Machine Learning
The Science of Selection under Uncertainty

Yevgeny Seldin

January 30, 2026

To my parents – Elena Markman and Anatoly Seldin

Contents

Preface

Learning, whether natural or artificial, is a process of selection. It starts with a set of candidate options and selects the more successful ones. In the case of machine learning the selection is done based on empirical estimates of prediction accuracy of candidate prediction rules on some data. Due to randomness of data sampling the empirical estimates are inherently noisy, leading to selection under uncertainty. The book provides statistical tools to obtain theoretical guarantees on the outcome of selection under uncertainty. We start with concentration of measure inequalities, which are the main statistical instrument for controlling how much an empirical estimate of expectation of a function deviates from the true expectation. The book covers a broad range of inequalities, including Markov's, Chebyshev's, Hoeffding's, Bernstein's, Empirical Bernstein's, Unexpected Bernstein's, kl, and split-kl. We then study the classical (offline) supervised learning and provide a range of tools for deriving generalization bounds, including Occam's razor, Vapnik-Chervonenkis analysis, and PAC-Bayesian analysis. The latter is further applied to derive generalization guarantees for weighted majority votes. After covering the offline setting, we turn our attention to online learning. We present the space of online learning problems characterized by environmental feedback, environmental resistance, and structural complexity. A common performance measure in online learning is regret, which compares performance of an algorithm to performance of the best prediction rule in hindsight, out of a restricted set of prediction rules. We present tools for deriving regret bounds in stochastic and adversarial environments, and under full information and bandit feedback.

Reading Guide

The book is used in teaching parts of three courses at the University of Copenhagen: "Machine Learning A", "Machine Learning B", and "Online and Reinforcement Learning". The material is split in the following way:

Machine Learning A: Chapter 1, Chapter 2, Sections 3.1—3.3, and Sections 4.1—4.4.

Machine Learning B: Parts of Chapter 3 and Chapter 4 not covered in Machine Learning A.

Online and Reinforcement Learning: Chapter 7.

Both "Machine Learning B" and "Online and Reinforcement Learning" assume that the reader is familiar with the material of "Machine Learning A", but they are independent of each other and can be read in any order.

Chapter 5 assumes that the reader is familiar with Chapters 1 and 2, but otherwise independent of the rest of the material.

Acknowledgements

I would like to thank my co-lecturers and students for asking excellent questions and for constantly inspiring me to find better ways to present the material, as well as for fishing out errors and typos in the text. Special thanks goes to my father, Anatoly Seldin, for his lifelong curiosity and engaging discussions, which, among other things, served an inspiration for Chapter 1.

The readers are more than welcome to propose further suggestions and report any typos to me at seldin@di.ku.dk. Your feedback will serve everyone who will read the book after you.

Chapter 1

Introduction

Machine Learning lies at the intersection of Computer Science and Statistics. In a "classical" form it starts with a set of potential prediction rules and uses data to select a prediction rule that is expected to provide good predictions on new data, see Figure 1.1. This process involves three key elements: *design*, *computation*, and *statistics*.

Design Learning starts with a set of potential prediction rules. It could be a set of parameters defining linear separators, a set of neural networks, decision trees, or anything else, as well as any combination of the above. The set may come with a prioritization (a "prior knowledge"), which would say that some rules are more likely than others to be good prediction rules. It is obviously desirable for the set to contain a good prediction rule, because it is the set that learning procedure selects from. However, there are additional desiderata dictated by computation and statistics, as discussed next. Design of a good set of candidate prediction rules is normally based on domain knowledge and experience.

Statistics Having the set of candidate prediction rules the next task is to estimate their quality, and this is where statistics comes in. The estimation is done by using annotated data given to the algorithm. The algorithm can take any prediction rule and calculate how well it preforms on the data provided to the algorithm, and use the outcome as an estimate of how well the same rule would preform on new data. The challenge is that the estimates based on the data are random samples of the expected performance of the algorithm on new data, and so they are inherently uncertain (which is why the book is called "The science of selection under uncertainty"). They may happen to be better or worse than their expected value, and thus confuse the selection process. Imagine that you had a set of archers and you had to select an archer to send to an arch shooting competition. And imagine that you would ask each archer to shoot once and send the one with the best shot to the competition. For each of the archers the trial shot may happen to be better or worse than their expected performance. So, on the one hand, you want

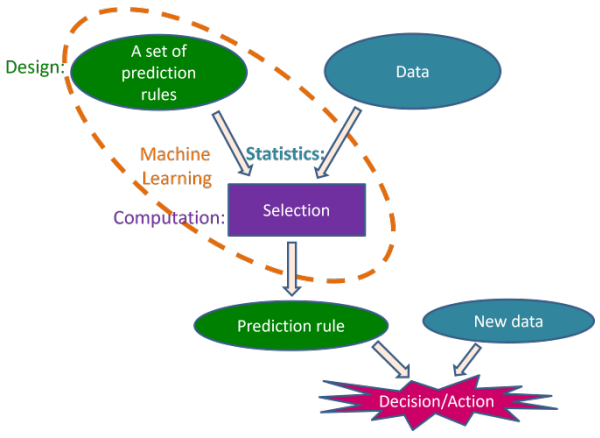


Figure 1.1: A "classical" machine learning process.

the selection set to be sufficiently large, so that with sufficiently large probability you would have some good archers in the set. But, on the other hand, for any poor archer there is a small probability that on the selection day they would accidentally hit the bull eye and get selected. And if the selection set contains too many poor archers, with *high* probability at least one of them would have a lucky selection day and get selected instead of a good archer, and ruin the final competition. In the context of a learning process, the archers are the prediction rules, their trial shots are their predictions on the training data, and the competition is the prediction of the selected rule on a new data point. And there is a statistical trade-off between starting with a sufficiently rich set of candidate prediction rules in the design phase to have a good candidate in the set, but at the same time not an overly large to control the probability of weak candidates accidentally wining the selection. In other words, even for a poor prediction rule, with small probability the training data may happen to fit the rule well and get predicted better than expected, and if there are too many poor prediction rules, with high probability at least one of them may happen to be "lucky" and confuse the selection process.

(We note that it is not the plain number of prediction rules that makes a difference, because if they are sufficiently similar their probabilities of being "lucky" may be correlated, but it is "richness" of the set, which is discussed in details in Chapter 4.)

Computation Finally comes the computational aspect of the selection process. If the set of candidate prediction rules is finite and small, it may be computationally feasible to estimate the quality of all of them. But if the set is large, or even potentially infinite, there is a need in computationally efficient procedures to find candidates that are estimated to deliver good predictions. And this is where Computer Science comes into play.

Design + Statistics + Computation At the end, all the three elements should fit together. Learning needs to start from a candidate set of prediction rules that is sufficiently rich to include good prediction rules, but, at the same time, not overly rich, so that excessive amount of poor prediction rules would not confuse the selection process. Alternatively, the candidate set may come with prioritization that would give sufficient preference to potentially good prediction rules. And the candidate set has to be sufficiently structured or small to support computationally efficient selection.

