Chapter 5

Three Learning Principles

The study of learning from data highlights some general principles that are fascinating concepts in their own right. Having gone through the mathematical analysis and empirical illustrations of the first few chapters, we have a good foundation from which to articulate some of these principles and explain them in concrete terms.

In this chapter, we will discuss three principles. The first one is related to the choice of model and is called Occam's razor. The other two are related to data; sampling bias establishes an important principle about obtaining the data, and data snooping establishes an important principle about handling the data. A genuine understanding of these principles will protect you from the most common pitfalls in learning from data, and allow you to interpret generalization performance properly.

5.1 Occam's Razor

Although it is not an exact quote of Einstein's, it is often attributed to him that "An explanation of the data should be made *as simple as possible, but no simpler.*" A similar principle, *Occam's Razor*, dates from the 14th century and is attributed to William of Occam, where the 'razor' is meant to trim down the explanation to the bare minimum that is consistent with the data.

In the context of learning, the penalty for model complexity which was introduced in Section 2.2 is a manifestation of Occam's razor. If $E_{\text{in}}(g) = 0$, then the explanation (hypothesis) is consistent with the data. In this case, the most plausible explanation, with the lowest estimate of E_{out} given in the VC bound (2.14), happens when the complexity of the explanation (measured by $d_{\text{VC}}(\mathcal{H})$) is as small as possible. Here is a statement of the underlying principle.

The simplest model that fits the data is also the most plausible.

Applying this principle, we should choose as simple a model as we think we can get away with. Although the principle that simpler is better may be intuitive, it is neither precise nor self-evident. When we apply the principle to learning from data, there are two basic questions to be asked.

1. What does it mean for a model to be simple?
2. How do we know that simpler is better?

Let's start with the first question. There are two distinct approaches to defining the notion of complexity, one based on a family of objects and the other based on an individual object. We have already seen both approaches in our analysis. The VC dimension in Chapter 2 is a measure of complexity, and it is based on the hypothesis set $\mathcal{H}$ as a whole, i.e., based on a family of objects. The regularization term of the augmented error in Chapter 4 is also a measure of complexity, but in this case it is the complexity of an individual object, namely the hypothesis h.

The two approaches to defining complexity are not encountered only in learning from data; they are a recurring theme whenever complexity is discussed. For instance, in information theory, *entropy* is a measure of complexity based on a family of objects, while *minimum description length* is a related measure based on individual objects. There is a reason why this is a recurring theme. The two approaches to defining complexity are in fact related.

When we say a family of objects is complex, we mean that the family is 'big'. That is, it contains a large variety of objects. Therefore, each individual object in the family is *one of many*. By contrast, a simple family of objects is 'small'; it has relatively few objects, and each individual object is *one of few*.

Why is the sheer number of objects an indication of the level of complexity? The reason is that both the number of objects in a family and the complexity of an object are related to how many parameters are needed to specify the object. When you increase the number of parameters in a learning model, you simultaneously increase how diverse $\mathcal{H}$ is and how complex the individual h is. For example, consider 17th order polynomials versus 3rd order polynomials. There is more variety in 17th order polynomials, and at the same time the individual 17th order polynomial is more complex than a 3rd order polynomial.

The most common definitions of object complexity are based on the number of bits needed to describe an object. Under such definitions, an object is simple if it has a short description. Therefore, a simple object is not only intrinsically simple (as it can be described succinctly), but it also has to be one of few, since there are fewer objects that have short descriptions than there are that have long descriptions, as a matter of simple counting.

Exercise 5.1

Consider hypothesis sets $\mathcal{H}_1$ and $\mathcal{H}_{100}$ that contain Boolean functions on 10 Boolean variables, so $\mathcal{X} = \{-1, +1\}^{10}$. $\mathcal{H}_1$ contains all Boolean functions

which evaluate to $+1$ on exactly one input point, and to -1 elsewhere; $\mathcal{H}_{100}$ contains all Boolean functions which evaluate to $+1$ on exactly 100 input points, and to -1 elsewhere.

(a) How big (number of hypotheses) are $\mathcal{H}_1$ and $\mathcal{H}_{100}$?

