Topic 4: Statistical learning

4.1 Supervised machine learning

4.1.1 Machine learning as programming-by-example

There are many clearly defined tasks for which one can easily write a computer program to accomplish, such as sorting a list of numbers or deciding if a graph is connected. However, there are many real-world tasks that would challenge even the most experienced programmer for a variety of reasons, such as the lack of subject-matter experts that know how to solve the task, or the inability of the experts to fully articulate their methods of solution. Spam filtering is an example of such a task. There are many hand-written rules used to catch some forms of spam, but there are always new forms of spam that get around these rules. Furthermore, while a typical email user may be able to recognize which emails are spam, they may be hard pressed to give a precise specification.

(Supervised) machine learning is a data-driven methodology for creating programs to solve ill-defined tasks discussed above. A *(supervised) learning algorithm* (a.k.a. *learner*) finds a program that "generalizes" from a collection of input/output pairs called the *training data*; the hope is that the program provides the "correct" output on inputs that were not present in the training data. The quality of a program produced by a learning algorithm is typically assessed empirically on another collection of input/output pairs called the *test data*.

4.1.2 Statistical framework

A statistical framework for understanding learning algorithms starts by imagining a *population*, which is a probability distribution P over the space $\mathcal{X} \times \mathcal{Y}$ of possible input/output pairs. (As in Section 3.1.1, we also refer to the "output" as the "label"; we focus on the case where $\mathcal{Y}$ is a discrete set, especially $\mathcal{Y} = \{0, 1\}$.) The population represents the distribution of input/label pairs of interest. Consider a program $f \colon \mathcal{X} \to \mathcal{Y}$, which we assume for simplicity is a function, also called a *predictor* or *classifier*. For $(\mathbf{x}, \mathbf{y}) \sim P$, the joint distribution of $(\mathbf{x}, \mathbf{y}, f(\mathbf{x}))$ captures the behavior and performance of f on the population; however, this joint distribution is often too unwieldy to reason about. A drastic simplification is to focus only on $\mathbf{z}_f := \mathbb{1}\{f(\mathbf{x}) \neq \mathbf{y}\}$, which only indicates if f predicts a label different from $\mathbf{y}$ on input $\mathbf{x}$. Since $\mathbf{z}_f$ takes values in $\{0, 1\}$, its distribution is characterized by its expected value:

$$\mathrm{E}[\mathbf{z}_f] = \mathrm{E}\big[\mathbb{1}\{f(\mathbf{x}) \neq \mathbf{y}\}\big] = \Pr(f(\mathbf{x}) \neq \mathbf{y}).$$

This probability is called the *(population) error rate* of f (with respect to P), denoted by $\mathsf{err}(f; P)$. (We may drop the argument P when it can be inferred from context.) We focus on error rate as the primary performance measure in this statistical framework, although many other measures are also used in practice.

If the population P is "known", then we could imagine using this knowledge about P to construct a classifier f that has the smallest possible error rate. But knowledge of P is typically not known to the required level of detail. Machine learning instead relies on training data under an assumption that the data is representative of the population in some sense. A common starting point is to assume that the set of training data is an independent and identically distributed (iid) sample from

P; we refer to this as the *iid training data assumption*. Under such an assumption, the training data is random, and hence the classifier returned by the learning algorithm is also random. On account of this randomness and the reasonable desire to discount situations where the (random) training data is highly atypical, we will focus on "typical" properties of the (random) classifier. For example, if $\mathbf{f}$ is the classifier returned by a learning algorithm under the iid training data assumption, we may be interested in the expected error rate

$$\mathrm{E}\big[\mathsf{err}(\mathbf{f}; P)\big].$$

Alternatively, we may be interested in the 95th percentile of $\mathsf{err}(\mathbf{f}; P)$, i.e., the smallest ϵ such that

$$\Pr(\mathsf{err}(\mathbf{f}; P) \leq \epsilon) \geq 0.95.$$

So far, we have not made any assumption about P itself. However, the term "error rate" suggests that if $f(\mathbf{x}) \neq \mathbf{y}$ for $(\mathbf{x}, \mathbf{y}) \sim P$, then f has provided an "incorrect" label on input $\mathbf{x}$. But it is possible that both 0 and 1 have positive P-mass under the conditional distribution of $\mathbf{y}$ given $\mathbf{x}$. For example, suppose P corresponds to how an actual population of people assign spam/not-spam labels to emails, and for a given email message x, 60% of the people declare it to be spam, and 40% of the people declare it to be not-spam. To minimize error rate, the best that any classifier can do is to label the message x as spam, even though this goes against 40% of the conditional distribution of $\mathbf{y}$ when $\mathbf{x} = x$. In general, the *Bayes (optimal) classifier* $f^\star \colon \mathcal{X} \to \mathcal{Y}$, which assigns the label

$$f^\star(x) := \arg\max_{y \in \mathcal{Y}} \Pr(\mathbf{y} = y \mid \mathbf{x} = x)$$

to the input $x \in \mathcal{X}$ (breaking ties arbitrarily), has the smallest possible error rate called the *Bayes error rate*:

$$\mathsf{err}(f^\star; P) = \Pr(f^\star(\mathbf{x}) \neq \mathbf{y}) = \mathrm{E}\left[1 - \max_{y \in \mathcal{Y}} \Pr(\mathbf{y} = y \mid \mathbf{x})\right].$$

When $\mathcal{Y} = \{0, 1\}$, the Bayes classifier and error rate simplify to

$$f^\star(x) = \begin{cases} 1 & \text{if } \Pr(\mathbf{y} = 1 \mid \mathbf{x} = x) > 1/2, \\ 0 & \text{otherwise;} \end{cases}$$

and

$$\mathsf{err}(f^\star; P) = \mathrm{E}\Big[\min\{\Pr(\mathbf{y} = 0 \mid \mathbf{x}), \Pr(\mathbf{y} = 1 \mid \mathbf{x})\}\Big].$$

A different interpretation is to regard each input $x \in \mathcal{X}$ as having a "correct" label $c(x)$ given by a "ground truth" function $c^\star \colon \mathcal{X} \to \mathcal{Y}$ from some class of functions $\mathcal{C}$. If $\Pr(\mathbf{y} \neq c^\star(x) \mid \mathbf{x} = x) > 0$, there is some positive probability that the actual label $\mathbf{y}$ one observes for the input $\mathbf{x} = x$ in the population is incorrect. Such errors can be regarded as *label noise*. The assumption of no label noise is called the *$\mathcal{C}$-realizability assumption*. More precisely, it is the assumption on P with respect to $\mathcal{C}$ that there exists some $c^\star \in \mathcal{C}$ with $\Pr(\mathbf{y} \neq c^\star(\mathbf{x})) = 0$. Under the $\mathcal{C}$-realizability assumption:

- the Bayes classifier $f^\star$ and the ground truth function $c^\star$ do not disagree:

$$\Pr(f^\star(\mathbf{x}) \neq c^\star(\mathbf{x})) = 0;$$

- the Bayes error rate is zero;
- the error rate of any classifier $f\colon \mathcal{X} \to \mathcal{Y}$ satisfies

$$\mathsf{err}(f; P) = \Pr(f(\mathbf{x}) \neq c^\star(\mathbf{x})).$$

Also, we may equivalently specify P by the pair $(P_{\mathbf{x}}, c^\star)$, where $P_{\mathbf{x}}$ is the marginal distribution of $\mathbf{x}$.

4.1.3 Probably Approximately Correct criterion

The *Probably Approximately Correct (PAC)* criterion, based on the seminal computational learning framework of Valiant (1984)[1], is a quantitative success criterion for learning algorithms. Fix parameters $\epsilon, \delta \in (0, 1)$. A learning algorithm *$(\epsilon, \delta)$-PAC learns* a concept class $\mathcal{C} \subseteq \mathcal{Y}^{\mathcal{X}}$ if, for any distribution P over $\mathcal{X} \times \mathcal{Y}$ satisfying the $\mathcal{C}$-realizability assumption, under the iid training data assumption, the output of the learning algorithm $\mathbf{f}$ satisfies

$$\Pr(\mathsf{err}(\mathbf{f}; P) \leq \epsilon) \geq 1 - \delta.$$

Most learning algorithms may use any number of training data, and as a general rule, the more training data the better. The number of training data required of a learning algorithm to (ϵ, δ)-PAC learn a concept class $\mathcal{C}$ is called the *sample complexity*. The sample complexity may depend on properties of the concept class $\mathcal{C}$, the parameters ϵ and δ, and possibly other quantities.[2] A learning algorithm is said to be a *PAC learning algorithm for $\mathcal{C}$* if it can (ϵ, δ)-PAC learn $\mathcal{C}$ for any $\epsilon, \delta \in (0, 1)$, provided enough training data (e.g., with sample complexity depending on $1/\epsilon$, $1/\delta$, etc.).

It is also useful to consider a learning criterion that asks for the error rate of the output $\mathbf{f}$ of the learning algorithm to be at most ϵ in expectation over the random choice of the training data:

$$\mathrm{E}[\mathsf{err}(\mathbf{f}; P)] \leq \epsilon.$$

If this holds, we will say that the learning algorithm *ϵ-learns $\mathcal{C}$ in expectation*.[3]

4.1.4 Role of test data

The statistical framework may also be extended to describe the role of test data with the additional assumptions that (1) like the training data, the test data is also an iid sample from P, and (2) the test data is independent of the training data. The first part of the assumption allows one to estimate the error rate of any fixed classifier $f\colon \mathcal{X} \to \mathcal{Y}$ with estimation error decreasing as a function of the number of test data, simply using by the fraction of the test data on which f provides a different label. This simple estimate is called the *test error rate*. The second part of the assumption ensures that such an estimate is also applicable to the classifier produced by a learning algorithm that uses the training data.

