Nati Srebro at UT Austin 2011-05-31 Lecture 2 Notes Draft

Patrick W. Gallagher

August 31, 2011

Abstract

More on the post-hoc generalization guarantee for ERM. δ versus δ_{bad}. (Relative) learning guarantee. Approximation error. ERM Estimation error. Infinite hypothesis class size: unsatisfying argument via finite precision arithmetic. Why this argument is unsatisfying. Sample complexity result. Aside on tightness of guarantees. The trade-off estimation error and approximation error or between simplicity and complexity. Example hypothesis class complexity hierarchies. Parameter counts for example hypothesis class hierarchies. The error-behavior-versus-hypothesis-class-complexity graph. Structural risk minimization (SRM). Problems with SRM in practice. An alternative to SRM: train then validate. Graph of validation error. Validation set must be independent of training set. Additional data hold-out for final evaluation. Another alternative: cross-validation. Cross-validation is not statistically sound. Summary: Learning is optimization.

1 More on the post-hoc generalization guarantee for ERM

To reiterate: we can think of the post-hoc generalization guarantee as the "after you have found a predictor that you like on the sample data set (e.g. by using ERM), how well can you expect that predictor to perform on the source distribution?" probabilistic guarantee. For example, if the empirical risk minimizer $\hat{h}$ achieves empirical 01 risk $\hat{R}_{s,01}\left(\hat{h}\right)$ that is "low", how low can we expect the source distribution expected 01 risk $R_{01}\left(\hat{h}\right)$ to be?

From our derivation involving the union bound, the guarantee on post-hoc generalization was: With probability at least $1-\delta_{bad}$, we will have a sample set realization s for which

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

1.1 A note on δ (how often the deviation bound fails for a particular hypothesis) versus δ_{bad} (what is the probability that there is *some* hypothesis in our hypothesis class for which the deviation bound fails) from the previous lecture

A few comments about the deviation probabilities in our deviation bounds, δ and δ_{bad} : By comparing the details of Hoeffding's inequality to the details of the statement derived by using the union bound (the result of which we also referred to as the "union bound"), we observe that these probabilities are related by $\delta_{bad} = |\mathcal{H}|\,\delta$ (strictly, $\delta_{bad} \leq |\mathcal{H}|\,\delta$). This comes directly from the initial union bound: the probability δ_{bad} of at least one of $A_1, \ldots, A_{|\mathcal{H}|}$ occurring[1] is no greater than the sum of the probabilities δ of each of A_i occurring individually.

[1] Recall that A_i refers to the event $\left|\hat{R}_{\mathcal{S},01}\left(h_i\right) - R_{01}\left(h_i\right)\right| \geq \varepsilon$, where the ith hypothesis in $\mathcal{H}$ has sample set performance more than ε different from the source distribution performance.

Since δ_{bad} corresponds to the probability that we really need to be concerned with, from here on we will refer to δ_{bad} simply as δ. (If we ever subsequently need to refer refer to the δ in Hoeffding's inequality, we will do so as δ_{hoeff}). Thus, going forward we will express the guarantee on post-hoc generalization as:

With probability at least $1-\delta$, we will have a sample set realization s for which

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

And the relative learning guarantee as: With probability at least $1-\delta$, it will be the case that

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

1.2 A closer look at the guarantee on post-hoc generalization and at the (relative) learning guarantee

Having set down that change in variable naming, let's now take a closer look at the guarantee on post-hoc generalization just rewritten above. Of particular note is the dependence on the hypothesis class $\mathcal{H}$ (currently $\mathcal{H}$ must be finite but shortly we will see how to consider infinite hypothesis classes). This dependence of the guarantee on post-hoc generalization on the hypothesis class follows directly from the fact that we are following the Empirical Risk Minimization principle to select an hypothesis *from a specified hypothesis class*. After all, the Empirical Risk Minimizer is really the "Empirical Risk Minimizer out of the hypotheses in a specific hypothesis class $\mathcal{H}$":

$$\mathbf{ERM}_{\mathcal{H}}(s) = \hat{h}_{\mathcal{H}} \triangleq \underset{h\in\mathcal{H}}{\operatorname{argmin}}\ \hat{R}_{s,01}(h)$$

If we wanted to be even more emphatic about all of the dependencies involved, we could write $\hat{h}_{\mathcal{H},s,01}$: meaning the hypothesis that, for the sample set realization s, minimizes the empirical 01 risk when compared to all other hypotheses in class $\mathcal{H}$.

Throughout our consideration of minimizing the empirical 01 risk, we need to keep in mind that our real goal is to find an hypothesis that achieves good *expected* 01 risk. Abstractly, we could think of "plucking" an hypothesis that we hope will display good performance on the source distribution from any imaginable possible hypotheses; the term "plucking" is meant to convey that (in the abstract) if we can think up an hypothesis, we can consider using it. Thinking completely abstractly, there is not any overriding reason to limit the hypotheses we will consider, for instance by making sure that the hypothesis classes are not "arbitrarily flexible".

However, once we are presented with the problem of selecting an hypothesis (that we hope will perform well on future data) based on a particular sample set realization s, we *cannot* view things so abstractly if we want (probabilistic) guarantees like those above. We are essentially *forced* to consider limits, not only in the sense that we will choose from some specific collection of hypothesis classes, but *also* in the sense that those individual hypothesis classes must not be "too flexible" (relative to the sample set size). Much of our attention in what follows will be directed to exploring why these limits are necessary, and what explicit forms the limits will need to take. For example, in the ERM guarantees above, the explicit limitation is: if we seek to be able to use the sample set performance as a good guide for the source distribution performance, we must have an hypothesis class $\mathcal{H}$ that is finite, and the number of hypotheses that $\mathcal{H}$ contains should not be "too large" relative to the sample set size m.[2]

[2]If the the number of hypotheses that $\mathcal{H}$ contains is not "too large" relative to the sample set size m, then with probability $1-\delta$ we will have a realized sample set s for which the sample set performance is a good guide for the source distribution performance, where "good" means the difference is not more than $\sqrt{\frac{\log|\mathcal{H}|+\log\frac{2}{\delta}}{2m}}$.

1.2.1 The approximation error (depends on the chosen hypothesis class $\mathcal{H}$ and on the problem source distribution $p_{X,Y}(x,y)$)

Thus, in Empirical Risk Minimization we have to limit ourselves to a specific hypothesis class $\mathcal{H}$ (and to impose limits on the class itself) in order to get a (probabilistic) post-hoc guarantee on the difference of the expected 01 risk $R_{01}\left(\hat{h}_{\mathcal{H}}\right)$ from the empirical 01 risk $\hat{R}_{s,01}\left(\hat{h}_{\mathcal{H}}\right)$ of the empirical risk minimizer in $\mathcal{H}: \hat{h}_{\mathcal{H}}$. The same limitation is necessary in the relative learning guarantee.

Side comment:
The guarantees we are considering differ in their "tangibility".
The post-hoc guarantee is *tangible* in that it provides us with a probabilistic bound on the relation between something we can *measure* (the sample set performance $\hat{R}_{s,01}\left(\hat{h}_{\mathcal{H}}\right)$ of the empirical risk minimizer $\hat{h}_{\mathcal{H}}$) and something we care about but cannot measure (the expected 01 loss $R_{01}\left(\hat{h}_{\mathcal{H}}\right)$ of the empirical risk minimizer $\hat{h}_{\mathcal{H}}$).
The relative learning guarantee is *intangible* in that it provides us with a probabilistic bound on the relation between two things that we care about but cannot generally measure: the source distribution performance $R_{01}\left(\hat{h}_{\mathcal{H}}\right)$ of the empirical risk minimizer $\hat{h}_{\mathcal{H}}$ and the source distribution performance $R_{01}\left(h^{\star}_{\mathcal{H}}\right)$ of the expected 01 loss minimizer $h^{\star}_{\mathcal{H}}$.

Looking specifically at the relative learning guarantee (the probabilistic bound on how much the expected 01 risk $R_{01}\left(\hat{h}_{\mathcal{H}}\right)$ of the empirical 01 risk minimizer $\hat{h}_{\mathcal{H}}$ might differ from the expected 01 risk $R_{01}\left(h^{\star}_{\mathcal{H}}\right)$ of the expected 01 risk minimizer $h^{\star}_{\mathcal{H}}$) we will now more closely examine the dependency of this bound on the hypothesis class $\mathcal{H}$. In particular, we note that the probabilistic bound on how large the expected 01 loss of the empirical 01 risk minimizing hypothesis $\hat{h}_{\mathcal{H}}$ is likely to be has two components: that is, $R_{01}\left(\hat{h}_{\mathcal{H}}\right)$ is (with high probability) less than the sum of $R_{01}\left(h^{\star}_{\mathcal{H}}\right)$ and $2\sqrt{\frac{\log|\mathcal{H}|+\log\frac{2}{\delta}}{2m}}$. We will now consider each of these two components in turn. The component $R_{01}\left(h^{\star}_{\mathcal{H}}\right)$ represents the best expected 01 loss possible when we limit ourselves to the hypothesis class $\mathcal{H}$, and is called the *approximation error* (induced by limiting ourselves to $\mathcal{H}$)[3], and can be thought of as an indication of how much we have lost by restricting ourselves to $\mathcal{H}$. The specific value that the "approximation error" $R_{01}\left(h^{\star}_{\mathcal{H}}\right)$ takes on is determined both by the hypothesis class $\mathcal{H}$ that we are restricting to, and also by the source distribution itself.

[3]Note that when we call $R_{01}\left(h^{\star}_{\mathcal{H}}\right)$ the *approximation error*, we are in some sense implicitly assuming that if we knew the "true" hypothesis h^{true}, we would be able to perfectly predict the correct label for every example in the source distribution. That is, saying that the expected risk $R_{01}\left(h^{\star}_{\mathcal{H}}\right)$ corresponds exactly to error-from-approximation implicitly implies that there is no "noise" in the labels-against-which-correctness-is-to-be-judged, so that we would have $R_{01}\left(h^{true}\right)=0$. If the "true" hypothesis h^{true} does not achieve 0 expected loss on the source distribution, then part of $R_{01}\left(h^{\star}_{\mathcal{H}}\right)$ is unavoidable, and the remaining "excess" part of $R_{01}\left(h^{\star}_{\mathcal{H}}\right)$ is due to "approximation error" that comes from restricting ourselves to considering only hypotheses from the hypothesis class $\mathcal{H}$. Explicitly, when $R_{01}\left(h^{true}\right)\neq 0$, we could emphasize this by writing

$$\begin{aligned} R_{01}\left(h^{\star}_{\mathcal{H}}\right) &= \text{"approximation error"} + R_{01}\left(h^{true}\right) \\ &= \text{"excess-error-induced-by-restricting-to-}\mathcal{H}\text{"} + \text{"fundamental-and-unavoidable-error-from-label-source-noise"}. \end{aligned}$$

Despite this, even in situations where there is unavoidable label noise and $R_{01}\left(h^{true}\right)=0$, by tradition the term "approximation error" is still used to refer to the best possible performance in when restricting to $\mathcal{H}$, $R_{01}\left(h^{\star}_{\mathcal{H}}\right)$.

Approximation error term.
The "approximation error" term, depending on the chosen hypothesis class $\mathcal{H}$ and the problem source distribution $p_{X,Y}$, is defined as the expected risk (where we can use any risk, and not just the 01 risk) on the source distribution of the hypothesis that minimizes the expected risk:

$$\text{Approximation-Error}\left(\mathcal{H}, p_{X,Y}\right) \triangleq R\left(h^{\star}_{\mathcal{H}}\right),$$
$$\text{where } h^{\star}_{\mathcal{H}} \triangleq \underset{h \in \mathcal{H}}{\operatorname{argmin}}\ R\left(h\right).$$

Recall that the source distribution $p_{X,Y}$ comes in when we are considering the expected risks $R\left(h\right)$ or $R\left(h^{\star}_{\mathcal{H}}\right)$. We could have written $R_{p_{X,Y}}\left(h\right)$ and $R_{p_{X,Y}}\left(h^{\star}_{\mathcal{H}}\right)$ to emphasize this.

