning. Using machine learning in practice requires that you make use of your own prior knowledge and experimentation to solve problems. But with the tools of machine learning, you can do amazing things!

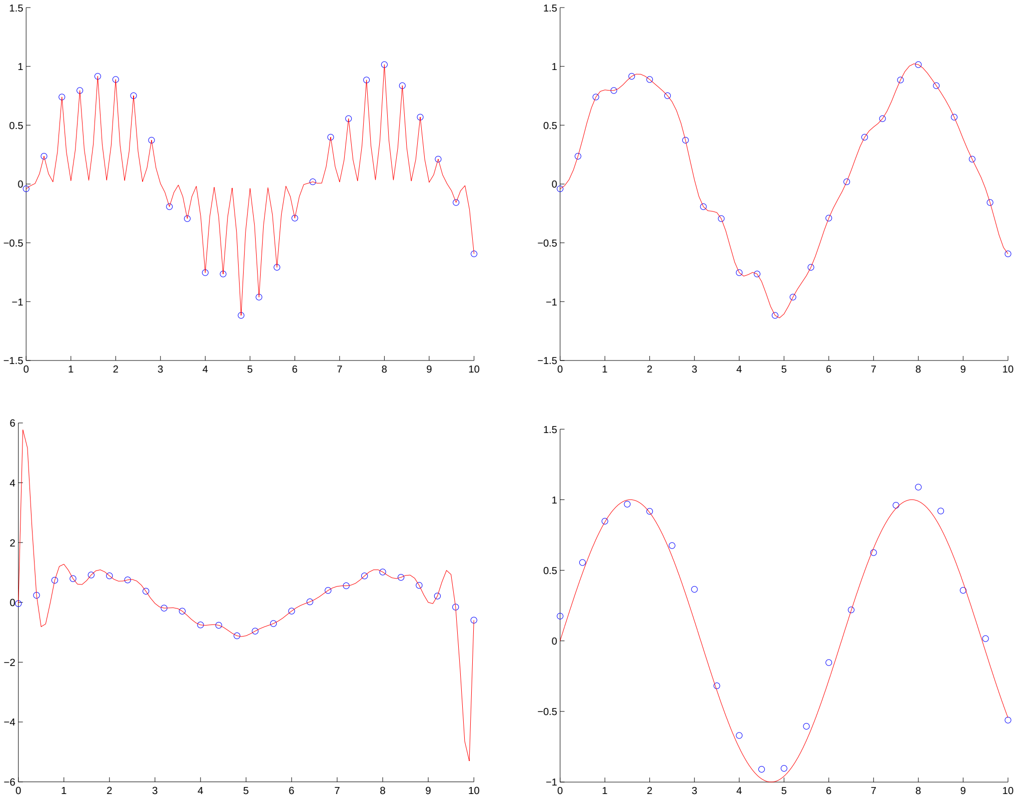

Figure 1: A simple regression problem. The blue circles are measurements (the training data), and the red curves are possible fits to the data. There is no one "right answer." The solution we prefer depends on the problem. Ideally we want to find a model that provides good predictions for new inputs (i.e., locations on the x-axis for which we had no training data). We will often prefer simple, smooth models like the one shown in the lower right.