(b) How many bits are needed to specify one of the hypotheses in $\mathcal{H}_1$?

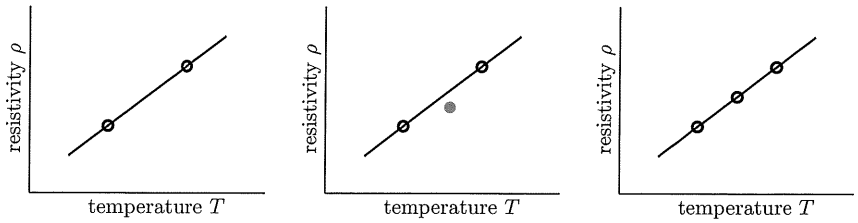
(c) How many bits are needed to specify one of the hypotheses in $\mathcal{H}_{100}$?

We now address the second question. When Occam's razor says that simpler is better, it doesn't mean simpler is more elegant. It means simpler has a better chance of being right. Occam's razor is about performance, not about aesthetics. If a complex explanation of the data performs better, we will take it.

The argument that simpler has a better chance of being right goes as follows. We are trying to fit a hypothesis to our data $\mathcal{D} = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_N, y_N)\}$ (assume y_n's are binary). There are fewer simple hypotheses than there are complex ones. With complex hypotheses, there would be enough of them to shatter $\mathbf{x}_1, \cdots, \mathbf{x}_N$, so it is certain that we can fit the data set regardless of what the labels $y_1, \cdots, y_N$ are, even if these are completely random. Therefore, fitting the data does not mean much. If, instead, we have a simple model with few hypotheses and we still found one that perfectly fits the dichotomy $\mathcal{D} = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_N, y_N)\}$, this is surprising, and therefore it means something.

Occam's Razor has been formally proved under different sets of idealized conditions. The above argument captures the essence of these proofs; if something is less likely to happen, then when it does happen it is more significant. Let us look at an example.

Example 5.1. Suppose that one constructs a physical theory about the resistivity of a metal under various temperatures. In this theory, aside from some constants that need to be determined, the resistivity ρ has a linear dependence on the temperature T. In order to verify that the theory is correct and to obtain the unknown constants, 3 scientists conduct the following three experiments and present their data to you.



Scientist 1 Scientist 2 Scientist 3

It is clear that Scientist 3 has produced the most convincing evidence for the theory. If the measurements are exact, then, Scientist 2 has managed to falsify the theory and we are back to the drawing board. What about Scientist 1? While he has not falsified the theory, has he provided any evidence for it? The answer is no, for we can reverse the question. Suppose that the theory was not correct, what could the data have done to prove him wrong? Nothing, since any two points can be joined by a line. Therefore, the model is not just likely to fit the data in this case, it is certain to do so. This renders the fit totally insignificant when it does happen. □

This example illustrates a concept related to Occam's Razor, which is the *axiom of non-falsifiability.* The axiom asserts that the data should have some chance of falsifying a hypothesis, if we are to conclude that it can provide evidence for the hypothesis. One way to guarantee that every data set has some chance at falsification is for the VC dimension of the hypothesis set to be less than N, the number of data points. This is discussed further in Problem 5.1. Here is another example of the same concept.

Example 5.2. Financial firms try to pick good traders (predictors of whether the market will go up or not). Suppose that each trader is tested on their prediction (up or down) over the next 5 days and those who perform well will be hired. One might think that this process should produce better and better traders on Wall Street. Viewed as a learning problem, consider each trader to be a prediction hypothesis. Suppose that the hiring pool is 'complex'; we are interviewing 2^5 traders who happen to be a diverse set of people such that their predictions over the next 5 days are all different. Necessarily one of these traders gets it all correct, and will be hired. Hiring the trader through this process may or may not be a good thing, since the process will pick someone even if the traders are just flipping coins to make their predictions. A perfect predictor always exists in this group, so finding one doesn't mean much. If we were interviewing only two traders, and one of them made perfect predictions, that would mean something. □

Exercise 5.2

Suppose that for 5 weeks in a row, a letter arrives in the mail that predicts the outcome of the upcoming Monday night football game. You keenly watch each Monday and to your surprise, the prediction is correct each time. On the day after the fifth game, a letter arrives, stating that if you wish to see next week's prediction, a payment of $50.00 is required. Should you pay?