1.2.2 ERM estimation error (of the empirical risk minimizer relative to the expected risk minimizer) and the *bound* on ERM estimation error

The second component of the probabilistic bound on $R_{01}\left(\hat{h}_{\mathcal{H}}\right)$ is the term $2\sqrt{\frac{\log|\mathcal{H}|+\log\frac{2}{\delta}}{2m}}$, which also has a dependence on the hypothesis class $\mathcal{H}$ (as well as on the probability δ that the bound will be violated and the sample size m). This part of the bound in the relative learning guarantee is a probabilistic bound on how much $R_{01}\left(\hat{h}_{\mathcal{H}}\right)$ and $R_{01}\left(h^{\star}_{\mathcal{H}}\right)$ will differ; it provides a bound on the *ERM estimation error*. Thus, it indicates how much *might* be lost by attempting to find the best (in terms of expected 01 loss) hypothesis in $\mathcal{H}$, $h^{\star}_{\mathcal{H}}$, when only getting to use a sample set realization s to do so. Again, the use of the term "estimation error" is due to the fact that we are not actually minimizing the expected error; we are only minimizing an *estimate* of the expected error.

ERM estimation error.
The **ERM** estimation error, depending on the chosen hypothesis class $\mathcal{H}$, the problem source distribution $p_{X,Y}$, and the sample set realization s is defined as the difference between the best-in-class expected risk $R\left(h^{\star}_{\mathcal{H}}\right)$ (of the expected risk minimizer $h^{\star}_{\mathcal{H}}$ in $\mathcal{H}$) and the estimated-best-in-class expected risk $R\left(\hat{h}_{s,\mathcal{H}}\right)$ (of the empirical risk minimizer $\hat{h}_{s,\mathcal{H}}$[a] from $\mathcal{H}$):

$$\text{ERM-Estimation-Error}\left(\mathcal{H}, p_{X,Y}, s\right) \triangleq R\left(h^{\star}_{\mathcal{H}}\right) - R\left(\hat{h}_{s,\mathcal{H}}\right),$$
$$\text{where } h^{\star}_{\mathcal{H}} \triangleq \underset{h \in \mathcal{H}}{\operatorname{argmin}}\ R\left(h\right)$$
$$\text{and } \hat{h}_{s,\mathcal{H}} \triangleq \underset{h \in \mathcal{H}}{\operatorname{argmin}}\ \hat{R}_{s}\left(h\right)$$

Recall that the source distribution $p_{X,Y}$ comes in when we are evaluating the expected risks $R\left(h\right)$ or $R\left(h^{\star}_{\mathcal{H}}\right)$. We could have written $R_{p_{X,Y}}\left(h\right)$, $R_{p_{X,Y}}\left(\hat{h}_{\mathcal{H}}\right)$, and $R_{p_{X,Y}}\left(h^{\star}_{\mathcal{H}}\right)$ to emphasize this.

[a] We are writing $\hat{h}_{s,\mathcal{H}}$ to emphasize something that is true by definition for ERM, and so typically not explicitly indicated. Namely, ERM selects the hypothesis from $\mathcal{H}$ that gives the best sample average error on the provided sample set realization s. If we had observed a different sample set, say s', we would find that typically a different hypothesis would be selected by ERM, i.e. typically $\hat{h}_{s,\mathcal{H}} \neq \hat{h}_{s',\mathcal{H}}$ for $s \neq s'$. The hypothesis that ERM selects thus always depends on the particular sample set, and we are temporarily being very explicit about this dependence.

Bound on ERM estimation error (a term on the RHS of the relative learning guarantee).
The probabilistic bound on the ERM estimation error, depending on the probability δ that the bound will be violated, the (cardinality of) the chosen hypothesis class $\mathcal{H}$, and the sample size m, is given by:

$$\text{Bound-on-ERM-Estimation-Error}\left(\delta, \mathcal{H}, m\right) \triangleq 2\sqrt{\frac{\log|\mathcal{H}| + \log\frac{2}{\delta}}{2m}}.$$

This is just the right-hand-side of the relative learning guarantee bound (with a different description to emphasize our view of it here). As before, it just says that with probability $1-\delta$, we will have gotten a sample set realization s of size m for which

$$R\left(h^{\star}_{\mathcal{H}}\right) - R\left(\hat{h}_{s,\mathcal{H}}\right) \leq 2\sqrt{\frac{\log|\mathcal{H}| + \log\frac{2}{\delta}}{2m}}.$$

Note in particular that if δ and $\mathcal{H}$ are fixed, increasing the sample size m to infinity will result in a bound of 0, so that we would then have $0 \leq R\left(h^{\star}_{\mathcal{H}}\right) - R\left(\hat{h}_{s,\mathcal{H}}\right) \leq 0$, implying $R\left(h^{\star}_{\mathcal{H}}\right) = R\left(\hat{h}_{s,\mathcal{H}}\right)$.
We can interpret the sample size dependence as saying "the larger the sample size m, the better the estimate we can expect to have". We can interpret the the dependence on hypothesis class size as saying "the larger the hypothesis class size $|\mathcal{H}|$, the harder it is to estimate *uniformly* over all predictors in the class, and the larger the ERM estimation error could possibly be".

To be sure that Empirical Risk Minimization was guaranteed to select the expected-01-loss-minimizing hypothesis in $\mathcal{H}$, we would need to see every example in the source distribution (typically an uncountably infinite number of examples); using only a finite sample data set s will lead to some lost performance relative to the expected-01-loss-minimizing hypothesis in $\mathcal{H}$. The "infinite sample" empirical 01 risk minimizer $\hat{h}_{\mathcal{H}}$ calculated for a data sample made up of the entire source population would be the same as the best-in-class predictor $h^{\star}_{\mathcal{H}}$ and thus the expected 01 loss of the "infinite sample" empirical 01 risk minimizer would be the same as the best-in-class expected 01 loss $R_{01}\left(h^{\star}_{\mathcal{H}}\right)$.

2 Reconsidering hypothesis class size (from finite to infinite)

As we have highlighted before, and will highlight again in the future, our abstract goal in learning is to find a predictor with good performance on future examples from the source distribution. Stated in this way, there is no restriction to a particular hypothesis class involved; if you were to just hand me a (demonstrably) good predictor, that would have been fine. Once we begin to talk about deciding on a particular *practical* procedure for choosing a predictor with good performance on the source distribution, the state of affairs changes. Consideration of a particular hypothesis class quickly comes up in nearly all practical approaches, and it certainly comes up for the particular case of Empirical Risk Minimization. Once we talk about Empirical Risk Minimization, it is always Empirical Risk Minimization by an hypothesis from a specified hypothesis class.

When we consider Empirical Risk Minimization, we now observe that we cannot simply choose *any* hypothesis class to search over (at least if we seek to be confident in the future performance of the ERM-selected hypothesis $\hat{h}_{\mathcal{H}}$). In particular, in order to have the (absolute value form of the) post-hoc guarantee that with probability at least $1-\delta$ we will have a sample set realization s for which $\left|\hat{R}_{s,01}\left(\hat{h}_{\mathcal{H}}\right) - R_{01}\left(\hat{h}_{H}\right)\right| \leq \sqrt{\frac{\log|\mathcal{H}|+\log\frac{2}{\delta}}{2m}}$ (that is, the absolute difference between the empirical risk of the empirical risk minimizer and the expected risk of the empirical risk minimizer is bounded by $\sqrt{\frac{\log|\mathcal{H}|+\log\frac{2}{\delta}}{2m}}$), the form of the bounding term tells us that we should not have an

hypothesis class that is "too large" relative to our sample data set size m.

If we failed make certain that $\log|\mathcal{H}|$ was not too large with respect to $2m$, then our bound would be quite large, and thus we would not be able to expect the empirical risk of $\hat{h}_{\mathcal{H}}$ would be indicative of the expected risk of $\hat{h}_{\mathcal{H}}$: the bound would tell us that they *could* be very different. We will begin to talk about the dependence between the bounding term $\sqrt{\frac{\log|\mathcal{H}|+\log\frac{2}{\delta}}{2m}}$ and the specified hypothesis class $\mathcal{H}$ by seeing how we might approach a case that seemingly completely defies our ability to produce a bound: specifically, hypothesis classes with an infinite number of hypotheses. Our initial approach in this lecture will be essentially correct and conceptually simple, but somewhat kludgy and perhaps unsatisfying; we will pursue cleaner, more satisfying approaches in subsequent lectures.

2.1 Infinite hypothesis classes

While we will now move on to the case of infinite hypothesis classes, there are perfectly reasonable hypothesis classes that are finite; finiteness does not automatically imply poor performance. The most prominent finite hypothesis classes are probably decisions trees (also called classification trees[1]) and decision lists. While the machine learning research community does not currently devote much attention to decision trees on their own, they remain very popular in practical data mining. While we will not go into any details of these methods here, the hypothesis class corresponding to any specific classification tree setting turns out to encompass a finite number of hypotheses. Thus, for these classifiers the count of all trees gives a good measure of the complexity of the hypothesis class, and can be immediately used as $|\mathcal{H}|$ in the bound terms above.

Notwithstanding the existence of reasonable hypothesis classes that have finite cardinality, there are many hypothesis classes of interest that are not finite. The specific infinite hypothesis classes that we will be most interested are oriented affine separator classes (also frequently referred to as "linear" separators). These classes take the input domain to be $\mathcal{X} = \mathbb{R}^d$,[4] and the hypothesis class $\mathcal{H}$ contains mappings from the input domain$\mathcal{X}$ to the label set $\mathcal{Y} \in \{-1, +1\}$ by taking the sign of an affine function of $x \in \mathcal{X}$:

$$\mathcal{H} = \left\{ x \mapsto \text{sign}\left(w^T x + b\right) \mid w \in \mathbb{R}^d,\ b \in \mathbb{R} \right\}.$$

In this setting, we represent each hypothesis (or predictor) in $\mathcal{H}$ as a particular pair (w, b).

In this hypothesis class, each hypothesis considers all input vectors in some half-space (specified by (w, b)) to have label "+1" and all the input vectors in the complementary half-space to have label "−1". What is the cardinality of this hypothesis class? Even if take $w \in \mathbb{R}^1$ and $b = 0$, there are still an infinite number of hypotheses in this class. Even if we go even further and take $w \in [0, 1]$ and $b = 0$, there are *still* an infinite number of possible values for w, and thus an infinite number of possible hypotheses.

Since even for this simple hypothesis class (of oriented affine separators) there are an infinite number of hypotheses in the class, the bounds developed (from Hoeffding's inequality and the union bound) above do not apply. We will consider several possible approaches to obtaining performance guarantees even in the infinite hypothesis class case; most of them can be described informally as "Actually, if we look at things correctly, the class doesn't *really* have an infinite number of distinct behaviors". Our first approach will be based on an argument from finite-precision arithmetic. This will be fairly straightforward, and will give the right idea; however, it will be fundamentally unsatisfactory. The next approach will be based on the notion of Vapnik-Chervonenkis dimension; we will postpone the details for now.

[4] We should perhaps view $\mathbb{R}^d$ as the domain of the *mapping* of the actual real-world input object to a representation that the computer can work with, but for convenience we will write x rather than $\phi(x)$.

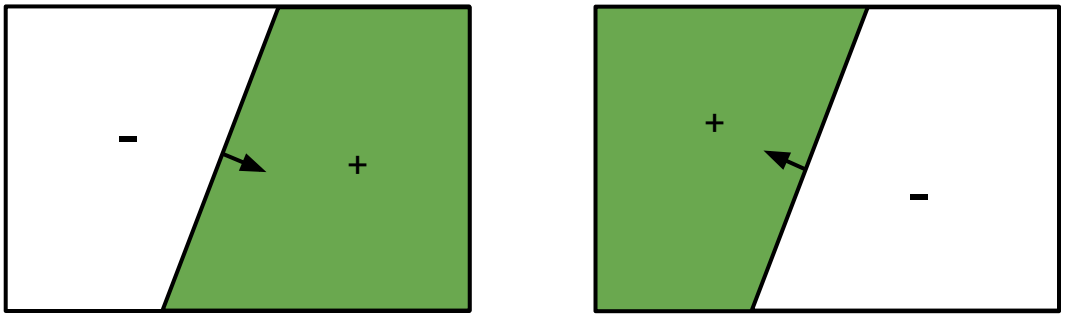

Figure 2.1: Example oriented affine separators in $\mathbb{R}^2$.