[1] The term "Probably Approximately Correct" was first used in this context by Angluin (1988).

[2] "Efficient PAC learning" puts further requirements on the computational complexity of the learning algorithm; for now, we defer computational considerations until later.

[3] This success criterion has been previously used in the probabilistic prediction model of Haussler et al. (1994).

4.2 PAC learning from online concept learning

4.2.1 Batch versus online learning

The statistical framework for learning described in Section 4.1 is also called "batch learning", because the learner is applied to an entire batch of training data to produce a classifier, and only after that is the classifier used to make predictions on other inputs. In contrast, in "online learning", learning and prediction are always ongoing. In particular, in each round t of the online concept learning scenario, the labeled example (x_t, y_t) is used both for evaluating the learner (i.e., we assess if $\hat{y}_t \neq y_t$), and also (possibly) used to update information being tracked by the learner that may inform predictions in future rounds (i.e., "learning"). For simplicity, we restrict attention to the case where $\mathcal{Y} = \{0, 1\}$.

4.2.2 From online to batch learning

It turns out that "batch learning" is no harder than "online learning" in the following sense. If there is learning algorithm for a concept class $\mathcal{C}$ in the online concept learning scenario, then that algorithm can be used to construct a second learning algorithm for $\mathcal{C}$ for the statistical setting (Algorithm 4.1). This latter algorithm (ϵ, δ)-PAC learns $\mathcal{C}$ with a sample complexity that is at most linear in the mistake bound of the first algorithm. The process of deriving a batch learning algorithm from an online learning algorithm is called an *online-to-batch conversion*; this particular conversion method is due to Helmbold and Warmuth (1995).

Algorithm 4.1: Batch learning algorithm B based on online learning algorithm A (for online concept learning scenario over domain $\mathcal{X}$).

Input: training data $((x_1, y_1), \ldots, (x_T, y_T)) \in (\mathcal{X} \times \{0, 1\})^T$.

- Simulate A in the online concept learning scenario using example (x_t, y_t) in round t, for a total of T rounds.
  - Let $f_t \colon \mathcal{X} \to \{0, 1\}$ denote the predictor used by A in round t (determined by the internal state of A at the start of round t). Note that f_{T+1} is well-defined even though we do not simulate round $T + 1$.
- Return $f \colon \mathcal{X} \to \{0, 1\}$, where for all $x \in \mathcal{X}$,

$$f(x) := \begin{cases} 1 & \text{if } |\{t \in [T+1] \mid f_t(x) = 1\}| > (T+1)/2, \\ 0 & \text{otherwise.} \end{cases}$$

The predictor f returned by Algorithm 4.1 can be regarded as a majority vote classifier over $f_1, \ldots, f_{T+1}$. Learning algorithms like this that combine many different predictors to produce a single predictor are called *ensemble methods* or *aggregation methods*. One downside of such methods is that computing a prediction with the predictor requires evaluating many other predictors.

Some relief comes when Algorithm 4.1 is used with an online learning algorithm that is *conservative*, meaning that it only makes an "update" to its internal state when a mistake is made. In such a case, the number of different predictors among $f_1, \ldots, f_{T+1}$ is at most $M + 1$, so f can be regarded as a weighted majority vote over at most $M + 1$ different predictors.

Theorem 4.1. *Suppose there is a learning algorithm A and an integer $M \geq 0$ such that, in the online concept learning scenario with the $\mathcal{C}$-realizability assumption, A makes at most M mistakes. Then there is a learning algorithm B (Algorithm 4.1) such that B ϵ-learns $\mathcal{C}$ in expectation for*

$$\epsilon = \frac{2M}{T+1}$$

where T is the number of training data.

Corollary 4.1. *In the setup of Theorem 4.1, for any $\epsilon, \delta \in (0,1)$, if the number of training data is at least $2M/(\epsilon\delta)$, then B (ϵ,δ)-PAC learns $\mathcal{C}$.*

Proof of Theorem 4.1. Let P be a probability distribution over $\mathcal{X} \times \{0,1\}$, and suppose

$$((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_{T+1}, \mathbf{y}_{T+1})) \sim P^{T+1}$$

is an iid sample from P of size $T+1$. For each t, define $\mathbf{S}_t := ((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_t, \mathbf{y}_t))$ to be the first t examples. The training data is $\mathbf{S}_T$, although we will also consider $(\mathbf{x}_{T+1}, \mathbf{y}_{T+1})$ in the analysis.

Given $\mathbf{S}_T$ as input, the algorithm B, shown in Algorithm 4.1, simulates A in the online concept learning scenario for T rounds using the random example $(\mathbf{x}_t, \mathbf{y}_t)$ in round t. Let $\mathbf{f}_t$ denote the predictor to be used by A in round t; this predictor depends (only) on the first $t-1$ training examples $\mathbf{S}_{t-1}$, so it is generally a random variable (except for $t=1$). Since $(\mathbf{x}_t, \mathbf{y}_t)$ and $\mathbf{S}_{t-1}$ are independent, so are $(\mathbf{x}_t, \mathbf{y}_t)$ and $\mathbf{f}_t$. Moreover, since $(\mathbf{x}_t, \mathbf{y}_t) \sim P$, we have

$$\mathsf{err}(\mathbf{f}_t; P) = \Pr(\mathbf{f}_t(\mathbf{x}_t) \neq \mathbf{y}_t \mid \mathbf{S}_{t-1}) = \mathrm{E}[\mathbb{1}\{\mathbf{f}_t(\mathbf{x}_t) \neq \mathbf{y}_t\} \mid \mathbf{S}_{t-1}].$$

Note that $\mathsf{err}(\mathbf{f}_1; P), \ldots, \mathsf{err}(\mathbf{f}_{T+1}; P)$ are random variables; the sum of their expected values is bounded as follows:

$$\begin{aligned}
\sum_{t=1}^{T+1} \mathrm{E}[\mathsf{err}(\mathbf{f}_t; P)] &= \sum_{t=1}^{T+1} \mathrm{E}\big[\Pr(\mathbf{f}_t(\mathbf{x}_t) \neq \mathbf{y}_t \mid \mathbf{S}_{t-1})\big] \\
&= \sum_{t=1}^{T+1} \mathrm{E}\big[\mathrm{E}[\mathbb{1}\{\mathbf{f}_t(\mathbf{x}_t) \neq \mathbf{y}_t\} \mid \mathbf{S}_{t-1}]\big] \\
&= \sum_{t=1}^{T+1} \mathrm{E}[\mathbb{1}\{\mathbf{f}_t(\mathbf{x}_t) \neq \mathbf{y}_t\}] \\
&= \mathrm{E}\left[\sum_{t=1}^{T+1} \mathbb{1}\{\mathbf{f}_t(\mathbf{x}_t) \neq \mathbf{y}_t\}\right] \\
&\leq M, \qquad (4.1)
\end{aligned}$$

where the inequality follows from the fact that A has mistake bound M (even when including a hypothetical round $T+1$).

Now let $\mathbf{f} \colon \mathcal{X} \to \{0,1\}$ be the function returned by the algorithm B. Consider the prediction of $\mathbf{f}$ on $\mathbf{x}_{T+1}$: The definition of $\mathbf{f}$ implies the inequality

$$\mathbb{1}\{\mathbf{f}(\mathbf{x}_{T+1}) \neq \mathbf{y}_{T+1}\} \leq \frac{2}{T+1} \sum_{t=1}^{T+1} \mathbb{1}\{\mathbf{f}_t(\mathbf{x}_{T+1}) \neq \mathbf{y}_{T+1}\},$$

and therefore

$$\mathrm{E}[\mathbb{1}\{\mathbf{f}(\mathbf{x}_{T+1}) \neq \mathbf{y}_{T+1}\}] \leq \mathrm{E}\left[\frac{2}{T+1}\sum_{t=1}^{T+1}\mathbb{1}\{\mathbf{f}_t(\mathbf{x}_{T+1}) \neq \mathbf{y}_{T+1}\}\right].$$

Since $(\mathbf{x}_{T+1}, \mathbf{y}_{T+1}) \sim P$ and $(\mathbf{x}_{T+1}, \mathbf{y}_{T+1})$ is independent of $\mathbf{f}_1, \ldots, \mathbf{f}_{T+1}, \mathbf{f}$, the left-hand side is

$$\mathrm{E}[\mathbb{1}\{\mathbf{f}(\mathbf{x}_{T+1}) \neq \mathbf{y}_{T+1}\}] = \mathrm{E}\big[\mathrm{E}[\mathbb{1}\{\mathbf{f}(\mathbf{x}_{T+1}) \neq \mathbf{y}_{T+1}\} \mid \mathbf{S}_T]\big] = \mathrm{E}\big[\mathsf{err}(\mathbf{f}; P)\big],$$

and the right-hand side is

$$\begin{aligned}
\mathrm{E}\left[\frac{2}{T+1}\sum_{t=1}^{T+1}\mathbb{1}\{\mathbf{f}_t(\mathbf{x}_{T+1}) \neq \mathbf{y}_{T+1}\}\right] &= \frac{2}{T+1}\sum_{t=1}^{T+1}\mathrm{E}[\mathbb{1}\{\mathbf{f}_t(\mathbf{x}_{T+1}) \neq \mathbf{y}_{T+1}\}] \\
&= \frac{2}{T+1}\sum_{t=1}^{T+1}\mathrm{E}\big[\mathrm{E}[\mathbb{1}\{\mathbf{f}_t(\mathbf{x}_{T+1}) \neq \mathbf{y}_{T+1}\} \mid \mathbf{S}_T]\big] \\
&= \frac{2}{T+1}\sum_{t=1}^{T+1}\mathrm{E}\big[\mathsf{err}(\mathbf{f}_t; P)\big] \\
&\leq \frac{2M}{T+1},
\end{aligned}$$

with the second inequality coming from (4.1). □

To get Corollary 4.1 from Theorem 4.1, we apply *Markov's inequality* (Proposition 4.1).

