Nati Srebro at UT Austin 2011-05-31 Lecture 1 Notes

Patrick W. Gallagher

August 31, 2011

Abstract

Machine learning overview. Advantages of learning approach over expert knowledge approach. Examples of machine learning. Spam example. The continuum of expert knowledge. Source distribution. Conditional probability. Marginal probability. Expected loss of a predictor. 01 loss. The idea of a learning rule. The average loss on the sample. Empirical risk minimization (ERM). Example hypothesis classes. Feature mapping for abstract objects. Initial analysis of ERM. Hoeffding's inequality. The flaw with the initial analysis. Complexity (here, cardinality) of the hypothesis class. Gender prediction example. Correcting the flaw: The union bound. Post-hoc guarantees versus learning guarantees.

1 What is machine learning?

What is machine learning? There are several possible views, but the view that we will take here is that machine learning is an engineering paradigm. In particular, it is an engineering paradigm where we we use data to *automatically* construct a complex system. The machine is learning from the data. This automatic construction using data stands in contrast to the alternative: using not data and automatic construction, but "expert knowledge" and explicit hand-programming of the complex system.

To highlight the distinction between the machine learning data-centric approach and the "expert knowledge" approach, let's consider an example. A classic problem that has been attacked using both the machine learning data-centric approach and the "expert knowledge" approach is Optical Character Recognition (OCR) for typed characters in the Latin alphabet. How would a non-machine-learning "expert knowledge" approach to Optical Character Recognition for typed Latin characters proceed? You would have to sit down and program a typed Latin alphabet character recognizer from scratch. You would not feed any examples to the system. It would take you a long time. You might eventually do a reasonably decent job.

You might program the system with an explicit rule that identifies the lines in a given character. Thus, sitting at your computer and thinking about an "A", you might program a rule that says "If you have a character with two lines sloping away from a common top point and another horizontal line joining those lines, the system should say that it's an 'A'." If the system sees one big circle it should say it's an "O" and so forth for the other letters. This would be a completely expert-knowledge-based approach. You explicitly use your knowledge of characters to program the system. You don't use any data and it is probably fairly time-consuming and painful. You program quite a bit.

A machine-learning data-centric approach would say, "Forget most of what you personally 'know' about characters. Don't explicitly code rules". Instead, just take lots of data and use some machine learning method to automatically develop a system. Your data might consist of ten thousand examples of "A", ten thousand examples of "B", ten thousand examples of "C", and so forth. Throw the examples at the machine learning method and let the computer figure out rules for what makes an "A" an "A", what makes a "B" a "B", and so forth, and what the

learning process outputs is essentially a program that takes an image and tells me what letter it is. So, the basic point here is that we constructed this system, this recognition system, using data instead of expert knowledge and careful programming. The example of optical character character recognition highlights several advantages of a data-centric machine-learning approach over an "expert knowledge" explicitly hand-coded approach.

1.1 The advantages of machine learning

The first advantage of machine learning is that the process is much *easier*, as there is much less programming involved. A second advantage comes from the fact that experts often believe that they know what is going on in the data, whether or not that is actually the case. With a focus on learning-from-data, we can often avoid this sort of "theory blindness" and directly find out about (possibly unusual) characteristics of the data that might otherwise not be represented in the "expert knowledge". Beyond this, a data-centric approach is definitely more *adaptive* to changes in the data. For example, if you want to do character recognition for handwritten Latin characters instead of for typed Latin characters, the "expert knowledge" from the typed Latin character setting is unlikely to be more than marginally helpful in the handwritten Latin character setting. On the other hand, a machine learning system set up to learn characteristics directly from data can simply be shown the handwritten data and learn its characteristics. With a machine learning system, all you need in order to recognize handwritten characters is to show the system lots of handwritten character data.

You can even consider a more extreme example and recognize an alphabet that you yourself have *no* "expert knowledge" about. If you want to recognize the Chinese alphabet or the Arabic alphabet, or any other alphabet, you don't even need to know what these letters are yourself: all you need is data. You need some examples and you can construct a character recognition system for another alphabet. Machine learning depends much more on data than on any "expert knowledge". We can train systems to do things that we don't even know how to do ourselves.

One final advantage of machine learning is that in many applications it simply yields much better performance. Character recognition systems were originally designed using a template-matching approach, where the templates were each hand-specified. Currently, almost all character recognition systems are based in (one way or another) on a machine learning approach, and they work much better. This observation — that a data-centric machine learning approach yields better performance — applies to many examples beyond just Optical Character Recognition. We now consider several examples to which machine learning is applied.

1.2 Examples of machine learning

- **Character recognition.** Given an image of a character, correctly identify the character.
- **Spam recognition.** Given an email, correctly identify the email as spam or not-spam [1, 2].
- **Speech recognition.** Given an audio of speech, identify the words being said.
  - In speech recognition, systems were not initially built using learning approaches. Current systems are more and more based on learning approaches. Among examples of early speech systems in the 1950s were various games that responded to sounds. The approach at the time is typically described as "template-matching", where the system attempted to match the waveform of the sound or of its Fourier spectrum to some library of hand-specified waveforms or spectra. We will consider the (now dominant) learning approaches that go beyond the "memory-based" template-matching approach that attempts to identify the words in a new audio sample by matching the new audio sample to a fixed library of examples. A learning-from-data approach would instead use those examples to try to extrapolate useful features of the available data that will be useful on new samples. Although memory-based approaches can be successful in some contexts, we will be talking about the more broadly applicable approach of learning-from-data.

- **Machine translation.** Given a sample of text in one language, produce text in another language with the same meaning.

  - Until relatively recently, machine translation systems involved comparatively little learning. Instead, most systems were based on large dictionaries and hand-specified rules of grammar, with linguists writing up grammar trees and matching grammars between languages, and dictionaries between languages [6]. A recent prominent example of the learning approach is Google Translate [3], but many other current translation systems use a similar approach. The result is that translation is now studied much more as a machine learning task. For example, take some text in English and take some text in French known to correspond to the English text. With many such corresponding texts, we can let the computer figure out how to map between the English and the French. Current systems are programmed with comparatively little linguistic knowledge. There is some linguistic knowledge incorporated, but mostly it's just a system trying to figure out how to map between one language to another based on examples. (Search: "Manning NLP video")

- **Computer vision.**

  - Up until the mid 1990s, computer vision was mostly based on geometry and identifying edges and lines in images. Starting with some seminal work on face recognition in the late 1990s [11, 10]and continuing to the present with almost every other application in vision, vision has been turned into a largely learning-based field. Instead of trying to figure out geometrically what geometry makes the face, we just give the computer a bunch of faces and let it figure out "In these images, this is what makes up a face". Tasks that are now accomplished using the learning approach include object recognition (identifying chairs, bicycles, cars, pedestrians) and identifying faces. From a modern perspective, the previous tasks perhaps seem naturally suited to a learning approach. However, even tasks like depth reconstruction or 3-D reconstruction which might initially appear to be best handled with geometry are now being successfully accomplished with learning.
  - In particular, if I give you two stereo images, you should be able to reconstruct the actual 3-D landscape. That is, you should be able to identify the depth, the distance to each object in the image. This is a very geometric task, and was traditionally done with geometry. A few years ago it was shown this can be done with a fairly pure learning approach. This learning approach doesn't know anything about geometry. It doesn't know anything about triangulating points with different views. It is just trained based on stereo images and their depth field and learns a mapping from pairs of stereo images to their depth field. There is even some work based on only one image [8]. Thus, even in situations where you really could build a system based on expert (here, geometric) knowledge, this approach is much more difficult and typically yields a less effective system. (See also [4])

- **Control systems.**

  - Another recent example is control systems. One of the most famous at the moment is flying a helicopter (Youtube: "stanfordhelicopter"). Flying a helicopter is a very complex task. Apparently, it is much harder than flying a plane. It is a very complicated control task, both for a human pilot and for the control system that goes into the helicopter itself. Among other aspects of the problem, the mapping from the blade rotation speed and the angle of the blades to what the helicopter then does is fairly complex. One result of this complexity is that helicopter autopilot systems are much, much less common than airplane autopilot systems are, and traditionally required extensive hand specification. The approach taken by the Stanford group (among others) for their model helicopter is based on learning: an experienced pilot

provides training data by flying the helicopter through a variety of maneuvers. Based on this training data, the learning system learns the mapping from the rotor control signals to the trajectory that the helicopter then takes. Once this learning has taken place, the system can be told "I want to follow the following trajectory, what control signals should be sent to the rotors to achieve that trajectory". The learning system knows nothing about physics or aerodynamics; it just learns the control mapping, and then uses this learned mapping to follow specified trajectories.

- **Ranking web search results.** Given a search query return a ranking of web pages by relevance/"goodness".
  - Returning a list of web pages containing a given query is not itself a learning problem. What *is* a learning problem is figuring out what search results are "good" and providing a ranking that reflects this "goodness" [7]. This ranking problem is a somewhat different from prediction-type problems, but is another area in which a lot of effort in learning is currently being spent.
- **Recommender systems.** Given a matrix with a small subset of entries filled with numbers from 1 through 5, and the remaining entries "missing", determine "good" values for the missing entries. This description is very abstract, but this abstraction corresponds to the fact that most current recommender systems use only the above information. In particular, they do not use any information about the characteristics of the users and they do not use any information about the characteristics of the movies. (Although such systems are now being pursued for future use.)
  - One prominent example is the Netflix movie recommender system. With movie recommender systems, we know the ratings given by system users for movies they have watched. Although each person typically only gives ratings for a small fraction of the total number of movies, we would like to make use of the person's ratings and the ratings of other users to make recommendations about other movies the person might like. Again, current recommender systems operate without knowing any characteristics of the movies themselves, and without any knowledge of the users themselves.
- **Computational biology.**
  - In the setting of computational biology applications such as inferring regulatory networks or inferring which elements in the genomic sequence are most important, there is, some sense, very little "expert knowledge". In this setting, the amount of information that could be derived by asking "experts" is relatively limited. The whole point in this setting is that (currently) no one knows the answers; no one actually knows what the gene regulatory networks look like or exactly how important various gene sequences are. We are trying to build a computer system that will be able to answer these questions, and here we only have the data to help us.
- **functional Magnetic Resonance Imaging (fMRI).** Given the Blood Oxygen Level Dependent (BOLD) signal strengths at various voxels in the subject's brain, predict whether the word the subject is looking at is a noun or a verb [12].

1.3 The spam example and the continuum of expert knowledge

Before we begin writing any definitions on the board, let's take a closer look at the spam example to highlight that in general, a system could incorporate amounts of "expert knowledge" along a continuum. A typical spam filter takes a large number of email examples and tries to extract rules from the examples that provide a characterization of what makes something spam or not-spam. Let us consider possible approaches to building such a spam filter.

We begin with a large number of emails that we know are spam and a large number of emails that we know are not spam. From this starting point, we want to find a decision rule that correctly classifies future emails as spam or not-spam. A fairly low-level approach involving essentially no expert knowledge would be to base our decision on nothing but the characters in the email. For example, we might note that in the training set spam emails, the relative frequency of the letter "v" was much higher than the relative frequency of the letter "v" in the training set not-spam emails. From this observation, we could decide to measure the relative frequency of the letter "v" in future emails and classify all messages where "v"s make up more than 2% of the characters as spam; anything below this threshold would be classified as not-spam.

As consideration of the above example might suggest, an approach that makes a decision based only on character-level information is unlikely to perform very well. Instead of just considering the characters in the emails, we could likely do better by considering entire words in the emails, and basing our decision rule on word-level information. For example, we might observe that the word "Nigerian" occurs much more frequently in spam than not-spam. We could then decide that if a future email contains the word "Nigerian", it is much more likely to be spam. By providing the spam filter with word-level features, we will make the task much easier for the machine learning algorithm. Basically what we are doing here is helping out the machine learning algorithm by telling it "Look, we know something about language. We don't know what makes spam or not, but we know that these characters really make up words. The important thing is not the individual characters but the words. So, why don't you start working based on the words." This is an example of using a relatively small amount of expert knowledge in an otherwise primarily learning-from-data approach.

We could be even more helpful to the learning method by introducing some extra features we suspect might be relevant. For example, beyond just the words in the email, we could also include features like the count how many words are in all capital letters. We can even go even further by giving information about whether recipients appear in an address book. In each of these cases, we are providing the learning method with additional features that we suspect will be relevant to the task of classifying email as spam or not-spam; the algorithm itself will determine how it actually uses that information.