2.2 Our first way to deal with infinite hypothesis class size: The argument from finite-precision computer arithmetic

We now begin with the first approach to deal with infinite hypothesis class size. To being, let us take a closer look at the hypothesis class of oriented affine separators in $\mathbb{R}^2$. When viewed as an abstract mathematical object this is an infinite hypothesis class. The hypothesis class contains an uncountably infinite number of hypotheses, corresponding to each possible value for (w, b). Although as a mathematical object the hypothesis class contains an uncountably infinite number of hypotheses, when you actually use the class, you do with with a computer. Since computers use finite-precision arithmetic, we will thus in practice have a (very large but) finite number of possible values for (w, b). To be more explicit, we note that each hypothesis corresponds to a pair (w, b). Since $w \in \mathbb{R}^d$ and $b \in \mathbb{R}^1$, each hypothesis is thus parametrized by $d + 1$ parameters, and each parameter is a real number. For convenience we will refer to the total parameter count as $D = d + 1$. Each of these D parameters is a real number, which in the computer will be (almost certainly) represented according to the IEEE 754 standard for floating point arithmetic. Moreover, we will assume that each floating point number thus represented has 64 bits.

At this point, we observe that there are 2^{64} distinct 64-bit floating point numbers[5]. Since each of the D parameters can take on only 2^{64} distinct values, the actual number of distinct hypotheses in the class of oriented affine separators can be no more than $\left(2^{64}\right)^D$.[6] We can write this explicitly as

$$|\mathcal{H}| \leq \left(2^{64}\right)^D = 2^{D \cdot 64}.$$

We now have established that when representing these hypotheses in the computer we effectively only have a finite number of hypotheses, and so we can use the bound $2\sqrt{\frac{\log|\mathcal{H}| + \log\frac{2}{\delta}}{2m}}$ derived previously. In that bound, the important quantity is the log cardinality $\log|\mathcal{H}|$ of the class. Using the result $|\mathcal{H}| \leq 2^{D \cdot 64}$, we find (recalling that when we

[5] 2 possible values (*i.e.* 0 or 1) for the first bit, times 2 possible values for the second but, and so on.
[6] 2^{64} distinct values for the first parameter, 2^{64} distinct values for the second parameter, and so one.

write log we always mean the natural logarithm):

$$\ln |\mathcal{H}| \leq \ln \left(2^{64}\right)^{D} = \ln 2^{D \cdot 64} \approx 45 \cdot D.$$

This tells us that when we use floating point representations of the parameters, **the "effective" log cardinality of the hypothesis class is linear in the number of parameters**. Even though we set this discussion in the context of affine separators, the same argument applies to to any hypothesis class whose member hypotheses are represented by D floating point numbers. Thus, for hypotheses represented in the computer, in some sense we can always replace the log cardinality of the hypothesis class by a somewhat arbitrary constant (here 45, deriving from the use of 64-bit floating point numbers), times the number of parameters D.

Indeed, for any reasonable measure of the precision with which we can represent hypotheses classes (whether in terms of drawing a line on a chalkboard or storing the parameters according to the IEEE 754 standard for floating point arithmetic, or anything else "reasonable" that you can think of), we can make an argument that the log cardinality of the *represented-in-practice* (*e.g.* drawn as a line on a board) hypothesis class is linear in the number of parameters used to describe each hypothesis in the class. (Think of how many distinguishable lines can you draw on the chalk board; there will certainly be less than the equivalent of 64-bits of precision.)

In accordance with this observation that the log cardinality of the computer-represented hypothesis class is effectively linear in the number of parameters used to describe each hypothesis in the class, we can think about putting $45 \cdot D$ in place of $|\mathcal{H}|$ in the bound $2\sqrt{\frac{\log|\mathcal{H}|+\log\frac{2}{\delta}}{2m}}$. Thus, our rough accounting of the effective log cardinality would be interpreted as yielding the following relative learning guarantee: With probability at least $1-\delta$, it will be the case that

$$R_{01}\left(\hat{h}\right) \leq R_{01}\left(h^{\star}\right) + 2\sqrt{\frac{45 \cdot D + \log\frac{2}{\delta}}{2m}}.$$

[7]

The basic idea from this argument is that we can essentially replace $\log |\mathcal{H}|$ by $c \cdot D$, where c is some constant determined by the particular practical representation we decide to use.

[7] Among other things, we can determine a lower bound on number of samples required to yield a relative learning bound of no more than ε :

$$\begin{aligned} & 2\sqrt{\frac{45 \cdot D + \log\frac{2}{\delta}}{2m}} \leq \varepsilon \\ \Longrightarrow & \frac{45 \cdot D + \log\frac{2}{\delta}}{2m} \leq \left(\frac{\varepsilon}{2}\right)^2 \\ & \Longrightarrow m \geq \frac{2\left(45 \cdot D + \log\frac{2}{\delta}\right)}{\varepsilon^2}. \end{aligned}$$

Thus, with probability at least $1-\delta$, if we have used a sample set size m larger than $\frac{2\left(45 \cdot D + \log\frac{2}{\delta}\right)}{\varepsilon^2}$, it will be the case that $R_{01}\left(\hat{h}\right) - R_{01}\left(h^{\star}\right) \leq \varepsilon$. In a sense, despite the direction of the inequality (i.e. $m \geq$), this about is actually a *lower* bound. This comes because when we use $m \geq \frac{2\left(45 \cdot D + \log\frac{2}{\delta}\right)}{\varepsilon^2}$, we can *guarantee* (with probability $1-\delta$) that we will have $R_{01}\left(\hat{h}\right) - R_{01}\left(h^{\star}\right) \leq \varepsilon$; nothing says that m *has* to be larger than $\frac{2\left(45 \cdot D + \log\frac{2}{\delta}\right)}{\varepsilon^2}$ for us to have $R_{01}\left(\hat{h}\right) - R_{01}\left(h^{\star}\right) \leq \varepsilon$; it is certainly *possible* that empirical risk minimization on a smaller sample set size might have yielded an hypothesis for which $R_{01}\left(\hat{h}\right) - R_{01}\left(h^{\star}\right) \leq \varepsilon$.

2.3 The unsatisfying aspects of the argument from finite-precision arithmetic

While the argument from finite-precision arithmetic is not incorrect, it does seem to leave something to be desired. We will highlight two particular issues, one relatively minor and another that is fairly major.

2.3.1 Relatively minor: The constant (*e.g.* in $45 \cdot D$) is arbitrary

One somewhat unsatisfying aspect of the argument from finite-precision (or from practical representation in any form) is that we eventually will get an arbitrary constant in our bound. In the 64-bit representation case we ended up with the number 45 as the constant, but we certainly are unlikely to receive data that is measured with enough accuracy to require all 64 of those bits[8]. Even 32 bits is likely to be far more than the amount of accuracy in your measured data. While these sorts of considerations do not change the fact that the "qualitative" implications of $\log|\mathcal{H}| \leq c \cdot D$, are fairly correct, the arbitrariness of the constant c indicates that the quantitative statement of the bound is not very meaningful. If we plug in numbers for a subset of D, δ, m, or ε (when we have expressed one of these variables in terms of the others), the resulting numbers should be taken with a very large grain of salt. This observation is extremely minor when compared to the next fact that we will examine: as a measure of the effective size (or complexity or capacity) of an hypothesis class, the number of parameters D fails to capture some important behavior.

2.3.2 Relatively major: Important distinctions are not captured (as illustrated by the sign-of-sine hypothesis class)

We can get a better idea of what the statement: "as a measure of the effective size (or complexity or capacity) of an hypothesis class, the number of parameters D fails to capture some important behavior" is getting at by considering a particular example. The reasoning from finite-precision arithmetic would imply that the effective size (or complexity) of an hypothesis class is closely connected to the number of parameters, so that a class consisting of hypotheses with 10 parameters should be considered "more complex" than a class where the hypotheses have only 2 parameters. We will now see an example of a seemingly very simple 2 parameter hypothesis class that has complexity much greater that we might expect[9]: the sign-of-sine hypothesis class. Looking at sign-of-sine hypothesis class will show us that counting parameters can be a very poor indication of the complexity of an hypothesis class.

For the sign-of-sine hypothesis class, we will take

$$\begin{aligned}
&\mathcal{X} = \mathbb{R}, \\
&\mathcal{Y} = \{-1, +1\}, \\
&\mathcal{H} = \{x \mapsto \mathbf{sign}\left[\sin\left(\omega x + \theta\right)\right] \mid \omega \in \mathbb{R},\ \theta \in \mathbb{R}\}.
\end{aligned}$$

Hypotheses in this class are specified by the real numbers (ω, θ); each hypothesis takes an input x from the real line and predicts $\mathbf{sign}\left[\sin\left(\omega x + \theta\right)\right]$ for xs label.

According to our idea of hypothesis class complexity from above, we would say that the sign-of-sine hypothesis class above has the same complexity as any other hypothesis class (on inputs from the real line) whose hypotheses were specified using 2 parameters. For example, the sign-of-sine hypothesis class would be said to have the same level of complexity as the "positive line segment" class specified as: predict $+1$ for $x \in [l, u]$ and predict -1 otherwise, with $l \in \mathbb{R}$ and $u \in \mathbb{R}$. Clearly, there is a very important sense in which the sign-of-sine class is much more complex

[8] Particle physics experiments may be one of the rare exceptions.

[9] Notwithstanding the point about finite-precision representation, the sign-of-sine class complexity is effectively *unlimited* in a much stronger sense than simply implied by the hypothesis being parametrized by real numbers. Understanding what this stronger sense of "unlimited" is will highlight a need to change our characterization of the "effective complexity" of an hypothesis class to more accurately capture what really matters.

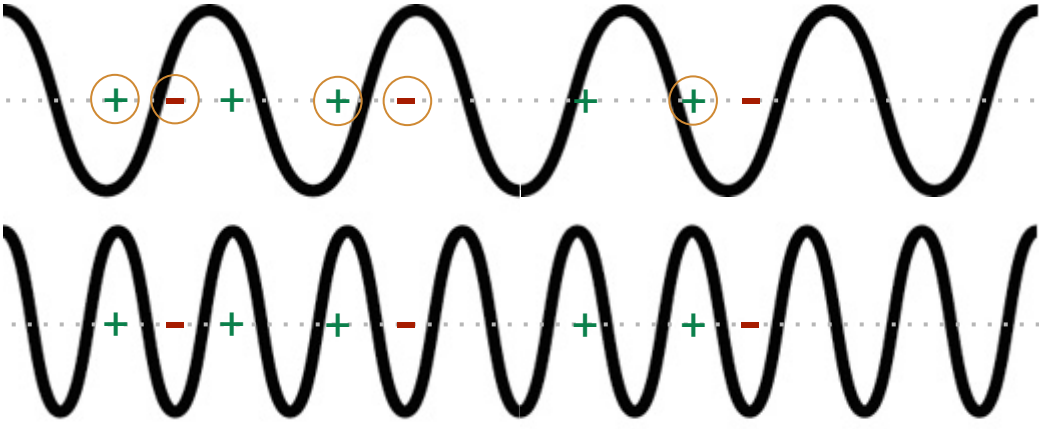

Figure 2.2: The "sign of sine" hypothesis class example. An example data set that is not correctly classified by the top sine wave. Mistakes are circled in yellow. However, we can select values for ω and θ so that all of the examples are correctly classified, as in the bottom sine wave. (Before computer representation), for *any* $+1/-1$ labeling of *any* size sample set of distinct (non-overlaid) points on the real line, there is an hypothesis in the sign-of-sine class that will perfectly label all of the points in the sample set. From this, we can conclude that no learning is possible using this class, no matter how large the sample set size.

than the positive-line-segment class. This greater complexity for the sign-of-sine class is be just as true in the "in-the-abstract", mathematical sense as it is when both classes are represented using finite-precision arithmetic; in order to remain focused on the main points, we will thus focus on the "in-the-abstract" case. At any level of representational precision the complexity of the sign-of-sine class is "greater" than the positive-line-segments class. We will consider this alternative notion of complexity (the Vapnik-Chervonenkis or VC dimension) in detail in the next lecture.

In summary: Hand-wavey arguments about the number of bits fail to differentiate between the sign-of-sine class and *e.g.* the positive-line-segment class. This is the second reason why the argument-from-finite precision is unsatisfactory (as is *any* argument based on the precision of the representation). The VC dimension will allow us to characterize the complexity of an hypothesis class in a much more rigorous way. In particular, we will see that the "effective complexity" as measured (by the VC dimension) of the sign-of-sine hypothesis class, while the VC dimension for the positive-line-segments class is 2.

2.4 A short comment on the "effective dimension" measured by the Vapnik-Chervonenkis dimension

This concept of VC dimension will get at a notion of the "effective" or "relevant" number of parameters of an hypothesis. It is possible to show that for any "reasonable" hypothesis class (*e.g.* for oriented affine separators), the VC dimension is actually equal to the number of parameters. Moreover, by using the VC dimension, we will see how to get a bound of the same form as we saw above, insofar as the number of the parameters will show up linearly under the square root. That is, we will see that for "reasonable" hypothesis classes we will get a bound with linear dependence (under the square root) on the number of parameters, without having to refer to finite-precision representation.