Proposition 4.1 (Markov's inequality)**.** *For any non-negative real-valued random variable* $\mathbf{z}$ *and any* $t > 0$*,*

$$\Pr(\mathbf{z} \geq t) \leq \frac{\mathrm{E}[\mathbf{z}]}{t}.$$

Proof. We clearly have $\mathbf{z} \geq \mathbb{1}\{\mathbf{z} \geq t\} \cdot t$ since $\mathbf{z}$ is non-negative and $t > 0$. Now take expectation of both sides: by linearity of expectation,

$$\begin{aligned}
\mathrm{E}[\mathbf{z}] &\geq \mathrm{E}[\mathbb{1}\{\mathbf{z} \geq t\}] \cdot t \\
&= \Pr(\mathbf{z} \geq t) \cdot t.
\end{aligned}$$

□

Proof of Corollary 4.1. Let $\mathbf{f}$ be the output of algorithm B; then $\mathsf{err}(\mathbf{f}; P)$ is a non-negative random variable. By Proposition 4.1, Theorem 4.1, and the assumption $T \geq 2M/(\epsilon\delta)$,

$$\Pr(\mathsf{err}(\mathbf{f}; P) \geq \epsilon) \leq \frac{\mathrm{E}\big[\mathsf{err}(\mathbf{f}; P)\big]}{\epsilon} \leq \frac{2M}{T+1} \cdot \frac{1}{\epsilon} < \delta.$$

□

The sample complexity of Algorithm 4.1 has a suboptimal linear dependence on $1/\delta$. There is a very simple alternative online-to-batch conversion that replaces the $1/\delta$ with $\log(1/\delta)$:

1. Repeat Algorithm 4.1 $k := \lceil \log_2(2/\delta) \rceil$ times on independent training data sets, all of size $\lceil 8M/\epsilon \rceil$; let $\mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(k)}$ be the resulting predictors.

    - By Corollary 4.1 and independence, the probability that $\mathsf{err}(\mathbf{f}^{(i)}; P) > \epsilon/2$ for all $i \in [k]$ is at most 2^{-k}, which is at most $\delta/2$ by the choice of k.

- Therefore, with probability at least $1 - \delta/2$, there exists some $i \in [k]$ such that

$$\mathsf{err}(\mathbf{f}^{(i)}; P) \leq \epsilon/2.$$

2. Consider yet another independent training data set to serve as "test data"; return the $\mathbf{f}^{(i)}$ that makes the fewest mistakes on this data set.

   - The collection of data used in this way is often called *validation data*.

The second step has been left underspecified. How many validation data should be used? Even if the first step succeeds—meaning that some predictor has error rate at most $\epsilon/2$—could it be that the second step returns a predictor with error rate more than ϵ? (Why did we purposely leave some "slack" in both ϵ and δ in the first step?) Some probabilistic tools similar to Markov's inequality will be helpful for answering these questions.[4]

4.3 Detour: Tools to bound the probability of large deviations

4.3.1 Chebyshev's inequality

Markov's inequality is often used with non-negative functions of random variables. A simple demonstration of this use is *Chebyshev's inequality*.

Proposition 4.2 (Chebyshev's inequality)**.** *For any real-valued random variable* $\mathbf{z}$ *and any* $t > 0$*,*

$$\Pr(|\mathbf{z} - \mathrm{E}(\mathbf{z})| \geq t) \leq \frac{\mathrm{var}(\mathbf{z})}{t^2}.$$

Proof. Apply Markov's inequality (Proposition 4.1) to the random variable $(\mathbf{z} - \mathrm{E}(\mathbf{z}))^2$. □

We could have instead applied Markov's inequality to $|\mathbf{z} - \mathrm{E}(\mathbf{z})|$, but it is often easier to analyze $\mathrm{var}(\mathbf{z}) = \mathrm{E}[(\mathbf{z} - \mathrm{E}(\mathbf{z}))^2]$ than it is to analyze $\mathrm{E}[|\mathbf{z} - \mathrm{E}(\mathbf{z})|]$.

Example 4.1. Let $\mathbf{z} = \sum_{i=1}^n \mathbf{x}_i$ be the sum of n iid $\{0,1\}$-valued (i.e., Bernoulli) random variables, with $\mathbf{x}_i \sim \mathrm{Bernoulli}(p)$ for some $p \in [0,1]$.[5] Then

$$\mathrm{E}(\mathbf{z}) = \sum_{i=1}^n \mathrm{E}(\mathbf{x}_i) = np, \qquad \mathrm{var}(\mathbf{z}) = \sum_{i=1}^n \mathrm{var}(\boldsymbol{\sigma}_i) = np(1-p) \leq \frac{n}{4}.$$

In fact, we have only used the pairwise independence of $\mathbf{x}_1, \ldots, \mathbf{x}_n$ to compute $\mathrm{var}(\mathbf{z})$. So by Chebyshev's inequality, for any $c > 0$,

$$\Pr\big(|\mathbf{z} - \mathrm{E}(\mathbf{z})| > c\sqrt{n}\big) \leq \frac{1}{4c^2}.$$

By equating the right-hand side to $\delta \in (0,1)$ and solving for c, we find that $|\mathbf{z} - \mathrm{E}(\mathbf{z})| \leq O(\sqrt{n/\delta})$ with probability at least $1 - \delta$. ◊

[4] The sample complexity of the suggested method will turn out to be at most $M \log(1/\delta)/\epsilon$ (up to constants). There is another online-to-batch conversion due to Angluin (1988) that has sample complexity at most $M \log(M/\delta)/\epsilon$, which is a bit worse, but has the advantage of returning one of the f_t produced by algorithm A, as opposed to an ensemble predictor. Yet another method due to Littlestone (1989) has an improved sample complexity of $(M + \log(1/\delta))/\epsilon$.

[5] The distribution of $\mathbf{z}$ is the *binomial distribution* with n trials and success probability p, typically written $\mathbf{z} \sim \mathrm{Binomial}(n, p)$.

We can also apply Markov's inequality with higher and higher powers of $|\mathbf{z} - \mathrm{E}(\mathbf{z})|$, which requires analysis of correspondingly higher moments (i.e., $\mathrm{E}[|\mathbf{z} - \mathrm{E}(\mathbf{z})|^k]$ for $k = 3, 4, \dots$).

Example 4.2. Let $\mathbf{z}$ be the random variable from Example 4.1. Then

$$\mathrm{E}\big((\mathbf{z} - \mathrm{E}(\mathbf{z}))^4\big) = \sum_{(i,j,k,l) \in [n]^4} \mathrm{E}((\mathbf{x}_i - p)(\mathbf{x}_j - p)(\mathbf{x}_k - p)(\mathbf{x}_l - p)).$$

Each expected value is non-zero only if the indices i, j, k, l are either identical, or if they form two distinct matched pairs (e.g., $i = j \neq k = l$). In the former case, the value is $p(1-p)^4 + (1-p)p^4$, which is at most $1/12$ (achieved when $p = (1 - 1/\sqrt{3})/2$). In the latter case, the value is $p^2(1-p)^2$, which is at most $1/16$ (achieved when $p = 1/2$). (This computation uses four-wise independence of $\mathbf{x}_1, \dots, \mathbf{x}_n$.) The number of $(i, j, k, l) \in [n]^4$ of the first type is n; the number of $(i, j, k, l) \in [n]^4$ of the second type is $\binom{4}{2}\binom{n}{2}$. Therefore

$$\mathrm{E}\big((\mathbf{z} - \mathrm{E}(\mathbf{z}))^4\big) \leq n \cdot \frac{1}{12} + \binom{4}{2}\binom{n}{2} \cdot \frac{1}{16} = \frac{n(9n-5)}{48}.$$

So a fourth-moment version of Chebyshev's inequality implies

$$\Pr\big(|\mathbf{z} - \mathrm{E}(\mathbf{z})| > c\sqrt{n}\big) \leq \frac{9 - 5/n}{c^4}.$$

For large enough c, this is an improvement over what was obtained using the standard version of Chebyshev's inequality. Indeed, again setting the right-hand side to δ, we find $|\mathbf{z} - \mathrm{E}(\mathbf{z})| \leq O(\sqrt{n}/\delta^{1/4})$ with probability at least $1 - \delta$. $\lozenge$

The bounds obtained in Examples 4.1 and 4.2 for the binomial distribution are unsatisfying because, by the Central Limit Theorem, it is known that as $n \to \infty$, the Binomial(n, p) distribution converges (in a certain precise sense) to a normal distribution, and deviations of c multiples of the standard deviation from the mean have probability that go to zero like $\exp(-c^2)$ in a normal distribution (as opposed to only $1/\operatorname{poly}(c)$). The reason is that the arguments in these examples only exploit low-order independence of the summands, rather than full mutual independence.

4.3.2 Logarithmic moment generating function and the Cramér-Chernoff method

We can analytically use all moments simultaneously through the *logarithmic moment generating function (log mgf; a.k.a. cumulant generating function)* for $\mathbf{z}$:

$$\mathsf{K}_{\mathbf{z}}(\lambda) := \ln \mathrm{E}[\exp(\lambda \mathbf{z})].$$

Note that $\mathsf{K}_{\mathbf{z}}(0) = 0$. The function might not be finite for $\lambda \neq 0$, although if it is finite at a negative input $\lambda_1 < 0$ and a positive input $\lambda_2 > 0$, then it is finite at all in-between inputs $\lambda \in [\lambda_1, \lambda_2]$ and also infinitely-differentiable on (λ_1, λ_2). We will assume henceforth that $\mathsf{K}_{\mathbf{z}}$ is finite on such an open interval containing 0.