(a) How many possible predictions of win-lose are there for 5 games?

(b) If the sender wants to make sure that at least one person receives correct predictions on all 5 games from him, how many people should he target to begin with?

(c) After the first letter 'predicting' the outcome of the first game, how many of the original recipients does he target with the second letter?

(d) How many letters altogether will have been sent at the end of the 5 weeks?

(e) If the cost of printing and mailing out each letter is \$0.50, how much would the sender make if the recipient of 5 correct predictions sent in the \$50.00?

(f) Can you relate this situation to the growth function and the credibility of fitting the data?

Learning from data takes Occam's Razor to another level, going beyond "as simple as possible, but no simpler." Indeed, we may opt for 'a simpler fit than possible', namely an imperfect fit of the data using a simple model over a perfect fit using a more complex one. The reason is that the price we pay for a perfect fit in terms of the penalty for model complexity in (2.14) may be too much in comparison to the benefit of the better fit. This idea was illustrated in Figure 3.7, and is a manifestation of overfitting. The idea is also the rationale behind the recommended policy in Chapter 3: *first* try a linear model – one of the simplest models in the arena of learning from data.

5.2 Sampling Bias

A vivid example of sampling bias happened in the 1948 US presidential election between Truman and Dewey. On election night, a major newspaper carried out a telephone poll to ask people how they voted. The poll indicated that Dewey won, and the paper was so confident about the small error bar in its poll that it declared Dewey the winner in its headline. When the actual votes were counted, Dewey lost – to the delight of a smiling Truman.

©Associated Press

This was not a case of statistical anomaly, where the newspaper was just incredibly unlucky (remember the δ in the VC bound?). It was a case where the sample was doomed from the get-go, regardless of its size. Even if the experiment were repeated, the result would be the same. In 1948, telephones were expensive and those who had them tended to be in an elite group that favored Dewey much more than the average voter did. Since the newspaper did its poll by telephone, it inadvertently used an in-sample distribution that was different from the out-of-sample distribution. That is what sampling bias is.

If the data is sampled in a biased way, learning will produce a similarly biased outcome.

Applying this principle, we should make sure that the training and testing distributions are the same; if not, our results may be invalid, or, at the very least, require careful interpretation.

If you recall, the VC analysis made very few assumptions, but one assumption it did make was that the data set $\mathcal{D}$ is generated from the same distribution that the final hypothesis g is tested on. In practice, we may encounter data sets that were not generated under those ideal conditions. There are some techniques in statistics and in learning to compensate for the 'mismatch' between training and testing, but not in cases where $\mathcal{D}$ was generated with the exclusion of certain parts of the input space, such as the exclusion of households with no telephones in the above example. There is nothing that can be done when this happens, other than to admit that the result will not be reliable – statistical bounds like Hoeffding and VC require a match between the training and testing distributions.

There are many examples of how sampling bias can be introduced in data collection. In some cases it is inadvertently introduced by an oversight, as in the case of Dewey and Truman. In other cases, it is introduced because certain types of data are not available. For instance, in our credit example of Chapter 1, the bank created the training set from the database of previous customers and how they performed for the bank. Such a set necessarily excludes those who applied to the bank for credit cards and were rejected, because the bank does not have data on how they *would have performed* if they were accepted. Since future applicants will come from a mixed population including some who would have been rejected in the past, the 'test set' comes from a different distribution than the training set, and we have a case of sampling bias. In this particular case, if no data on the applicants that were rejected is available, nothing much can be done other than to acknowledge that there is a bias in the final predictor that learning will produce, since a representative training set is just not available.

Exercise 5.3

In an experiment to determine the distribution of sizes of fish in a lake, a net might be used to catch a representative sample of fish. The sample is

then analyzed to find out the fractions of fish of different sizes. If the sample is big enough, statistical conclusions may be drawn about the actual distribution in the entire lake. Can you smell ☺ sampling bias?