The process of manually deciding to include "higher-level" features described above highlights the fact that there is really a continuum between a "learning-only" approach and an "expert-knowledge-only" approach. By providing the machine learning method with features that we suspect will be more relevant to its prediction task, we are using a relatively small amount of expert knowledge in an otherwise learning-from-data approach. Even when we considered working at the level of characters, that is already that is giving the computer *some* of our expert knowledge, because in practice email doesn't come in characters, it comes in bits. We are already telling the learning method that these bits actually make up characters. If you want to go without *any* expert knowledge, the "pure" learning-from-data approach would begin not with bits but with *compressed* bits. We can't expect the learning approach to work based on just the raw raw data itself. In fact, in many cases, there is really no such thing as the raw data. When I take an image, there are many ways to represent the image (different resolutions, different ISO speeds, different compression formats, etc ...). There is really no "raw" way to represent the image.

This is just to say that we will consider methods that are primarily learning from data, while also incorporating *some* expert knowledge. This places us near to the learning-from-data end of the continuum of expert knowledge. On one end of this continuum, you base a system *only* on expert knowledge, with no learning whatsoever. For example, you just hand-program what an "A" looks and what a "B" looks. There is no incorporation of data, and no sense of learning from that data. On the other end of the continuum, you get purely "raw" data (such as the compressed bits of an email) and you want the machine learning method to do *everything* (decompressing the data, noting that the resulting bits make characters, noting that the characters make words, etc...).

One might ask whether it is even *possible* for a machine learning method to do "everything". It turns out that one can prove that "bias-free learning is futile"[13]. That is, if you are on the far extreme of pure learning-from-data , incorporating *no* prior knowledge, if the learning method is only provided with a bitstream without any indication

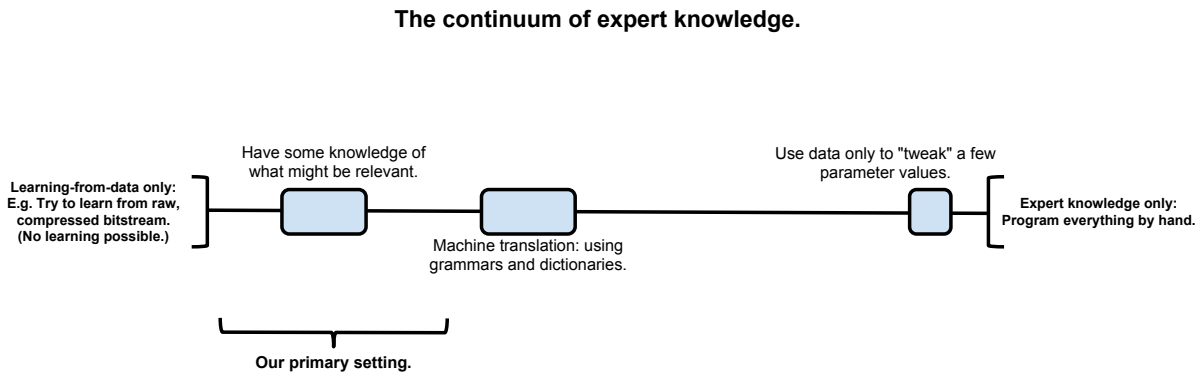

Figure 1.1: The continuum of expert knowledge in machine learning.

of what in the bitstream could possibly make a good classification rule, then no learning is possible. In order to learn from data, it is necessary to inject *some* form of prior knowledge. Thus, the approaches we will consider incorporate some prior knowledge by providing the learning method with features that we suspect to be a super-set of relevant features of the data. We don't know precisely how these features might be relevant, but we know enough to capture some super-set of the relevant things. Based on this provided super-set of relevant features of the data, we then want the method to learn how to successfully use these features to predict.

Even though we are primarily taking a learning-from-data view, we will still be incorporating a little (or even more than a little) prior knowledge. In the spam filter example, we would incorporate a modicum of prior knowledge by using word-level features. In machine translation, we might be incorporating even more prior knowledge, by including some grammatical knowledge, some dictionaries. Our focus is still on learning from the data.

Before we move on, let us also briefly consider what an approach near the expert-knowledge end of the continuum would be like. In this setting, you might have a set of hand-specified functions to model the behavior of a system, completely determined except for the values of a handful of parameter values. In this case, you could consider using a training data set to determine reasonable values for these last parameters.

The underlying message here is that all of these example problems can be approached by methods on a continuum of expert knowledge. The ideas that we will be considering in this course will be relevant to understanding any method on the continuum that does any learning at all. To help focus our attention, however, we will primarily consider the "mostly-learning" side of things, where we begin with a fairly basic super-set of relevant features and we seek methods that can successfully learn from these very basic features.

2 Definitions

Thus far, we have been talking about fairly high-level examples. We are now going to switch to being much more concrete. In particular, although we noted that learning in general is the use of data to solve some task, where that task can vary from flying a helicopter to translating sentences from French to English, we will focus solely on *prediction* tasks like character recognition. Thus, our task will be to use the data to find a "good" predictor. First, we need to ask "what is a predictor?"

2.1 A predictor

A predictor is mapping from the initial, abstract object space to some label set, where for character recognition the domain of abstract input objects, $\mathcal{X}$, consists of possible images of letters and the label set, $\mathcal{Y}$, consists of the the twenty-six letters of the Latin alphabet. In other problems, the label could be a real-valued number, or a person's height, or their income, or how much they like a film; in general, the label could anything we want to predict. Although many label sets are possible, for simplicity we will be considering the simplest label set: binary labels $\{+1, -1\}$. We can view these as corresponding to predicting, for example, whether something is the letter "G" $(+1)$ or not the letter "G" (-1), or whether a given image contains a face $(+1)$ or does not contain a face (-1).

2.2 The source joint distribution $p_{X,Y}(x, y)$ and its components

When we ask "Is this predictor good?", we are implicitly asking "Is this predictor going to give accurate predictions of the labels for future data?". This means that in order to determine whether a predictor is "good" we need to a means of referring to this future data. The view we will take in most of the course is Statistical Learning, wherein we refer to future data as coming from some unknown *source* joint distribution $p_{X,Y}$ over input objects (which we usually just take to be elements of $\mathbb{R}^d$, or of a subset of $\mathbb{R}^d$, such as the positive orthant[1]) and their corresponding labels, which we write as the joint distribution $p_{X,Y}(x, y)$, where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. In the character recognition setting, for example, the source joint distribution would over all possible images of characters and the corresponding letter label. Such a character recognition source distribution would assign much more probability to ("image containing a circular shape", "O") than to ("image containing a circular shape","T"). After closer consideration of the source joint distribution $p_{X,Y}(x, y)$ we might further be motivated to view it as really having two components: the conditional probability $p_{Y|X}(y|x)$ of the label random variable Y given the appearance random variable X, and the marginal probability $p_X(x)$ of the input image. That is, $p_{X,Y}(x, y) = p_{Y|X}(y|x) \cdot p_X(x)$.

2.2.1 The source conditional probability distribution $p_{Y|X}(y|x)$

Having thus factored the source joint distribution, we observe that the probability of the label random variable Y conditioned on the input image random variable X, $p_{Y|X}(y|x)$, is what really defines "correctness" for a predictor: If I am handed a particular image, say "image of an A", then $p_{Y|X}(y|X = \text{"image of an A"})$ (alternately written as just $p_{Y|X}(y|\text{"image of an A"})$) indicates how probable each of the possible labels is, given that the input image random variable was observed to have the value "image of an A". In many cases, this will essentially be a deterministic probability distribution, so the conditional distribution of the random variable Y after having observed the value of the random variable X, $p_{Y|X}(y|x)$, will be a 1 (in the Y discrete case) e.g. for the situation $Y = \text{"A"}$ and 0 for all of the other label from the label set (That is, $p_{Y|X}(Y = \text{"A"}|X = \text{"image of an } A\text{"}) = p_{Y|X}(\text{"A"}|\text{"image of an } A\text{"}) = 1$, $p_{Y|X}(Y = \text{"B"}|X = \text{"image of an A"}) = p_{Y|X}(\text{"B"}|\text{"image of an A"}) = 0$, and so on). This determinism need not always be the case, since some times there really will be *inherent* uncertainty in the task we are trying to do, and this conditional distribution captures that possibility. For an extremely fuzzy input image, there might be no single really correct answer, so that $p_{Y|X}(y|X = \text{"very fuzzy image, maybe of an 'o'"})$ would then assign significant probability to more than one label (perhaps to "c" and "o").

[1] It is sometimes useful to realize that we only *represent* input objects as vectors of d real numbers. For example, a tiff-format image of an "A" may well have come from an actual "A" written or printed on a piece of paper somewhere. We will typically favor the much more specific account of input objects as being vectors in $\mathbb{R}^d$ over the much less specific idea of the input objects coming from real-world objects. We may consider there to be some sort of feature mapping $\phi(\cdot)$ that refers to the process by the original real-world object is mapped to a specific representation in the computer: $\phi(\text{"letter A on page of book"}) \mapsto \text{"gray-level values for pixels in image of letter A"}$.

2.2.2 The source marginal probability distribution $p_X(x)$

The other part of this source joint distribution $p_{X,Y}(x,y)$, is the marginal probability $p_X(x)$ of the random input variable X that can take on values in the set of possible input objects. In the character recognition example, this marginal distribution indicates which input letter images are likely, and which input letter images are not likely. It might seem that this marginal probability distribution should be irrelevant, because we just want to correctly predict labels after we are handed input objects. Indeed, we might well ask "If I am only predicting labels for inputs that I am handed, why does it matter how likely different inputs are?" The relevance arises from the fact that we want our predictor to be "good" for (input,label) pairs from the joint source distribution. If in the character recognition task we know that all of the images will be of some letter, so that $p_X(X = \text{"image of keyboard cat"}) = p_X(\text{"image of keyboard cat"}) = 0$, then we also know that $p_{X,Y}(X = \text{"image of keyboard cat"}, y) = p_{X,Y}(\text{"image of keyboard cat"}, y) = 0$ for all possible label realizations y that the random variable Y could take on [2]. In general, we don't need to care what label our predictor would guess for an image that is very unlikely.

2.3 The 01 loss of a predictor h on an observed (input, label) pair (x,y)

For our first means of evaluating how well a predictor h[3] is doing, we consider evaluating the performance of h on a given (input,label) pair[4] (x,y). A natural means of evaluating performance would be to say that if the label $h(x)$ (that our predictor h guesses for the observed input value x) does not match the provided label y, we incur a loss of 1. If the prediction $h(x)$ does match the provided label y, we incur 0 loss. The loss function that represents this measure of performance is called the 01 loss: $\mathbf{loss}_{01} : \{-1,1\} \times \{-1,1\} \to \{0,1\}$, defined as

$$\mathbf{loss}_{01}(h(x), y) = \begin{cases} 1 & \text{if } h(x) \neq y \\ 0 & \text{if } h(x) = y \end{cases}$$

2.4 The risk or expected loss of a predictor with respect to a given source joint distribution

Since we now have a way to tell how well a predictor h is doing on any particular observed (input, label) pair (x,y), we can ask how well we expect to do (on average) over the entire (admittedly unknown) source joint distribution $p_{X,Y}(x,y)$. Measuring the performance of a predictor h by evaluating its expected loss over pairs (x,y) drawn from the source joint distribution $p_{X,Y}(x,y)$ yields the *risk* $R[h(\cdot)]$ of that predictor on that source joint distribution:

$$R[h(\cdot)] = \mathbb{E}_{(X,Y)\sim p_{X,Y}}[\mathbf{loss}(h(X), Y)].$$

[2] Since

$$\begin{aligned} p_{X,Y}(X = \text{"image of keyboard cat"}, y) &= p_{Y|X}(y|X = \text{"image of keyboard cat"}) \cdot p_X(X = \text{"image of keyboard cat"}) \\ &= p_{Y|X}(y|X = \text{"image of keyboard cat"}) \cdot 0 \\ &= 0. \end{aligned}$$

[3] The h is from "hypothesis", an alternate term sometimes used to describe a predictor. We can think of a predictor or a prediction rule as an hypothesis about the labels of input objects.