The book focuses primarily on the statistical aspect of the selection process. We provide tools for quantifying, controlling, and reducing the uncertainty, as well as tools for prioritizing prediction rules in the selection process. Chapter 7 departs from the "classical" batch learning paradigm and moves into the realm of online learning, while still maintaining a strong focus on the statistical aspect of taking actions (making selections) under uncertainty.

Chapter 2

Supervised Learning

The most basic and widespread form of machine learning is supervised learning. In the classical batch supervised learning setting the learner is given an annotated sample, which is used to derive a prediction rule for annotating new samples. We start with a simple informal example and then formalize the problem.

Let's say that we want to build a prediction rule that will use the average grade of a student in home assignments, say on a 100-points scale, to predict whether the student will pass the final exam. Such a prediction rule could be used for preliminary filtering of students to be allowed to take the final exam. The annotated sample could be a set of average grades of students from the previous year with indications of whether they have passed the final exam. The prediction rule could take a form of a threshold grade (a.k.a. decision stump), above which the student is expected to pass and below fail.

Now assume that we want to take a more refined approach and look into individual grades in each assignment, say, 5 assignments in total. For example, different assignments may have different relevance for the final exam or, maybe, some students may demonstrate progression throughout the course, which would mean that their early assignments should not be weighted equally with the later ones. In the refined approach each student can be represented by a point in a 5-dimensional space. The one-dimensional threshold could be replaced by a separating hyperplane, which separates the 5-dimensional space of grades into a linear subspace, where most students are likely to pass, and the complement, where they are likely to fail. An alternative approach is to look at "nearest neighbors" of a student in the space of grades. Given a grade profile of a student (the point representing the student in the 5-dimensional space) we look at students with the closest grade profile and see whether most of them passed or failed. This is known as the *K Nearest Neighbors* algorithm, where K is the number of neighbors we look at. But how many neighbors K should we look at? Considering the extremes gives some intuition about the problem. Taking just one nearest neighbor may be unreliable. For example, we could have a good student that accidentally failed the final exam and then all the neighbors will be marked as "expected to fail". Going in the other extreme and taking all the students in as neighbors is also undesirable, because effectively it will ignore the individual profile altogether. So a good value of K should be somewhere between 1 and n, where n is the size of the annotated set. But how to find it? Well, read on and you will learn how to approach this question formally.

2.1 The Supervised Learning Setting

We start with a bunch of notations and then illustrate them with examples.

- $\mathcal{X}$ - the sample space.
- $\mathcal{Y}$ - the label space.
- $X \in \mathcal{X}$ - unlabeled sample.
- $(X, Y) \in (\mathcal{X} \times \mathcal{Y})$ - labeled sample.
- $S = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ - a training set. We assume that (X_i, Y_i) pairs in S are sampled i.i.d. according to an unknown, but fixed distribution $p(X, Y)$.

- $h : \mathcal{X} \to \mathcal{Y}$ - a hypothesis, which is a function from $\mathcal{X}$ to $\mathcal{Y}$.
- $\mathcal{H}$ - a hypothesis set.
- $\ell(Y', Y)$ - the loss function for predicting Y' instead of Y.
- $\hat{L}(h, S) = \frac{1}{n} \sum_{i=1}^{n} \ell(h(X_i), Y_i)$ - the empirical loss (a.k.a. error or risk) of h on S. (In many textbooks S is omitted from the notation and $\hat{L}(h)$ or $\hat{L}_n(h)$ is used to denote $\hat{L}(h, S)$.)
- $L(h) = \mathbb{E}\left[\ell(h(X), Y)\right]$ - the expected loss (a.k.a. error or risk) of h, where the expectation is taken with respect to $p(X, Y)$.

The Learning Protocol

The classical supervised learning acts according to the following protocol:

1. The learner gets a training set S of size n sampled i.i.d. according to $p(X, Y)$.
2. The learner returns a prediction rule h.
3. New instances (X, Y) are sampled according to $p(X, Y)$, but only X is observed and h is used to predict the unobserved Y.

The goal of the learner is to return h that minimizes $L(h)$, which is the expected error on new samples.

Examples - Sample and Label Spaces

Let's say that we want to predict person's height based on age, gender, and weight. Then $\mathcal{X} = \mathbb{N} \times \{\pm 1\} \times \mathbb{R}$ and $\mathcal{Y} = \mathbb{R}$. If we want to predict gender based on age, weight, and height, then $\mathcal{X} = \mathbb{N} \times \mathbb{R} \times \mathbb{R}$ and $\mathcal{Y} = \{\pm 1\}$. If we want to predict the height of a baby at the age of 4 years based on his or her height at the ages of 1, 2, and 3 years, then $\mathcal{X} = \mathbb{R}^3$ and $\mathcal{Y} = \mathbb{R}$.

2.1.1 Classification, Regression, and Other Supervised Learning Problems

The most widespread forms of supervised learning are classification and regression. We also mention a few more, mainly to show that the supervised learning setting is much richer.

Classification A supervised learning problem is a classification problem when the output (label) space $\mathcal{Y}$ is binary. The goal of the learning algorithm is to separate between two classes: yes or no; good or bad; healthy or sick; male or female; etc. Most often the translation of the binary label into numerical representation is done by either taking $\mathcal{Y} = \{\pm 1\}$ or $\mathcal{Y} = \{0, 1\}$. Sometimes the setting is called *binary classification* to emphasize that $\mathcal{Y}$ takes just two values.

Regression A supervised learning problem is a regression problem when the output space $\mathcal{Y} = \mathbb{R}$. For example, prediction of person's height would be a regression problem.

Multiclass Classification When $\mathcal{Y}$ consists of a finite and typically unordered and relatively small set of values, the corresponding supervised learning problem is called multiclass classification. For example, prediction of a study program a student will apply for based on his or her grades would be a multiclass classification problem. Finite ordered output spaces, for example, prediction of age or age group can also be modeled as multiclass classification, but it may be possible to exploit the structure of $\mathcal{Y}$ to obtain better solutions. For example, it may be possible to exploit the fact that ages 22 and 23 are close together, whereas 22 and 70 are far apart; therefore, it may be possible to share some information between close ages, as well as exploit the fact that predicting 22 instead of 23 is not such a big mistake as predicting 22 instead of 70. Depending on the setting, it may be preferable to model prediction of ordered sets as regression rather than multiclass classification.

Structured Prediction Consider the problem of machine translation. An algorithm gets a sentence in English as an input and should produce a sentence in Danish as an output. In this case the output (the sentence in Danish) is not merely a number, but a structured object and such prediction problems are known as structured prediction.