There are other cases, arguably more common, where sampling bias is introduced by human intervention. It is not that uncommon for someone to throw away training examples they don't like! A Wall Street firm who wants to develop an automated trading system might choose data sets when the market was 'behaving well' to train the system, with the semi-legitimate justification that they don't want the noise to complicate the training process. They will surely achieve that if they get rid of the 'bad' examples, but they will create a system that can be trusted only in the periods when the market does behave well! What happens when the market is not behaving well is anybody's guess. In general, throwing away training examples based on their values, e.g., examples that look like outliers or don't conform to our preconceived ideas, is a fairly common sampling bias trap.

Other biases. Sampling bias has also been called selection bias in the statistics community. We will stick with the more descriptive term sampling bias for two reasons. First, the bias arises in how the data was *sampled*; second, it is less ambiguous because in the learning context, there is another notion of selection bias drifting around – *selection* of a final hypothesis from the learning model based on the data. The performance of the selected hypothesis on the data is optimistically biased, and this could be denoted as a selection bias. We have referred to this type of bias simply as bad generalization.

There are various other biases that have similar flavor. There is even a special type of bias for the research community, called publication bias! This refers to the bias in published scientific results because negative results are often not published in the literature, whereas positive results are. The common theme of all of these biases is that they render the standard statistical conclusions invalid because the basic premise for such conclusions, that the sampling distribution is the same as the overall distribution, does not hold any more. In the field of learning from data, it is sampling bias in the training set that we need to worry about.

5.3 Data Snooping

Data snooping is the most common trap for practitioners in learning from data. The principle involved is simple enough,

If a data set has affected any step in the learning process, its ability to assess the outcome has been compromised.

Applying this principle, if you want an unbiased assessment of your learning performance, you should keep a test set in a vault and never use it for learning in any way. This is basically what we have been talking about all along in training versus testing, but it goes beyond that. Even if a data set has not been 'physically' used for training, it can still affect the learning process, sometimes in subtle ways.

Exercise 5.4

Consider the following approach to learning. By looking at the data, it appears that the data is linearly separable, so we go ahead and use a simple perceptron, and get a training error of zero after determining the optimal set of weights. We now wish to make some generalization conclusions, so we look up the d_{VC} for our learning model and see that it is $d+1$. Therefore, we use this value of d_{VC} to get a bound on the test error.

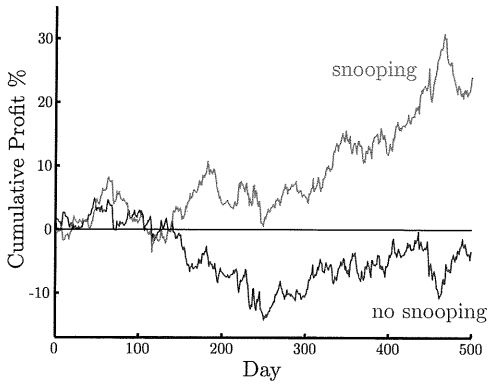
(a) What is the problem with this bound - is it correct?

(b) Do we know the d_{VC} for the learning model that we actually used? It is this d_{VC} that we need to use in the bound.

To avoid the pitfall in the above exercise, it is extremely important that you choose your learning model *before* seeing any of the data. The choice can be based on general information about the learning problem, such as the number of data points and prior knowledge regarding the input space and target function, but not on the actual data set $\mathcal{D}$. Failure to observe this rule will invalidate the VC bounds, and any generalization conclusions will be up in the air. Even a careful person can fall into the traps of data snooping. Consider the following example.

Example 5.3. An investment bank wants to develop a system for forecasting currency exchange rates. It has 8 years worth of historical data on the US Dollar (USD) versus the British Pound (GBP), so it tries to use the data to see if there is any pattern that can be exploited. The bank takes the series of daily changes in the USD/GBP rate, normalizes it to zero mean and unit variance, and starts to develop a system for forecasting the direction of the change. For each day, it tries to predict that direction based on the fluctuations in the previous 20 days. 75% of the data is used for training, and the remaining 25% is set aside for testing the final hypothesis.