[4] A given pair represents represents a particular example pair drawn from the source distribution over the input random variable X and the label random variable Y. A specific example pair (x,y) means that we have observed the random variable X to have taken on the value x, and the random variable Y to have taken on the value y. While we could emphasize the distinction between random variables and observations of the values those random variables take on by writing $(X = x,\ Y = y)$, we will typically use the standard simplification and instead write (x,y).

If we have a particular loss function in mind, we can be more specific. For example, the expected 01 loss yields the 01 risk (with respect to the source joint distribution $p_{X,Y}(x,y)$):

$$R_{01}[h(\cdot)] = \mathbb{E}_{(X,Y)\sim p_{X,Y}}[\mathbf{loss}_{01}(h(X),Y)].$$

For emphasis, the *risk* is sometimes referred to as the *expected risk* (to distinguish it from the *empirical risk* that we will introduce below). Other terms with the same meaning are *expected loss, generalization error*, or *source-distribution risk*. Having defined the risk of a predictor, we can now say that a predictor h is "good" on a particular source joint distribution if it has low risk $R[h(\cdot)]$ on that distribution. Considering the particular case of the 01 loss function $\mathbf{loss}_{01}$, we can observe an additional property of the 01 risk $R_{01}[h(\cdot)]$: it is the probability that the predictor h will incorrectly predict the label for any pair (x,y) drawn at random from the source joint distribution[5]:

$$R_{01}[h(\cdot)] = \mathbb{E}_{(X,Y)\sim p_{X,Y}}[\mathbf{loss}_{01}(h(X),Y)] = \mathbb{P}_{(X,Y)\sim p_{X,Y}}\{h(X)\neq Y\}.$$

Thus, a predictor h with low 01 risk also has low probability of incorrectly predicting the label. This equivalence between the risk and the probability of incorrect label prediction holds only for the 01 loss, and so when we subsequently consider other loss functions, we will evaluate predictors via their expected loss instead of their probability of incorrect label prediction.

Finally, observe that the risk depends on *both* the source joint distribution $p_{X,Y}(x,y)$ and the predictor h. However, we will be assuming that the source joint distribution is fixed. Thus, when we write the risk, we will only indicate the dependence on the predictor h : $R[h(\cdot)]$. For any predictor, we can ask what the risk is; we seek a predictor for which the risk is minimized.

2.5 A similarity and a difference between the primarily learning-from-data approach and the primarily expert-knowledge-based approach.

The definitions we have considered so far are not specific to the learning-from-data setting only; when designing an system based primarily on expert knowledge, your task is still to find a predictor with low expected loss. Whether we are incorporating more or less expert knowledge, the performance metric is the same. The difference is that in the learning-from-data setting we want to find a predictor with low expected loss primarily by learning from data.

3 The abstract ideal case of learning from the true source joint distribution

Where are we now? Viewed ideally, we now have a procedure for choosing a "best" predictor. In this ideal case, we assume that we have complete knowledge of the true source joint distribution $p_{X,Y}(x,y)$. We next choose a loss function $\mathbf{loss}(\cdot,\cdot)$. Now, we can decide that the "best" predictor $h^\star$ is the predictor that minimizes the expected value of the loss function on source joint distribution $p_{X,Y}(x,y)$. If the loss function really characterizes what we care about, the process of learning in this ideal setting can be viewed as a mapping from the source joint distribution and this loss function to an expected-loss-minimizing predictor:

$$[p_{X,Y}(x,y), \mathbf{loss}(\cdot,\cdot)] \mapsto h^\star.$$

[5]Since

$$\begin{aligned}\mathbb{E}_{(X,Y)\sim p_{X,Y}}[\mathbf{loss}_{01}(h(X),Y)] &= 0\cdot\mathbb{P}_{(X,Y)\sim p_{X,Y}}\{h(X)=Y\} + 1\cdot\mathbb{P}_{(X,Y)\sim p_{X,Y}}\{h(X)\neq Y\}\\ &= \mathbb{P}_{(X,Y)\sim p_{X,Y}}\{h(X)\neq Y\}\end{aligned}$$

How does actual practice differ from this ideal setting? The primary difference is that we do not ever have complete knowledge of the true source joint distribution; all that we actually have is a sample data set drawn from the true source joint distribution. In order to move forward with the process of learning a predictor, we need to make some assumptions about how this sample data set is drawn.

3.1 Sampling assumptions

It is possible to make various assumptions about the sampling of the data set. We will specifically be considering the setting of Statistical Learning Theory[9]. In this setting, we assume that we are given a particular observed sample data set s of m (input, label) pairs, written $s = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, where each point is an observation of the joint random variables (X_i, Y_i) that are each independently and identically distributed (i.i.d.) according to the source joint distribution $p_{X,Y}(x, y)$, *assumed to be the same source joint distribution that we will measure the expected loss with respect to*:

$$\mathcal{S}_{(X_i,Y_i)\sim p} = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_m, Y_m)\}$$

We could alternately express this assumption as "each random variable pair (X_i, Y_i) is sampled independently according to $p_{X,Y}$":

$$(X_i, Y_i) \underset{\text{ind.}}{\sim} p_{X,Y},$$

or as "our sample data set random variable $\mathcal{S}$ consists of m points being sampled i.i.d. from the source joint distribution $p_{X,Y}$":

$$\mathcal{S} \sim p_{X,Y}^m.$$

Observations/realizations of the "sample set" random variable $\mathcal{S}$ will be written as s, as indicated above. One of the assumptions considered above is often violated in actual practice: often, the sample data set is *not* drawn from the same distribution that we will measure our expected loss on. Despite this, essentially any type of analysis of machine learning methods assumes that the sample data set *is* drawn from the same distribution that we use to measure the error[6].

3.2 The abstract idea of a learning rule or inductive principle for choosing a predictor based on a sample data set

Since in actual practice the only knowledge of the true source joint distribution comes through the particular observed sample data set s, the process of learning will now look a little different. Again considered abstractly, and assuming that we have a loss function that characterizes what we care about, the process of learning from a sample data set is a mapping from a particular observed sample data set s and a loss function $\mathbf{loss}(\cdot, \cdot)$ to a predictor h

$$[s, \mathbf{loss}(\cdot, \cdot)] \mapsto h.$$

This will be our view of what a learning algorithm does: it takes a loss function and a sample data set of labeled examples and it outputs a predictor. Learning in general can be viewed as much more complex. The algorithm might get as input something that is not simply (input,label) pairs, its loss function might not simply compare a prediction to a label, and its output might not exactly be a predictor. However, for this course, we will primarily

[6]Why do we want these distributions to be the same? Clearly, if you want to build an OCR system and you intend to use the system on typewritten letters, it would be better to learn from a sample data set of typewritten letters instead of a sample data set of handwritten letters.

be considering just simple supervised learning in which we would like to find a good predictor based on a labeled sample data set.

When we were assuming that we had complete knowledge of the true source joint distribution, we said that the goal of learning is find a predictor that minimizes the expected loss on the true source joint distribution. Clearly, when we do not have complete knowledge of the true source joint distribution, this is not possible. All that we have is the sample data set. In this setting, it is perhaps reasonable to say that we should choose our predictor to minimize the expected loss on what we do have access to, and hope that this predictor will also do well on the true source joint distribution.

3.3 The empirical risk or average loss on the sample data set

The expected loss of a predictor h on a particular observed sample data set $s = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ could also be referred to as the *empirical risk* $\hat{R}_s[h(\cdot)]$ of the predictor on the sample data set, or as the "average loss on the sample data set", and is written

$$\hat{\mathbb{E}}_s[\mathbf{loss}(h(X), Y)] \triangleq \hat{R}_s[h(\cdot)] \triangleq \frac{1}{m}\sum_{i=1}^{m} \mathbf{loss}(h(x_i), y_i)$$

For each hypothesis, I can calculate the empirical error, the average of the loss on each of the m points in the sample data set. Empirical expectation $\hat{\mathbb{E}}_s$ just means "average over the observed sample set s". We will be using the "hat" " ˆ " to indicate that the quantity thus marked is determined by the sample data set.

The abstract method of selecting a predictor was to choose the predictor that minimizes the expected loss with respect to the true source joint distribution. In practice, our only knowledge of the true source joint distribution comes from the observed sample data set s. We do not know the expected loss; however, we can *estimate* the expected loss by computing the average loss on the sample data set. We can use this estimate of the expected loss in a learning rule to select a specific predictor: select the predictor $\hat{h}$ that minimizes the average loss on the sample data set (the "hat" indicates the dependence on the sample data set). This is a typical approach to learning, and the most common name for this approach is Empirical Risk Minimization (ERM) [9] (recall that "empirical risk" just refers to the average loss on the sample data set).

3.4 A specific learning rule: Empirical risk minimization (ERM)

Thus far we have just said "select the predictor", but typically we need to specify a particular class from which we select the predictor. In particular, as the first step of Empirical Risk Minimization we choose some hypothesis class $\mathcal{H}$. We view $\mathcal{H}$ as a set of predictors, written as

$$\mathcal{H} = \{h \ : \ \mathcal{X} \to \mathcal{Y}\} \subseteq \mathcal{Y}^{\mathcal{X}}$$

Speaking technically , this hypothesis class is a subset of all functions from the input domain $\mathcal{X}$ to the label domain $\mathcal{Y}$[7].]] And, the Empirical Risk Minimization learning rule for selecting a particular predictor (the Empirical Risk Minimizer $\hat{h}$) in the hypothesis class $\mathcal{H}$, given a sample data set s and a loss function $\mathbf{loss}(\cdot,\cdot)$can be written as follows:

$$\mathbf{ERM}_{\mathcal{H}}(s) = \hat{h} = \underset{h\in\mathcal{H}}{\operatorname{argmin}} \hat{R}_s(h) = \underset{h\in\mathcal{H}}{\operatorname{argmin}} \frac{1}{m}\sum_{i=1}^{m} \mathbf{loss}(h(x_i), y_i)$$

[7] $\mathcal{Y}^{\mathcal{X}}$ denotes the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$; this notation is related to the way in which the set of all subsets of a set, say $\mathcal{M}$, is indicated: $2^{\mathcal{M}}$ in connection with size of the power set for finite sets $\mathcal{M}$.

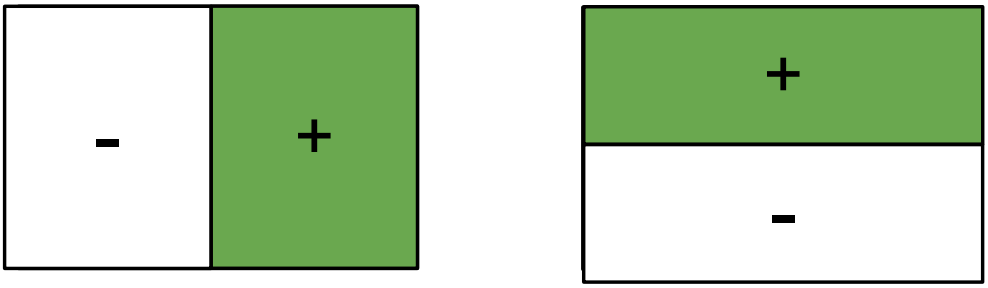

Figure 4.1: Examples from the hypothesis class of axis-aligned (greater-than-or-equal-to) hyperplanes in $\mathbb{R}^2$.

This is just a mapping from the observed sample data set s to a predictor $\hat{h}$. In particular, it is a mapping to the predictor $\hat{h}$ in our hypothesis class $\mathcal{H}$ that achieves minimum average loss on the observed sample data set s (equivalently described as minimum empirical risk $\hat{R}_s(h)$). Thus, one of the first things that we do in a learning problem is to limit ourselves only to hypotheses in a certain class. What sort of hypothesis classes might we consider?

4 Example hypothesis classes (in $\mathbb{R}^2$)

For the moment, we will continue to consider the binary labels, so that the label set is $\mathcal{Y} = \{+1, -1\}$. We could consider a wide variety of input domains, but for simplicity we will begin by considering $\mathcal{X} = \mathbb{R}^2$. With this input domain and label set, we can now consider some specific examples of hypothesis classes.