2.1.2 The Loss Function $\ell(Y', Y)$

The loss function (a.k.a. the error function) encodes how much the user of an algorithm cares about various kinds of mistakes. Most literature on binary classification, including this book, uses the *zero-one loss* defined by

$$\ell(Y', Y) = \mathbb{1}(Y' \neq Y) = \begin{cases} 1, & \text{if } Y' \neq Y \\ 0, & \text{otherwise,} \end{cases}$$

where $\mathbb{1}$ is the indicator function. Common loss functions in regression are the *square loss*

$$\ell(Y', Y) = (Y' - Y)^2$$

and the *absolute loss*

$$\ell(Y', Y) = |Y' - Y|.$$

The above loss functions are convenient general choices, but not necessarily the right choice for a particular application. For example, imagine that you design an algorithm for fire alarm that predicts "fire / no fire". Assume that the cost of a house is 3,000,000 DKK and the cost of calling in a fire brigade is 2,000 DKK. Then the loss function would be

$\ell(Y', Y) =$

Y' \ Y	no fire	fire
no fire	0	3,000,000
fire	2,000	0

The loss for making the correct prediction is zero, but the loss of *false positive* (predicting fire when in reality there is no fire) and *false negative* (predicting no fire when the reality is fire) are not symmetric anymore.

Pay attention that the loss depends on how the predictions are used and the loss table depends on the user. For example, if the same alarm is installed in a house that is worth 10,000,000 DKK, the ratio between the cost of false positives and false negatives will be very different and, as a result, the optimal prediction strategy will not necessarily be the same.

2.2 K Nearest Neighbors for Binary Classification

One of the simplest algorithms for binary classification is K Nearest Neighbors (K-NN). The algorithm is based on an externally provided distance function $d(\mathbf{x}, \mathbf{x}')$ that computes distances between pairs of points $\mathbf{x}$ and $\mathbf{x}'$. For example, for points in $\mathbb{R}^d$ the distance could be the Euclidean distance $d(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\| = \sqrt{\sum_{j=1}^{d} (x_j - x'_j)^2} = \sqrt{(\mathbf{x} - \mathbf{x}')^T (\mathbf{x} - \mathbf{x}')}$, where $\mathbf{x} = (x_1, \ldots, x_d)$ and x_j is the j-th coordinate of vector $\mathbf{x}$. Other choices of distance measures are possible and, in general, lead to different predictions. The choice of the distance measure $d(\mathbf{x}, \mathbf{x}')$ is the key for success or failure of K-NN, but we leave the topic of selection of d outside the scope of the book.

K-NN algorithm takes as input a set of training points $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ and predicts the label of a target point $\mathbf{x}$ based on the majority vote of K points from S, which are the closest to $\mathbf{x}$ in terms of the distance measure $d(\mathbf{x}_i, \mathbf{x})$.

The ordering of d_i-s in Step 3 is identical to the ordering of d_i^2 and for the Euclidean distance we can save the computation of the square root by working with squared distances.

The hypothesis space $\mathcal{H}$ is implicit in the K-NN algorithm. It is the space of all possible partitions of the sample space $\mathcal{X}$. The output hypothesis h is parametrized by all training points $h_S = h_{\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}}$. In the sequel we will see other prediction rules that operate with more explicit hypothesis spaces, for example, a space of all linear separators.

Algorithm 1 K Nearest Neighbors (K-NN) for Binary Classification with $\mathcal{Y} = \{\pm 1\}$

1: **Input:** A set of labeled points $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ and a target point $\mathbf{x}$ that has to be classified.
2: Calculate the distances $d_i = d(\mathbf{x}_i, \mathbf{x})$.
3: Sort d_i-s in ascending order and let $\sigma : \{1, \ldots, n\} \to \{1, \ldots, n\}$ be the corresponding permutation of indices. In other words, for any pair of indices $i < j$ we should have $d_{\sigma(i)} \leq d_{\sigma(j)}$.
4: The output of K-NN is $y = \text{sign}\left(\sum_{i=1}^{K} y_{\sigma(i)}\right)$. It is the majority vote of K points that are the closest to $\mathbf{x}$. Note that we can calculate the output of K-NN for all K in one shot.

2.2.1 How to Pick K in K-NN?

One of the key questions in K-NN is how to pick K. It is instructive to consider the extreme cases to gain some intuition. In 1-NN the prediction is based on a single sample $(\mathbf{x}_i, y_i)$ which happens to be closest to the target point $\mathbf{x}$. This may not be the best thing to do. Imagine that you are admitted to a hospital and a diagnostic system determines whether you are healthy or sick based on a single annotated patient that has the symptoms closest to yours (in distance measure d). You would likely prefer to be diagnosed based on the majority of diagnoses of several patients with similar symptoms. At the other extreme, in n-NN, where n is the number of samples in S, the prediction is based on the majority of labels y_i within the sample S, without even taking any particular $\mathbf{x}$ into account. So the desirable K is somewhere between 1 and n, but how to find it?

Let $h_{\texttt{K-NN}}$ denote the prediction rule of K-NN. As K goes from 1 to n, K-NN provides n different prediction rules, $h_{\texttt{1-NN}}$, $h_{\texttt{2-NN}}$, $\ldots$, $h_{\texttt{n-NN}}$ (or half of that if we only take the odd values of K). Recall that we are interested in finding K that minimizes the expected loss $L(h_{\texttt{K-NN}})$ and that $L(h_{\texttt{K-NN}})$ is unobserved. We can calculate the empirical loss $\hat{L}(h_{\texttt{K-NN}}, S)$ for any K. However, $\hat{L}(h_{\texttt{1-NN}}, S)$ is always zero[1] and in general the empirical error of K-NN is an underestimate of its expected error and we need other tools to estimate $L(h_{\texttt{K-NN}})$. We start developing these tools in the next section and continue throughout the book.

2.3 Validation

Whenever we select a hypothesis $\hat{h}^*_S$ out of a hypothesis set $\mathcal{H}$ based on empirical performances $\hat{L}(h, S)$, the empirical performance $\hat{L}(\hat{h}^*_S, S)$ becomes a biased estimate of $L(\hat{h}^*_S)$. This is clearly observed in 1-NN, where $\hat{L}(h_{\texttt{1-NN}}, S) = 0$, but $L(h_{\texttt{1-NN}})$ is most often not zero (we remind that the hypothesis space in 1-NN is the space of all possible partitions of the sample space $\mathcal{X}$ and $h_{\texttt{1-NN}}$ is the hypothesis that achieves the minimal empirical error in this space). The reason is that when we do the selection we pick $\hat{h}^*_S$ that is best suited for S (it achieves the minimal $\hat{L}(h, S)$ out of all h in $\mathcal{H}$). Therefore, from the perspective of $\hat{h}^*_S$ the new samples (X, Y) are not "similar" to the samples (X_i, Y_i) in S. A bit more precisely, (X, Y) is not exchangeable with (X_i, Y_i), because if we would exchange (X_i, Y_i) with (X, Y) it is likely that $\hat{h}^*_S$, the hypothesis that minimizes $\hat{L}(h, S)$, would be different. Again, this is very clear in 1-NN: if we change one sample (X_i, Y_i) in S we get a different prediction rule $h_{\texttt{1-NN}}$. We get back to this topic in much more details in Chapter 4 after we develop some mathematical tools for analyzing the bias in Chapter 3. For now we present a simple solution for estimating $L(\hat{h}^*_S)$ and motivate why we need the tools from Chapter 3.