The test shows great success. The final hypothesis has a hit rate (percentage of time getting the direction right) of 52.1%. This may seem modest, but in the world of finance you can make a lot of money if you get that hit rate consistently. Indeed, over the 500 test days (2 years worth, as each year has about 250 trading days), the cumulative profit of the system is a respectable 22%.



When the system is used in live trading, the performance deteriorates significantly. In fact, it loses money. Why didn't the good test performance continue on the new data? In this case, there is a simple explanation and it has to do with data snooping. Although the bank was careful to set aside test points that were not used for training in order to properly evaluate the final hypothesis, the test data had in fact affected the training process in a subtle way. When the original series of daily changes was normalized to zero mean and unit variance, *all of the data* was involved in this step. Therefore, the test data that was extracted had already contributed to the choices made by the learning algorithm by contributing to the values of the mean and the variance that were used in normalization. Although this seems like a minor effect, it *is* data snooping. When you plot the cumulative profit on the test set with or without that snooping step, you see how snooping resulted in an over-optimistic expectation compared to the realistic expectation that avoids snooping.

It is not the normalization that was a bad idea. It is the involvement of test data in that normalization, which contaminated this data and rendered its estimate of the final performance inaccurate. □

One of the most common occurrences of data snooping is the reuse of the same data set. If you try learning using first one model and then another and then another on the same data set, you will eventually 'succeed'. As the saying goes, if you torture the data long enough, it will confess ☺. If you try all possible dichotomies, you will eventually fit any data set; this is true whether we try the dichotomies directly (using a single model) or indirectly (using a sequence of models). The effective VC dimension for the series of trials will not be that of the last model that succeeded, but of the entire union of models that could have been used depending on the outcomes of different trials.

Sometimes the reuse of the same data set is carried out by different people. Let's say that there is a public data set that you would like to work on. Before you download the data, you read about how other people did with this data set

using different techniques. You naturally pick the most promising techniques as a baseline, then try to improve on them and introduce your own ideas. Although you haven't even seen the data set yet, you are already guilty of data snooping. Your choice of baseline techniques was affected by the data set, through the actions of others. You may find that your estimates of the performance will turn out to be too optimistic, since the techniques you are using have already proven well-suited to *this particular* data set.

To quantify the damage done by data snooping, one has to assess the penalty for model complexity in (2.14) taking the snooping into consideration. In the public data set case, the effective VC dimension corresponds to a much bigger hypothesis set than the $\mathcal{H}$ that your learning algorithm uses. It covers all hypotheses that were considered (and mostly rejected) by everybody else in the process of coming up with the solutions that they published and that you used as your baseline. This is a potentially huge set with very high VC dimension, hence the generalization guarantees in (2.14) will be much worse than without data snooping.

Not all data sets subjected to data snooping are equally 'contaminated'. The bounds in (1.6) in the case of a choice between a finite number of hypotheses, and in (2.12) in the case of an infinite number, provide guidelines for the level of contamination. The more elaborate the choice made based on a data set, the more contaminated the set becomes and the less reliable it will be in gauging the performance of the final hypothesis.

Exercise 5.5

Assume we set aside 100 examples from $\mathcal{D}$ that will not be used in training, but will be used to select one of three final hypotheses g_1, g_2, g_3 produced by three different learning algorithms that train on the rest on the data. Each algorithm works with a different $\mathcal{H}$ of size 500. We would like to characterize the accuracy of estimating $E_{\text{out}}(g)$ on the selected final hypothesis if we use the same 100 examples to make that estimate.

(a) What is the value of M that should be used in (1.6) in this situation?

(b) How does the level of contamination of these 100 examples compare to the case where they would be used in training rather than in the final selection?

In order to deal with data snooping, there are basically two approaches.

1. Avoid data snooping: A strict discipline in handling the data is required. Data that is going to be used to evaluate the final performance should be 'locked in a safe' and only brought out after the final hypothesis has been decided. If intermediate tests are needed, separate data sets should be used for that. Once a data set has been used, it should be treated as contaminated as far as testing the performance is concerned.