An important property of the log mgf is that it plays very nicely with sums of independent random variables: if $\mathbf{z} = \mathbf{x}_1 + \cdots + \mathbf{x}_n$ where $\mathbf{x}_1, \dots, \mathbf{x}_n$ are independent, then

$$\mathsf{K}_{\mathbf{z}}(\lambda) = \mathsf{K}_{\mathbf{x}_1}(\lambda) + \cdots + \mathsf{K}_{\mathbf{x}_n}(\lambda).$$

Distribution of $\mathbf{z}$	$\mathsf{K}_{\mathbf{z}}(\lambda)$	Taylor expansion of $\mathsf{K}_{\mathbf{z}}(\lambda)$
Normal(μ, σ^2)	$\mu\lambda + \frac{\sigma^2}{2}\lambda^2$	$\mu\lambda + \frac{\sigma^2}{2}\lambda^2$
Bernoulli(p)	$\ln(1 + p(\exp(\lambda) - 1))$	$p\lambda + \frac{p(1-p)}{2}\lambda^2 + \frac{p(1-p)(1-2p)}{6}\lambda^3 + \cdots$
Poisson(μ)	$\mu(\exp(\lambda) - 1)$	$\mu\lambda + \frac{\mu}{2}\lambda^2 + \frac{\mu}{6}\lambda^3 + \cdots$

Table 4.1: Logarithmic moment generating functions for a few well-known distribution families.

This uses the mutual independence of $\mathbf{x}_1, \ldots, \mathbf{x}_n$ (not just k-wise independence for $k < n$).

For many well-known families of distributions, formulas for the log mgf are known. See the first three columns in Table 4.1 for a few examples.

The first terms of the Taylor expansions in Table 4.1 are suggestive of a pattern about $\mathsf{K}'_{\mathbf{z}}(0)$ that turns out to hold in general:

$$\mathsf{K}'_{\mathbf{z}}(0) = \mathrm{E}(\mathbf{z}).$$

Indeed,

$$\mathsf{K}'_{\mathbf{z}}(\lambda) = \frac{\mathrm{E}\big[\mathbf{z}\exp(\lambda\mathbf{z})\big]}{\mathrm{E}\big[\exp(\lambda\mathbf{z})\big]} = \mathrm{E}\big[\mathbf{z}\exp(\lambda\mathbf{z} - \mathsf{K}_{\mathbf{z}}(\lambda))\big] = \mathrm{E}\big[\mathbf{z}_\lambda\big],$$

which can be seen as the mean of a related random variable $\mathbf{z}_\lambda$ whose distribution is the same as that of $\mathbf{z}$ except "tilted" by $z \mapsto \exp(\lambda z - \mathsf{K}_{\mathbf{z}}(\lambda))$ in the following sense: letting F_λ denote the cumulative distribution function (CDF) for $\mathbf{z}_\lambda$ (so F_0 is the CDF for $\mathbf{z}$), we have

$$\frac{\mathrm{d}F_\lambda}{\mathrm{d}z}(z) = \exp(\lambda z - \mathsf{K}_{\mathbf{z}}(\lambda))\frac{\mathrm{d}F_0}{\mathrm{d}z}(z). \tag{4.2}$$

The second derivative of $\mathsf{K}_{\mathbf{z}}$ is

$$\mathsf{K}''_{\mathbf{z}}(\lambda) = \frac{\mathrm{E}\big[\mathbf{z}^2\exp(\lambda\mathbf{z})\big]}{\mathrm{E}\big[\exp(\lambda\mathbf{z})\big]} - \left(\frac{\mathrm{E}\big[\mathbf{z}\exp(\lambda\mathbf{z})\big]}{\mathrm{E}\big[\exp(\lambda\mathbf{z})\big]}\right)^2 = \mathrm{var}(\mathbf{z}_\lambda),$$

and

$$\mathsf{K}''_{\mathbf{z}}(0) = \mathrm{var}(\mathbf{z}).$$

Since variance is always non-negative, this also shows that $\mathsf{K}_{\mathbf{z}}$ is a convex function. (In fact, it is strictly convex as long as $\mathbf{z}$ is not a constant.) Higher-order derivatives of $\mathsf{K}_{\mathbf{z}}$ at zero give higher-order "cumulants" of $\mathbf{z}$.

Proposition 4.3 (Cramér-Chernoff method)**.** *Let $\mathbf{z}$ be a real-valued random variable. If $t > \mathrm{E}(\mathbf{z})$, then*

$$\Pr(\mathbf{z} \geq t) \leq \exp\Big(-\sup_{\lambda>0}\{t\lambda - \mathsf{K}_{\mathbf{z}}(\lambda)\}\Big) = \exp\Big(-\sup_{\lambda\in\mathbb{R}}\{t\lambda - \mathsf{K}_{\mathbf{z}}(\lambda)\}\Big).$$

If $t < \mathrm{E}(\mathbf{z})$, then

$$\Pr(\mathbf{z} \leq t) \leq \exp\Big(-\sup_{\lambda<0}\{t\lambda - \mathsf{K}_{\mathbf{z}}(\lambda)\}\Big) = \exp\Big(-\sup_{\lambda\in\mathbb{R}}\{t\lambda - \mathsf{K}_{\mathbf{z}}(\lambda)\}\Big).$$

Distribution of $\mathbf{z}$	$\mathsf{K}^*_{\mathbf{z}-\mathrm{E}(\mathbf{z})}(\varepsilon)$
Normal(μ, σ^2)	$\frac{\varepsilon^2}{2\sigma^2}$
Bernoulli(p)	$\mathsf{RE}(\text{Bernoulli}(p+\varepsilon), \text{Bernoulli}(p))$ for $\varepsilon \in [-p, 1-p]$
Poisson(μ)	$\mu \cdot \mathsf{h}_{\mathrm{B}}(\frac{\varepsilon}{\mu})$ for $\varepsilon \geq -\mu$, where $\mathsf{h}_{\mathrm{B}}(a) := (1+a)\ln(1+a) - a$ for $a \geq -1$

Table 4.2: Rate functions for the distribution families in Table 4.1 (after centering).

Proof. We start with the bound on $\Pr(\mathbf{z} \geq t)$ for $t > \mathrm{E}(\mathbf{z})$. Consider $\lambda > 0$, and apply Markov's inequality (Proposition 4.1) to $\exp(\lambda \mathbf{z})$:

$$\begin{aligned}\Pr(\mathbf{z} \geq t) &= \Pr(\exp(\lambda \mathbf{z}) \geq \exp(\lambda t)) \\ &\leq \frac{\mathrm{E}\big[\exp(\lambda \mathbf{z})\big]}{\exp(\lambda t)} \\ &= \exp\Big(-(t\lambda - \mathsf{K}_{\mathbf{z}}(\lambda))\Big).\end{aligned}$$

This holds for all $\lambda > 0$, so we may replace $t\lambda - \mathsf{K}_{\mathbf{z}}(\lambda)$ in the bound with $\sup_{\lambda>0}\{t\lambda - \mathsf{K}_{\mathbf{z}}(\lambda)\}$. To see why the supremum in the bound can be taken over all $\lambda \in \mathbb{R}$, as opposed to just $\lambda > 0$, it suffices to show that

$$\sup_{\lambda \leq 0}\{t\lambda - \mathsf{K}_{\mathbf{z}}(\lambda)\} \leq 0 < \sup_{\lambda > 0}\{t\lambda - \mathsf{K}_{\mathbf{z}}(\lambda)\}.$$

This will use the assumption that $t \geq \mathrm{E}(X)$; we leave this to Exercise 4.1.

A completely symmetric argument proves the bound on $\Pr(\mathbf{z} \leq t)$ for $t < \mathrm{E}(\mathbf{z})$. □

Exercise 4.1. Show that, for $t > \mathrm{E}(\mathbf{z})$,

$$\sup_{\lambda \leq 0}\{t\lambda - \mathsf{K}_{\mathbf{z}}(\lambda)\} \leq 0 < \sup_{\lambda > 0}\{t\lambda - \mathsf{K}_{\mathbf{z}}(\lambda)\},$$

and for $t < \mathrm{E}(\mathbf{z})$,

$$\sup_{\lambda \geq 0}\{t\lambda - \mathsf{K}_{\mathbf{z}}(\lambda)\} \leq 0 < \sup_{\lambda < 0}\{t\lambda - \mathsf{K}_{\mathbf{z}}(\lambda)\}.$$ ◊

4.3.3 Rate function and Cramér's theorem

The supremum quantity appearing in Proposition 4.3 is the *Fenchel conjugate* of $\mathsf{K}_{\mathbf{z}}$:

$$\mathsf{K}^*_{\mathbf{z}}(t) := \sup_{\lambda \in \mathbb{R}}\{t\lambda - \mathsf{K}_{\mathbf{z}}(\lambda)\} = \begin{cases} \sup_{\lambda>0}\{t\lambda - \mathsf{K}_{\mathbf{z}}(\lambda)\} & \text{if } t > \mathrm{E}(\mathbf{z}), \\ \sup_{\lambda<0}\{t\lambda - \mathsf{K}_{\mathbf{z}}(\lambda)\} & \text{if } t < \mathrm{E}(\mathbf{z}), \\ 0 & \text{if } t = \mathrm{E}(\mathbf{z}). \end{cases}$$

This function $\mathsf{K}^*_{\mathbf{z}}$—a.k.a. the *rate function* or the *Cramér transform* of $\mathbf{z}$—is convex because it is the supremum of linear functions. Table 4.2 shows the rate functions for the distribution families from Table 4.1.

