Classification

David Wagner is the primary author of this chapter.

Machine learning is a class of techniques for automatically finding patterns in data and using it to draw inferences or make predictions. You have already seen linear regression, which is one kind of machine learning. This chapter introduces a new one: *classification*.

Classification is about learning how to make predictions from past examples. We are given some examples where we have been told what the correct prediction was, and we want to learn from those examples how to make good predictions in the future. Here are a few applications where classification is used in practice:

- For each order Amazon receives, Amazon would like to predict: ***is this order fraudulent?*** They have some information about each order (e.g., its total value, whether the order is being shipped to an address this customer has used before, whether the shipping address is the same as the credit card holder's billing address). They have lots of data on past orders, and they know which of those past orders were fraudulent and which weren't. They want to learn patterns that will help them predict, as new orders arrive, whether those new orders are fraudulent.

- Online dating sites would like to predict: ***are these two people compatible?*** Will they hit it off? They have lots of data on which matches they've suggested to their customers in the past, and they have some idea which ones were successful. As new customers sign up, they'd like to make predictions about who might be a good match for them.

- Doctors would like to know: ***does this patient have cancer?*** Based on the measurements from some lab test, they'd like to be able to predict whether the particular patient has cancer. They have lots of data on past patients, including their lab measurements and whether they ultimately developed cancer, and from that, they'd like to try to infer what measurements tend to be characteristic of cancer (or non-cancer) so they can diagnose future patients accurately.

- Politicians would like to predict: ***are you going to vote for them?*** This will help them focus fundraising efforts on people who are likely to support them, and focus get-out-the-vote efforts on voters who will vote for them. Public databases and commercial databases have a lot of information about most people: e.g., whether they own a home or rent; whether they live in a rich neighborhood or poor neighborhood; their interests and hobbies; their shopping habits; and so on. And political campaigns have surveyed some voters and found out who they plan to vote for, so they have some examples

where the correct answer is known. From this data, the campaigns would like to find patterns that will help them make predictions about all other potential voters.

All of these are classification tasks. Notice that in each of these examples, the prediction is a yes/no question – we call this *binary classification*, because there are only two possible predictions.

In a classification task, each individual or situation where we'd like to make a prediction is called an *observation*. We ordinarily have many observations. Each observation has multiple *attributes*, which are known (for example, the total value of the order on Amazon, or the voter's annual salary). Also, each observation has a *class*, which is the answer to the question we care about (for example, fraudulent or not, or voting for you or not).

When Amazon is predicting whether orders are fraudulent, each order corresponds to a single observation. Each observation has several attributes: the total value of the order, whether the order is being shipped to an address this customer has used before, and so on. The class of the observation is either 0 or 1, where 0 means that the order is not fraudulent and 1 means that the order is fraudulent. When a customer makes a new order, we do not observe whether it is fraudulent, but we do observe its attributes, and we will try to predict its class using those attributes.

Classification requires data. It involves looking for patterns, and to find patterns, you need data. That's where the data science comes in. In particular, we're going to assume that we have access to *training data*: a bunch of observations, where we know the class of each observation. The collection of these pre-classified observations is also called a training set. A classification algorithm is going to analyze the training set, and then come up with a classifier: an algorithm for predicting the class of future observations.

Classifiers do not need to be perfect to be useful. They can be useful even if their accuracy is less than 100%. For instance, if the online dating site occasionally makes a bad recommendation, that's OK; their customers already expect to have to meet many people before they'll find someone they hit it off with. Of course, you don't want the classifier to make too many errors — but it doesn't have to get the right answer every single time.