Overfitting and Underfitting

A common danger in machine learning is *overfitting*—producing a model that performs well on the data you train it on but generalizes poorly to any new data. This could involve learning *noise* in the data. Or it could involve learning to identify specific inputs rather than whatever factors are actually predictive for the desired output.

The other side of this is *underfitting*—producing a model that doesn't perform well even on the training data, although typically when this happens you decide your model isn't good enough and keep looking for a better one.

In Figure 11-1, I've fit three polynomials to a sample of data. (Don't worry about how; we'll get to that in later chapters.)

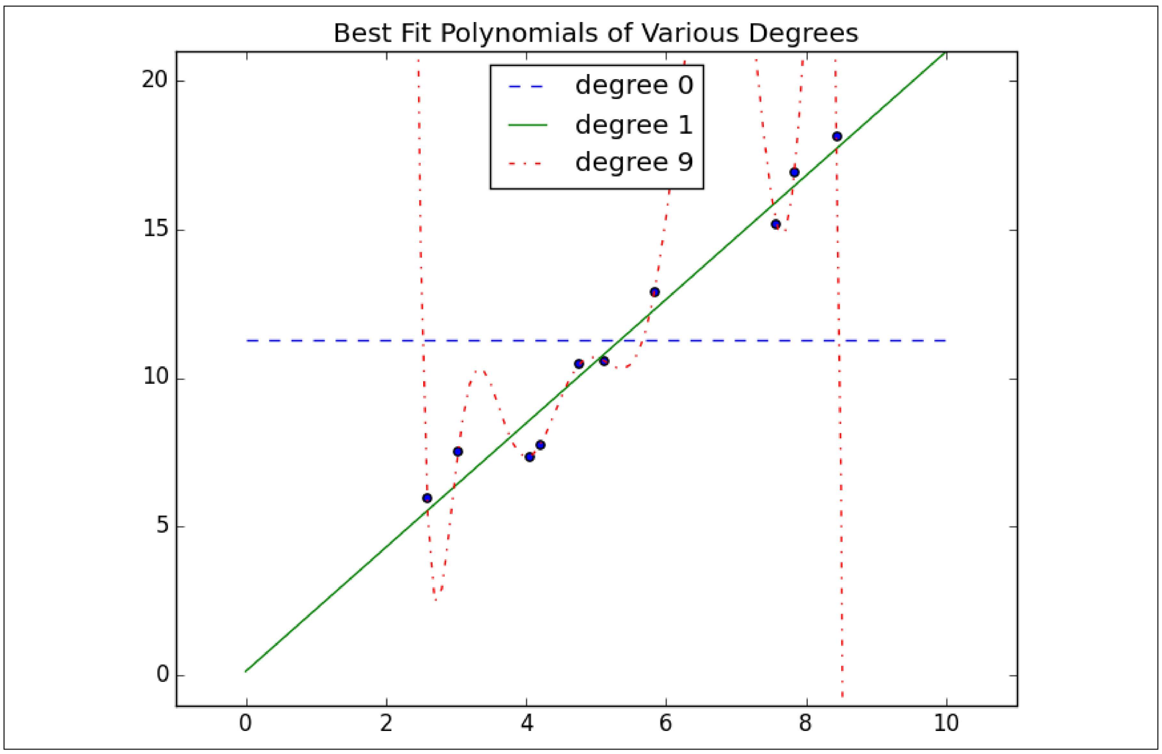

Figure 11-1. Overfitting and underfitting

The horizontal line shows the best fit degree 0 (i.e., constant) polynomial. It severely *underfits* the training data. The best fit degree 9 (i.e., 10-parameter) polynomial goes through every training data point exactly, but it very severely *overfits*; if we were to pick a few more data points, it would quite likely miss them by a lot. And the degree 1 line strikes a nice balance; it's pretty close to every point, and—if these data are representative—the line will likely be close to new data points as well.

Clearly, models that are too complex lead to overfitting and don't generalize well beyond the data they were trained on. So how do we make sure our models aren't too