4.1 Axis-aligned greater-than-or-equal-to hyperplanes with input domain $\mathcal{X} = \mathbb{R}^2$

We can begin by considering the hypothesis class of "axis-aligned greater-than-or-equal-to hyperplanes in $\mathbb{R}^2$". Hypotheses in this class will predict that a particular example x is positive by looking at either the first coordinate x_1 or the second coordinate x_2 and asking whether the coordinate being consider has a value greater than or equal to a threshold θ. Hypotheses in this class are parametrized by which coordinate they look at ($i = 1$ or $i = 2$) and the by threshold value θ:

$$\begin{aligned}
\mathcal{X} &= \mathbb{R}^2 \\
\mathcal{Y} &= \{+1, -1\} \\
\mathcal{H} &= \{x_i \geq \theta \mid i \in \{1, 2\}, \theta \in \mathbb{R}\}
\end{aligned}$$

Having written down the hypothesis class, we might now ask "What do hypotheses in this class look like?"

Predictors in this class will assign a positive label to one side of the axis-aligned hyperplane specified by the coordinate being considered and the threshold θ. Anything on the other side is considered "negative".

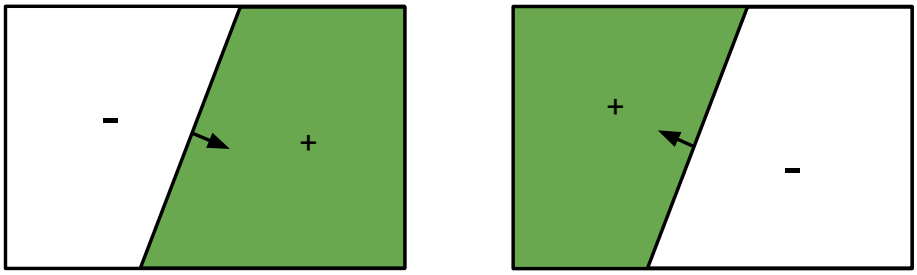

Figure 4.2: Examples from the hypothesis class of oriented affine oriented hyperplanes in $\mathbb{R}^2$.

4.2 Axis-aligned *oriented* hyperplanes with input domain $\mathcal{X} = \mathbb{R}^2$

The hypothesis class was very limited: it could not even express the prediction "Everything with first coordinate less than or equal to 0 has label +1"! We could expand the previous hypothesis class to include less-than-or-equal-to predictions by considering "oriented" axis-aligned hyperplanes. For hypotheses in this class, once we have selected the coordinate consider, we can say that the "positive" class (with label value+1) are those input examples for which either $x_i \geq \theta$ or $x_i \leq \theta$.

4.3 Oriented affine (sometimes called "linear") hyperplanes with input domain $\mathcal{X} = \mathbb{R}^2$

One hypothesis class that we will look at quite a bit is that of oriented affine[8] hyperplanes. In $\mathbb{R}^2$, we would write this class as

$$\begin{aligned}
\mathcal{X} &= \mathbb{R}^2 \\
\mathcal{Y} &= \{+1, -1\} \\
\mathcal{H} &= \left\{ x \mapsto \textbf{sign}\left(w^T x + b\right) \mid w \in \mathbb{R}^2, b \in \mathbb{R} \right\}
\end{aligned}$$

Predictors in this class map the observed input x to the sign of $\left(w^T x + b\right)$ for $w \in \mathbb{R}^2, b \in \mathbb{R}$. Essentially, we have some affine function of x, and we predict the label based on whether that affine function is greater than 0 or less than 0.

This gives us predictors which separate the input domain affinely into a positive class and a negative class.

4.4 The above hypothesis classes with input domain $\mathcal{X} = \mathbb{R}^d$

Each of the above hypothesis classes can be straightforwardly generalized from input domain $\mathbb{R}^2$ to input domain $\mathbb{R}^d$.

[8] Strictly speaking, the term "linear" refers to the case where the parameter $b = 0$: so that we just have $w^T x$. "Affine" indicates that we have a linear term plus an offset: $w^T x + b$. Frequently, however, both "linear" and "affine" are used to refer to the case of a linear term plus an offset.

4.5 Other hypothesis classes with input domain $\mathcal{X} = \mathbb{R}^2$

Generally, with input domain $\mathbb{R}^2$, I can think about the hypothesis classes as indicating the set examples with label +1. Instead of affine separators, I could talk about squares, rectangles, circles, or other geometric objects. Alternately, we could consider functions that involve higher degree polynomials of the coordinates of the input x. Further, all of these examples could be also be considered in $\mathbb{R}^d$.

4.6 Hypothesis classes with input domain $\mathcal{X} = \{0,1\}^d$

The input domain $\mathcal{X}$ does not always need to be in some $\mathbb{R}^d$. Rather than think of each input object as some point in $\mathbb{R}^d$, the input object could be made up of binary features:

$$\mathcal{X} = \{0,1\}^d$$

In this input domain, natural hypothesis classes would be an "or" of a few of the coordinates or an "and" of a few of these coordinates, or a formula including up to some specified length of the coordinates, or of decision trees with up to a certain number of nodes considering the coordinates.

5 Feature mapping

One important aspect of the hypothesis class that we should address is the question of what features of the input points the hypothesis actually operates on. In each of the example hypothesis classes above, we said that the input xs are vectors in $\mathbb{R}^d$. However, in many situations we encounter in practice, the input xs are not simply vectors in $\mathbb{R}^d$. More often, the input x is a much more complicated or abstract object such as an image or an email message.

When our actual initial input x is some abstract object, we must take some steps to re-represent that object in a format our predictor can act on. This re-representation process is called a "feature map" and is typically denoted as a function $\phi(\cdot)$. The feature mapping function $\phi(\cdot)$ maps abstract input object x to a form that the predictor can operate on: we will focus on mappings to $\mathbb{R}^d$, but other alternatives are possible, such as a mapping to $\{0,1\}^d$.

If we have a feature mapping in place, the predictors in the hypothesis classes act not on x but on $\phi(x)$. For example

$$\begin{aligned}\phi(x) &\in \mathbb{R}^2\\ \mathcal{Y} &= \{+1,-1\}\\ \mathcal{H} &= \{\phi(x_i) \geq \theta \mid i \in \{1,2\}, \theta \in \mathbb{R}\}\end{aligned}$$

A feature mapping is clearly useful when our input object xs are not things that we can directly do math on, such as sound or an image. However, we will see situations where it might be worthwhile to define a feature mapping even for input objects that are already in $\mathbb{R}^2$.

The feature mapping is not simply an afterthought. In a sense, part of what defines an hypothesis class is the particular feature mapping function $\phi(\cdot)$ used to specify what the hypothesis class gets to operate on. If we go back to the example of a spam filter for email messages, we could ask "Is our prediction based on the characters? On the words? On the word counts?

We can see the feature mapping function as what takes the original compressed bitstream of the email and subsequently yields the features that the predictors will use (characters, words, word counts, etc...). In this view, the feature mapping specifies what features are available for the predictor to use, and the hypothesis class specifies what we are allowed to do with these features.

6 Analysis of ERM, (flawed) version 1

Having set up the ground rules that we will follow for the time being, we can now get back to asking "How well does the principle of ERM do at giving us a predictor that will have low expected loss on the true source joint distribution?". We will see that the the empirical risk of a predictor h (the average loss of h on the sample data set) is actually a reasonably good estimate of the expected risk of the predictor (the expected loss of h on the true source joint distribution)

To reiterate, the expected loss of a predictor is a property of the predictor h and the of the true source distribution $p_{X,Y}(x,y)$. We don't know the distribution $p_{X,Y}$. We just claimed that the empirical risk $\hat{R}_s(h)$ is a good estimator of the expected loss of that predictor.

We will now give some details about how well empirical risk estimates the expected risk, with a particular focus on the 01 loss. If we focus on the 01 loss and we consider some specific predictor h, the expected 01 risk (or the 01 generalization error, or the expected 01 loss with respect to the true source distribution) is $R_{01}[h(\cdot)] = \mathbb{E}_{(X,Y)\sim p_{X,Y}}[\mathbf{loss}_{01}\{h(X), Y\}]$, and this is equal to the probability of incorrectly predicting a label, $\mathbb{P}_{(X,Y)\sim p_{X,Y}}\{h(X) \neq Y\}$.

If we now consider the sample data set of size m, we note the following: I have m examples, and for each example, the predictor has probability $\mathbb{P}_{(X,Y)\sim p_{X,Y}}\{h(X) \neq Y\}$ of incorrectly predicting the label. For simplicity, suppose that this expected mistake probability is 0.2. Thus, with each example I have probability 0.2 of making a mistake.

We now ask, "On average, how many mistakes do I expect to make on my sample data set of m examples?" This question corresponds exactly to having a binomial random variable ("heads", you are correct, "tails" you are incorrect) that we average over m trials. As our number of samples (trials) grows, the average outcome will converge fairly rapidly to the probability of actually making a mistake. Note also that this connection between the average loss and the probability of making a mistake relied on our use of the 01 loss.

For a more general loss, the sample average loss is an empirical expectation and this converges to its expectation on the source distribution fairly rapidly. In fact, we can quantify the convergence of the sample average loss to the expected loss. We can do this quantification for any loss, but we will first focus on the case of the binary 01 loss.

6.1 Hoeffding's inequality

Consider a specific predictor h. This predictor has an expected 01 loss $R_{01}(h)$ with respect to the true source joint distribution $p_{X,Y}(x,y)$. This expected 01 loss is the quantity we actually care about. However, we only have access to an *estimate* of the expected 01 loss of the predictor h, $R_{01}(h)$, in the form of the sample average 01 loss $\hat{R}_{s,01}(h)$ of the predictor computed on a sample s of m examples drawn i.i.d. from the true source distribution.

There are many such samples of size m that can be drawn from the true source distribution. The predictor h will have a different average 01 loss on each such sample. If the sample average 01 loss of the predictor h is usually close to the expected 01 loss of the predictor h, then we can feel more confident about using the sample average 01 loss in place of the expected 01 loss.

As it turns out, we can bound the probability that the predictor's sample average 01 loss computed on samples of size m will differ by more than ε from the predictor's expected 01 loss. We do this via Hoeffding's inequality for (averages of) *bounded* independent random variables[5]. This inequality depends heavily on the fact that the 01 loss is *bounded* (between 0 and 1).

Concentration of measure (from "Terence Tao 254A, Notes 1"):
We will primarily be using Hoeffding's inequality (stated below) in these lectures; however, this inequality is just one member of a large group of results collectively referred to as "large deviation inequalities" connected to "concentration of measure [read as: "probability]" results. The following excerpt from Terence Tao's course notes provides a clear statement of the context of these results:

> Suppose we have a large number of scalar random variables $X_1, \ldots, X_n$, which each have bounded size on average (e.g. their mean and variance could be $O(1)$). What can one then say about their sum $X_1 + \ldots + X_n$? If each individual summand X_i varies in an interval of size $O(1)$, then their sum of course varies in an interval of size $O(n)$. However, a remarkable phenomenon, known as *concentration of measure*, asserts that assuming a sufficient amount of independence between the component variables $X_1, \ldots, X_n$, this sum sharply concentrates in a much narrower range, typically in an interval of size $O(\sqrt{n})$. This phenomenon is quantified by a variety of *large deviation inequalities* that give upper bounds (often exponential in nature) on the probability that such a combined random variable deviates significantly from its mean. The same phenomenon applies not only to linear expressions such as the sum random variable $X_1 + \ldots + X_n$, but more generally to nonlinear combinations $F(X_1, \ldots, X_n)$ of such variables, provided that the nonlinear function F is sufficiently regular (in particular, if it is Lipschitz, either separately in each variable, or jointly in all variables).
>
> The basic intuition here is that it is difficult for a large number of independent variables $X_1, \ldots, X_n$ to "work together" to simultaneously pull a sum $X_1 + \ldots + X_n$ or a more general combination $F(X_1, \ldots, X_n)$ too far away from its mean. Independence here is the key; concentration of measure results typically fail if the X_i are too highly correlated with each other.