3 Sample complexity

While we will get to VC dimension later, at the moment we have already established bounds that are qualitatively correct. These bounds give us the relative learning guarantee: when we use the Empirical Risk Minimization approach to select an hypothesis $\hat{h}_{\mathcal{H}}$ based on a sample data set realization s, with probability at least $1-\delta$, the expected 01 loss of $\hat{h}_{\mathcal{H}}$ on the source distribution will be within $2\sqrt{\frac{\log|\mathcal{H}|+\log\frac{2}{\delta}}{2m}}$ of the expected 01 loss of $h^{\star}_{\mathcal{H}}$:

With probability at least $1-\delta$, it will be that case that

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

We can take another close look at the terms on the right hand side above, whose sum bounds the expected 01 loss of $\hat{h}_{\mathcal{H}}$. The first term $R_{01}\left(h^{\star}_{\mathcal{H}}\right)$ (the "approximation error" induced by restricting to $\mathcal{H}$) indicates the best expected 01 loss we can possibly achieve when restricting our hypotheses to be in $\mathcal{H}$. When we consider a new hypothesis class $\mathcal{H}_{new}$ that is a more complex superset of the previous hypothesis class $\mathcal{H}_{prev}$, so that $\mathcal{H}_{prev} \subset \mathcal{H}_{new}$, we hope (and expect to typically observe) that this term will decrease; considering a superset can never increase the expected loss (in the worst case the expected loss will stay the same).

On the other hand, the second term $2\sqrt{\frac{\log|\mathcal{H}|+\log\frac{2}{\delta}}{2m}}$ (the "estimation error" induced by applying ERM to a finite sample data set) will only increase when the complexity of the hypothesis class grows while the other quantities m and δ are held fixed. We might well ask whether there is any situation in which we could consider larger and larger hypothesis class supersets so as to hopefully decrease the "approximation error" $R_{01}\left(h^{\star}_{\mathcal{H}}\right)$, while still ensuring that our estimation error $2\sqrt{\frac{\log|\mathcal{H}|+\log\frac{2}{\delta}}{2m}}$ was not overly large. The answer is that we can, of course: we just need to increase the size of the sample data set to which we apply ERM.

We can be more precise than just saying that we need to increase the size of the sample set in order to be able to consider larger hypothesis classes while still keeping low estimation error: we can "invert" the relative learning guarantee to get a "sample complexity" bound. This will tell us how large a sample we need in order to be at least $1-\delta$ sure that following the ERM principle will select an hypothesis that achieves expected 01 loss that is ε-near the best possible expected 01 loss in $\mathcal{H}$.

If we want to use ERM to select $\hat{h}_{\mathcal{H}}$ with performance that is at least $1-\delta$ probable to be within ε of best possible hypothesis in the class $h^{\star}_{\mathcal{H}}$, we would write:

We want it to be the case that with probability at least $1-\delta$, we will have

$$R_{01}\left(\hat{h}_{\mathcal{H}}\right) - \inf_{h\in\mathcal{H}} R_{01}\left(h\right) \leq \varepsilon$$
$$R_{01}\left(\hat{h}_{\mathcal{H}}\right) - R_{01}\left(h^{\star}_{\mathcal{H}}\right) \leq \varepsilon$$

By examining our previous bound, we see that we will have the desired ε-nearness result if $2\sqrt{\frac{\log|\mathcal{H}|+\log\frac{2}{\delta}}{2m}} \leq \varepsilon$.[10]

[10] Because, if we have: 1) with probability at least $1-\delta$, it will be that case that

$$R_{01}\left(\hat{h}_{\mathcal{H}}\right) - R_{01}\left(h^{\star}_{\mathcal{H}}\right) \leq 2\sqrt{\frac{\log|\mathcal{H}| + \log\frac{2}{\delta}}{2m}},$$

and we also have 2) that $2\sqrt{\frac{\log|\mathcal{H}|+\log\frac{2}{\delta}}{2m}} \leq \varepsilon$, taking these two statements together implies:

Thus, we can ask, "If I hold ε, $\mathcal{H}$, and δ fixed, what is the minimum sample size m necessary for the relative learning bound to 'kick in'?"; that is, after size we will have probability at least $1-\delta$ confidence that the performance difference will be less than the bound term $2\sqrt{\frac{\log|\mathcal{H}|+\log\frac{2}{\delta}}{2m}}$. When that minimum sample size is exceeded, as long as we also have $2\sqrt{\frac{\log|\mathcal{H}|+\log\frac{2}{\delta}}{2m}} \leq \varepsilon$, will will thereby have an at least $1-\delta$ guarantee of ε-nearness. To restate once more: if we have a sample set size for which the previous inequality $2\sqrt{\frac{\log|\mathcal{H}|+\log\frac{2}{\delta}}{2m}} \leq \varepsilon$, holds, then with probability at least $1-\delta$, the ERM-selected hypothesis will display expected 01 loss performance that is ε-near to the best expected 01 loss performance achievable in $\mathcal{H}$.

We will now see what the implication of these observations is for identifying the minimum sample size needed to ensure that we are at least $1-\delta$ confident that we will have ε-nearness:

$$
\begin{aligned}
2\sqrt{\frac{\log|\mathcal{H}|+\log\frac{2}{\delta}}{2m}} &\leq \varepsilon \\
&\implies \varepsilon^2 \geq \frac{4\left(\log|\mathcal{H}|+\log\frac{2}{\delta}\right)}{2m} \\
&\implies m \geq \frac{2\left(\log|\mathcal{H}|+\log\frac{2}{\delta}\right)}{\varepsilon^2} \\
&\implies m \geq \frac{2\left(45\cdot D+\log\frac{2}{\delta}\right)}{\varepsilon^2} \\
&\implies m \geq O\left(\frac{D+\log\frac{2}{\delta}}{\varepsilon^2}\right)
\end{aligned}
$$

We interpret this expression as follows: while we may, of course, possibly get a sample of size smaller than $\frac{2\left(45\cdot D+\log\frac{2}{\delta}\right)}{\varepsilon^2}$ for which we will observe that the performance of the ERM-selected hypothesis is within ε of the best-in-class performance, once our sample size exceeds $\frac{2\left(45\cdot D+\log\frac{2}{\delta}\right)}{\varepsilon^2}$, we will be at least $1-\delta$ confident that the ERM-selected hypothesis achieves ε-near performance. We might even say that that $\frac{2\left(45\cdot D+\log\frac{2}{\delta}\right)}{\varepsilon^2}$ is an upper bound on the number of samples needed to be at least $1-\delta$ confident about ε-nearness (since for m larger than that number of samples, we definitely have at least $1-\delta$ confidence; more than that, we *might* even have achieved ε-nearness with a smaller sample size).

3.1 Interpretation: How many samples are needed to learn "almost" as well as the best hypothesis in the class?

Results of the above form "once you have a sample size larger than a specific amount, you can be at least $1-\delta$ confident that the ERM-selected hypothesis will achieve performance that is ε-near to the best-in-class performance" are referred to as "sample complexity" bounds. Two particularly interesting features of the sample complexity result

With probability at least $1-\delta$, it will be that case that

$$
\begin{aligned}
R_{01}\left(\hat{h}_{\mathcal{H}}\right) - R_{01}\left(h^{\star}_{\mathcal{H}}\right) &\overset{w.p.\ 1-\delta}{\leq} 2\sqrt{\frac{\log|\mathcal{H}|+\log\frac{2}{\delta}}{2m}} \overset{w.p.\ 1}{\leq} \varepsilon \\
\implies R_{01}\left(\hat{h}_{\mathcal{H}}\right) - R_{01}\left(h^{\star}_{\mathcal{H}}\right) &\overset{w.p.\ 1-\delta}{\leq} \varepsilon
\end{aligned}
$$

above are that we did not need to assume much beyond independence of the example pairs that we will see[11] and that we can often actually put in numbers for the important quantities: ε, $|\mathcal{H}|$, and δ.[12] When our sample size is greater than that specific amount, we can be at least $1-\delta$ confident.

The statement above also contains the aspect of the complexity bound implication that is slightly less than we might ideally wish for: by using the ERM principle on a sample larger than the indicated size, we only then know that the ERM-selected hypothesis performance is within ε of the best-in-class performance — we remain in the dark about whether the ERM-selected hypothesis will display performance within ε of the best hypothesis from any other hypothesis class (other than those that are subsets of the class $\mathcal{H}$ from which ERM selected).

4 Brief aside: How tight are these guarantees (when viewed in sample complexity form)? What can be tighter?

One might well ask whether there is room for improvement in the post-hoc (sample complexity) guarantee and the relative learning (sample complexity) guarantee in the forms above. It turns out that the expressions for the sample complexity guarantees above are "almost" tight for hypotheses selected using the principle of Empirical Risk Minimization.

Let us consider two aspects of the sample-complexity-bound expression $m \geq O\left(\frac{D+\log\frac{2}{\delta}}{\varepsilon^2}\right)$ First, the linear dependence of the bound on the "effective dimension" D of hypotheses from the class $\mathcal{H}$ is essentially "tight".[13] There is thus not really room for improvement in terms of the linear dependence on "effective dimension".

Second, it turns out that there *is* some room for improvement as regards the dependence on the specified accuracy level ε. In the expression $m \geq O\left(\frac{D+\log\frac{2}{\delta}}{\varepsilon^2}\right)$, if we wanted to be able to guarantee (with probability at least $1-\delta$) performance that was $\varepsilon_{new} = \frac{\varepsilon_{old}}{2}$-near instead of ε_{old}-near, the appearance of the "squared" on the accuracy level implies that we would need 4 times as many samples for the (probability at least $1-\delta$) performance guarantee to "kick-in" at the new $\varepsilon_{new} = \frac{\varepsilon_{old}}{2}$ accuracy level.

It turns out that this inverse-and-squared dependence on the specified accuracy level ε actually can be improved (albeit modestly). Although we will omit the details, we will write down the improved expression.

[11] We also needed to assume either boundedness or (less restrictively) subgaussianity of the corresponding "loss function" random variables derived by computing the loss function values on the "example pair" random variables.

[12] We might for example, typically decide to specify $\delta = 2e^{-7} \approx .002$, so that $\ln\frac{2}{\delta} = 7$. Only the value of $|\mathcal{H}|$ might be problematic, for reasons that we have discussed above and will discuss at more length below. For the moment, we will just use the qualitative result from the argument-from-finite-precision: $\log|\mathcal{H}| = c \cdot D$. That is, the log cardinality of the hypothesis class depends linearly on the number of parameters necessary to indicate a specific hypothesis in the class.

[13] That is, one can imagine an hypothesis class for which the inequality should be an equality.

Improved dependence of the sample complexity on ε :
With the more correct dependence, we will see that, rather than depending on ε^2, the expression will actually look more like $\frac{1}{\varepsilon}\left(\frac{R_{01}(h^\star_{\mathcal{H}})+\varepsilon}{\varepsilon}\right)$.

$$m \geq O\left(\left[\frac{\log|\mathcal{H}| + \log\frac{2}{\delta}}{\varepsilon}\right]\left[\frac{R_{01}\left(h^\star_{\mathcal{H}}\right)+\varepsilon}{\varepsilon}\right]\right)$$
$$\Longrightarrow m \geq O\left(\left[\frac{D + \log\frac{2}{\delta}}{\varepsilon}\right]\left[\frac{R_{01}\left(h^\star_{\mathcal{H}}\right)+\varepsilon}{\varepsilon}\right]\right).$$

Analogously to the previous expression, this more correct expression can be interpreted as saying: when the sample size m is greater than $O\left(\frac{\log|\mathcal{H}|+\log\frac{2}{\delta}}{\varepsilon}\frac{R(h^\star)+\varepsilon}{\varepsilon}\right)$, with probability at least $1-\delta$, it will be the case that the ERM-selected hypothesis will achieve performance that is within ε of the best-in-class performance (and it is also possible that ε-nearness will be achieved for some sample sets of size smaller m).
The point here is that this improved sample complexity expression definitely shows better dependence than $\frac{1}{\varepsilon^2}$. Moreover, the dependence can be *much* better if $R_{01}\left(h^\star_{\mathcal{H}}\right)$ is "very small" or 0.
If you can somehow be certain that the best-in-class performance is zero (*i.e.* $R_{01}\left(h^\star_{\mathcal{H}}\right)=0$, then the term $\frac{R(h^\star)+\varepsilon}{\varepsilon}$ becomes $\frac{0+\varepsilon}{\varepsilon}=1$, so that the complexity-bound-dependence would be be $\frac{1}{\varepsilon}$. That is, if we somehow know that $R_{01}\left(h^\star_{\mathcal{H}}\right)=0$, the revised expression above tells us that in order to be able to guarantee (with probability $1-\delta$) performance that was $\varepsilon_{new}=\frac{\varepsilon_{old}}{2}$-near instead of ε_{old}-near, we would need 2 times as many samples (instead of the previous 4 times) for the (probability at least $1-\delta$) performance guarantee to "kick-in" at the new $\varepsilon_{new}=\frac{\varepsilon_{old}}{2}$ accuracy level.