The solution is to split the sample set S into training set $S^{\texttt{train}}$ and validation set $S^{\texttt{val}}$. We can then find the best hypothesis for the training set, $h^*_{S^{\texttt{train}}}$, and validate it on the validation set by computing $\hat{L}(h^*_{S^{\texttt{train}}}, S^{\texttt{val}})$. Note that from the perspective of $h^*_{S^{\texttt{train}}}$ the samples in $S^{\texttt{val}}$ are exchangeable with any new samples (X, Y). If we exchange $(X_i, Y_i) \in S^{\texttt{val}}$ with another sample (X, Y) coming from the same distribution, $h^*_{S^{\texttt{train}}}$ will stay the same and in expectation $\mathbb{E}\left[\ell(h^*_{S^{\texttt{train}}}(X_i), Y_i)\right] = \mathbb{E}\left[\ell(h^*_{S^{\texttt{train}}}(X), Y)\right]$, meaning that on average $\hat{L}(h^*_{S^{\texttt{train}}}, S^{\texttt{val}})$ will also stay the same (only on average, the exact value may change). Therefore, $\hat{L}(h^*_{S^{\texttt{train}}}, S^{\texttt{val}})$ is an unbiased estimate of $L(h^*_{S^{\texttt{train}}})$. (We get back to this point in much more details in Chapter 4.)

[1] This is because the closest point in S to a sample point $\mathbf{x}_i$ is $\mathbf{x}_i$ itself and we assume that S includes no identical points with dissimilar labels, which is a reasonable assumption if $\mathcal{X} = \mathbb{R}^d$.

Now we get to the question of how to split S into $S^{\texttt{train}}$ and $S^{\texttt{val}}$, and again it is very instructive to consider the extreme cases. Imagine that we keep a single sample for validation and use the remaining $n-1$ samples for training. Let's say that we keep the last sample, (X_n, Y_n), for validation, then $\hat{L}(h^*_{S^{\texttt{train}}}, S^{\texttt{val}}) = \ell(h^*_{S^{\texttt{train}}}(X_n), Y_n)$ and in the case of zero-one loss it is either zero or one. Even though $\hat{L}(h^*_{S^{\texttt{train}}}, S^{\texttt{val}})$ is an unbiased estimate of $L(h^*_{S^{\texttt{train}}})$, it clearly does not represent it well. At the other extreme, if we keep $n-1$ points for validation and use the single remaining point for training we run into a different kind of problem: a classifier trained on a single point is going to be extremely weak. Let's say that we have used the first point, (X_1, Y_1), for training. In the case of K-NN classifier, as well as most other classifiers, $h^*_{S^{\texttt{train}}}$ will always predict Y_1, no matter what input it gets. The validation error $\hat{L}(h^*_{S^{\texttt{train}}}, S^{\texttt{val}})$ will be a very good estimate of $L(h^*_{S^{\texttt{train}}})$, but this is definitely not a classifier we want.

So how many samples from S should go into $S^{\texttt{train}}$ and how many into $S^{\texttt{val}}$? Currently there is no "gold answer" to this question, but in Chapters 3 and 4 we develop mathematical tools for intelligent reasoning about it. An important observation to make is that for h independent of (X, Y) the zero-one loss $\ell(h(X), Y)$ is a Bernoulli random variable with bias $\mathbb{P}(\ell(h(X), Y) = 1) = L(h)$. Furthermore, when h is independent of a set of samples $\{(X_1, Y_1), \ldots, (X_m, Y_m)\}$ (i.e., these samples are not used for selecting h), the losses $\ell(h(X_i), Y_i)$ are independent identically distributed (i.i.d.) Bernoulli random variables with bias $L(h)$. Therefore, when $S^{\texttt{val}}$ is of size m, the validation loss $\hat{L}(h^*_{S^{\texttt{train}}}, S^{\texttt{val}})$ is an average of m i.i.d. Bernoulli random variables with bias $L(h^*_{S^{\texttt{train}}})$. The validation loss $\hat{L}(h^*_{S^{\texttt{train}}}, S^{\texttt{val}})$ is observed, but the expected loss that we are actually interested in is unobserved. One of the key questions that we are interested in is how far $\hat{L}(h^*_{S^{\texttt{train}}}, S^{\texttt{val}})$ can be from $L(h^*_{S^{\texttt{train}}})$. We have already seen that $m = 1$ is too little. But how large should it be, 10, 100, 1000? Essentially this question is equivalent to asking how many times do we need to flip a biased coin in order to get a satisfactory estimate of its bias. In Chapter 3 we develop concentration of measure inequalities that answer this question.

Another technical question is which samples should go into $S^{\texttt{train}}$ and which into $S^{\texttt{val}}$? From the theoretical perspective we assume that S is sampled i.i.d. and, therefore, it does not matter. We can take the first $n-m$ samples into $S^{\texttt{train}}$ and the last m into $S^{\texttt{val}}$ or split in any other way. From a practical perspective the samples may actually not be i.i.d. and there could be some parameter that has influenced their order in S. For example, they could have been ordered alphabetically. Therefore, from a practical perspective it is desirable to take a random permutation of S before splitting, unless the order carries some information we would like to preserve. For example, if S is a time-ordered series of product reviews and we would like to build a classifier that classifies them into positive and negative, we may want to get an estimate of temporal variation and keep the order when we do the split, i.e., train on the earlier samples and validate on the later.

2.3.1 Test Set: It's not about how you call it, it's about how you use it!

Assume that we have split S into $S^{\texttt{train}}$ and $S^{\texttt{val}}$; we have trained $h_{\texttt{1-NN}}, \ldots, h_{\texttt{n-NN}}$ on $S^{\texttt{train}}$; we calculated $\hat{L}(h_{\texttt{1-NN}}, S^{\texttt{val}}), \ldots, \hat{L}(h_{\texttt{n-NN}}, S^{\texttt{val}})$ and picked the value K^* that minimizes $\hat{L}(h_{\texttt{K-NN}}, S^{\texttt{val}})$. Is $\hat{L}(h_{\texttt{K*-NN}}, S^{\texttt{val}})$ an unbiased estimate of $L(h_{\texttt{K*-NN}})$?

This is probably one of the most conceptually difficult points about validation, at least when you encounter it for the first time. While for each $h_{\texttt{K-NN}}$ individually $\hat{L}(h_{\texttt{K-NN}}, S^{\texttt{val}})$ is an unbiased estimate of $L(h_{\texttt{K-NN}})$, the validation loss $\hat{L}(h_{\texttt{K*-NN}}, S^{\texttt{val}})$ is a *biased* estimate of $L(h_{\texttt{K*-NN}})$. This is because $S^{\texttt{val}}$ was used for selection of K^* and, therefore, $h_{\texttt{K*-NN}}$ depends on $S^{\texttt{val}}$. So if we want to get an unbiased estimate of $L(h_{\texttt{K*-NN}})$ we have to reserve some "fresh" data for that. So we need to split S into $S^{\texttt{train}}$, $S^{\texttt{val}}$, and $S^{\texttt{test}}$; train the K-NN classifiers on $S^{\texttt{train}}$; pick the best K^* based on $\hat{L}(h_{\texttt{K-NN}}, S^{\texttt{val}})$; and then compute $\hat{L}(h_{\texttt{K*-NN}}, S^{\texttt{test}})$ to get an unbiased estimate of $L(h_{\texttt{K*-NN}})$.