Hoeffding's Inequality (via Wikipedia)
Let $T_1, \ldots, T_m$ be *independent* scalar random variables. Assume further that the T_i are *bounded* so that $T_i \in [a_i, b_i]$.[a] Then, for the empirical mean of these m bounded variables,

$$\overline{T} = \frac{1}{m}\sum_{i=1}^{m} T_i,$$

we have the inequality

$$\mathbb{P}\left\{\left|\overline{T} - \mathbb{E}\left[\overline{T}\right]\right| \geq \varepsilon\right\} \leq 2\exp\left(-\frac{2\varepsilon^2 m^2}{\sum_{i=1}^{m}(b_i - a_i)^2}\right)$$

In our setting, we make the following identifications. For each bounded independent random variable T_i, we have the 01 loss of the predictor h on the i^{th} example $(X_i, Y_i) : \mathbf{loss}_{01}(h(X_i), Y_i)$ (and a function of a random variable is itself a random variable). Note that because the examples (X_i, Y_i) are i.i.d. random variables, the losses for the examples are themselves independent random variables. More than that, the each 01 loss is a *bounded* random variable. Whatever the value taken on by the joint random variables (X_i, Y_i), the corresponding 01 loss is certainly bounded between 0 and 1: $\mathbf{loss}_{01}(h(X_i), Y_i) \in [0, 1]$, so that $a_i = 0$ and $b_i = 1$. The empirical mean random variable of these m bounded random variables is now just the sample average loss random variable:

$$\hat{R}_{\mathcal{S},01}(h) = \frac{1}{m}\sum_{i=1}^{m} \mathbf{loss}_{01}(h(X_i), Y_i).$$

Further, the expected value (over all realizations s of of the "sample set" random variable $\mathcal{S}$ distributed as p^m) of the sample average loss random variable is just the expected 01 loss:

$$\begin{aligned}
\mathbb{E}_{\mathcal{S}\sim p^m_{X,Y}}\left[\hat{R}_{\mathcal{S},01}(h)\right] &= \mathbb{E}_{\mathcal{S}\sim p^m_{X,Y}}\left[\frac{1}{m}\sum_{i=1}^{m}\mathbf{loss}_{01}(h(X_i), Y_i)\right] \\
&= \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}_{\mathcal{S}\sim p^m_{X,Y}}\left[\mathbf{loss}_{01}(h(X_i), Y_i)\right] \\
&= \frac{1}{m}\sum_{i=1}^{m} R_{01}(h) = \frac{1}{m} m \cdot R_{01}(h) \\
&= R_{01}(h).
\end{aligned}$$

Plugging these identifications into the inequality above, we find

$$\mathbb{P}_{\mathcal{S}\sim p^m_{X,Y}}\left\{\left|\hat{R}_{\mathcal{S},01}(h) - R_{01}(h)\right| \geq \varepsilon\right\} \leq 2e^{-2\varepsilon^2 m}.$$

Again, note that the *boundedness* of the 01 loss, which led to the boundedness of the sample average loss random variable, was sufficient to establish the result above. The reasoning above would not have worked as stated for a loss function that could take on an arbitrarily large value; as an example, we will later emphasize convex loss functions and the reasoning stated above would not work for any reasonable convex loss (a loss function that gives 0 loss to both correct and incorrect predictions is convex and bounded, but thoroughly pointless). When we get to convex loss functions, we will thus be motivated to consider alternative lines of reasoning.

[a] There are other inequalities, analogous to Hoeffding's inequality, for which the assumption that the random variables are bounded is not required. These alternative inequalities often come in the form of bounds on the sum of sub-Gaussian random variables. Search "Terence Tao 254A, Notes 1" for many more details.

Thus, Hoeffding's inequality gives us a bound on the probability that the sample average 01 loss (which we will have access to) will differ from the expected 01 loss (which is what we really care about). Alternately, for reasonably large sample size m, we can say that Hoeffding's inequality tells us that for *most* observed sample data sets of size m, the average 01 loss of h on those sample data sets will be "close" to the expected 01 loss of h on the source distribution.

Sometimes it will be more useful to interpret Hoeffding's inequality from a different perspective.

Re-interpreting Hoeffding's inequality.
Hoeffding's inequality was expressed in terms of a bound on the probability of deviation of the empirical 01 loss random variable from the expected 01 loss:

$$\mathbb{P}_{\mathcal{S}\sim p_{X,Y}^m}\left\{\left|\hat{R}_{\mathcal{S},01}(h) - R_{01}(h)\right| \geq \varepsilon\right\} \leq 2e^{-2\varepsilon^2 m}.$$

This could equivalently be stated

$$\mathbb{P}_{\mathcal{S}\sim p_{X,Y}^m}\left\{\left|\hat{R}_{\mathcal{S},01}(h) - R_{01}(h)\right| < \varepsilon\right\} \geq 1 - 2e^{-2\varepsilon^2 m}.$$

This tells us that with probability at least $1 - 2e^{-2\varepsilon^2 m}$, we will have a sample set realization s for which

$$\left|\hat{R}_{s,01}(h) - R_{01}(h)\right| < \varepsilon.$$

(That is, when we make observations of the "sample set" random variable $\mathcal{S}$, the probability is at least $1 - 2e^{-2\varepsilon^2 m}$ that empirical risk of the hypothesis h evaluated on the specific sample set realization s is within ε of the expected risk of h.) We can get a more commonly used statement by equating the probability of a large difference with a value δ :

$$\delta = 2e^{-2\varepsilon^2 m},$$

which we can solve for the "deviation amount" ε :

$$\begin{aligned}
\frac{\delta}{2} &= e^{-2\varepsilon^2 m} \\
\implies \frac{2}{\delta} &= e^{2\varepsilon^2 m} \\
\implies \log\frac{2}{\delta} &= 2\varepsilon^2 m \\
\implies \frac{\log\frac{2}{\delta}}{2m} &= \varepsilon^2 \\
\implies \varepsilon &= \sqrt{\frac{\log\frac{2}{\delta}}{2m}}.
\end{aligned}$$

Where we use log to mean the natural logarithm. Restating our previous statement once more, but now in terms of the probability δ instead of the deviation amount ε, we get the following:
With probability at least $1 - \delta$, we will have a sample set realization s for which

$$\left|\hat{R}_{s,01}(h) - R_{01}(h)\right| < \sqrt{\frac{\log\frac{2}{\delta}}{2m}}.$$

When we consider a particular hypothesis h, we can thus use our reformulation of Hoeffding's inequality to state that with probability at least $1 - \delta$[9] the absolute value of the difference between the expected 01 loss $R_{01}(h)$ and

[9]Where we take δ to be some small probability that "things will go wrong" and we will get a "misleading" sample data set. A "misleading" sample data set is a sample of m points for which the sample average 01 loss of h differs from the expected 01 loss of

the average 01 loss on a sample data set of size m $\hat{R}_{s,01}(h)$, is less than $\sqrt{\frac{\log\frac{2}{\delta}}{2m}}$: With probability at least $1-\delta$, we will have a sample set realization s for which

$$\left|R_{01}(h) - \hat{R}_{s,01}(h)\right| < \sqrt{\frac{\log\frac{2}{\delta}}{2m}}.$$

For example, if we specify that we want the probability that "things will go wrong" to be no more than $\delta = 0.001$ and we have drawn a sample of size $m = 10,000$, the above statement becomes: With probability 0.999, $\left|R(h) - \hat{R}_s(h)\right| < 0.019$. If we want $\delta = 0.001$ and still have $m = 10,000$, we can state that with probability 0.99999, $\left|R(h) - \hat{R}_s(h)\right| < 0.025$. That is, for 99.999% of the possible sample data sets of size $10,000$ points, the empirical risk of h and the expected risk of h will be within 0.025. For a fixed size of sample data set, a smaller δ value leads to a larger bound on the absolute difference. For a fixed δ value, increasing the size of the sample data set leads to a smaller bound on the absolute difference.

To emphasize: throughout all of this, the expected 01 loss of the hypothesis h, $R_{01}(h)$, is evaluated with respect to the true source distribution p, and so remains fixed. On the other hand, the average 01 loss of the hypothesis h over possible sample data sets ($i.e.$ as a function of the "sample set" random variable $\mathcal{S}$) , $\hat{R}_{\mathcal{S},01}(h)$, is a random variable because it is a function of the the "sample set" random variable. There are many realizations s of the "sample set" random variable $\mathcal{S}$ (that is, many possible sets of m (input,label) pairs that we could draw from the source distribution), and testing an hypothesis h on each possible realized/observed sample data set s will have a corresponding sample average 01 loss.

Summing up: The source distribution is fixed, so $R_{01}(h)$ is fixed, is just a number. The sample data set random variable $\mathcal{S}$ is made up of m pairs each distributed i.i.d. from the source distribution, and so when we write $\hat{R}_{\mathcal{S},01}(h)$, we are emphasizing that this is a function of a random variable, and so is itself a random variable. $\hat{R}_{\mathcal{S},01}(h)$ is a function of the sample set random variable $\mathcal{S}$; $\hat{R}_{s,01}(h)$ is a function of a realization s of the sample set random variable $\mathcal{S}$. Once more: s is a realization of the "sample set" random variable $\mathcal{S}$; $\hat{R}_{s,01}(h)$ is a realization of the "empirical risk" random variable $\hat{R}_{\mathcal{S},01}(h)$. Finally, we will always assume that the m (input, label) random variable pairs (X_i, Y_i) each are distributed independently from the source distribution $p_{X,Y}$.

6.2 Consideration of Hoeffding's inequality

Having looked at Hoeffding's inequality, we might well ask whether it provides any justification for using the Empirical Risk Minimization learning rule. If we know that we can consider Hoeffding's inequality for each hypothesis and it will tell us that (with probability $1-\delta$) the empirical risk for that hypothesis is close to the expected risk for that hypothesis, does this imply that the hypothesis that minimizes the empirical risk will also minimize the expected risk? If it did, then picking the hypothesis that minimizes the empirical risk would be justified.

Basically, if the empirical error is a good estimator for the expected error and we want to minimize the expected error, it seems that minimizing the empirical error would be a good thing. If we start with the statement that for any hypothesis h, with $1-\delta$, we will have a sample set realization s for which $\left|R_{01}(h) - \hat{R}_{s,01}(h)\right| < \sqrt{\frac{\log\frac{2}{\delta}}{2m}}$, can we conclude that in fact the *expected* 01 risk of the empirical 01 risk minimizer is going to be "close" ($i.e.$ within $\sqrt{\frac{\log\frac{2}{\delta}}{2m}}$) to the *empirical* 01 risk of the empirical 01 risk minimizer? And, if this "closeness" holds, can we say that

h by more than $\sqrt{\frac{\log\frac{2}{\delta}}{2m}}$. In this case, the performance on the sample data set does not accurately reflect the performance on the source distribution: an hypothesis that would perform very poorly on the source distribution might perform extremely well on such a "misleading" sample data set.

we thereby establish that the *empirical* 01 risk minimizing hypothesis "almost" minimizes the *expected* 01 risk? We now consider a plausible but incorrect argument that purports to establish just the result discussed above. We will subsequently consider an example that will highlight the flaw in this reasoning, and point us towards a correct line of argument.