Exercise 4.2. Prove that $\mathsf{K}^*_{\mathbf{z}}$ is non-increasing on the interval $(-\infty, \mathrm{E}(\mathbf{z}))$ and non-decreasing on the interval $(\mathrm{E}(\mathbf{z}), \infty)$. ◊

When Proposition 4.3 is applied to sums of n iid random variables, we see that the probability of large deviations from the mean is at most a quantity that is exponentially small in n.

Corollary 4.2. *Let $\mathbf{z} := \sum_{i=1}^{n} \mathbf{x}_i$ be the sum of n iid random variables $\mathbf{x}_1, \ldots, \mathbf{x}_n$, with $\mu := \mathrm{E}(\mathbf{x}_1)$. For any $\epsilon > 0$,*

$$\Pr(\mathbf{z} \geq n(\mu + \epsilon)) \leq \exp\bigl(-n\mathsf{K}^*_{\mathbf{x}_1}(\mu + \epsilon)\bigr),$$
$$\Pr(\mathbf{z} \leq n(\mu - \epsilon)) \leq \exp\bigl(-n\mathsf{K}^*_{\mathbf{x}_1}(\mu - \epsilon)\bigr).$$

Proof. By independence of $\mathbf{x}_1, \ldots, \mathbf{x}_n$, for each $\tau \in \{\mu - \epsilon, \mu + \epsilon\}$,

$$\begin{aligned}\mathsf{K}^*_{\mathbf{z}}(n\tau) &= \sup_{\lambda \in \mathbb{R}} \{n\tau\lambda - \mathsf{K}_{\mathbf{z}}(\lambda)\} \\ &= n \sup_{\lambda \in \mathbb{R}} \{\tau\lambda - \mathsf{K}_{\mathbf{x}_1}(\lambda)\} \\ &= n\mathsf{K}^*_{\mathbf{x}_1}(\tau).\end{aligned}$$

The claim now follows from Proposition 4.3. □

Corollary 4.2 is asymptotically tight in the following sense: for all $\epsilon' > \epsilon > 0$,

$$\liminf_{n\to\infty} \frac{1}{n} \ln \Pr(\mathbf{z} \geq n(\mu + \epsilon)) \geq -\mathsf{K}^*_{\mathbf{x}_1}(\mu + \epsilon'),$$
$$\liminf_{n\to\infty} \frac{1}{n} \ln \Pr(\mathbf{z} \leq n(\mu - \epsilon)) \geq -\mathsf{K}^*_{\mathbf{x}_1}(\mu - \epsilon').$$

These asymptotic lower bounds together with Corollary 4.2 are referred to as *Cramér's theorem*.

Example 4.3. Let $\mathbf{z}$ be the random variable from Example 4.1. For any $\varepsilon \in [-p, 1-p]$, we have

$$\mathsf{K}^*_{\mathbf{x}_1}(p + \varepsilon) = \mathsf{RE}(\text{Bernoulli}(p + \varepsilon), \text{Bernoulli}(p)).$$

By an inequality called *Pinsker's inequality* (Lemma 4.2), we have

$$\mathsf{RE}(\text{Bernoulli}(p + \varepsilon), \text{Bernoulli}(p)) \geq 2 \cdot \mathsf{TV}(\text{Bernoulli}(p + \varepsilon), \text{Bernoulli}(p))^2 = 2\varepsilon^2,$$

where $\mathsf{TV}(Q, P)$ the *total variation distance* between Q and P (i.e., the maximum difference in probabilities assigned to any event by Q and P). By Corollary 4.2, for any $\epsilon > 0$ such that $p + \epsilon \leq 1$,

$$\Pr(\mathbf{z} \geq n(p + \epsilon)) \leq \exp\bigl(-2n\epsilon^2\bigr).$$

Setting the right-hand side to $\delta/2$ and solving for ϵ, we find that $\mathbf{z} \leq np + \sqrt{n \ln(2/\delta)/2}$ with probability at least $1 - \delta/2$. Also, for any $\epsilon > 0$ such that $p - \epsilon \geq 0$,

$$\Pr(\mathbf{z} \leq n(p - \epsilon)) \leq \exp\bigl(-2n\epsilon^2\bigr),$$

so we find that $\mathbf{z} \geq np - \sqrt{n \ln(2/\delta)/2}$ with probability at least $1 - \delta/2$. (These inequalities are the same as what we would get using *Hoeffding's inequality*, discussed in Section 4.3.4.) By applying the union bound[6] to the complements of these two events, we conclude $|\mathbf{z} - \mathrm{E}(\mathbf{z})| \leq \sqrt{n \ln(2/\delta)/2}$ with probability at least $1 - \delta$. ◊

[6] If A and B are events from the same probability space, then $\Pr(A \cup B) \leq \Pr(A) + \Pr(B)$. See Section 4.4.1 for more discussion.

4.3.4 Subgaussianity and Hoeffding's inequality

For some random variables, it is easy to obtain an analytically simple upper bound on the log mgf.

Proposition 4.4. *If $\mathsf{K}_{\mathbf{z}}$ is upper-bounded (point-wise) by a function U, then $\mathsf{K}^*_{\mathbf{z}}$ is lower-bounded (point-wise) by U^*, the Fenchel conjugate of U.*

Random variables whose log mgf are upper-bounded by $v\lambda^2/2$ are said to be *subgaussian with variance proxy v (a.k.a. v-subgaussian)*, i.e., the log mgf is upper-bounded by that of $\mathrm{Normal}(0, v)$. It is easy to check that if $\mathbf{z}$ is v-subgaussian, then $\mathsf{K}^*_{\mathbf{z}}(t) \geq t^2/(2v)$.

The sum of independent subgaussian random variables has variance proxy equal to the sum of the variance proxies of the summands. This is compatible with the *2-stability* property of the normal distribution: if $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are independent with $\mathbf{x}_i \sim \mathrm{Normal}(0, v_i)$ for each $i \in [n]$, then their sum $\mathbf{z} = \sum_{i=1}^n \mathbf{x}_i$ satisfies $\mathbf{z} \sim \mathrm{Normal}(0, v_1 + \cdots + v_n)$.

Corollary 4.3. *If $\mathbf{z}$ is the sum of n independent mean-zero random variables $\mathbf{x}_1, \ldots, \mathbf{x}_n$, where $\mathbf{x}_i$ is v_i-subgaussian for each $i \in [n]$, then for any $\epsilon > 0$,*

$$\max\{\Pr(\mathbf{z} \geq n\epsilon), \Pr(\mathbf{z} \leq -n\epsilon)\} \leq \exp\bigl(-n\epsilon^2/(2\bar{v})\bigr),$$

where $\bar{v} := \frac{1}{n}\sum_{i=1}^n v_i$.

Bounded (centered) random variables are subgaussian, with variance proxy equal to the variance of the random variable that is uniformly distributed over the two endpoints of the range.

Lemma 4.1 (Hoeffding's lemma)**.** *If $\mathbf{x}$ is a mean-zero random variable taking values in $[a, b]$ for some $a \leq b$, then $\mathbf{x}$ is $(b-a)^2/4$-subgaussian.*

Proof. First, observe that any random variable $\mathbf{x}'$ taking values in $[a, b]$ has $\mathrm{var}(\mathbf{x}') \leq (b-a)^2/4$. To see this, let $c := (a+b)/2$, and observe that $|\mathbf{x}' - c| \leq |b-a|/2$, and hence $\mathrm{var}(\mathbf{x}') \leq \mathrm{E}[(\mathbf{x}' - c)^2] \leq (b-a)^2/4$.

Next, by Taylor's theorem with explicit remainder,

$$\mathsf{K}_{\mathbf{x}}(\lambda) = \mathsf{K}_{\mathbf{x}}(0) + \mathsf{K}'_{\mathbf{x}}(0)\lambda + \frac{\mathsf{K}''_{\mathbf{x}}(\lambda')}{2}\lambda^2$$

for some λ' between 0 and λ. Recall that

$$\mathsf{K}_{\mathbf{x}}(0) = 0, \qquad \mathsf{K}'_{\mathbf{x}}(0) = \mathrm{E}[\mathbf{x}] = 0, \qquad \mathsf{K}''_{\mathbf{x}}(\lambda') = \mathrm{var}(\mathbf{x}_{\lambda'}),$$

where $\mathbf{x}_{\lambda'}$ is a tilted version of $\mathbf{x}$ with CDF given in (4.2). But $\mathbf{x}_{\lambda'}$ also takes values in $[a, b]$, so $\mathrm{var}(\mathbf{x}_{\lambda'}) \leq (b-a)^2/4$ as argued above. Therefore $\mathsf{K}_{\mathbf{x}}(\lambda) \leq (b-a)^2/8$. □

The use of Hoeffding's lemma (Lemma 4.1) with Corollary 4.3 is called *Hoeffding's inequality*.

Corollary 4.4 (Hoeffding's inequality)**.** *If $\mathbf{z}$ is the sum of independent mean-zero random variables $\mathbf{x}_1, \ldots, \mathbf{x}_n$, where each $\mathbf{x}_i$ takes values in $[a_i, b_i]$ for some $a_i \leq b_i$, then for all $\epsilon > 0$,*

$$\Pr(\mathbf{z} > n\epsilon) \leq \exp\bigl(-2n\epsilon^2/c^2\bigr)$$

where $c^2 := \frac{1}{n}\sum_{i=1}^n (b_i - a_i)^2$.

Hoeffding's lemma is also useful for proving Pinsker's inequality (which was used in Example 4.3).