It's not about how you call it, it's about how you use it! Some people think that if you call some data a test set it automatically makes loss estimates on this set unbiased. This is not true. Imagine that you have split S into $S^{\texttt{train}}$, $S^{\texttt{val}}$, and $S^{\texttt{test}}$; you trained K-NN on $S^{\texttt{train}}$, picked the best value K^* using $S^{\texttt{val}}$, and estimated the loss of $h_{\texttt{K*-NN}}$ on $S^{\texttt{test}}$. And now you are unhappy with the result and you want to try a different learning method, say a neural network. You go through the same steps: you train networks with various parameter settings on $S^{\texttt{train}}$, you validate them on $S^{\texttt{val}}$, and you pick the best parameter set θ^* based on the validation loss. Finally, you compute the test loss of the neural network parametrized by θ^* on $S^{\texttt{test}}$. It happens to be lower than the test loss of K^*-NN and you decide to go with the neural network. Does the empirical loss of the neural network on $S^{\texttt{test}}$ represent an unbiased

estimate of its expected loss? No! Why? Because our choice to pick the neural network was based on its superior performance relative to $h_{\texttt{K}^*\texttt{-NN}}$ on $S^{\texttt{test}}$, so $S^{\texttt{test}}$ was used in selection of the neural network. Therefore, there is dependence between $S^{\texttt{test}}$ and the hypothesis we have selected, and the loss on $S^{\texttt{test}}$ is biased. If we want to get an unbiased estimate of the loss we have to find new "fresh" data or reserve such data from the start and keep it in a locker until the final evaluation moment. Alternatively, we can correct for the bias and in Chapter 4 we will learn some tools for making the correction. The main take-home message is: ***It is not about how you call a data set, S^{train}, S^{val}, or S^{test}, it is the way you use it which determines whether you get unbiased estimates or not!*** In some cases it is possible to get unbiased estimates or to correct for the bias already with $S^{\texttt{train}}$, and sometimes there is bias even on $S^{\texttt{test}}$ and we need to correct for that.

2.3.2 Cross-Validation

Sometimes it feels wasteful to use only part of the data for training and part for validation. A *heuristic* way around it is cross-validation. In the standard N-fold cross-validation setup the data S are split into N non-overlapping folds $S_1, \ldots, S_N$. Then for $i \in \{1, \ldots, N\}$ we train on all folds except the i-th and validate on S_i. We then take the average of the N validation errors and pick the parameter that achieves the minimum (for example, the best K in K-NN). Finally, we train a model with the best parameter we have selected in the cross-validation procedure (for example the best K^* in K-NN) using all the data S.

The standard cross-validation procedure described above is a heuristic and has no theoretical guarantees. It is fairly robust and widely used in practice, but it is possible to construct examples, where it fails. In Chapter 4 we describe a modification of the cross-validation procedure, which comes with theoretical generalization guarantees and is empirically competitive with the standard cross-validation procedure.

2.4 Perceptron - Basic Algorithm for Linear Classification

Linear classification is another basic family of classification strategies. Let $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \{\pm 1\}$. A hyperplane in $\mathbb{R}^d$ is described by a tuple $(\mathbf{w}, b)$, where $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$. The points $\mathbf{x}$ on the hyperplane are described by the equation

$$\mathbf{w}^T\mathbf{x} + b = 0.$$

A linear classifier $h = (\mathbf{w}, b)$ assigns label $+1$ to all points on the "positive" side of the hyperplane and -1 on the "negative" side of the hyperplane. Specifically,

$$h(\mathbf{x}) = \text{sign}\left(\mathbf{w}^T\mathbf{x} + b\right).$$

Homogeneous classifiers We distinguish between *homogeneous* linear classifiers and non-homogeneous linear classifiers. A homogeneous linear classifier is described by a hyperplane passing through the origin. From the mathematical point of view it means that $b = 0$.

We note that any linear classifier in $\mathbb{R}^d$ can be transformed into a homogeneous linear classifier in $\mathbb{R}^{d+1}$ by the following transformation

$$\mathbf{x} \to (\mathbf{x}; 1)$$
$$\{\mathbf{w}, b\} \to (\mathbf{w}; b)$$

(where by ";" we mean that we append a row to a column vector). In other words, we append "1" to the $\mathbf{x}$ vector and combine $\mathbf{w}$ and b into one vector in $\mathbb{R}^{d+1}$. Note that $\mathbf{w}^T\mathbf{x} + b = (\mathbf{w}; b)^T(\mathbf{x}; 1)$ and, therefore, the predictions of the transformed model are identical to predictions of the original model. Through this transformation any learning algorithm for homogeneous classifiers can be directly applied to learning non-homogeneous classifiers.

Hypothesis space The hypothesis space in linear classification is the space of all possible separating hyperplanes. If we are talking about homogeneous linear classifiers then it is restricted to hyperplanes passing through the origin. Thus, for homogeneous linear classifiers $\mathcal{H} = \mathbb{R}^d$ and for general linear classifiers $\mathcal{H} = \mathbb{R}^{d+1}$.

Perceptron algorithm Perceptron is the simplest algorithm for learning homogeneous separating hyperplanes. It operates under the ***assumption that the data are separable by a homogeneous hyperplane***, meaning that there exists a hyperplane passing through the origin that perfectly separates positive points from negative.

Algorithm 2 Perceptron

1: **Input:** A training set $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$
2: **Initialization:** $\mathbf{w}_1 = \mathbf{0}$ (where $\mathbf{0}$ is the zero vector)
3: $t = 1$
4: **while** exists $(\mathbf{x}_{i_t}, y_{i_t})$, such that $y_{i_t}(\mathbf{w}_t^T \mathbf{x}_{i_t}) \leq 0$ **do**
5: $\quad \mathbf{w}_{t+1} = \mathbf{w}_t + y_{i_t}\mathbf{x}_{i_t}$
6: $\quad t = t + 1$
7: **end while**
8: **Return:** $\mathbf{w}_t$

Note that a point $(\mathbf{x}, y)$ is classified correctly if $y\mathbf{w}^T\mathbf{x} > 0$ and misclassified if $y\mathbf{w}^T\mathbf{x} \leq 0$. Thus, the selection step (line 4 in the pseudocode) picks a misclassified point, as long as there exists such. The update step (line 5 in the pseudocode) rotates the hyperplane $\mathbf{w}$, so that the classification is "improved". Specifically, the following property is satisfied: if $(\mathbf{x}_{i_t}, y_{i_t})$ is the point selected at step t then $y_{i_t}\mathbf{w}_{t+1}^T\mathbf{x}_{i_t} > y_{i_t}\mathbf{w}_t^T\mathbf{x}_{i_t}$ (verification of this property is left as an exercise to the reader). Note this property does not guarantee that after the update $\mathbf{w}_{t+1}$ will classify $(\mathbf{x}_{i_t}, y_{i_t})$ correctly. But it will rotate in the right direction and after sufficiently many updates $(\mathbf{x}_{i_t}, y_{i_t})$ will end up on the right side of the hyperplane. Also note that while the classification of $(\mathbf{x}_{i_t}, y_{i_t})$ is improved, it may go the opposite way for other points. As long as the data are linearly separable, the algorithm will eventually find the separation.