An incorrect argument for justifying ERM using Hoeffding's inequality:
For any particular hypothesis h, with probability at least $1-\delta$, we can bound the absolute difference between the expected 01 risk of h and the empirical 01 risk of h :

$$\left|R_{01}(h)-\hat{R}_{s,01}(h)\right|<\sqrt{\frac{\log\frac{2}{\delta}}{2m}}$$

Suppose that we determine which hypothesis in our hypothesis class minimizes the empirical 01 risk: $\hat{h}$, defined as $\hat{h}\triangleq\underset{h\in\mathcal{H}}{\operatorname{argmin}}\hat{R}_{s,01}(h)$, (the minimizer of $\hat{R}_{s,01}(\cdot)$, which is the empirical 01 risk on the observed sample data set s that we were given). Having selected this *empirical* 01 risk minimizing hypothesis out of our hypothesis class $\mathcal{H}$, we now place $\hat{h}$ into Hoeffding's inequality. This would be expressed as: For the empirical 01 risk minimizing hypothesis $\hat{h}$, we have with probability at least $1-\delta$, the following bound on the absolute difference between $R_{01}\left(\hat{h}\right)$ (the *expected* 01 risk of the empirical 01 risk minimizer $\hat{h}$) and $\hat{R}_{s,01}\left(\hat{h}\right)$ (the *empirical* 01 risk of the empirical 01 risk minimizer $\hat{h}$):

$$\left|R_{01}\left(\hat{h}\right)-\hat{R}_{s,01}\left(\hat{h}\right)\right|<\sqrt{\frac{\log\frac{2}{\delta}}{2m}}.$$

Which implies

$$(1)\ R_{01}\left(\hat{h}\right)<\hat{R}_{s,01}\left(\hat{h}\right)+\sqrt{\frac{\log\frac{2}{\delta}}{2m}}.$$

Suppose that we now determine which hypothesis in our hypothesis class minimizes the *expected* 01 risk: $h^{\star}$, defined as $h^{\star}\triangleq\underset{h\in\mathcal{H}}{\operatorname{argmin}}R_{01}(h)$, (the minimizer of $R_{01}(\cdot)$, which is the expected 01 risk over the true source distribution). Having selected this *expected* 01 risk minimizing hypothesis out of our hypothesis class $\mathcal{H}$, we now place $h^{\star}$ into Hoeffding's inequality. This would be expressed as: For the expected 01 risk minimizing hypothesis $h^{\star}$, we have, with probability at least $1-\delta$, the following bound on the absolute difference between $R_{01}(h^{\star})$ (the *expected* 01 risk of the expected 01 risk minimizer $h^{\star}$) and $\hat{R}_{s,01}(h^{\star})$ (the *empirical* 01 risk the expected 01 risk minimizer $h^{\star}$):

$$\left|R_{01}(h^{\star})-\hat{R}_{s,01}(h^{\star})\right|<\sqrt{\frac{\log\frac{2}{\delta}}{2m}}.$$

Which implies

$$(2)\ \hat{R}_{s,01}(h^{\star})<R_{01}(h^{\star})+\sqrt{\frac{\log\frac{2}{\delta}}{2m}}$$

We also observe, from the definition of the empirical 01 risk minimizer $\hat{h}$, (recalling: $\hat{h}\triangleq\underset{h\in\mathcal{H}}{\operatorname{argmin}}\hat{R}_{s,01}(h)$), that we know the empirical 01 risk minimizer $\hat{h}$ has lower (or equal) empirical 01 risk when compared to all other hypotheses, and in particular, when compared to the expected 01 risk minimizer $h^{\star}$:

$$(3)\ \hat{R}_{s,01}\left(\hat{h}\right)\leq\hat{R}_{s,01}(h^{\star}).$$

Is the following correct?: With probability at least$1-\delta$, we will have a sample set realization s such that

$$R_{01}\left(\hat{h}\right) \overset{(1)}{\leq} \hat{R}_{s,01}\left(\hat{h}\right) + \sqrt{\frac{\log\frac{2}{\delta}}{2m}} \overset{(3)}{\leq} \hat{R}_{s,01}\left(h^{\star}\right) + \sqrt{\frac{\log\frac{2}{\delta}}{2m}} \overset{(2)}{\leq} R_{01}\left(h^{\star}\right) + \sqrt{\frac{\log\frac{2}{\delta}}{2m}} + \sqrt{\frac{\log\frac{2}{\delta}}{2m}}$$

where $h^{\star} \triangleq \operatorname{argmin} R_{01}\left(h\right)$ is the hypothesis that minimizes the expected 01 risk. If the above were correct, it would mean that the expected loss of the empirical 01 risk minimizing hypothesis would be (with probability at least $1-\delta$) no more $2\sqrt{\frac{\log\frac{2}{\delta}}{2m}}$ worse than the best possible expected 01 risk, $R_{01}\left(h^{\star}\right)$. This would then appear to justify the ERM principle.

I indicated previously that the argument was not correct. What is the flaw? Why can we not use the reasoning above to justify the ERM principle? What have we neglected to consider?

6.3 The flaw in the argument: Hoeffding's inequality holds for each h separately; it does *not* hold for all $h \in \mathcal{H}$ concurrently

One way of expressing the problem is to note that there are actually *two* different ways that things might go "wrong", and Hoeffding's inequality only considers *one* of those ways. With a particular hypothesis, you might have a sample data set that makes it look misleadingly good (Hoeffding's inequality addresses this). For a particular sample data set, out of all the hypotheses in your hypothesis class, there might be one hypothesis (or more than one) that happens to look misleadingly good on that particular sample data set (Hoeffding's inequality does not address this at all).

The step in the reasoning above where we determined which hypothesis $\hat{h}$ in our hypothesis class $\mathcal{H}$ minimizes the empirical 01 risk is precisely the setting of the second problem. We have the sample data set that we were given. For this particular sample data set, we consider the empirical 01 risk of each in turn. If an hypothesis exists in the hypothesis class that does extremely well on the particular data set we were given, that hypothesis will be selected by the ERM principle, regardless of how well that hypothesis would perform on the true source distribution. This need not always worry us. For example, if our hypothesis class consists of exactly one hypothesis, and that hypothesis happens to attain average 01 loss of 0 on a sample data set of 1,000 example pairs, it seems unlikely that that performance would be strongly different from the performance on the true source distribution. Clearly, there are other situations that will be problematic: if our hypothesis class contained 2^{1000} hypothesis, each corresponding to a specific possible $(+1,-1)$ labeling of all 1,000 input values in our sample data set, then we know that there is an hypothesis in the class that will achieve average 01 loss of 0 on the sample data set, no matter what the labels on the example pairs were. For such a "rich" hypothesis class (relative to the sample data set size), average 01 loss of 0 is essentially meaningless; nonetheless, the ERM principle will choose precisely that hypothesis achieving average 01 loss of 0.

6.4 What did we do wrong? We neglected the complexity (here, the cardinality) of the hypothesis class

When the hypothesis class is "diverse"/"complex" and large (relative to the sample data set size), we can expect the ERM principle to choose an hypothesis that has "meaningless" good performance on the sample data set. In particular, there will be no reason to expect the ERM-chosen hypothesis to have similarly good expected 01 risk on the true source distribution. Thus, we must account for the complexity and (relative) size of the hypothesis class in any performance evaluation that we consider. Not considering the complexity/relative size of the hypothesis

class over which we selected $\hat{h}$ using the ERM principle meant that we were not actually justified in saying that $R_{01}\left(\hat{h}\right) \overset{(1)}{\leq} \hat{R}_{s,01}\left(\hat{h}\right) + \sqrt{\frac{\log \frac{2}{\delta}}{2m}}$. We will soon consider what we *are* justified in saying, but first we will look at a concrete example that will illustrate what we mean by "complexity"/size (relative to the sample data set size) of the hypothesis class and the specific problem that occurs when we do not take the complexity of the hypothesis class into account.

6.4.1 Gender prediction: An example involving complexity (here, cardinality) of the hypothesis class, relative to sample data set size

We will now consider a specific prediction task: the input object will be a person; the label will be "male" or "female". The input domain will be "people in North America". Thus,

$$\mathcal{X} = \{\text{"people in North America"}\}$$
$$\mathcal{Y} = \{\text{"male","female"}\}.$$

A particular sample data set of three people from our source distribution of people in North America might be {("Brian Greene","male"),("Mary Roach","female"), ("Carl Zimmer","male")}. Our sample data set will be of size 40, consisting of the people in the room when this lecture was given. We will use the principle of Empirical Risk Minimization on this sample data set to select a particular hypothesis from an hypothesis class.

To illustrate the notion of the complexity of an hypothesis class, we will consider four different possible classes of hypotheses. Each of the four hypothesis classes will have a finite number of possible hypotheses, but that finite number will range from quite small to rather fairly large.

Possible hypothesis classes to consider:

$$\mathcal{H}_{birth} = \{\text{"predictors based only on month and day of birthdate"}\}$$
$$\mathcal{H}_{nation} = \{\text{"predictors based only on nationality"}\}$$
$$\mathcal{H}_{hair} = \{\text{"predictors based only on short or long hair"}\}$$
$$\mathcal{H}_{phone} = \{\text{"predictors based only on last four digits of phone number"}\}$$

For example, a possible hypothesis from $\mathcal{H}_{nation}$ would assign, for each of the 196 nations on Earth, a predicted gender for all of the people from that nation, such as: "If Canadian, predict male", "If French, predict female", "If Bolivian, predict male", and so on. A possible hypothesis from $\mathcal{H}_{hair}$ would be "If hair is short, predict male" and "If hair is long, predict female". Likewise, an hypothesis from $\mathcal{H}_{birth}$ would assign a predicted gender to all people born on each of the 365 (or 366) days of the year, and hypotheses from $\mathcal{H}_{phone}$ would assign, for each of the 10,000 possible four digit combinations, a particular predicted gender to all people with the same last four digits of their phone number.

Given these four possible classes of hypotheses and our knowledge of the true source distribution (the population of North America), we can confidently suppose that one of these classes contains an hypothesis that is likely to be the best. By "best" we mean that it will achieve the lowest expected 01 loss when applied to the entire population of North America (out of all the hypotheses in each of these four classes). The class is $\mathcal{H}_{hair}$, and the hypothesis that we expect to be "best" is "If hair is short, predict male" and "If hair is long, predict female". Having established what we believe to be $h^\star$ (the hypothesis out of these four hypothesis classes that will minimize the *expected* 01 loss when applied to the entire population of North America) we can now ask what the Empirical Risk Minimization principle would select in this setting.

One of these hypothesis classes in particular is likely to contain an hypothesis that achieves a sample average 01 loss of 0 on a sample data set of size 40: $\mathcal{H}_{phone}$. The 40 people in the room are very likely to each have a distinct last four digits of their phone number. If the last four digits *are* distinct for each person, we just need to identify a particular hypothesis that has matching predictions of the corresponding gender values. Such an hypothesis will thus "meaninglessly" achieve sample average 01 loss of 0 on a sample data set of size 40. In fact, when its performance considered on the entire population of North America, our knowledge suggests that such an hypothesis will have expected 01 loss of 0.5, because we believe that phone numbers are allocated at random between men and women.

To contrast, the hypothesis that we think will do best when applied to the entire population of North America is unlikely to achieve sample average 01 loss of 0 on a sample data set of size 40, because we know that there are some women with short hair and some men with long hair. The particular sample data set in the room had 1 female with short hair, and 1 male with long hair, yielding sample average 01 loss of $\frac{2}{40} = 0.05$ for the *expected* 01 loss minimizing hypothesis from $h^\star$. It seems likely that this sample data set performance will be quite similar to the expected performance on the entire population.

This situation, where an hypothesis in $\mathcal{H}_{phone}$ achieves 0 sample average 01 loss, is likely to occur most of the time for relatively small sample data set sizes[10]. When this happens, the principle of Empirical Risk Minimization will select *that* "meaninglessly good" hypothesis (from $\mathcal{H}_{phone}$), rather than the hypothesis from $\mathcal{H}_{hair}$ that we think will do best on the entire population.

Why does Empirical Risk Minimization select the "meaninglessly good" hypothesis? It is essentially a matter of (relative sizes of) numbers. In particular, there are $2^{10,000}$ different predictors based on last four digits of a person's phone number. For the sample data set of the 40 people in the room, there are only 2^{40} *possible* gender "labellings". It is very likely that the 40 people in the room will have distinct values for the last four digits of their phone numbers; if the number values are distinct, one of the $2^{10,000}$ different predictors in $\mathcal{H}_{phone}$ will predict *perfectly*. This perfect predictor will be selected by the Empirical Risk Minimization principle.

When an hypothesis class is too "complex" or large (relative to the sample data set size), it is highly likely to contain "meaninglessly good" predictors. We must take this hypothesis class size/"complexity" into account instead of just selecting an hypothesis based on minimizing the empirical 01 risk. We will see how the use of the union bound (in the context of Hoeffding's inequality) allows us to incorporate the hypothesis class size (for hypothesis classes with a finite number of hypotheses).

6.5 Correcting the flaw by using the union bound (via Andrew Ng)

As the first step toward a performance bound that will take into account the size of the hypothesis class, let us recall the initial form of Hoeffding's inequality. In the form we initially saw it, Hoeffding's inequality told us that for a particular hypothesis h_i in our (finite) hypothesis class $\mathcal{H}$, we can bound the probability of h_i having "loss-estimation-ε-closeness-failure-for-h_i" as:

$$\mathbb{P}_{\mathcal{S}\sim p_{X,Y}^m}\left\{\left|\hat{R}_{\mathcal{S},01}\left(h_i\right) - R_{01}\left(h_i\right)\right| \geq \varepsilon\right\} \leq 2e^{-2\varepsilon^2 m}.$$

In words, this shows that for the particular hypothesis h_i, the probability of event "sample average 01 loss is farther than ε from the expected 01 loss" (alternately describable as "loss-estimation-ε-closeness-failure-for-h_i") is bounded by $2e^{-2\varepsilon^2 m}$.