For our purposes in this class, the improved dependence on the specified accuracy level ε is of limited importance. Much more important for us is the linear dependence of the sample-complexity expression on the "effective dimension"; alternately stated as: of the linear dependence on the log cardinality.

What is perhaps the most important related result (which we will again not cover in detail) is that it is possible to establish/prove that "You might as well use Empirical Risk Minimization". We can state this slightly more specifically as: However you decide to select an hypothesis based on a sample data set, no matter what "learning rule" or "inductive principle" you use, you will not be able to use these examples substantially more effectively that you would have if you had used the ERM principle. The number of samples that you will need (when selecting an hypothesis using *any* learning rule) to guarantee (with probability at least $1-\delta$) performance that is ε-near to the best-in-class performance *cannot* be much less than what we indicated above for the case where we select an hypotheses using the ERM principle. Thus, "you might as well use Empirical Risk Minimization".

5 The trade-off between estimation error and "approximation error"

For the results considered above, we assume that we have *already* settled on a specific hypothesis class to use. For this *particular* hypothesis class, we then ask questions about relative learning, such as "What sample size m do I need to ensure with probability at least $1-\delta$ that $R_{01}\left(\hat{h}_{\mathcal{H}}\right) - R_{01}\left(h^\star_{\mathcal{H}}\right) \leq \varepsilon$?" Alternately, we could ask questions about the post-hoc guarantee, such as "If using the ERM principle on a sample set selects hypothesis $\hat{h}_{\mathcal{H}}$ that achieves sample average 01 loss of $\ell\%$, how much worse is the expected 01 loss of $\hat{h}_{\mathcal{H}}$ likely to be?"

In contrast to the above situation of pre-selected single hypothesis classes, it is commonly the case in practice that we have a *choice* of hypothesis classes. If we are trying to find a predictor with low expected 01 loss, and we are not limited to only considering hypotheses from some specific class, we then can first consider a range of

Figure 5.1: Example decision boundaries in the class of degree-at-most-k polynomials. Linear. Parabolic. Higher order.

hypothesis classes and only subsequently select an hypothesis from one of the class; we can consider "richer"/"more complex" hypothesis classes or "less rich"/"less complex" hypothesis classes. As we will see, this decision of which hypothesis class to choose from will reveal a trade-off.

With a "richer" hypothesis class, the expected estimation error will be larger, but we are likely to see smaller "approximation error"; with a "less rich" hypothesis class, the expected estimation error will be smaller but we are likely to see larger "approximation error".

5.1 Simplicity to complexity via a hierarchy of hypothesis classes: Example hierarchies

In order to get a better feeling for the trade-offs mentioned above, let us now consider a situation where we have such a hierarchy of hypothesis classes, where each level of the hierarchy corresponds to a "richer" superset of hypotheses.

5.1.1 Example hypothesis class hierarchy: sign-of-degree-k-polynomial .

We will consider a hierarchy of hypothesis classes, with "complexity" indexed by an integer k. The corresponding hypothesis class $\mathcal{H}_k$ will consist of predictions based on the sign of a polynomial of degree at most k, where k is a non-negative integer:

$$\begin{aligned}\mathcal{X} &= \mathbb{R}^d \\ \mathcal{Y} &= \{+1, -1\} \\ \mathcal{H}_k &= \{x \mapsto \text{sign}\left[\text{poly}_k\left(x\right)\right] \mid \text{poly}_k\left(\cdot\right) \text{ is a polynomial with degree at most } k,\ k \in \mathbb{Z} \cup \{0\}\}\end{aligned}$$

The important aspect of this class for our purposes is that as we increase k, we get a hierarchy of classes, with each larger value of k yielding an hypothesis class that is a superset of the previous hypothesis classes; for example $\mathcal{H}_5 \subset \mathcal{H}_6$, since polynomial of degree no more than 5 could also be described as a polynomial of degree no more than 6 (as well as no more than 7, 8, etc...).

5.1.2 Parameter count for the sign-of-degree-at-most-k-polynomial hypothesis classes

What is the parameter count D_k for hypotheses in the kth sign-of-degree-at-most-k-polynomial hypothesis class $\mathcal{H}_k$? The parameter count D_k for $\mathcal{H}_k$ is given by the count of all the coefficients for each of the possible terms in the degree-at-most-k-polynomial, so it can be no greater than $(d+1)^k$: that is, $D_k \leq (d+1)^k$, where d is the input dimension.

The point here is that for a larger value of the maximum-allowed-degree k, the corresponding $\mathcal{H}_k$ includes "richer" polynomials. For $k = 1$, the possibilities are just affine functions of the input. For $k = 2$, the corresponding class also includes parabolas. The cost for this richness is the increased parameter count, which leads a higher estimation error bound.

Figure 5.2: Affine separators with zero norm constraint on the parameter vector w. $k = 1$ vertical. $k = 1$ horizontal. $k = 2$.

5.1.3 Example hypothesis class hierarchy: Affine separators with zero norm constraint

We can further consider another hypothesis class hierarchy, in which $\mathcal{H}_k$ consists of affine predictors where you can use at most k features:

$$\begin{aligned}\mathcal{X} &= \mathbb{R}^d \\ \mathcal{Y} &= \{+1, -1\} \\ \mathcal{H}_k &= \left\{x \mapsto \text{sign}\left[w^T x + b\right] \;\middle|\; w \in \mathbb{R}^d,\ b \in \mathbb{R},\ \|w\|_0 \leq k,\ k \in \mathbb{Z} \cup \{0\}\right\},\end{aligned}$$

where the so-called[14] "zero norm" $\|\cdot\|_0$ counts the number of non-zero entries in the vector argument. By specifying $\|w\|_0 \leq k$, we are restricting the number of non-zero entries in w to be at most k. We are still looking at oriented affine separators, but instead of allowing arbitrary affine separators, $\mathcal{H}_k$ is only allowed to be a function of at most k non-zero features (although we do not specify in advance which features will be non-zero).

Although the limitations of the blackboard are somewhat severe, we can see a simple blackboard example of the above hypothesis class when $d = 2$ and $k = 1$. In the $d = 2$ case, when we set $k = 1$, we are indicating that the predictor is only allowed to depend on 1 of the 2 features of the input. This restriction means that predictors in this case will either be vertical separators or horizontal separators.

The important aspect of this class for our purposes is again that as we increase k, we get a hierarchy of classes, each larger value of k yielding an hypothesis class that is a superset of the previous hypothesis classes; for example $\mathcal{H}_{10} \subset \mathcal{H}_{11}$, since every affine separator whose w has no more than 10 non-zero entries also has no more than 11 non-zero entries (and no more than 12, 13, etc...).

5.1.4 Parameter count for the "affine separators with zero norm constraint" hypothesis classes

Let us now do a parameter count for the "affine separators with zero norm constraint" hypothesis classes. As written above in $\mathcal{H}_k = \left\{x \mapsto \text{sign}\left[w^T x + b\right] \;\middle|\; w \in \mathbb{R}^d,\ b \in \mathbb{R},\ \|w\|_0 \leq k,\ k \in \mathbb{Z} \cup \{0\}\right\}$, it might appear that we should say that the number of parameters for (w, b) is $d + 1$, because the vector w is d-dimensional and b is 1-dimensional. However, counting parameters in this way essentially ignores the restriction to having at most k non-zero entries in w. To see why this matters, consider the case where we have $d = 1000$ but $k = 10$. It seems strange to say that the parameter count is $1000 + 1$ when we know that only 10 of the 1000 entries in w are non-zero; it hardly seems necessary to separately tell where all of the 0s are. We will now pursue a more sensible parameter count for (w, b).

We can motivate this more sensible parameter count by considering how we might specify a particular hypothesis in this class; for b, we just use a real number. The case of w is somewhat more involved. We observe that when $k \ll d$, we are much better off identifying the k-sparse w vector by first indicating which (at most) k of the d coordinates of w are non-zero and then telling the values in those non-zero coordinates. Each of the k non-zero locations in w can be specified by the index of that location, *i.e.* by an integer in the range $[1, d]$. Thus, we could choose to represent the k-sparse vector w by k integers telling the non-zero locations, and then telling the values at those locations using k real numbers. In all, we would represent (w, b) using k integers, and $k + 1$ real numbers. The $k + 1$ real numbers can be handled according to the same argument we used above for parameters represented in floating-point: the

[14] "So-called" because the "zero norm" $\|\cdot\|_0$ fails the property of "positive homogeneity" required to be a norm. This property requires that $\|\alpha v\| = |\alpha| \|v\|$ for any scalar α. For the "zero norm" this fails because multiplying a vector w by a (non-zero) scalar α does not change the number of non-zero entries in that vector. That is, if $\|w\|_0 = k$, it will also be the case that $\|\alpha w\|_0 = k$ for any non-zero scalar α.

number of distinct values for each of the $k+1$ 64-bit floating point numbers is $\left(2^{64}\right)^{k+1} = 2^{64\cdot(k+1)}$. On the other hand, to represent (in the computer) the integer indices in the range $[1, d]$, we require $\lceil \log_2 d \rceil$ bits for each of the k location indices. There are $2^{\lceil \log_2 d \rceil}$ possible binary numbers that could indicate each of the k non-zero locations out of the total d; the number of distinct possibilities here is thus $\left(2^{\lceil \log_2 d \rceil}\right)^k = 2^{k\cdot\lceil \log_2 d \rceil}$. The total number of distinct hypotheses in the class $\mathcal{H}_k$ consisting of "affine separators with zero norm at most k" is the product of the number $2^{k\cdot\lceil \log_2 d \rceil}$ of distinct non-zero-location specifications times the number $2^{64\cdot(k+1)}$ of distinct floating point number values at each of those locations, giving: $|\mathcal{H}_k| = 2^{k\cdot\lceil \log_2 d \rceil} 2^{64\cdot(k+1)} = 2^{k\cdot\lceil \log_2 d \rceil + 64\cdot(k+1)}$, so that the natural log cardinality is $\ln|\mathcal{H}_k| = k \cdot \lceil \log_2 d \rceil \cdot \ln 2 + 64 \cdot \ln 2 \cdot (k+1) = O\left(k \cdot \lceil \log_2 d \rceil + (k+1)\right)$.[15]

5.2 Recap of hierarchy of hypothesis classes

In both of the cases above, we have an indexed hierarchy of hypothesis classes that increase in complexity as we increase the "complexity level parameter" k. In the first case, k indicated the largest allowed degree of a polynomial; in the second case, k indicated the largest allowed number of non-zero entries of the parameter w of an affine separator. For both of these complexity-level-specified-by-k hierarchies, increasing k yields a superset, since $k_{smaller} < k_{larger} \Longrightarrow \mathcal{H}_{k_{smaller}} \subset \mathcal{H}_{k_{larger}}$.

6 Considering the graph of behaviors over a hierarchy of hypothesis classes of increasing complexity. Quantities of interest: approximation error, training error, estimation error, generalization error (of the empirical risk minimizer), validation error of the empirical risk minimizer.

We have now introduced a variety of quantities of interest as regards the performance of predictors. We will soon graph them together to see how each of the quantities relates to the others is to graph them together. As we increase k and thereby allow more complex hypotheses, we will see what the effect of the wider class is on various sample average 01 loss or expected 01 loss amounts. For example, how does the "approximation error" $R_{01}\left(h_k^\star\right)$ behave when k increases? We first show this on its own for the integer k case, and subsequently with other quantities in the continuous k case.