The algorithm does not specify the order in which misclassified points are selected. Two natural choices are sequential and random. We leave it as an exercise to the reader to check which of the two choices leads to faster convergence of the algorithm.

2.5 Exercises

Exercise 2.1 (*Make your own*)**.** Imagine that you would like to write a learning algorithm that would predict the final grade of a student in the Machine Learning course based on their profile, for example, their grades in prior courses, their study program, etc. Such an algorithm would have been extremely useful: we would save significant time on grading and predict the final grade when the student just signs up for the course. We expect that the students would also appreciate such service and avoid all the worries about their grades. Anyhow, if you were to make such an algorithm.

1. What profile information would you collect and what would be the sample space $\mathcal{X}$?
2. What would be the label space $\mathcal{Y}$?
3. How would you define the loss function $\ell(y, \hat{y})$?
4. Assuming that you want to apply K-Nearest-Neighbors, how would you define the distance measure $d(x, x')$?
5. How would you evaluate the performance of your algorithm? (In terms of the loss function you have defined earlier.)
6. Assuming that you have achieved excellent performance and decided to deploy the algorithm, would you expect any issues coming up? How could you alleviate them?

There is no single right answer to the question. The main purpose is to help you digest the definitions we are working with. Your answer should be short, no more than 2-3 sentences for each bullet point. For example, it is sufficient to mention 2-3 items for the profile information, you should not make a page-long list.

Exercise 2.2 (*Digits Classification with K Nearest Neighbors*)**.** In this question you will implement and apply the K Nearest Neighbors learning algorithm to classify handwritten digits. You should make your own implementation (rather than use libraries), but it is allowed to use library functions for vector and matrix operations. Apart from implementation of the K-NN algorithm, the question aims to improve your skills of working with vector operations in Python.

Preparation

- Download `MNIST-5-6-Subset.zip` file.[2]
  The file contains:
  - `MNIST-5-6-Subset.txt`
  - `MNIST-5-6-Subset-Labels.txt`
  - `MNIST-5-6-Subset-Light-Corruption.txt`
  - `MNIST-5-6-Subset-Moderate-Corruption.txt`
  - `MNIST-5-6-Subset-Heavy-Corruption.txt`
- `MNIST-5-6-Subset.txt` is a space-separated file of real numbers (written as text). It contains a 784×1877 matrix, written column-by-column (the first 784 numbers in the file correspond to the first column; the next 784 numbers are the second column, and so on).
  - Each column in the matrix is a 28×28 grayscale image of a digit, stored column-by-column (the first 28 out of 784 values correspond to the first column of the 28×28 image, the next 28 values correspond to the second column, and so on). In the appendix you can find a Python script that serves as an illustration of one way to load and visualize the data.
- `MNIST-5-6-Subset-Labels.txt` is a space-separated file of 1877 integers. The numbers label the images in `MNIST-5-6-Subset.txt` file: the first number ("5") is the number drawn in the image corresponding to the first column; the second number corresponds to the second column, and so on.
- `Light-Corruption`, `Moderate-Corruption`, and `Heavy-Corruption` are corrupted versions of the digits in `MNIST-5-6-Subset.txt`, the order is preserved. It is a good idea to visualize the corrupted images to get a feeling of the corruption magnitude.

Detailed Instructions We pursue several goals in this question:

- Get your hands on implementation of K-NN and practice vector operations in Python.
- Explore fluctuations of the validation error as a function of the size of a validation set. (**Task#1**)
- Explore the impact of data corruption on the optimal value of K. (**Task#2**, optional)

IMPORTANT: Please, remember to include axis labels, legends and appropriate titles in your plots!

Task #1 In order to explore fluctuations of the validation error as a function of the size of the validation set, we use the following construction:

- Implement a Python function `knn(training_points, training_labels, test_points, test_labels)` that takes as input a $d \times m$ matrix of training points `training_points`, where m is the number of training points and d is the dimension of each point ($d = 784$ in the case of digits), a vector `training_labels` of the corresponding m training labels, a $d \times n$ matrix `test_points` of n test points, and their labels `test_labels` (you will need to convert the labels from $\{5, 6\}$ to $\{-1, 1\}$). The function should return a vector of length m, where each element represents the average error of K-NN on the test points for the corresponding value of K for $K \in \{1, \ldots, m\}$. **Include a printout**

[2]The file can be downloaded at `https://drive.google.com/file/d/1ztcOra97a6-udEvOB1QbgFXIMypcNXbs/view`. It is a subset of digits '5' and '6' from the famous MNIST dataset (LeCun et al., 1994).

of your implementation of the function in the report. (Only this function, not all of your code, the complete code should be included in the `.zip` file.) Ideally, the function should have no for-loops, check the practical advice at the end of the question.

- Use the first m digits for training the K-NN model. Take $m = 50$.
- Consider five validation sets, where for $i \in \{1, \ldots, 5\}$ the set i consists of digits $m + (i-1) \times n + 1, \ldots, m + i \times n$, and where n is the size of each of the five validation sets (we will specify n in a moment). The data split is visualized below.

Training data (m points)	Validation set #1 (n points)	Validation set #2 (n points)	Validation set #3 (n points)	Validation set #4 (n points)	Validation set #5 (n points)

- Calculate the validation error for each of the sets as a function of K, for $K \in \{1, \ldots, m\}$. Plot the validation error for each of the five validation sets as a function of K in the same figure (you will get five lines in the figure).
- Execute the experiment above with $n \in \{10, 20, 40, 80\}$. You will get four figures for the four values of n, with five lines in each figure. **Include these four figures in your report.**
- Create a figure where for each $n \in \{10, 20, 40, 80\}$ you plot the variance of the validation error over the five validation sets, as a function of K. You will get four lines in this figure, one for each n. **Include this figure in your report.** (Clarification, in case you got confused: fix n and K, then you have five numbers corresponding to validation errors on the five validation sets. You should compute the variance of these five values. Now keep n fixed, take $K \in \{1, \ldots, m\}$, and compute the variance as a function of K, i.e., compute it for each K separately. This gives you one line. And then each $n \in \{10, 20, 40, 80\}$ gives you a line, so you get four lines.)
- What can you say about fluctuations of the validation error as a function of n? **Answer in the report.**
- What can you say about the prediction accuracy of K-NN as a function of K? **Answer in the report.**
- A high-level comment: a more common way of visualizing variation of outcomes of experiment repetitions is to plot the mean and error bars, but this form of visualization makes it too easy for humans to ignore the error bars and concentrate just on the mean, see the excellent book of Kahneman (2011). The visualization you are asked to provide in this question makes it hard to ignore the variation.

