vious ratings. It starts with random factors, then tunes these factors to make them more and more aligned with how viewers have rated movies before, until they are ultimately able to *predict* how viewers rate movies in general. The factors we end up with may not be as intuitive as 'comedy content', and in fact can be quite subtle or even incomprehensible. After all, the algorithm is only trying to find the best way to predict how a viewer would rate a movie, not necessarily explain to us how it is done. This algorithm was part of the winning solution in the million-dollar competition.

1.1.1 Components of Learning

The movie rating application captures the essence of learning from data, and so do many other applications from vastly different fields. In order to abstract the common core of the learning problem, we will pick one application and use it as a metaphor for the different components of the problem. Let us take credit approval as our metaphor.

Suppose that a bank receives thousands of credit card applications every day, and it wants to automate the process of evaluating them. Just as in the case of movie ratings, the bank knows of no magical formula that can pinpoint when credit should be approved, but it has a lot of data. This calls for learning from data, so the bank uses historical records of previous customers to figure out a good formula for credit approval.

Each customer record has personal information related to credit, such as annual salary, years in residence, outstanding loans, etc. The record also keeps track of whether approving credit for that customer was a good idea, i.e., did the bank make money on that customer. This data guides the construction of a successful formula for credit approval that can be used on future applicants.

Let us give names and symbols to the main components of this learning problem. There is the input $\mathbf{x}$ (customer information that is used to make a credit decision), the unknown target function $f: \mathcal{X} \to \mathcal{Y}$ (ideal formula for credit approval), where $\mathcal{X}$ is the input space (set of all possible inputs $\mathbf{x}$), and $\mathcal{Y}$ is the output space (set of all possible outputs, in this case just a yes/no decision). There is a data set $\mathcal{D}$ of input-output examples $(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_N, y_N)$, where $y_n = f(\mathbf{x}_n)$ for $n = 1, ..., N$ (inputs corresponding to previous customers and the correct credit decision for them in hindsight). The examples are often referred to as data points. Finally, there is the learning algorithm that uses the data set $\mathcal{D}$ to pick a formula $g: \mathcal{X} \to \mathcal{Y}$ that approximates f. The algorithm chooses g from a set of candidate formulas under consideration, which we call the hypothesis set $\mathcal{H}$. For instance, $\mathcal{H}$ could be the set of all linear formulas from which the algorithm would choose the best linear fit to the data, as we will introduce later in this section.

When a new customer applies for credit, the bank will base its decision on g (the hypothesis that the learning algorithm produced), not on f (the ideal target function which remains unknown). The decision will be good only to the extent that g faithfully replicates f. To achieve that, the algorithm