6.1 "Approximation error" $R_{01}\left(h_k^\star\right)$ behavior

How does the "approximation error" $R_{01}\left(h_k^\star\right)$, the expected 01 loss of the best-in-class-$\mathcal{H}_k$ hypothesis, behave as we increase the complexity parameter k? Does it increase with k? Decrease? Increase and then decrease? Decrease and then increase? It decreases. We can see this by starting out considering the best-in-class performance when the class is quite "small". In a small class we are quite likely to have high approximation error: the best performance in a small class need not be very good. If we begin to allow larger and larger hypothesis classes by increasing the value of k, we expect the "approximation error" to decrease. Moreover, because we have an inclusive hierarchy where each larger-k hypothesis class contains all of the elements of the smaller-k hypothesis classes, the approximation error is definitely going to decrease monotonically. When you minimize something over a specified set, allowing the minimization to range over a larger, inclusive set will only decrease the best value; even if none of the newly added

[15] This is again a qualitatively correct result, by which we mean that a more theoretically principled approach to determining the effective dimensionality (*e.g.* the yet-to-be-defined-VC dimension) would give a similar big "O" result.

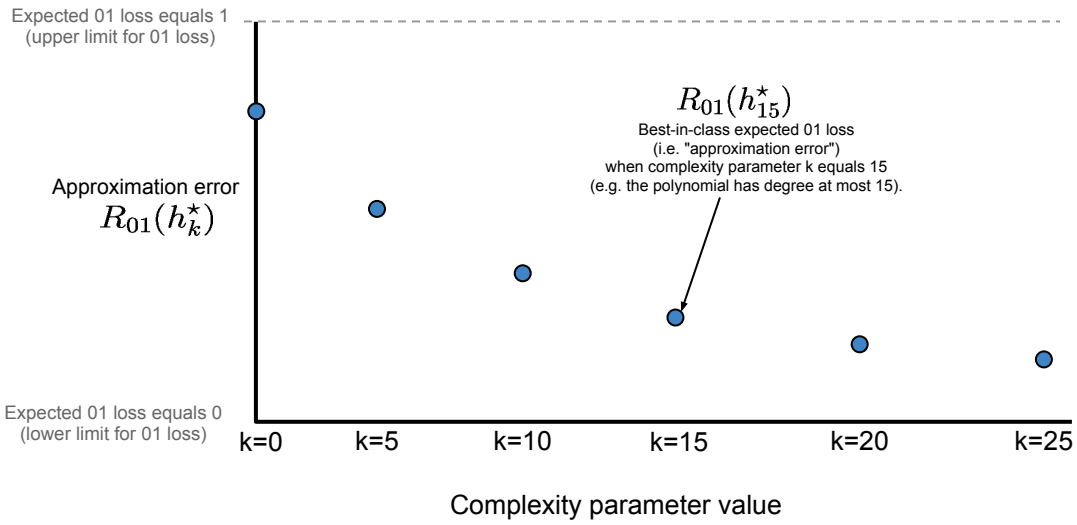

Figure 6.1: Qualitative plot of "approximation error" $R_{01}\left(h_k^\star\right)$ as a function of increasing integer-valued complexity parameter k, as per the example hypothesis class hierarchies above.

elements do a better job, you can always just select the previous best performer. In sum: the "approximation error" is definitely non-increasing. It might, however, not ever go to 0. There are two reasons for this. Firstly, if the "true" hypothesis is not in any of the hypothesis classes indexed by k, we will always have a non-zero approximation error. Secondly, even if the "true" hypothesis is in one of the classes indexed by k, there may be fundamental, unavoidable "noise" in the label source distribution; even the "true" hypothesis will not achieve zero error.

6.2 Training error $\hat{R}_{s,01}\left(\hat{h}_k\right)$ behavior of ERM-selected hypothesis

How do we expect the training error (the empirical 01 risk/sample average 01 loss of the empirical risk minimizer $\hat{h}_k$) to look[16]? If we fix our sample data set, the training error would also be monotonically decreasing as we increased the allowed complexity via k, as we are again performing minimization over increasingly larger sets. We also expect the training error $\hat{R}_{s,01}\left(\hat{h}_k\right)$ to typically be smaller than the approximation error $R_{01}\left(h_k^\star\right)$ (although from our previous bounds, we saw that it was certainly possible for the training error to be greater than the approximation error for some sample data sets; however, since the training error is measured for an Empirical Risk *Minimizing* hypothesis, we typically expect the training error to be smaller than the approximation error {those are the hypotheses more likely to be selected by ERM}).

In the low-complexity regime towards the left of the graph, we can be reasonably confident that ERM will select as empirical risk minimizer $\hat{h}_k$ something that shows expected 01 loss performance not too far from the expected 01 loss performance expected risk minimizer $h_k^\star$. However, as we increase the size of our hypothesis class

[16]If we were to consider a *different* sample set of size m for each complexity level k, the plot of training error would of course display significant variance (assuming that the sample size m was relatively modest). By fixing the particular sample set, we can focus on the effect of allowing more "complexity" as indexed by k, as shown for a fixed data set.

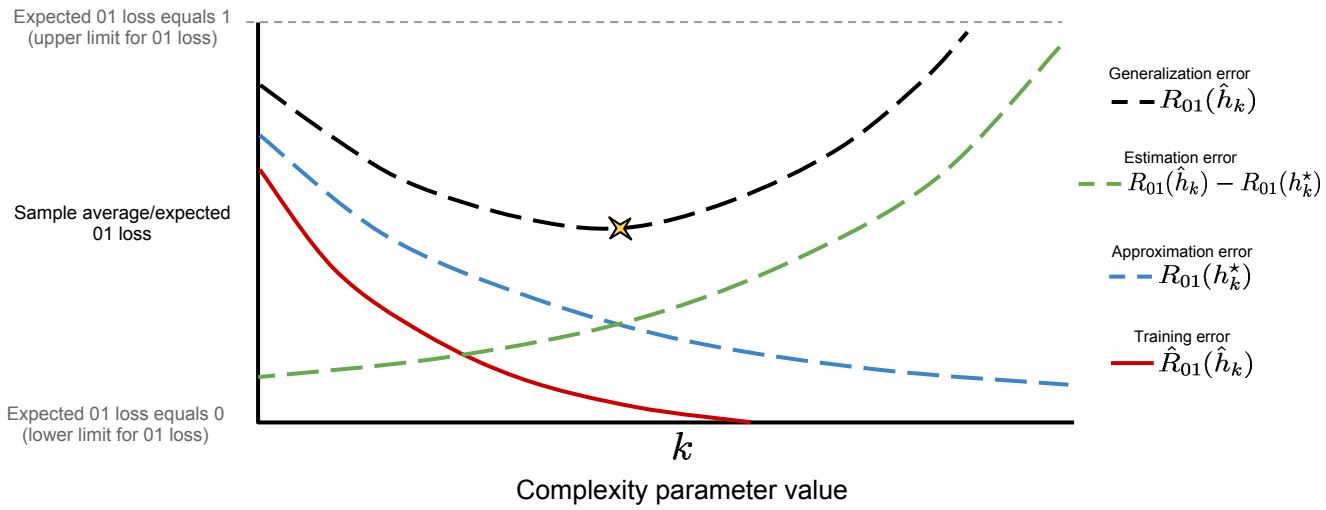

Figure 6.2: Qualitative plot of quantities of interest. For a fixed sample set of a specific size m, and shown as a function of the now-continuous complexity parameter k. Dashed lines indicate unobservable/unmeasurable quantities. Included: Approximation error $R_{01}\left(h_k^\star\right)$; training error $\hat{R}_{s,01}\left(\hat{h}_k\right)$ (the empirical error of the empirical risk minimizer); estimation error $R_{01}\left(\hat{h}_k\right) - R_{01}\left(h_k^\star\right)$; generalization error $R_{01}\left(\hat{h}_k\right)$ of the empirical risk minimizer. Each quantity is displayed as a function of the k parameter indexing increasingly complex, inclusive hypothesis classes $\mathcal{H}_k$. The yellow "X" marks the performance of the hypothesis we would like to be able to identify during learning: the hypothesis for which the generalization error (*i.e.* the future performance) is minimized.

size, the probabilistic "estimation error" bound will also increase, indicating that there is a higher chance that the ERM-selected hypothesis will display very different expected performance than the best-in-class performer. One often-associated indicator of this state of affairs is 0 training error. We frequently observe that hypotheses with 0 training error have quite poor generalization error performance, as detailed below.

6.3 Estimation error behavior

Recall that we defined the quantity $R_{01}\left(\hat{h}_k\right) - R_{01}\left(h_k^\star\right)$ as the estimation error. This tells us how far the expected 01 loss performance of the ERM-selected hypothesis $\hat{h}_k$ is from the expected 01 loss performance of the best-in-class-$\mathcal{H}_k$ hypothesis $h_k^\star$. Recall also that the relative learning guarantee gives us a bound on the estimation error of the form $2\sqrt{\frac{\log|\mathcal{H}_k|+\log\frac{2}{\delta}}{2m}}$. How do we expect the estimation error to change as we increase complexity via k?

We expect it to increase. One way to see why this should be is to note that as ERM is applied to select from more and more complex classes (while not performing this selection based on a larger and larger sample data set), it becomes more likely that the hypothesis selected by ERM shows "misleadingly good" performance on the sample data set; *i.e.* that the ERM-selected hypothesis will "look good" on the sample set while being much worse on future data from the source distribution. We also note that the probabilistic bound on the estimation error $2\sqrt{\frac{\log|\mathcal{H}_k|+\log\frac{2}{\delta}}{2m}}$ will grow as the hypothesis class grows in complexity (via $|\mathcal{H}_k|$); for a given confidence level $1-\delta$, the *bound* term on the estimation error must increase with k in order to remain correct, which indicates a corresponding increase in the possibility of a large estimation error.

6.4 Generalization error $R_{01}\left(\hat{h}_k\right)$ behavior of ERM-selected hypothesis

The generalization error $R_{01}\left(\hat{h}_k\right)$ is bounded in the relative learning guarantee: With probability at least $1-\delta$, it will be that case that $R_{01}\left(\hat{h}_k\right) \leq R_{01}\left(h_k^\star\right) + 2\sqrt{\frac{\log|\mathcal{H}_k|+\log\frac{2}{\delta}}{2m}}$. We note that as the k increases, the $2\sqrt{\frac{\log|\mathcal{H}_k|+\log\frac{2}{\delta}}{2m}}$ part of the bound increases, so that (if the decrease in approximation error $R_{01}\left(h_k^\star\right)$ does not offset this) it becomes possible to observe much larger generalization error. Expressed alternately, we could say that as k increases, it might become more likely to observe much larger estimation error $R_{01}\left(\hat{h}_k\right) - R_{01}\left(h_k^\star\right)$; we might say this is because when we consider more hypotheses, we are more likely to encounter one with "meaninglessly good" performance on the sample data set s.

How might we expect generalization error to typically behave? We can break the generalization error $R_{01}\left(\hat{h}_k\right)$ into the sum of the approximation error $R_{01}\left(h_k^\star\right)$ and the estimation error $R_{01}\left(\hat{h}_k\right) - R_{01}\left(h_k^\star\right))$[17] When we have a "simple" class with a small k value, we expect a moderately higher generalization error becomes the approximation error term will be somewhat high in this case, while the estimation error is likely to be modest (perhaps best seen by noting that when $|\mathcal{H}_k|$ is small, the bound on the estimation $2\sqrt{\frac{\log|\mathcal{H}_k|+\log\frac{2}{\delta}}{2m}}$ is correspondingly modest) . When we have a "complex" class with a high k value, we expect a much smaller approximation error $R_{01}\left(h_k^\star\right)$ because we can consider a much broader range of hypotheses; however, we are likely to pay a price in estimation error for this large $|\mathcal{H}_k|$. This (possible) price is perhaps best seen through the corresponding increase in the bound term

[17]Or we could refer to the relative learning guarantee to express the generalization error as being, with probability $1-\delta$, bounded as follows: $R_{01}\left(\hat{h}_k\right) \leq R_{01}\left(h_k^\star\right) + 2\sqrt{\frac{\log|\mathcal{H}_k|+\log\frac{2}{\delta}}{2m}}$. That is, instead of decomposing the generalization error into the "approximation error" and a *bound* on the estimation error.

$2\sqrt{\frac{\log|\mathcal{H}_k|+\log\frac{2}{\delta}}{2m}}$. When selecting from a wide range of hypotheses and evaluating those hypotheses by their sample set performance, we become more likely to select an hypothesis that looks "misleadingly good" on the sample set, while displaying much worse performance on the source distribution[18].