Task #2 (optional, not for submission) In order to explore the influence of corruptions on the performance of K-NN and on the optimal value of K, we use this construction:

- Take the uncorrupted set, take m as before and $n = 80$, and construct training and validation sets as above. Plot five lines for the five validation sets, as a function of K, for $K \in \{1, \ldots, m\}$. **Include this figure in your report.**
- Repeat the experiment with the `Light-Corruption` set (both training and test images should be taken from the lightly corrupted set), then with the `Moderate-Corruption` set, and then with the `Heavy-Corruption` set. **Include one figure for each of the corrupted sets in your report.**
- Discuss how corruption magnitude influences the prediction accuracy of K-NN and the optimal value of K. **Answer in the report.**

Optional, not for submission: You are very welcome to experiment further with the data.

Practical Advice Check Chapter E for practical advice on how to make a vectorized implementation of K-NN. In interpreter programming languages like Python using vectorized implementations is much more efficient than using for-loops.

Chapter 3

Concentration of Measure Inequalities

Concentration of measure inequalities are one of the main tools for analyzing learning algorithms. This chapter is devoted to a number of concentration of measure inequalities that form the basis for the results discussed in later chapters.

3.1 Markov's Inequality

Markov's Inequality is the simplest and relatively weak concentration inequality. Nevertheless, it forms the basis for many much stronger inequalities that we will see in the sequel, and for some distributions it is actually tight (see Exercise 3.1).

Theorem 3.1 (Markov's Inequality)**.** *For any non-negative random variable X and $\varepsilon > 0$:*

$$\mathbb{P}(X \geq \varepsilon) \leq \frac{\mathbb{E}[X]}{\varepsilon}.$$

Proof. Define a random variable $Y = \mathbb{1}(X \geq \varepsilon)$ to be the indicator function of whether X exceeds ε. Then $Y \leq \frac{X}{\varepsilon}$ (see Figure 3.1). Since Y is a Bernoulli random variable, $\mathbb{E}[Y] = \mathbb{P}(Y = 1)$ (see Appendix B). We have:

$$\mathbb{P}(X \geq \varepsilon) = \mathbb{P}(Y = 1) = \mathbb{E}[Y] \leq \mathbb{E}\left[\frac{X}{\varepsilon}\right] = \frac{\mathbb{E}[X]}{\varepsilon}.$$

Check yourself: where in the proof do we use non-negativity of X and strict positiveness of ε? □

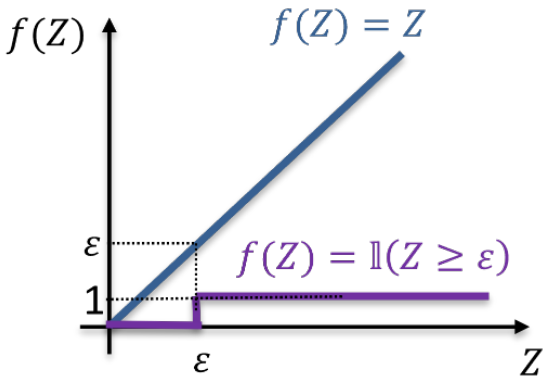


Figure 3.1: Relation between the identity function and the indicator function.

By denoting the right hand side of Markov's inequality by δ we obtain the following equivalent statement. For any non-negative random variable X:

$$\mathbb{P}\left(X \geq \frac{1}{\delta}\mathbb{E}\left[X\right]\right) \leq \delta.$$

Example. We would like to bound the probability that we flip a fair coin 10 times and obtain 8 or more heads. Let $X_1, \ldots, X_{10}$ be i.i.d. Bernoulli random variables with bias $\frac{1}{2}$. The question is equivalent to asking what is the probability that $\sum_{i=1}^{10} X_i \geq 8$. We have $\mathbb{E}\left[\sum_{i=1}^{10} X_i\right] = 5$ (the reader is invited to prove this statement formally) and by Markov's inequality

$$\mathbb{P}\left(\sum_{i=1}^{10} X_i \geq 8\right) \leq \frac{\mathbb{E}\left[\sum_{i=1}^{10} X_i\right]}{8} = \frac{5}{8}.$$

We note that even though Markov's inequality is weak, there are situations in which it is tight. We invite the reader to construct an example of a random variable for which Markov's inequality is tight.

3.2 Chebyshev's Inequality

Our next stop is Chebyshev's inequality, which exploits variance to obtain tighter concentration.

Theorem 3.2 (Chebyshev's inequality)**.** *For any $\varepsilon > 0$*

$$\mathbb{P}(|X - \mathbb{E}\left[X\right]| \geq \varepsilon) \leq \frac{\mathbb{V}\left[X\right]}{\varepsilon^2}.$$

Proof. The proof uses a transformation of a random variable. We have that $\mathbb{P}(|X - \mathbb{E}\left[X\right]| \geq \varepsilon) = \mathbb{P}\Big((X - \mathbb{E}\left[X\right])^2 \geq \varepsilon^2\Big)$, because the first statement holds if and only if the second holds. In addition, using Markov's inequality and the fact that $(X - \mathbb{E}\left[X\right])^2$ is a non-negative random variable we have

$$\mathbb{P}(|X - \mathbb{E}\left[X\right]| \geq \varepsilon) = \mathbb{P}\Big((X - \mathbb{E}\left[X\right])^2 \geq \varepsilon^2\Big) \leq \frac{\mathbb{E}\left[(X - \mathbb{E}\left[X\right])^2\right]}{\varepsilon^2} = \frac{\mathbb{V}\left[X\right]}{\varepsilon^2}.$$

Check yourself: where in the proof did we use the positiveness of ε? □

In order to illustrate the relative advantage of Chebyshev's inequality compared to Markov's consider the following example. Let $X_1, \ldots, X_n$ be n independent identically distributed Bernoulli random variables and let $\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$ be their average. We would like to bound the probability that $\hat{\mu}_n$ deviates from $\mathbb{E}\left[\hat{\mu}_n\right]$ by more than ε (this is the central question in machine learning). We have $\mathbb{E}\left[\hat{\mu}_n\right] = \mathbb{E}\left[X_1\right] = \mu$ and by independence of X_i-s and Theorem B.26 we have $\mathbb{V}\left[\hat{\mu}_n\right] = \frac{1}{n^2}\mathbb{V}\left[n\hat{\mu}_n\right] = \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{V}\left[X_i\right] = \frac{1}{n}\mathbb{V}\left[X_1\right]$. By Markov's inequality

$$\mathbb{P}(\hat{\mu}_n - \mathbb{E}\left[\hat{\mu}_n\right] \geq \varepsilon) = \mathbb{P}(\hat{\mu}_n \geq \mathbb{E}\left[\hat{\mu}_n\right] + \varepsilon) \leq \frac{\mathbb{E}\left[\hat{\mu}_n\right]}{\mathbb{E}\left[\hat{\mu}_n\right] + \varepsilon} = \frac{\mathbb{E}\left[X_1\right]}{\mathbb{E}\left[X_1\right] + \varepsilon}.$$