2. Account for data snooping: If you have to use a data set more than once, keep track of the level of contamination and treat the reliability of

your performance estimates in light of this contamination. The bounds (1.6) and (2.12) can provide guidelines for the relative reliability of different data sets that have been used in different roles within the learning process.

Data snooping versus sampling bias. Sampling bias was defined based on how the data was obtained before any learning; data snooping was defined based on how the data affected the learning, in particular how the learning model is selected. These are obviously different concepts. However, there are cases where sampling bias occurs as a consequence of 'snooping' – looking at data that you are not supposed to look at. Here is an example.

Consider predicting the performance of different stocks based on historical data. In order to see if a prediction rule is any good, you take all currently traded companies and test the rule on their stock data over the past 50 years. Let us say that you are testing the "buy and hold" strategy, where you would have bought the stock 50 years ago and kept it until now. If you test this 'hypothesis', you will get excellent performance in terms of profit. Well, don't get too excited! You inadvertently biased the results in your favor by picking only *currently traded companies*, which means that the companies that did not make it are not part of your evaluation. When you put your prediction rule to work, it will be used on all companies whether they will survive or not, since you cannot identify which companies today will be the 'currently traded' companies 50 years from now. This is a typical case of sampling bias, since the problem is that the training data is not representative of the test data. However, if we trace the origin of the bias, we did 'snoop' in this case by looking at future data of companies to determine which of these companies to use in our training. Since we are using information in training that we would not have access to in real trading, this is viewed as a form of data snooping.

5.4 Problems

Problem 5.1 The idea of *falsifiability* – that a claim can be rendered false by observed data – is an important principle in experimental science.

> **Axiom of Non-Falsifiability.** *If the outcome of an experiment has no chance of falsifying a particular proposition, then the result of that experiment does not provide evidence one way or another toward the truth of the proposition.*

Consider the proposition "There is $h \in \mathcal{H}$ that approximates f as would be evidenced by finding such an h with in-sample error zero on $\mathbf{x}_1, \cdots, \mathbf{x}_N$." We say that the proposition is falsified if no hypothesis in $\mathcal{H}$ can fit the data perfectly.

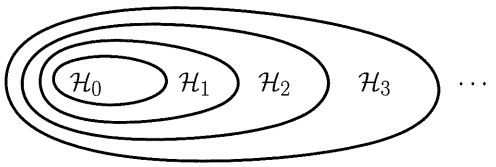
(a) Suppose that $\mathcal{H}$ shatters $\mathbf{x}_1, \cdots, \mathbf{x}_N$. Show that this proposition is not falsifiable *for any* f.

(b) Suppose that f is random ($f(\mathbf{x}) = \pm 1$ with probability $\frac{1}{2}$, independently on every $\mathbf{x}$), so $E_{\text{out}}(h) = \frac{1}{2}$ for every $h \in \mathcal{H}$. Show that

$$\mathbb{P}[\text{falsification}] \geq 1 - \frac{m_{\mathcal{H}}(N)}{2^N} .$$

(c) Suppose $d_{\text{vc}} = 10$ and $N = 100$. If you obtain a hypothesis h with zero E_{in} on your data, what can you 'conclude' from the result in part (b)?

Problem 5.2 Structural Risk Minimization (SRM) is a useful framework for model selection that is related to Occam's Razor. Define a *structure* – a nested sequence of hypothesis sets:



The SRM framework picks a hypothesis from each $\mathcal{H}_i$ by minimizing E_{in}. That is, $g_i = \underset{h \in \mathcal{H}_i}{\text{argmin}}\, E_{\text{in}}(h)$. Then, the framework selects the final hypothesis by minimizing E_{in} *and* the model complexity penalty Ω. That is, $g^* = \underset{i=1,2,\cdots}{\text{argmin}}(E_{\text{in}}(g_i) + \Omega(\mathcal{H}_i))$. Note that $\Omega(\mathcal{H}_i)$ should be non-decreasing in i because of the nested structure.

(a) Show that the in-sample error $E_{\text{in}}(g_i)$ is non-increasing in i.

(b) Assume that the framework finds $g^* \in \mathcal{H}_i$ with probability p_i. How does p_i relate to the complexity of the target function?