7 Structural risk minimization: Which complexity level should we use?

Having introduced the notion of having a hierarchy of hypothesis classes and then examined the qualitative behavior of various quantities of interest when we consider increasingly complex hypothesis classes, we must now ask: what particular level of complexity should we use. We mentioned above that we seek the level k such that the hypothesis that ERM would select from that class $\mathcal{H}_k$ would give the best performance on future data from the source distribution (i.e. the complexity level k for which the generalization error of the hypothesis that ERM selects from $\mathcal{H}_k$ will be smallest). The problem with using the generalization error to pick the complexity level is that we do not have access to the generalization error; it is an unobservable quantity. In fact, of the quantities plotted above, we can only actually observe the training error $\hat{R}_{s,01}\left(\hat{h}_k\right)$ of the hypothesis that ERM selects from $\mathcal{H}_k$. If we could directly observe performance that hypotheses will have on future data (*i.e.* the generalization error) we wouldn't need to do Empirical Risk Minimization in the first place — we would just select the hypothesis h with the lowest generalization error $R_{01}(h)$.

Should we pick the complexity by selecting the k for which the training error is minimized? It is the only on of the quantities that we can observe, after all. This approach to selecting the appropriate complexity is problematic; we can see that this is so by observing that the training error will decrease monotonically as we increase the complexity k. If we were to pick k such that the training error of the hypothesis that ERM selects from $\mathcal{H}_k$ would be minimized, we would always pick the largest allowed value of k. If nothing else, our post-hoc guarantee that with probability $1-\delta$, it will be the case that $R_{01}\left(\hat{h}_k\right) - \hat{R}_{s,01}\left(\hat{h}_k\right) \leq \sqrt{\frac{\log|\mathcal{H}_k|+\log\frac{2}{\delta}}{2m}}$ tells us that using the largest k possible could lead to very bad performance on future data, even though $\hat{h}_k$ does well on the sample data set. That is, even if $\hat{R}_{s,01}\left(\hat{h}_k\right)$ is modest, the term $\sqrt{\frac{\log|\mathcal{H}_k|+\log\frac{2}{\delta}}{2m}}$ could be very large when k is large, and thus so could the generalization error.

7.1 A closer look at the post-hoc guarantee on the generalization error $R_{01}\left(\hat{h}_k\right)$

At this point, our situation is as follows: we would like to find the complexity level k so that applying ERM to select an hypothesis from $\mathcal{H}_k$ results in the smallest generalization error for the selected hypothesis. We cannot observe the generalization error. However, we do have a post-hoc learning guarantee that relates the generalization error and the training error (which we can observe): with probability $1-\delta$, it will be the case that $R_{01}\left(\hat{h}_k\right) \leq \hat{R}_{s,01}\left(\hat{h}_k\right) + \sqrt{\frac{\log|\mathcal{H}_k|+\log\frac{2}{\delta}}{2m}}$.

We note that on the right hand side of the post-hoc guarantee (that probabilistically bounds the generalization error) both terms are things we have access to. We certainly have access to the training error $\hat{R}_{s,01}\left(\hat{h}_k\right)$. Of the variables in the other term, we get to specify δ, we know our sample size m, and we also know what our hypothesis

[18] With a bigger class, we can find hypotheses with better and better performance on the sample data set. However, at some point, the *generalization* error probably starts increasing again because the complexity is increasing; thus the required number of samples to estimate well is also increasing. In the present setting, we are holding the number of samples fixed, so that the estimation error becomes progressively worse.

class is (and so we would know $\log|\mathcal{H}_k|$ if the class is finite, or the corresponding parameter count D_k if the class is not finite). The observation that both of these quantities are observable suggests instead of trying to minimize the (unobservable) generalization error, we can select the complexity level k and the ERM-selected hypothesis $\hat{h}_k$ that together minimize the bound on the generalization error: $\hat{R}_{s,01}\left(\hat{h}_k\right) + \sqrt{\frac{\log|\mathcal{H}_k|+\log\frac{2}{\delta}}{2m}}$. This approach (or inductive principle) is referred to as Structural Risk Minimization.

7.2 Structural Risk Minimization defined

We are now seek to going to search both over the hypothesis classes in our hierarchy and over hypotheses in those classes. The approach that we will consider is referred to as Structural Risk Minimization (SRM), and consists of minimizing the sum of the training and and the bound term: $\hat{R}_{s,01}\left(\hat{h}_k\right) + \sqrt{\frac{\log|\mathcal{H}_k|+\log\frac{2}{\delta}}{2m}}$ or, in keeping with a view in terms of the parameter count, $\hat{R}_{s,01}\left(\hat{h}_k\right) + \sqrt{\frac{\log|\mathcal{H}_k|+\log\frac{2}{\delta}}{2m}}$.

We could describe this SRM approach as: select the complexity parameter k and the hypothesis $h \in \mathcal{H}_k$ such that the sum of the empirical error of h, $\hat{R}_{s,01}(h)$ plus our bound $\sqrt{\frac{45D_k+\log\frac{2}{\delta}}{2m}}$ is minimized. Note that in the case of a finite hypothesis class, we would have $\log|\mathcal{H}_k|$ in place of D_k. We can write this approach as follows:

$$\mathbf{SRM}_{k,\mathcal{H}_k}(s) = \hat{h}_k \triangleq \underset{k,\ h\in\mathcal{H}_k}{\operatorname{argmin}} \left\{ \hat{R}_{s,01}(h) + \sqrt{\frac{45\cdot D_k + \log\frac{2}{\delta}}{2m}} \right\}.$$

Although the definition and description given here is relatively straightforward, the term Structural Risk Minimization is sometimes used to refer to different definitions; despite their differences, these approaches do share a similar basic idea: Instead of minimizing just the sample average 01 loss on its own, we are also going to pay attention to "overoptimistic" the sample average 01 loss might be as an indication of the expected 01 loss on the source distribution. Essentially, this method can be described as taking into account both the sample average 01 loss of the hypothesis being considered and also the complexity of the class from which the hypothesis is taken.

If we find an hypothesis from a "simple" class that also achieves low training error, we can be more confident that that hypothesis will do well on future data from the source distribution. For example, we might consider $\mathcal{H}_k$ to be polynomials of degree at most k; if we find an hypothesis with low degree and low training error, that is the hypothesis that SRM would select (even if ERM would select an hypothesis with higher degree but lower training error). As its description might convey, this is qualitatively a reasonable approach. However, it does have some problems as regards its practical application.

7.3 Problems with applying structural risk minimization in practice

Despite being a reasonable approach in its general idea, applying SRM in practice has a significant stumbling block: the bound term $\sqrt{\frac{45\cdot D_k+\log\frac{2}{\delta}}{2m}}$ is very loose (it can even lead to an estimate on the generalization error $R_{01}\left(\hat{h}_k\right)$ that is greater than 1). Despite being correct, the bounds in the forms we have considered are so loose that using them in an SRM approach as described above does not typically yield good results. We could have guessed at this looseness, even if only from the arbitrariness of the constant "45"; the arbitrariness of this constant was perhaps only modestly troubling conceptually. However, if we intend to use these bounds in practice (numerically) to select an hypothesis, this sort of arbitrariness is problematic: for example, if we change the representation from 64-bit to 32-bit, the 45 will change and we will possibly select a different hypothesis.

Even if we were to do the math as carefully as we can and thereby get the tightest number possible, it will still not be the case that the resulting bound will be *numerically* tight. Beyond that, there is the fact that all of these bounds are in some sense worst case bounds: they hold for any hypothesis class, for any source distribution. If we were to consider specific cases of source distribution, we will find that the bounds we have considered are likely to be seriously over-estimating. These problems mean that SRM is rarely useful in practice; typically, the hypothesis and k value selected by minimizing the sum above is not actually very good on the source distribution.

7.4 What to do instead of SRM: Train then validate

Given that SRM typically leads to less-than-stellar results in practice, we need to consider another approach. We want to pick a class that will have an hypothesis with good generalization error. We don't get direct access to the expected 01 loss on the source distribution (the generalization error), and if we were to choose a class that has an hypothesis with low sample average 01 loss, we would end up selecting the most complex class allowed (for the reasons discussed above). The alternative approach that we are considering attempts to get around not having direct access to the generalization error by trying to get an estimate of it.

Suppose that you had an arrangement so that whenever you wanted to know how an hypothesis would do on future data, you could request another sample of size m — this sample would yield an estimate of the performance on the source distribution. While the ERM selected hypothesis might look "misleadingly" good on the sample used to select it, it is unlikely to look similarly "misleadingly good" on the new data sample. As it turns out, we can partially replicate just such an approach to estimating the generalization error: the idea is to carve out a "future data sample" from the original sample data set that we are provided at the start. We select an hypothesis using the rest of the data and then look to the "future data sample" to estimate generalization error.

This process of "setting aside" part of the sample data set to use as a "future data" sample is referred to as a *validation* approach.

We being a validation approach with our initially provided sample data set s. We proceed to randomly divide this sample data set into two parts: a "training" set s_{train} that we will use to select the hypothesis, and an independent "validation" set s_{val}, that we will use to estimate the generalization error of the selected hypothesis. That is, $s = \{s_{\text{train}}, s_{\text{val}}\}$.

Having divided up our initial sample data set, for each k that we are considering, we calculate $\hat{h}_k = \mathbf{ERM}_{\mathcal{H}_k}(s_{\text{train}})$, the minimizer of the sample average 01 loss calculated *only* on the training set s_{train}. If we wanted to be particularly explicit about the fact that $\hat{h}_k$ is selected using the training set s_{train} only, we could write $\hat{h}_{k,s_{\text{train}}}$: we only use s_{train} when selecting the hypothesis $\hat{h}_{k,s_{\text{train}}}$ from each $\mathcal{H}_k$. Having arrived at the candidate $\hat{h}_k$s (each of which is ERM-on-s_{train}-selected out of the corresponding $\mathcal{H}_k$) for all allowed k values, we now want to select between the different $\hat{h}_k$ (empirical risk minimizers on the training set). We perform this selection by evaluating each $\hat{h}_k$s performance on the validation set s_{val} and then selecting the $\hat{h}_k$ with minimum validation set average 01 loss $\hat{R}_{s_{\text{val}},01}\left(\hat{h}_k\right)$. At the end of this process, we return the best-on-the-validation-set hypothesis $\hat{h}_{\hat{k}}$.[19]

[19] To strongly emphasize that $\hat{h}_{\hat{k}}$ was selected based on both the training set $s_{\text{train}}$and the validation set s_{val}, we could write $\hat{h}_{\hat{k},\{s_{\text{train}},s_{\text{val}}\}}$instead.

The validation approach:
Begin by randomly dividing the original sample data set s into training and validation sets: $s = \{s_{\text{train}}, s_{\text{val}}\}$

1. For each value of the complexity parameter k, calculate the hypothesis in the complexity class selected by k that minimizes the training set s_{train} average 01 loss:

$$\hat{h}_{k,s_{\text{train}}} \triangleq \underset{h \in \mathcal{H}_k}{\text{argmin}} \; \hat{R}_{s_{\text{train}},01}(h)$$

2. Select the complexity parameter value $\hat{k}_{s_{\text{val}}}$ whose corresponding hypothesis $\hat{h}_{\hat{k}}$ minimizes the validation set s_{val} average 01 loss:

$$\hat{k}_{s_{\text{val}}} \triangleq \underset{k}{\text{argmin}} \; \hat{R}_{s_{\text{val}},01}\left(\hat{h}_{k,s_{\text{train}}}\right)$$

3. Return $\hat{h}_{\hat{k}}$.

Note that the "hat" on the k is meant to indicate that the value of the parameter has been selected using sample data; in this case, using the validation set s_{val}.

7.5 Graphing the validation error

We expect the validation set error to vary around the generalization error; the expectation of the validation error over validation sets carved out from all possible sample data sets will be equal to the generalization error, but any particular validation error will typically vary around that expectation. The validation error will be an unbiased estimator of the generalization error (of each $\hat{h}_{k,s_{\text{train}}}$) so long as the validation set s_{val} is independent of the training set s_{train}.