When we use the ERM principle to select an hypothesis from an hypothesis class, we only have a problem when the selected hypothesis happens to be one of the hypotheses which happens to "slip through" or has "loss-estimation-ε-closeness-failure-for-h_i"; an event that we can expect to happen with probability no more than $2e^{-2\varepsilon^2 m}$.

[10] Say, for sample data set sizes less than some fraction of 10,000 people determinable by an analog to the birthday problem.

An hypothesis that "slips through" can have sample performance much better (*i.e.* misleadingly better) than its expected performance on the source distribution[11]. If the "misleadingly better" performance of that hypothesis on the sample data set is the *best* performance (out of all hypotheses in the hypothesis class) on the sample data set, the ERM principle will select that hypothesis, and we will then get poor performance when we apply this hypothesis to the source distribution. By contrast, if *no* hypothesis in the class "slips through", then we can be confident that when ERM selects an hypothesis, the hypothesis' performance on the sample data set is representative of its performance on the source distribution.

The more hypotheses in the hypothesis class, the more likely it is that one of them will "slip through" and get "misleadingly good" performance on the sample data set, and thereby be selected by the ERM principle. When an hypothesis "slips through" we might say that ERM can possibly "break" by selecting the "misleadingly good" hypothesis. If *none* of the hypotheses "slip through", they *all* display performance on the sample data set that is "representative" of their performance on the source distribution; when none of the hypotheses "slip through", we know that in this case ERM cannot break from seeing "misleadingly good" hypothesis performance[12]. If we can establish (a bound on) the probability that *none* of the hypotheses "slips through", we will thus have a bound on the probability of being in the setting where we know that ERM cannot break from seeing "misleadingly good" hypothesis performance on the sample data set (because we are in the setting where all of the hypotheses display "representative" performance on the sample data set).

We will see how the union bound can be used to establish a bound this probability.

The union bound.
Let $A_1,\ A_2, \ldots, A_k$ be k different events (that need not be independent). Then, the union bound states

$$\mathbb{P}\left\{A_1 \cup \cdots \cup A_k\right\} \leq \mathbb{P}\left\{A_1\right\} + \ldots + \mathbb{P}\left\{A_k\right\}.$$

In probability theory, the union bound is usually stated as an axiom, but it also makes intuitive sense: The probability of any one of k events happening is at most the sums of the probabilities of the k different events.

Having stated the generic form of the union bound, we note that the events we currently care about are of the form "hypothesis h_i looks misleadingly good". Thus, let event A_i be the event "hypothesis h_i looks misleadingly good" or "hypothesis h_i slipped through", explicitly written out as the as event $\left|\hat{R}_{\mathcal{S},01}\left(h_i\right) - R_{01}\left(h_i\right)\right| \geq \varepsilon$. Hoeffding's inequality tells us that for any particular one of these A_i, it holds true that $\mathbb{P}_{\mathcal{S}\sim p^m_{X,Y}}\left\{A_i\right\} \leq 2e^{-2\varepsilon^2 m}$. We want to establish a bound on the probability that *none* of these events occurs.

To do this, we first use use the union bound to establish a bound on the probability that *some* hypothesis in the hypothesis class "slips through". Since the complement of the event "*there is at least one* hypothesis in the hypothesis class that slips through" is the event "*none* of the hypotheses in the hypothesis class slip through", the probability bound for the "nothing slips through" event is 1 minus the probability bound for the "something slips through" event.

With this in mind, we first use the union bound to establish the probability bound on the "something slipped

[11] In the absolute-value form of Hoeffding's inequality above, an hypothesis can also have "misleadingly worse" performance, but we will not be considering the effect of that event here. The effect of "misleadingly worse" performance is typically only a problem if it affects the hypothesis that we "should have chosen" and the "second-place" hypothesis chosen instead is much less good in terms of its actual performance on the source distribution.

[12] Note that we might still have gotten an "atypical" sample data set. The initial bound from Hoeffding's inequality tells us how likely we are to have gotten an "atypical" sample.

through" event:

$$\begin{aligned}\mathbb{P}_{\mathcal{S}\sim p_{X,Y}^m}\left\{\exists h_i \in \mathcal{H} \mid \left|\hat{R}_{\mathcal{S},01}\left(h_i\right) - R_{01}\left(h_i\right)\right| \geq \varepsilon\right\} &= \mathbb{P}_{\mathcal{S}\sim p_{X,Y}^m}\left\{A_1 \cup \cdots \cup A_{|\mathcal{H}|}\right\} \\ &\leq \sum_{i=1}^{|\mathcal{H}|} \mathbb{P}_{\mathcal{S}\sim p_{X,Y}^m}\left\{A_i\right\} \\ &\leq \sum_{i=1}^{|\mathcal{H}|} 2e^{-2\varepsilon^2 m} \\ &\leq 2\left|\mathcal{H}\right| e^{-2\varepsilon^2 m}.\end{aligned}$$

If we then subtract 1 from both sides, we find that the probability of the "nothing slipped through" event is

$$\begin{aligned}\mathbb{P}_{\mathcal{S}\sim p_{X,Y}^m}\left\{\neg\exists h_i \in \mathcal{H} \mid \left|\hat{R}_{\mathcal{S},01}\left(h_i\right) - R_{01}\left(h_i\right)\right| \geq \varepsilon\right\} &= \mathbb{P}_{\mathcal{S}\sim p_{X,Y}^m}\left\{\forall h_i \in \mathcal{H},\ \left|\hat{R}_{\mathcal{S},01}\left(h_i\right) - R_{01}\left(h_i\right)\right| \leq \varepsilon\right\} \\ &\geq 1 - 2\left|\mathcal{H}\right| e^{-2\varepsilon^2 m}.\end{aligned}$$

So, with probability at least $1 - 2\left|\mathcal{H}\right| e^{-2\varepsilon^2 m}$ (frequently stated as "with high probability"), we will be in the setting where "nothing slipped through" and *all* of the hypotheses in the hypothesis class will display representative performance on the sample data set: the empirical risk random variable $\hat{R}_{\mathcal{S},01}\left(h_i\right)$ will take on a value within ε of $R_{01}\left(h_i\right)$ for all $h_i \in \mathcal{H}$. When all of the hypotheses in the hypothesis class show representative performance on the sample, we know that in this case ERM cannot break from seeing "misleadingly good" hypothesis performance [13]

This is called a uniform convergence result, because this is a bound that holds simultaneously for all (rather than for just one at a time) $h_i \in \mathcal{H}$.

In the derivation above there were three quantities of interest: the sample size m, the deviation amount ε, and the probability of error (previously denoted by δ). We note that we can bound any one of these three quantities in terms of the other two.

[13] But again, we might still have gotten an "atypical" sample data set.

Re-interpreting the inequality we got from the union bound.
The inequality we got was expressed in terms of a bound on the probability that "nothing slips through":

$$\mathbb{P}_{\mathcal{S}\sim p_{X,Y}^m}\left\{\forall h_i \in \mathcal{H},\ \left|\hat{R}_{\mathcal{S},01}\left(h_i\right) - R_{01}\left(h_i\right)\right| \leq \varepsilon\right\} \geq 1 - 2\left|\mathcal{H}\right| e^{-2\varepsilon^2 m}.$$

This tells us that with probability at least $1 - 2\left|\mathcal{H}\right| e^{-2\varepsilon^2 m}$, we will have gotten a sample set realization s where

$$\forall h_i \in \mathcal{H},\ \left|\hat{R}_{s,01}\left(h_i\right) - R_{01}\left(h_i\right)\right| \leq \varepsilon$$

We can get a more commonly used statement by equating the probability that "there is some hypothesis that is misleadingly good" (alternately describable as the probability that "there is some hypothesis is bad and could break ERM") with a value δ_{bad} :

$$\delta_{bad} = 2\left|\mathcal{H}\right| e^{-2\varepsilon^2 m},$$

which we can solve for the "deviation amount" ε :

$$\begin{aligned}
\frac{\delta_{bad}}{2\left|\mathcal{H}\right|} &= e^{-2\varepsilon^2 m} \\
\implies \frac{2\left|\mathcal{H}\right|}{\delta_{bad}} &= \left|\mathcal{H}\right| e^{2\varepsilon^2 m} \\
\implies \log\frac{2\left|\mathcal{H}\right|}{\delta_{bad}} &= 2\varepsilon^2 m \\
\implies \frac{\log\frac{2|\mathcal{H}|}{\delta_{bad}}}{2m} &= \varepsilon^2 \\
\implies \varepsilon &= \sqrt{\frac{\log\left|\mathcal{H}\right| + \log\frac{2}{\delta_{bad}}}{2m}}.
\end{aligned}$$

Where we use log to mean the natural logarithm. Restating our previous statement once more, but now in terms of the probability that some hypothesis "slips through", δ_{bad}, instead of the deviation amount ε, we get the following: With probability at least $1 - \delta_{bad}$, we will have gotten a sample set realization s where

$$\forall h_i \in \mathcal{H},\ \left|\hat{R}_{s,01}\left(h_i\right) - R_{01}\left(h_i\right)\right| \leq \sqrt{\frac{\log\left|\mathcal{H}\right| + \log\frac{2}{\delta_{bad}}}{2m}}.$$

That is, with probability at least $1 - \delta_{bad}$, we will be in the case where ERM won't break by selecting an hypothesis that is "misleadingly good" (because we will be in the case where all of the hypotheses display "representative" performance on the sample data set). With a slight abuse of terminology, this is called "the union bound".

We can now return to our note that there are actually *two* different ways that things might go "wrong": With a particular hypothesis, you might have a sample data set that makes it look misleadingly good (This is what Hoeffding's inequality by itself addressed). For a particular sample data set, out of all the hypotheses in your hypothesis class, there might be one hypothesis (or more than one) that happens to look misleadingly good on that particular sample data set (The union bound addresses this). When we combine the union bound and Hoeffding's inequality, we thereby get a bound that covers variation both over possible sample data sets and over possible

hypotheses in an hypothesis class.

6.6 What does the union bound tell us? (It is a post-hoc guarantee.)

What does this bound this tell us? It tells us that if we run our learning algorithm on a "sufficiently large" sample data set and we happen to find an hypothesis with low empirical error on this data set, we can be "confident" that that hypothesis will perform similarly well on the source distribution. To be more specific, we can specify a desired level of confidence (via δ_{bad}) and the corresponding expression for the bound ($\sqrt{\frac{\log|\mathcal{H}|+\log\frac{2}{\delta_{bad}}}{2m}}$) will indicate how large a sample data set we would need in order ensure that the performance of the hypotheses on the sample data set would be "representative".

For example, suppose that we have an hypothesis class of size $|\mathcal{H}| = 10,000$, and we want the probability of there being some "possibly misleading" hypothesis in $\mathcal{H}$ (such that its sample data set performance differs from its source distribution performance by more than $\varepsilon = 0.01$) to be no higher than $\delta_{bad} = e^{-7}$. We can plug so these values into the bound to find out how large of a sample data set we need to have in order to meet these requirements:

$$\varepsilon = 0.01 \leq \sqrt{\frac{\log 10,000 + \log\frac{2}{e^{-7}}}{2m}}$$
$$\implies m \geq \frac{2}{17} \cdot 10^4 \approx 1200.$$

Thus, if you have more than around 1200 examples in your sample data set, you can be very confident that all of the hypotheses in your class will perform similarly on the sample data set and on the source distribution. In this situation, if you happen to find an hypothesis that has low sample average loss on those 1200 points, you can be confident that the expected loss on the source distribution will differ by no more than $\varepsilon = 0.01$.

6.6.1 What does the union bound not tell us? (The union bound says nothing about whether the hypothesis class actually contains any hypotheses that have low expected loss.)

Note that this says nothing doesn't say anything (a priori) about whether a particular hypothesis class will actually *contain* any hypotheses that will have low expected loss on the source distribution. For this reason, this bound is sometimes referred to as a post-hoc guarantee. Only *after* you manage to find an hypothesis that has low sample average loss does it become a guarantee on the "goodness" of the expected loss of that hypothesis; specifically, it says that you can be "confident" that the expected loss will not differ from that "low" sample average loss by more than the specified ε (as long as the sample size is sufficiently large). The bound does not give any foreknowledge of good performance, but it provides its post-hoc guarantee with very few assumptions about the source distribution. (The $|\mathcal{H}|$ in this bound means that you also have to know your hypothesis class, but that is not much of a limitation since you should presumably know your hypothesis class in any case.)