Lemma 4.2 (Pinsker's inequality for Bernoulli distributions)**.** *Let* $Q = \text{Bernoulli}(q)$ *and* $P = \text{Bernoulli}(p)$*. Then*

$$\mathsf{RE}(Q, P) \geq 2 \cdot \mathsf{TV}(Q, P)^2 = 2(q-p)^2$$

Proof. From Table 4.2, we have

$$\mathsf{RE}(Q, P) = \sup_{\lambda \in \mathbb{R}} \{q\lambda - \ln \mathrm{E}[\exp(\lambda \mathbf{x})]\},$$

where $\mathbf{x} \sim P = \text{Bernoulli}(p)$. By Lemma 4.1, since $\mathbf{x} - p$ takes values in $[-p, 1-p]$,

$$\ln \mathrm{E}[\exp(\lambda \mathbf{x})] = p\lambda + \ln \mathrm{E}[\exp(\lambda(\mathbf{x}-p))] \leq p\lambda + \frac{\lambda^2}{8}, \quad \forall \lambda \in \mathbb{R}.$$

Therefore, for any $\lambda \in \mathbb{R}$,

$$\mathsf{RE}(Q, P) \geq q\lambda - p\lambda - \frac{\lambda^2}{8}.$$

Plugging in $\lambda := 4(q-p)$ (which maximizes the right-hand side) gives

$$\mathsf{RE}(Q, P) \geq 2(q-p)^2. \qquad \square$$

4.3.5 Poisson heuristic and Bennett's inequality

In Examples 4.1 to 4.3, we have treated p as a constant, focusing primarily on the dependence on n and δ. However, the well-known Poisson heuristic for the Binomial distribution suggests that when $p \asymp 1/n$, the distribution of $\mathbf{z}$ (which is Binomial(n, p)) behaves like Poisson(np). This holds more generally for sums of independent bounded random variables, as formalized below.

Lemma 4.3. *If* $\mathbf{z} = \sum_{i=1}^n \mathbf{x}_i$ *is the sum of mean-zero random variables* $\mathbf{x}_1, \ldots, \mathbf{x}_n$*, each taking values in* $(-\infty, b]$ *for some* $b > 0$*, then for all* $\lambda \geq 0$*,*

$$\mathsf{K}_{\mathbf{z}}(\lambda) \leq \frac{v}{b^2}(\exp(\lambda b) - \lambda b - 1)$$

where $v := \sum_{i=1}^n \mathrm{var}(\mathbf{x}_i)$*.*

Proof. Define the function $\phi(a) := (\exp(a) - a - 1)/a^2$ for $a \neq 0$, and $\phi(a) := 1/2$ for $a = 0$. It can be checked that ϕ is an increasing function. Then

$$\begin{aligned}
\mathsf{K}_{\mathbf{x}_i}(\lambda) &= \ln(\mathrm{E}[\exp(\lambda \mathbf{x}_i)]) \\
&\leq \mathrm{E}[\exp(\lambda \mathbf{x}_i)] - 1 && \text{(since } \ln(1+a) \leq a \text{ for } a > -1\text{)} \\
&= \mathrm{E}[\exp(\lambda \mathbf{x}_i) - \lambda \mathbf{x}_i - 1] && \text{(since } \mathrm{E}(\mathbf{x}_i) = 0\text{)} \\
&= \mathrm{E}\big[\phi(\lambda \mathbf{x}_i)\lambda^2 \mathbf{x}_i^2\big] \\
&\leq \lambda^2 \phi(\lambda b)\, \mathrm{E}[\mathbf{x}_i^2] && \text{(since } \mathbf{x}_i \leq b \text{ and } \phi \text{ is increasing)} \\
&= \frac{\mathrm{var}(\mathbf{x}_i)}{b^2}(\exp(\lambda b) - \lambda b - 1).
\end{aligned}$$

By independence,

$$\mathsf{K}_{\mathbf{z}}(\lambda) = \sum_{i=1}^n \mathsf{K}_{\mathbf{x}_i}(\lambda) \leq \frac{v}{b^2}(\exp(\lambda b) - \lambda b - 1). \qquad \square$$

Corollary 4.5 (Bennett's and Bernstein's inequalities)**.** *In the setting of Lemma 4.3, for all $t > 0$,*

$$\Pr(\mathbf{z} \geq t) \leq \exp\left(-\frac{v}{b^2}\mathsf{h}_{\mathrm{B}}\left(\frac{bt}{v}\right)\right) \leq \exp\left(-\frac{t^2}{2(v+bt/3)}\right)$$

where h_{B} is defined in Table 4.2.

Proof. Let $\tilde{\mathbf{z}} \sim \text{Poisson}(v/b^2)$. Then

$$\mathsf{K}_{\tilde{\mathbf{z}}-\mathrm{E}(\tilde{\mathbf{z}})}(\lambda b) = \mathsf{K}_{\tilde{\mathbf{z}}}(\lambda b) - \mathrm{E}(\tilde{\mathbf{z}})\lambda b = \frac{v}{b^2}(\exp(\lambda b) - \lambda b - 1).$$

Therefore by Lemma 4.3,

$$\mathsf{K}_{\mathbf{z}}(\lambda) \leq \mathsf{K}_{\tilde{\mathbf{z}}-\mathrm{E}(\tilde{\mathbf{z}})}(\lambda b), \quad \forall \lambda \geq 0.$$

Combining this inequality with Proposition 4.3 gives, for all $t > 0$,

$$\begin{aligned}
\Pr(\mathbf{z} \geq t) &\leq \exp\left(-\sup_{\lambda>0}\{t\lambda - \mathsf{K}_{\mathbf{z}}(\lambda)\}\right) \\
&\leq \exp\left(-\sup_{\lambda>0}\left\{t\lambda - \mathsf{K}_{\tilde{\mathbf{z}}-\mathrm{E}(\tilde{\mathbf{z}})}(\lambda b)\right\}\right) \\
&= \exp\left(-\sup_{\lambda b>0}\left\{\frac{t}{b}(\lambda b) - \mathsf{K}_{\tilde{\mathbf{z}}-\mathrm{E}(\tilde{\mathbf{z}})}(\lambda b)\right\}\right) = \exp\left(-\frac{v}{b^2}\mathsf{h}_{\mathrm{B}}\left(\frac{t}{v/b}\right)\right)
\end{aligned}$$

where the last step uses the rate function for Poisson(v/b^2) at t/b (see Table 4.2). The second claimed inequality uses the fact that

$$\mathsf{h}_{\mathrm{B}}(a) \geq \frac{a^2}{2(1+a/3)}, \quad \forall a \in \mathbb{R} \tag{4.3}$$

(see Exercise 4.3). Therefore

$$\exp\left(-\frac{v}{b^2}\mathsf{h}_{\mathrm{B}}\left(\frac{bt}{v}\right)\right) \leq \exp\left(-\frac{v}{b^2}\cdot\frac{b^2t^2}{2v^2(1+bt/(3v))}\right) = \exp\left(-\frac{t^2}{2(v+bt/3)}\right). \qquad \square$$

Exercise 4.3. Prove (4.3). ◊

Exercise 4.4. This exercise gives (another) proof of Pinsker's inequality. Let P and Q be probability distributions over a domain $\mathcal{X}$; for simplicity, assume they are discrete and that $Q \ll P$. Use P and Q to also denote their probability mass functions, and define the centered likelihood ratio function $R := Q/P - 1$ (i.e., $R(x) := Q(x)/P(x) - 1$ for $x \in \mathcal{X}$). Let $\mathbf{x} \sim P$.

(a) Show that $\mathrm{E}[\mathsf{h}_{\mathrm{B}}(R(\mathbf{x}))] = \mathsf{RE}(Q, P)$.

(b) Show that

$$\mathrm{E}\left[\frac{R(\mathbf{x})^2}{1+R(\mathbf{x})/3}\right] \geq \frac{\mathrm{E}[|R(\mathbf{x})|]^2}{\mathrm{E}[1+R(\mathbf{x})/3]}.$$

(c) Put the previous parts together with Exercise 4.3 to show

$$\mathsf{RE}(Q, P) \geq \left(\frac{1}{2}\sum_{x\in\mathcal{X}}|Q(x) - P(x)|\right)^2. \qquad \text{◊}$$

Example 4.4. Let $\mathbf{z}$ be the random variable from Example 4.1. We have $\mathbf{x}_i - p \leq 1$ for each $i \in [n]$, so by applying Corollary 4.5 to $\mathbf{z} - np$, we have

$$\Pr(\mathbf{z} - np \geq t) \leq \exp\left(-\frac{t^2}{2(np(1-p) + t/3)}\right).$$

Setting the right-hand side to $\delta/2$ and solving for ϵ, we find that $\mathbf{z} \leq np + \sqrt{2np(1-p)\ln(2/\delta)} + \ln(2/\delta)/3$ with probability at least $1 - \delta/2$.