To recap what is going on in the graph of the validation error, the plot is meant to indicate the qualitative behavior (as k increases) of validation set average 01 loss for the ERM-selected hypothesis out of $\mathcal{H}_k$: $\hat{R}_{s_{\text{val}},01}\left(\hat{h}_{k,s_{\text{train}}}\right)$. The subscript is "$k$" at this point and not "$\hat{k}$" because the $\hat{k}$ refers to the value of k for which the validation error $\hat{R}_{s_{\text{val}},01}\left(\hat{h}_{k,s_{\text{train}}}\right)$ was the smallest over all values of k considered (marked with the yellow star on the corresponding graph). The point here is that once we get to the step where the complexity parameter k is being chosen to minimize the empirical risk $\hat{R}_{s_{\text{val}},01}\left(\hat{h}_{k,s_{\text{train}}}\right)$ on the validation portion s_{val} of the sample data set s, we have already used s_{train} to select each of the $\hat{h}_{k,s_{\text{train}}}$.

7.6 Why the validation set must be independent of the training set

We can apply the bounds that we have for a fixed hypothesis (via Hoeffding's inequality) to relate the generalization error to the empirical error on some independent training set (here, on the validation portion s_{val} of the sample data set$s$). In order to apply the bound from Hoeffding's inequality to each of the $\hat{h}_{k,s_{\text{train}}}$ (and thereby get a guarantee on how well the performance of $\hat{h}_{k,s_{\text{train}}}$ on s_{val} will serve as a stand-in for the performance of $\hat{h}_{k,s_{\text{train}}}$ on the source distribution), we must consider Hoeffding's inequality for a set of data that is *independent* from the data (specifically s_{train}) that was used to select the hypothesis $\hat{h}_{k,s_{\text{train}}}$ [20].

[20] Refer to the motivation for why we need the union bound to see a previous case where this independence was important.

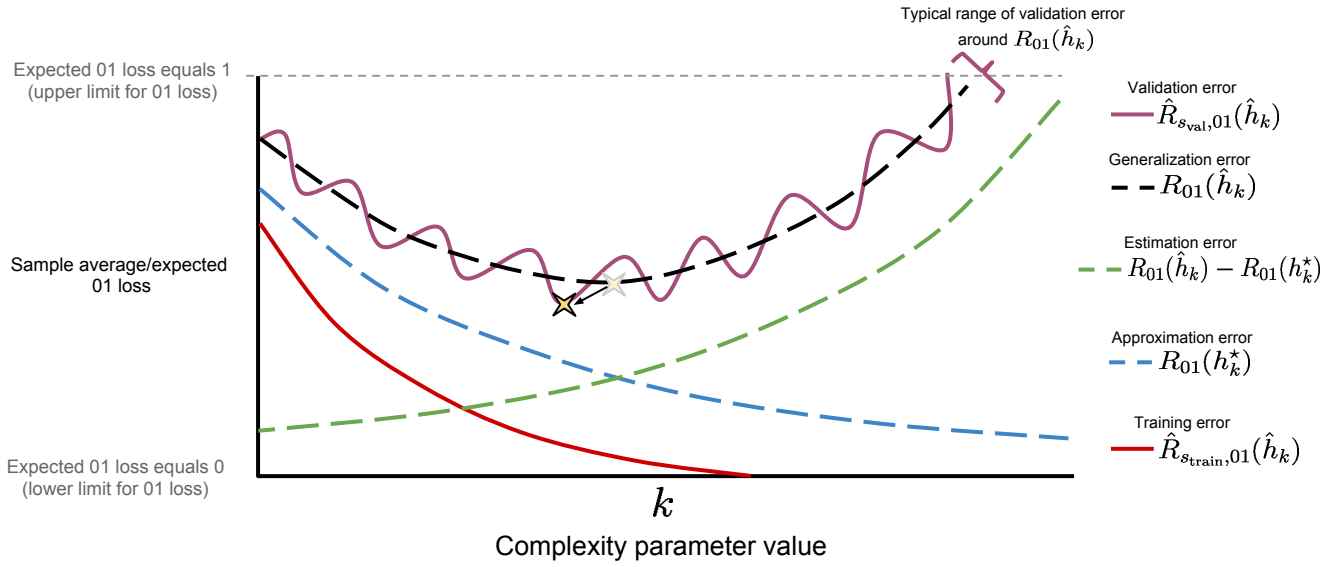

Figure 7.1: Qualitative plot now showing validation error. For a fixed sample set of a specific size m, and shown as a function of the now-continuous complexity parameter k. Dashed lines indicate unobservable/unmeasurable quantities. Included: Approximation error $R_{01}\left(h_k^\star\right)$; training error $\hat{R}_{s_{\text{train}},01}\left(\hat{h}_k\right)$ (the empirical error of the empirical risk minimizer); estimation error $R_{01}\left(\hat{h}_k\right) - R_{01}\left(h_k^\star\right)$; generalization error $R_{01}\left(\hat{h}_k\right)$ of the empirical risk minimizer; validation error $\hat{R}_{s_{\text{val}},01}\left(\hat{h}_{k,s_{\text{train}}}\right)$. Each quantity is displayed as a function of the k parameter indexing increasingly complex, inclusive hypothesis classes $\mathcal{H}_k$. The validation error is drawn "squiggly" to emphasize that it varies around its expected value (the generalization error). The grayed-out yellow "X" marks the performance of the hypothesis we would *like* to be able to identify during learning: the hypothesis for which the generalization error (*i.e.* the future performance) is minimized. The validation process selects the hypothesis whose performance is indicated with the non-grayed yellow "X".

In the strictest interpretation of this need (for an independent evaluation set in order to apply Hoeffding's inequality to the hypotheses $\hat{h}_{k,s_{\text{train}}}$, we have to *first* decide on h (in our case, by evaluating its performance on s_{train}) and only then draw a random validation set from the source distribution[21]Although we would like to have this conceptual process of "first draw s_{train} and only after using s_{train} should you draw s_{val} from the source distribution" actually be the case, in practice we have the entire sample data set s at the beginning. This is not really a problem, as you can still think of the process of setting aside s_{val} as conceptually "drawing s_{val} from the source distribution". Thus, you get the entire sample data set s at the start. You set aside s_{val}. You then use s_{train} to ERM-select $\hat{h}_{k,s_{\text{train}}}$ from each $\mathcal{H}_k$ (this is *conceptually* prior to drawing s_{val}). Once you have used s_{train} to ERM-select hypotheses $\hat{h}_{k,s_{\text{train}}}$ from each $\mathcal{H}_k$, only then do you return to the independent validation set s_{val}. As long as the validation set is not used in any way in the selection of the $\hat{h}_{k,s_{\text{train}}}$, the conceptual desire for a subsequently drawn validation set is satisfied by simply not using the validation set. Again: You are not allowed to touch the validation set when doing this empirical risk minimization task in which you select the hypotheses $\hat{h}_{k,s_{\text{train}}}$ for each of the $\mathcal{H}_k$ being considered.[22]

7.7 Being even more careful: Keep another holdout test set for final evaluation of expected performance of the validation-set selected predictor

If you want to get an accurate estimate of the source distribution performance $R_{01}\left(\hat{h}_{\hat{k},\{s_{\text{train}},s_{\text{val}}\}}\right)$ of the hypothesis $\hat{h}_{\hat{k},\{s_{\text{train}},s_{\text{val}}\}}$, you will need to compute that estimate on yet another separate holdout set; we can call this the test set s_{test}. We would then use $\hat{R}_{s_{\text{test}},01}\left(\hat{h}_{\hat{k},\{s_{\text{train}},s_{\text{val}}\}}\right)$ as our estimate of $R_{01}\left(\hat{h}_{\hat{k},\{s_{\text{train}},s_{\text{val}}\}}\right)$; that is, we use the average 01 loss on s_{test} to estimate the generalization error of the validation-process-selected hypothesis and k value, $\hat{h}_{\hat{k}}$. In this setting we would have $s = \{s_{\text{train}}, s_{\text{val}}, s_{\text{test}}\}$; we would partition our original sample data set into three independent parts, and only used the third part, here called the test set, in order to get an *independent* estimate $\hat{R}_{s_{\text{test}},01}\left(\hat{h}_{\hat{k},\{s_{\text{train}},s_{\text{val}}\}}\right)$ of the generalization error $R_{01}\left(\hat{h}_{\hat{k},\{s_{\text{train}},s_{\text{val}}\}}\right)$ of the validation-process-selected hypothesis and k value, $\hat{h}_{\hat{k}}$. This approach is statistically quite sound; however, there is another, even more popular approach that is somewhat less statistically sound.

7.8 A practical (but not statistically sound) alternative: Cross-validation

We will only discuss cross-validation briefly in this lecture, but we will provide a short overview (recommended: search "autonlab cv notes" for Andrew Moore's notes). We will use cross-validation as another approach to determine what complexity level k we should choose. Cross-validation begins similarly to our description of the "validation approach" described above: we take the sample data set s and randomly separate it into a training set and a validation set. Typically the training set is much larger than the validation set. However, instead of only doing this process once (of splitting the sample data set into training and validation sets), we do this T times. For each of the T train/validate splits, we follow the process describe in the "validation approach" section above.

[21]In our case, "first decide on h" takes the form of getting the $\hat{h}_{k,s_{\text{train}}}$ by using the training portion s_{train} of the sample data set s, and the data set that is independent from the data used to select the $\hat{h}_k$ is the validation portion s_{val} of the data sample s.

[22]If you further wanted to ensure *simultaneously* that the performance of the $\hat{h}_{k,s_{\text{train}}}$s on the validation set s_{val} is actually indicative *at the same time* of their performance on the source distribution, you would need to pursue the approach we took in the union bound argument but with the number of different values of k in place of the number of different hypotheses. Since we might typically have 100 values of k, we would have a factor of $\ln 100 \approx 5$ in the corresponding (union-derived) bound. This means that even with a relatively modest validation set size, you could be confident that the validation error estimates $\hat{R}_{s_{\text{val}},01}\left(\hat{h}_{k,s_{\text{train}}}\right)$ are actually good estimates of $R_{01}\left(\hat{h}_{k,s_{\text{train}}}\right)$ for all of the $\hat{h}_{k,s_{\text{train}}}$s at the same time.

For example, we might start out wanting to know how well k does. We train on $s_{\text{train}.t}$ and then validate on $s_{\text{val}.t}$ for each of the $t = 1 \ldots T$ splits of the sample data set. We might find that for the $t = 1$ split that the ERM-selected hypothesis from the k class $\mathcal{H}_1$ has average 01 loss on the $t = 1$ validation set $s_{\text{val}.1}$ of $\hat{R}_{s_{\text{train}.1},01}\left(\hat{h}_{k,s_{\text{train}.1}}\right)$. Likewise, for the $t = 2$ split, we would have validation set performance of $\hat{R}_{s_{\text{val}.2},01}\left(\hat{h}_{k,s_{\text{train}.2}}\right)$. At the end, for our estimate how an ERM-selected hypothesis from class $\mathcal{H}_k$ will perform, we will use the average of all of these T validation set performances: "How good do we think class k is?"$= \frac{1}{T}\sum_{t=1}^{T} \hat{R}_{s_{\text{val}.t},01}\left(\hat{h}_{k,s_{\text{train}.t}}\right)$. We go through that sequence of steps, to eventually average the T validation set performances, for each of the values of k that we are allowing. We then select the k for which this source-distribution performance estimate is minimized.

7.9 A brief discussion of what is problematic about cross-validation

The somewhat strange thing is that it need not be the case that, for instance it is entirely like that $\hat{h}_{k,s_{\text{train}.3}} \neq \hat{h}_{k,s_{\text{train}.4}}$. Each of the T splits is quite likely to select a different hypothesis.

Despite this somewhat strange result, you can work very hard and show that (under some assumptions) you are not going to get worse results using cross-validation than you would have if you simply use a single validation set.

The estimate of "how good class k is", written above as "How good do we think class k is?"$= \frac{1}{T}\sum_{t=1}^{T} \hat{R}_{s_{\text{val}.t},01}\left(\hat{h}_{k,s_{\text{train}.t}}\right)$, is not actually an estimate of expected 01 loss on the source distribution of some particular, fixed hypothesis. Instead, it is the generalization error of the best hypothesis in class k — as judged on the particular $s_{\text{train}.t}, s_{\text{val}.t}$ splits that we randomly made, and so changing as the split changes. For reasons of this sort, the analysis of cross-validation is problematic.

8 A short summing up of our progress thus far: Learning is optimization.

The topic of this course is machine learning and optimization. By this point, it should be clear that there is a strong connection between machine learning and optimization. For a specific example we observe that in Empirical Risk Minimization, learning boils down to solving an optimization problem**:** Minimize the empirical risk. As we saw when we considered the problem of selecting the complexity level k, learning is often more than solving a single optimization problem: instead we solve a whole sequence of related optimization problems.

References

[1] W.Y. Loh. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23, 2011.