6.6.2 Reiteration of the post-hoc nature of the union bound

Let's consider this just a little more. Again, this bound has very few assumptions, and is in a sense fairly practical. Practical how? Suppose have the same set up as in the example above: $\varepsilon = 0.01$, $\delta_{bad} = e^{-7}$, and$|\mathcal{H}| = 10,000$. If you run your learning algorithm on a sample of size $m = 1400$ and you happen to find a predictor in $\mathcal{H}$ with sample average loss of 0.01, you now can tell your boss something very specific. You can go back to your boss and tell him "Look, I know that with probability 0.999 ($= 1 - e^{-7}$), this predictor that got a sample average loss of 0.01 will not have an expected loss on the source distribution of more than 0.02 ($= 0.01 + \varepsilon$)." At this point, if you go and

ship a product containing this predictor, I can guarantee you with probability 0.999 that its expected loss is going to also be small.

This is certainly a nice statement to be able to make, but again you can only say this after you have actually found an hypothesis with low sample average loss. It is a post-hoc guarantee. However, we will now see how we can transform this post-hoc guarantee into a *(relative) learning* guarantee. This relative learning guarantee will tell us in what setting we will actually be able to do the learning. The steps in establishing the relative learning result will essentially match those of the flawed argument justifying ERM. However, where we previously ran into trouble because we were only using Hoeffding's inequality, this time we will use the union bound and so will end up with a correct result.

A *correct* argument for justifying ERM (using the union bound):
Via the union bound, we can say that with probability at least $1-\delta_{bad}$, *none* of the hypotheses in our class will have "misleading" performance on the sample average data set; all of the hypotheses will have sample average loss "near" their expected loss on the source distribution. Stated explicitly, we have:
With probability at least $1-\delta_{bad}$, we will have gotten a sample set realization s where

$$\forall h_i \in \mathcal{H},\ \left|\hat{R}_{s,01}\left(h_i\right) - R_{01}\left(h_i\right)\right| \leq \sqrt{\frac{\log|\mathcal{H}| + \log\frac{2}{\delta_{bad}}}{2m}}.$$

Now, since with "high probability" we are in the case where *all* hypotheses in the hypothesis class display performance on the observed sample data set s that is "representative", we know that the Empirical Risk Minimizing hypothesis $\hat{h}$ (*in particular*) displays "representative" performance on the sample data set. That is, for the hypothesis $\hat{h}$ that minimizes the empirical 01 risk $\hat{R}_{s,01}(\cdot)$, defined as $\hat{h} \triangleq \underset{h\in\mathcal{H}}{\operatorname{argmin}} \hat{R}_s(h)$, we apply the union bound. This would be expressed as: For the empirical 01 risk minimizing hypothesis $\hat{h}$, we have, with probability at least $1-\delta_{bad}$, we will have gotten a sample set realization s for which we have the following bound on the absolute difference between $R_{01}\left(\hat{h}\right)$ (the *expected* 01 risk of the empirical 01 risk minimizer $\hat{h}$) and $\hat{R}_{s,01}\left(\hat{h}\right)$ (the *empirical* 01 risk of the empirical 01 risk minimizer $\hat{h}$):

$$\left|R_{01}\left(\hat{h}\right) - \hat{R}_{s,01}\left(\hat{h}\right)\right| < \sqrt{\frac{\log|\mathcal{H}| + \log\frac{2}{\delta_{bad}}}{2m}}.$$

Which implies

$$(1)\ R_{01}\left(\hat{h}\right) < \hat{R}_{s,01}\left(\hat{h}\right) + \sqrt{\frac{\log|\mathcal{H}| + \log\frac{2}{\delta_{bad}}}{2m}}.$$

Likewise, for the hypothesis in our hypothesis class that minimizes the *expected* 01 risk $h^\star$, defined as $h^\star \triangleq \underset{h\in\mathcal{H}}{\operatorname{argmin}} R_{01}(h)$, (the minimizer of $R_{01}(\cdot)$, which is the expected 01 risk over the true source distribution), we can also apply the union bound. This would be expressed as: For the expected 01 risk minimizing hypothesis $h^\star$, we have, with probability at least $1-\delta_{bad}$, we will have gotten a sample set realization s where for which we will have the following bound on the absolute difference between $R_{01}(h^\star)$ (the *expected* 01 risk of the expected 01 risk minimizer $h^\star$) and $\hat{R}_{\mathcal{S},01}(h^\star)$ (the *empirical* 01 risk the expected 01 risk minimizer $h^\star$):

$$\left|R_{01}\left(h^\star\right) - \hat{R}_{s,01}\left(h^\star\right)\right| < \sqrt{\frac{\log|\mathcal{H}| + \log\frac{2}{\delta_{bad}}}{2m}}.$$

Which implies

$$(2)\ \hat{R}_{s,01}\left(h^\star\right) < R_{01}\left(h^\star\right) + \sqrt{\frac{\log|\mathcal{H}| + \log\frac{2}{\delta_{bad}}}{2m}}$$

We also observe, from the definition of the empirical 01 risk minimizer $\hat{h}$, (recalling: $\hat{h} \triangleq \underset{h\in\mathcal{H}}{\operatorname{argmin}} \hat{R}_{\mathcal{S}}(h)$), that we know the empirical 01 risk minimizer $\hat{h}$ has lower (or equal) empirical 01 risk than all other hypotheses, and in particular, the expected 01 risk minimizer $h^\star$:

$$(3)\ \hat{R}_{s,01}\left(\hat{h}\right) \leq \hat{R}_{s,01}\left(h^\star\right).$$

We can now use the (correct) reasoning established in the box above to establish a *(relative) learning* guarantee: With probability at least $1 - \delta_{bad}$, we will have gotten a sample set realization s where

$$\begin{aligned} R_{01}\left(\hat{h}\right) &\overset{(1)}{\leq} \hat{R}_{s,01}\left(\hat{h}\right) + \sqrt{\frac{\log|\mathcal{H}| + \log\frac{2}{\delta_{bad}}}{2m}} \\ &\overset{(3)}{\leq} \hat{R}_{s,01}\left(h^\star\right) + \sqrt{\frac{\log|\mathcal{H}| + \log\frac{2}{\delta_{bad}}}{2m}} \\ &\overset{(2)}{\leq} R_{01}\left(h^\star\right) + 2\sqrt{\frac{\log|\mathcal{H}| + \log\frac{2}{\delta_{bad}}}{2m}} \end{aligned}$$

6.7 Post-hoc guarantees and (relative) learning guarantees

From the above, we note two particular guarantees: a post-hoc learning guarantee and a relative learning guarantee. The post-hoc guarantee was: With probability at least $1 - \delta_{bad}$, we will have gotten a sample set realization s for which

$$R_{01}\left(\hat{h}\right) \leq \hat{R}_{s,01}\left(\hat{h}\right) + \sqrt{\frac{\log|\mathcal{H}| + \log\frac{2}{\delta_{bad}}}{2m}}.$$

We also arrived at a relative learning guarantee: With probability at least $1 - \delta_{bad}$, we will have

$$R_{01}\left(\hat{h}\right) \leq R_{01}\left(h^\star\right) + 2\sqrt{\frac{\log|\mathcal{H}| + \log\frac{2}{\delta_{bad}}}{2m}}.$$

(Note that there is no mention of a sample set realization). We discussed the post-hoc guarantee above; now we will consider the relative learning guarantee. What does the relative learning guarantee say? We can interpret the relative learning guarantee as saying: ERM will give you an hypothesis that has expected 01 loss that is almost as good as the hypothesis will best expected 01 loss. In terms of the chances for finding a predictor with good expected 01 loss on the source distribution, we can say that if there *is* a predictor in the hypothesis class $\mathcal{H}$ that has low expected 01 loss, then using Empirical Risk Minimization will give me a predictor $\hat{h}$ that is "almost as good" in terms of $\hat{h}$s expected 01 loss on the source distribution.

The problem with this relative learning guarantee is that it isn't very tangible. It tells you that the ERM principle will yield a predictor that does almost as well (in terms of performance on the source distribution) as the predictor in the hypothesis class with *best* performance on the source distribution. However, there is no guarantee that that "best" performance possible in your hypothesis class is actually any good in absolute terms. You simply don't know whether the expected-risk-minimizing hypothesis $h^\star$ actually achieves expected 01 risk $R_{01}\left(h^\star\right)$ that is actually "low enough" to be called good. However, if there *is* a good predictor in the class, the relative learning guarantee says that ERM will yield a predictor that is competitive with that good predictor in terms of performance on the source distribution.

These are the two types of guarantees we can look at: relative learning and post-hoc Note that in their present form, both of these guarantees depend on the size of the hypothesis class. This means that the present form of the bound is only meaningful for finite hypothesis classes; an infinite hypothesis class would result in an infinite bound. We will soon see how we can establish guarantees that will also hold for infinite hypothesis classes.

Post-hoc generalization guarantee.
You can think of the post-hoc generalization guarantee as the "after you have found something that you like (e.g. by using ERM), how well will you do on the source distribution?" probabilistic guarantee. Restating this question, we ask: If you find an hypothesis that does well on your sample data set, such as: if your Empirical Risk Minimizing hypothesis $\hat{h}$ has "pleasing" empirical 01 risk $\hat{R}_{s,01}\left(\hat{h}\right)$ on your sample data set (*i.e.* the empirical 01 risk minimizer $\hat{h}$ achieves empirical 01 risk $\hat{R}_{s,01}\left(\hat{h}\right)$ that is "low enough"), how will $\hat{h}$ do on the source distribution; what will $R_{01}\left(\hat{h}\right)$ be like? The bound below address this question.
With probability at least $1-\delta_{bad}$, we will have a sample set realization s for which the *post-hoc generalization guarantee* holds:

$$R_{01}\left(\hat{h}\right) \leq \hat{R}_{s,01}\left(\hat{h}\right) + \sqrt{\frac{\log|\mathcal{H}| + \log\frac{2}{\delta_{bad}}}{2m}}.$$

(Relative) learning guarantee.
You can think of the (relative) learning guarantee as the "If there is a good predictor in my chosen hypothesis class (in particular, if the expected 01 risk minimizer $h^\star$ achieves expected 01 risk $R_{01}\left(h^\star\right)$ on your source distribution that is "pleasing"), how well will the expected 01 risk of your empirical 01 risk minimizing hypothesis compare?" probabilistic guarantee. Or: How well will your Empirical Risk Minimizing hypothesis do *relative* to the best hypothesis in your class, in terms of expected 01 loss on the source distribution? The bound below address this question.
With probability at least $1-\delta_{bad}$, we have the *(relative) learning guarantee*

$$R_{01}\left(\hat{h}\right) \leq R_{01}\left(h^\star\right) + 2\sqrt{\frac{\log|\mathcal{H}| + \log\frac{2}{\delta_{bad}}}{2m}}.$$

References

[1] G.V. Cormack. Email spam filtering: A systematic review. *Foundations and Trends in Information Retrieval*, 1(4):335–455, 2007.

[2] J. Goodman, G.V. Cormack, and D. Heckerman. Spam and the ongoing battle for the inbox. *Communications of the ACM*, 50(2):24–33, 2007.

[3] A. Halevy, P. Norvig, and F. Pereira. The unreasonable effectiveness of data. *Intelligent Systems, IEEE*, 24(2):8–12, 2009.

[4] A. Hertzmann. Machine learning for computer graphics: A manifesto and tutorial. In *Computer Graphics and Applications, 2003. Proceedings. 11th Pacific Conference on*, pages 22–36. IEEE.

[5] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, pages 13–30, 1963.

[6] F. Jelinek. Some of my best friends are linguists. *Language resources and evaluation*, 39(1):25–34, 2005.

[7] T.Y. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.

[8] A. Saxena, S.H. Chung, and A.Y. Ng. Learning depth from single monocular images. In *In NIPS 18*, 2005.

[9] V.N. Vapnik. *The nature of statistical learning theory*. Springer Verlag, 2000.

[10] P. Viola and M.J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[11] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2001*, pages 511–518, 2001.

[12] X. Wang, R. Hutchinson, and T.M. Mitchell. Training fmri classifiers to discriminate cognitive states across multiple subjects. *Advances in neural information processing systems*, 16:709–716, 2004.

[13] D.H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7):1341–1390, 1996.