We can also apply Corollary 4.5 to $np - \mathbf{z} = \sum_{i=1}^{n}(p - \mathbf{x}_i)$, since $p - \mathbf{x}_i \leq 1$ for each $i \in [n]$ as well. This gives

$$\Pr(np - \mathbf{z} \geq t) \leq \exp\left(-\frac{t^2}{2(np(1-p) + t/3)}\right),$$

so we find that $\mathbf{z} \geq np - \sqrt{2np(1-p)\ln(2/\delta)} - \ln(2/\delta)/3$ with probability at least $1 - \delta/2$. By a union bound, we conclude $|\mathbf{z} - \mathrm{E}(\mathbf{z})| \leq \sqrt{2np(1-p)\ln(2/\delta)/2} + \ln(2/\delta)/3$ with probability at least $1 - \delta$. If $p = O(1/n)$, then the bound is $O(\log(1/\delta))$, which does not increase with n. ♢

4.4 PAC learning by minimizing training error rate

4.4.1 Union bound

We have already used the *union bound* in Section 4.3 (in Examples 4.3 and 4.4): for any events A and B in the same probability space, $\Pr(A \cup B) \leq \Pr(A) + \Pr(B)$. (In fact, this generalizes to unions of countably-many events.) If A and B are both small probability events, then $A \cup B$ is an "only-slightly-less-small" probability event. In our applications, A and B will be events that we want to show are unlikely to happen, so a typical use is as follows:

$$\Pr(\text{neither } A \text{ nor } B \text{ holds}) = 1 - \Pr(A \cup B) \geq 1 - (\Pr(A) + \Pr(B)).$$

This generalizes to "exponentially-many" events as long as each event is "exponentially unlikely". Applications of this principle pervade this section.

4.4.2 Learning by finding a consistent hypothesis

We now consider another strategy for learning in the statistical setting: find a classifier that is consistent with the training data. We call this the *Consistent Hypothesis (CH)* algorithm.

Algorithm 4.2: Consistent Hypothesis (CH) algorithm for learning with hypothesis class $\mathcal{H}$.
Input: training data $((x_1, y_1), \ldots, (x_T, y_T)) \in (\mathcal{X} \times \mathcal{Y})^T$.

- Return any $h \in \mathcal{H}$ such that $h(x_t) = y_t$ for all $t \in [T]$, or return $\perp$ if no such h exists.

Here, $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ is a function class called the *hypothesis class* used by CH. For the goal of PAC learning a concept class $\mathcal{C}$, it is typical to use $\mathcal{H}$ that contains $\mathcal{C}$ as a subset. A simple option is $\mathcal{H} = \mathcal{C}$, but sometimes it is computationally easier to use a different or larger class (see, e.g., Pitt and Valiant, 1988).

Theorem 4.2. *For any $\epsilon, \delta \in (0,1)$, if the number of training data is at least $\ln(|\mathcal{H}|/\delta)/\epsilon$, and if $\mathcal{C} \subseteq \mathcal{H}$, then CH (Algorithm 4.2 with hypothesis class $\mathcal{H}$) (ϵ, δ)-PAC learns $\mathcal{C}$.*

Proof. Let P be a probability distribution over $\mathcal{X} \times \mathcal{Y}$ satisfying $\mathcal{C}$-realizability, and suppose

$$\mathbf{S} := ((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_T, \mathbf{y}_T)) \sim P^T$$

is an iid sample from P of size T, where for some $c^\star \in \mathcal{C}$, we have $\mathbf{y}_t = c^\star(\mathbf{x}_t)$ for all $t \in [T]$. Since $\mathcal{C} \subseteq \mathcal{H}$, we have $c^\star \in \mathcal{H}$, and hence CH does not return $\perp$. Let $\mathbf{h}$ denote the hypothesis from $\mathcal{H}$ returned by CH on input $\mathbf{S}$. Our goal is to show that the probability that $\mathsf{err}(\mathbf{h}; P) > \epsilon$ is at most δ.

For each hypothesis $h \in \mathcal{H}$, let B_h be the event that $h(\mathbf{x}_t) = c^\star(\mathbf{x}_t)$ for all $t \in [T]$. We have, for each $t \in [T]$,

$$\Pr(h(\mathbf{x}_t) = c^\star(\mathbf{x}_t)) = 1 - \Pr(h(\mathbf{x}_t) \neq c^\star(\mathbf{x}_t)) = 1 - \mathsf{err}(h; P)$$

since $(\mathbf{x}_t, c^\star(\mathbf{x}_t)) \sim P$. Therefore, by independence,

$$\Pr(B_h) = (1 - \mathsf{err}(h; P))^T.$$

Let $\mathcal{H}_{>\epsilon} := \{h \in \mathcal{H} \mid \mathsf{err}(h; P) > \epsilon\}$. Define the event

$$B := \bigcup_{h \in \mathcal{H}_{>\epsilon}} B_h,$$

If the event B does not happen, then $\mathbf{h}$ could not have been selected from $\mathcal{H}_{>\epsilon}$, which in turn implies $\mathsf{err}(\mathbf{h}; P) \leq \epsilon$. Since each B_h for $h \in \mathcal{H}_{>\epsilon}$ has probability at most $\exp(-\epsilon T)$, we have

$$\Pr(B) = \Pr\left(\bigcup_{h \in \mathcal{H}_{>\epsilon}} B_h\right) \leq \sum_{h \in \mathcal{H}_{>\epsilon}} \Pr(B_h) \leq |\mathcal{H}_{>\epsilon}|(1-\epsilon)^T < |\mathcal{H}| \exp(-\epsilon T)$$

where the first inequality follows from the union bound, and the last inequality uses the fact $1 + a < e^a$ for $a > 0$. By assumption, the number of training data satisfies $T \geq \ln(|\mathcal{H}|/\delta)/\epsilon$, so the right-hand side is at most δ. □

Example 4.5. Suppose $\mathcal{C} = \mathcal{H} = \mathsf{MCONJ}_n$, and S is a collection of training examples labeled by some $c^\star \in \mathcal{C}$. To find a hypothesis from $\mathcal{H}$ consistent with S, simulate Attribute Elimination in the online concept learning scenario as follows. At the start of round t, let h_t be the hypothesis used by Attribute Elimination; recall that $h_t \in \mathsf{MCONJ}_n = \mathcal{H}$. Let (x_t, y_t) be any training example in S for which $h_t(x_t) \neq y_t$. If there are none, then h_t is consistent with every example in S. If there is one, then Attribute Elimination has made a mistake. Since Attribute Elimination can make at most n mistakes, it finds a consistent hypothesis from $\mathcal{H}$ after at most n rounds. It is clear that the overall time complexity is polynomial in n and $|S|$. This gives a polynomial-time algorithm for PAC learning MCONJ_n, with sample complexity

$$\frac{n \ln(2) + \ln(1/\delta)}{\epsilon}.$$ ◊

The argument in Example 4.5 uses the fact that the final classifier returned by Attribute Elimination come from $\mathcal{H}$ (and hence $\mathcal{C}$) so that the union bound over (events corresponding to) $\mathcal{H}$ covers the possible classifiers that the learner may return. Learning algorithms for $\mathcal{C}$ that return a classifier from $\mathcal{C}$, such as the one from Example 4.5, are called *proper learning algorithms*. The online-to-batch conversion method from Algorithm 4.1 does not generally yield a proper learning algorithm for $\mathcal{C}$, even if the online learning algorithm uses predictors that come from $\mathcal{C}$. A learning algorithm that is not proper is called an *improper learning algorithm*.

Note that Theorem 4.2 only needs the $\mathcal{C}$-realizability assumption and the assumption $\mathcal{C} \subseteq \mathcal{H}$ to ensure that CH does not return $\perp$. Essentially the same argument proves the following theorem.

Theorem 4.3. *Let* $\mathbf{S}$ *be an iid sample of size* T *from a distribution* P *over* $\mathcal{X} \times \mathcal{Y}$*. For any* $\delta \in (0,1)$*, with probability at least* $1-\delta$*, for all* $h \in \mathcal{H}$*,*

$$\mathsf{err}(h;\mathbf{S}) = 0 \implies \mathsf{err}(h;P) \leq \frac{\ln(|\mathcal{H}|/\delta)}{T}.$$

4.4.3 Learning by finding a hypothesis with low training error rate

Even if there is no hypothesis consistent with the training data, it may nevertheless be useful to return a hypothesis that makes few mistakes on the training data. This is called *Empirical Risk Minimization (ERM)* for archaic reasons.

Let $\mathsf{err}(h;S) := \frac{1}{|S|}\sum_{(x,y)\in S} \mathbb{1}\{h(x) \neq y\}$ denote the *empirical error rate* of h on a data set S from $\mathcal{X} \times \mathcal{Y}$ (a.k.a. *training error rate* when S is the training data).

Algorithm 4.3: Empirical Risk Minimization (ERM) for learning with hypothesis class $\mathcal{H}$.
Input: training data S from $\mathcal{X} \times \mathcal{Y}$.

- Return $h \in \mathcal{H}$ with minimal training error rate

$$\mathsf{err}(h;S) = \min_{h' \in \mathcal{H}} \mathsf{err}(h';S).$$

Note that if S is labeled by some $h^\star \in \mathcal{H}$, then ERM with $\mathcal{H}$ is the same as CH with $\mathcal{H}$, since the minimum training error rate will be zero. ERM is also used as a subroutine in other learning algorithms (e.g., the online-to-batch conversion method discussed at the end of Section 4.2.2; see also Exercise 4.6).

Theorem 4.4. *If ERM (Algorithm 4.3 with hypothesis class* $\mathcal{H}$*) is run with an iid sample of size* T *from a distribution* P *over* $\mathcal{X} \times \mathcal{Y}$*, then for any* $\delta \in (0,1)$*, with probability at least* $1-\delta$*, ERM returns a hypothesis* $\mathbf{h}$ *from* $\mathcal{H}$ *satisfying*

$$\mathsf{err}(\mathbf{h};P) \leq \min_{h \in \mathcal{H}} \mathsf{err}(h;P) + \sqrt{\frac{2\ln(2|\mathcal{H}|/\delta)}{T}}.$$

Proof. Let $\mathbf{S}$ be an iid sample of size T from P. Let $\delta' := \delta/|\mathcal{H}|$. For each $h \in \mathcal{H}$, Corollary 4.4 implies that with probability at least $1-\delta'$,

$$|T\mathsf{err}(h;P) - T\mathsf{err}(h;\mathbf{S})| \leq \sqrt{T\ln(2/\delta')/2}.$$

Therefore, by a union bound, the following holds with probability at least $1 - |\mathcal{H}|\delta' = 1-\delta$:

$$|\mathsf{err}(h;P) - \mathsf{err}(h;\mathbf{S})| \leq \sqrt{\frac{\ln(2/\delta')}{2T}}, \quad \forall h \in \mathcal{H}. \tag{4.4}$$

Henceforth we assume this event holds.