(c) Argue that the p_i's are unknown but $p_0 \leq p_1 \leq p_2 \leq \cdots \leq 1$.

(d) Suppose $g^* = g_i$. Show that

$$\mathbb{P}\left[|E_{\text{in}}(g_i) - E_{\text{out}}(g_i)| > \epsilon \mid g^* = g_i\right] \leq \frac{1}{p_i} \cdot 4m_{\mathcal{H}_i}(2N)e^{-\epsilon^2 N/8}.$$

Here, the conditioning is on selecting g_i as the final hypothesis by SRM. *[Hint: Use the Bayes theorem to decompose the probability and then apply the VC bound on one of the terms]*

You may interpret this result as follows: if you use SRM and end up with g_i, then the generalization bound is a factor $\frac{1}{p_i}$ worse than the bound you would have gotten had you simply started with $\mathcal{H}_i$.

Problem 5.3 In our credit card example, the bank starts with some vague idea of what constitutes a good credit risk. So, as customers $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ arrive, the bank applies its vague idea to approve credit cards for some of these customers. Then, only those who got credit cards are monitored to see if they default or not.

For simplicity, suppose that the first N customers were given credit cards. Now that the bank knows the behavior of these customers, it comes to you to improve their algorithm for approving credit. The bank gives you the data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$.

Before you look at the data, you do mathematical derivations and come up with a credit approval function. You now test it on the data and, to your delight, obtain perfect prediction.

(a) What is M, the size of your hypothesis set?

(b) With such an M, what does the Hoeffding bound say about the probability that the true performance is worse than 2% error for $N = 10000$?

(c) You give your g to the bank and assure them that the performance will be better than 2% error and your confidence is given by your answer to part (b). The bank is thrilled and uses your g to approve credit for new clients. To their dismay, more than half their credit cards are being defaulted on. Explain the possible reason(s) behind this outcome.

(d) Is there a way in which the bank could use your credit approval function to have your probabilistic guarantee? How? *[Hint: The answer is yes!]*

Problem 5.4 The S&P 500 is a set of the largest 500 companies currently trading. Suppose there are $10,000$ stocks currently trading, and there have been $50,000$ stocks which have ever traded over the last 50 years (some of these have gone bankrupt and stopped trading). We wish to evaluate the profitability of various 'buy and hold' strategies using these 50 years of data (roughly $12,500$ trading days).

Since it is not easy to get stock data, we will confine our analysis to today's S&P 500 stocks, for which the data is readily available.

(a) A stock is profitable if it went up on more than 50% of the days. Of your S&P stocks, the most profitable went up on 52% of the days ($E_{\text{in}} = 0.48$).

(i) Since we picked the best among 500, using the Hoeffding bound,

$$\mathbb{P}[|E_{\text{in}} - E_{\text{out}}| > 0.02] \leq 2 \times 500 \times e^{-2 \times 12500 \times 0.02^2} \approx 0.045.$$

There is a greater than 95% chance this stock is profitable. Where did we go wrong?

(ii) Give a better estimate for the probability that this stock is profitable. *[Hint: What should the correct M be in the Hoeffding bound?]*

(b) We wish to evaluate the profitability of 'buy and hold' for general stock trading. We notice that all of our 500 S&P stocks went up on at least 51% of the days.

(i) We conclude that buying and holding a stocks is a good strategy for general stock trading. Where did we go wrong?

(ii) Can we say *anything* about the performance of buy and hold trading?

Problem 5.5 You think that the stock market exhibits reversal, so if the price of a stock sharply drops you expect it to rise shortly thereafter. If it sharply rises, you expect it to drop shortly thereafter.

To test this hypothesis, you build a trading strategy that buys when the stocks go down and sells in the opposite case. You collect historical data on the current S&P 500 stocks, and your hypothesis gave a good annual return of 12%.

(a) When you trade using this system, do you expect it to perform at this level? Why or why not?

(b) How can you test your strategy so that its performance in sample is more reflective of what you should expect in reality?

Problem 5.6 One often hears "Extrapolation is harder than interpolation." Give a possible explanation for this phenomenon using the principles in this chapter. *[Hint: training distribution versus testing distribution.]*