Let $\mathbf{h}$ be the hypothesis from $\mathcal{H}$ returned by ERM when run on $\mathbf{S}$, and let $h^\star$ be any hypothesis in $\mathcal{H}$. Since both $\mathbf{h}$ and $h^\star$ are from $\mathcal{H}$, (4.4) gives

$$|\mathsf{err}(\mathbf{h};P) - \mathsf{err}(\mathbf{h};\mathbf{S})| \leq \sqrt{\frac{\ln(2/\delta')}{2T}}, \tag{4.5}$$

$$|\mathsf{err}(h^\star;P) - \mathsf{err}(h^\star;\mathbf{S})| \leq \sqrt{\frac{\ln(2/\delta')}{2T}}. \tag{4.6}$$

Then

$$\mathsf{err}(\mathbf{h};P) - \mathsf{err}(h^\star;P) = \underbrace{\big(\mathsf{err}(\mathbf{h};P) - \mathsf{err}(\mathbf{h};\mathbf{S})\big)}_{\text{bound using (4.5)}} + \underbrace{\big(\mathsf{err}(\mathbf{h};\mathbf{S}) - \mathsf{err}(h^\star;\mathbf{S})\big)}_{\text{at most 0 by choice of } \mathbf{h}} + \underbrace{\big(\mathsf{err}(h^\star;\mathbf{S}) - \mathsf{err}(h^\star;P)\big)}_{\text{bound using (4.6)}}.$$

Therefore

$$\mathsf{err}(\mathbf{h};P) - \mathsf{err}(h^\star;P) \le \sqrt{\frac{\ln(2/\delta')}{2T}} + 0 + \sqrt{\frac{\ln(2/\delta')}{2T}} = \sqrt{\frac{2\ln(2|\mathcal{H}|/\delta)}{T}}. \qquad \square$$

By using Example 4.4 instead of Example 4.3, we can improve the bound in Theorem 4.4 in cases where $\mathsf{err}(h^\star;P) = O(\log(|\mathcal{H}|/\delta)/T)$ (perhaps even zero).

Theorem 4.5. *There is an absolute constant $C > 0$ such that the following holds. If ERM (Algorithm 4.3 with hypothesis class $\mathcal{H}$) is run with an iid sample of size T from a distribution P over $\mathcal{X} \times \mathcal{Y}$, then for any $\delta \in (0,1)$, with probability at least $1-\delta$, ERM returns a hypothesis $\mathbf{h}$ from $\mathcal{H}$ satisfying*

$$\mathsf{err}(\mathbf{h};P) \le \mathsf{err}(h^\star;P) + C\left(\sqrt{\frac{\mathsf{err}(h^\star;P)\log(|\mathcal{H}|/\delta)}{T}} + \frac{\log(|\mathcal{H}|/\delta)}{T}\right)$$

for all $h^\star \in \mathcal{H}$.

Exercise 4.5. Prove Theorem 4.5. The following algebraic fact may be useful: for any non-negative numbers A, B, C,

$$A \le B + C\sqrt{A} \implies A \le B + C^2 + \sqrt{B}C. \qquad \lozenge$$

For example, if $\min_{h' \in \mathcal{H}} \mathsf{err}(h';P) \le \epsilon/2$, then using Theorem 4.5, we find that it suffices to have

$$T \ge \frac{C'\log(|\mathcal{H}|/\delta)}{\epsilon}$$

(for some other absolute constant $C' > 0$) to ensure that, with probability at least $1-\delta$, we have $\mathsf{err}(\mathbf{h};P) \le \epsilon$. If we had used Theorem 4.4, then the dependence on ϵ in the sample complexity would be $1/\epsilon^2$ instead of $1/\epsilon$.

An important concern about ERM is that, for many hypothesis classes $\mathcal{H}$ that one might want to use, there do not exist algorithmic implementations that either always correctly find a minimizer of the training error rate (even in an approximate sense), or always run in polynomial time. Nevertheless, the design of many heuristic learning algorithms are motivated by ERM, and thus ERM may serve as a theoretical idealization of these heuristics.

Exercise 4.6. Complete the analysis of the online-to-batch method at the end of Section 4.2.2. What should be the size of the validation data set? What is the overall sample complexity for achieving (ϵ,δ)-PAC learning? $\lozenge$

Finally, ERM can be used to achieve a stronger guarantee than (ϵ,δ)-PAC learning. Theorems 4.4 and 4.5 show that, with high probability (at least $1-\delta$), ERM returns $\mathbf{h}$ with error rate that is not much higher than the best achievable error rate using hypotheses from $\mathcal{H}$. This should be thought of as a statistical analogue of the notion of regret from online learning (cf. Section 2.1.1). These guarantees require no assumptions on the population P, i.e., they are agnostic about P.

A learning algorithm is said to *(ϵ, δ)-PAC agnostically learns* a concept class $\mathcal{C} \subseteq \mathcal{Y}^{\mathcal{X}}$ if, for any distribution P over $\mathcal{X} \times \mathcal{Y}$, under the iid training data assumption, the output of the learning algorithm $\mathbf{f}$ satisfies

$$\Pr\left(\mathsf{err}(\mathbf{f}; P) \leq \min_{c \in \mathcal{C}} \mathsf{err}(c; P) + \epsilon\right) \geq 1 - \delta.$$

(There is no assumption on $\min_{c \in \mathcal{C}} \mathsf{err}(c; P)$, although the guarantee is only interesting if it is non-trivial.) If ERM is used with a hypothesis class $\mathcal{H}$ satisfying $\mathcal{C} \subseteq \mathcal{H}$, then

$$\min_{h \in \mathcal{H}} \mathsf{err}(h; P) \leq \min_{c \in \mathcal{C}} \mathsf{err}(c; P).$$

Hence, in this case, ERM (ϵ, δ)-PAC agnostically learns $\mathcal{C}$ with sample complexity

$$\frac{2 \ln(2|\mathcal{H}|/\delta)}{\epsilon^2}.$$

4.4.4 Occam's razor

A variation on CH is to not return just any consistent hypothesis, but instead to return the "simplest" consistent hypothesis; this is the philosophical principle behind "Occam's razor". One possible formalization is to imagine that the hypotheses in $\mathcal{H}$ are ordered by "simplicity" in a list: $h_1, h_2, \ldots$. In fact, the hypothesis class could be infinite (but countable).

Theorem 4.6 (Occam's razor, simplified)**.** *Let $\mathbf{S}$ be an iid sample of size T from a distribution P over $\mathcal{X} \times \mathcal{Y}$, and let $h_1, h_2, \ldots$ be an enumeration of a hypothesis class $\mathcal{H}$. For any $\delta \in (0, 1)$, with probability at least $1 - \delta$, for all $i \in \mathbb{N}$,*

$$\mathsf{err}(h_i; \mathbf{S}) = 0 \implies \mathsf{err}(h_i; P) \leq \frac{\ln(i(i+1)/\delta)}{T}.$$

The ordering of the hypotheses can be thought of as bets about which hypotheses are more likely to be accurate. The earlier on the list that a consistent hypothesis appears, the better—in the sense of having a better error rate guarantee.

Proof of Theorem 4.6. For each $i \in \mathbb{N}$, let B_i be the event that h_i is consistent with $\mathbf{S}$, and let $\epsilon_i := \ln(i(i+1)/\delta)/T$. Following the argument from the proof of Theorem 4.2,

$$\Pr(B_i) \leq \exp(-\mathsf{err}(h_i; P)T).$$

We need to show that $\Pr(\bigcup_{i \in I} B_i) \leq \delta$, where $I := \{i \in \mathbb{N} \mid \mathsf{err}(h_i; P) > \epsilon_i\}$. By the union bound,

$$\begin{aligned}
\Pr\left(\bigcup_{i \in I} B_i\right) &\leq \sum_{i \in I} \Pr(B_i) \\
&\leq \sum_{i \in I} \exp(-\epsilon_i T) \\
&= \sum_{i \in I} \frac{\delta}{i(i+1)} \\
&= \delta \sum_{i \in I} \left(\frac{1}{i} - \frac{1}{i+1}\right) \\
&\leq \delta.
\end{aligned}$$

□

A similar approach may be used with sequences of hypotheses classes $(\mathcal{H}_i)_{i\in\mathbb{N}}$. Suppose these hypothesis classes are constructed such that $\mathcal{C} \subset \bigcup_{i\in\mathbb{N}} \mathcal{H}_i$, but you do not know which $\mathcal{H}_i$ contains the target concept for the $\mathcal{C}$-realizable distribution of interest. Given T iid training data from such a distribution, run CH with $\mathcal{H}_i$ starting with $i = 1$, and continuing until a consistent hypothesis is found. With probability at least $1 - \delta$, for every $i \in \mathbb{N}$, if a consistent hypothesis is found in $\mathcal{H}_i$, then its error rate is at most

$$\frac{\ln(|\mathcal{H}_i|/\delta)}{T} + \frac{\ln(i(i+1))}{T}.$$

The first term is the error rate guarantee for CH if you had known to use $\mathcal{H}_i$; the second term is the price for not having this benefit of hindsight. This strategy is a special case of a technique called *structural risk minimization* (Vapnik, 1985).