Note that as n grows the inequality stays the same. By Chebyshev's inequality we have

$$\mathbb{P}(\hat{\mu}_n - \mathbb{E}\left[\hat{\mu}_n\right] \geq \varepsilon) \leq \mathbb{P}(|\hat{\mu}_n - \mathbb{E}\left[\hat{\mu}_n\right]| \geq \varepsilon) \leq \frac{\mathbb{V}\left[\hat{\mu}_n\right]}{\varepsilon^2} = \frac{\mathbb{V}\left[X_1\right]}{n\varepsilon^2}.$$

Note that as n grows the right hand side of the inequality decreases at the rate of $\frac{1}{n}$. Thus, in this case Chebyshev's inequality is much tighter than Markov's and it illustrates that as the number of random variables grows the probability that their average significantly deviates from the expectation decreases. In the next section we show that this probability actually decreases at an exponential rate.

3.3 Hoeffding's Inequality

Hoeffding's inequality is a much more powerful concentration result.

Theorem 3.3 (Hoeffding's Inequality)**.** *Let $X_1, \ldots, X_n$ be independent real-valued random variables, such that for each $i \in \{1, \ldots, n\}$ there exist $a_i \leq b_i$, such that $X_i \in [a_i, b_i]$. Then for every $\varepsilon > 0$:*

$$\mathbb{P}\left(\sum_{i=1}^{n} X_i - \mathbb{E}\left[\sum_{i=1}^{n} X_i\right] \geq \varepsilon\right) \leq e^{-2\varepsilon^2 / \sum_{i=1}^{n}(b_i - a_i)^2} \tag{3.1}$$

and

$$\mathbb{P}\left(\sum_{i=1}^{n} X_i - \mathbb{E}\left[\sum_{i=1}^{n} X_i\right] \leq -\varepsilon\right) \leq e^{-2\varepsilon^2 / \sum_{i=1}^{n}(b_i - a_i)^2}. \tag{3.2}$$

By taking a union bound of the events in (3.1) and (3.2) we obtain the following corollary.

Corollary 3.4. *Under the assumptions of Theorem 3.3:*

$$\mathbb{P}\left(\left|\sum_{i=1}^{n} X_i - \mathbb{E}\left[\sum_{i=1}^{n} X_i\right]\right| \geq \varepsilon\right) \leq 2e^{-2\varepsilon^2 / \sum_{i=1}^{n}(b_i - a_i)^2}. \tag{3.3}$$

Equations (3.1) and (3.2) are known as "one-sided Hoeffding's inequalities" and (3.3) is known as "two-sided Hoeffding's inequality".

If we assume that X_i-s are identically distributed and belong to the $[0, 1]$ interval we obtain the following corollary (see Exercise 3.2).

Corollary 3.5. *Let $X_1, \ldots, X_n$ be independent random variables, such that $X_i \in [0, 1]$ and $\mathbb{E}[X_i] = \mu$ for all i, then for every $\varepsilon > 0$:*

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} X_i - \mu \geq \varepsilon\right) \leq e^{-2n\varepsilon^2} \tag{3.4}$$

and

$$\mathbb{P}\left(\mu - \frac{1}{n}\sum_{i=1}^{n} X_i \geq \varepsilon\right) \leq e^{-2n\varepsilon^2}. \tag{3.5}$$

Recall that by Chebyshev's inequality $\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$ converges to μ at the rate of n^{-1}. Hoeffding's inequality demonstrates that the convergence is actually much faster, at least at the rate of e^{-n}.

The proof of Hoeffding's inequality is based on Hoeffding's lemma.

Lemma 3.6 (Hoeffding's Lemma)**.** *Let X be a random variable, such that $X \in [a, b]$. Then for any $\lambda \in \mathbb{R}$:*

$$\mathbb{E}\left[e^{\lambda X}\right] \leq e^{\lambda \mathbb{E}[X] + \frac{\lambda^2 (b-a)^2}{8}}.$$

The function $f(\lambda) = \mathbb{E}\left[e^{\lambda X}\right]$ is known as the *moment generating function* of X, since $f'(0) = \mathbb{E}[X]$, $f''(0) = \mathbb{E}\left[X^2\right]$, and, more generally, $f^{(k)}(0) = \mathbb{E}\left[X^k\right]$. We provide a proof of the lemma immediately after the proof of Theorem 3.3.

Proof of Theorem 3.3. We prove the first inequality in Theorem 3.3. The second inequality follows by applying the first inequality to $-X_1, \ldots, -X_n$. The proof is based on Chernoff's bounding technique. For any $\lambda > 0$ the following holds:

$$\mathbb{P}\left(\sum_{i=1}^{n} X_i - \mathbb{E}\left[\sum_{i=1}^{n} X_i\right] \geq \varepsilon\right) = \mathbb{P}\left(e^{\lambda\left(\sum_{i=1}^{n} X_i - \mathbb{E}\left[\sum_{i=1}^{n} X_i\right]\right)} \geq e^{\lambda\varepsilon}\right) \leq \frac{\mathbb{E}\left[e^{\lambda\left(\sum_{i=1}^{n} X_i - \mathbb{E}\left[\sum_{i=1}^{n} X_i\right]\right)}\right]}{e^{\lambda\varepsilon}},$$

where the first step holds since $e^{\lambda x}$ is a monotonically increasing function for $\lambda > 0$ and the second step holds by Markov's inequality. We now take a closer look at the nominator:

$$\begin{aligned}
\mathbb{E}\left[e^{\lambda\left(\sum_{i=1}^n X_i - \mathbb{E}\left[\sum_{i=1}^n X_i\right]\right)}\right] &= \mathbb{E}\left[e^{\left(\sum_{i=1}^n \lambda(X_i - \mathbb{E}[X_i])\right)}\right] \\
&= \mathbb{E}\left[\prod_{i=1}^n e^{\lambda(X_i - \mathbb{E}[X_i])}\right] \\
&= \prod_{i=1}^n \mathbb{E}\left[e^{\lambda(X_i - \mathbb{E}[X_i])}\right] \qquad (3.6) \\
&\leq \prod_{i=1}^n e^{\lambda^2 (b_i - a_i)^2/8} \qquad (3.7) \\
&= e^{(\lambda^2/8)\sum_{i=1}^n (b_i - a_i)^2},
\end{aligned}$$

where (3.6) holds since $X_1, \ldots, X_n$ are independent and (3.7) holds by Hoeffding's lemma applied to a random variable $Z_i = X_i - \mathbb{E}[X_i]$ (note that $\mathbb{E}[Z_i] = 0$ and that $Z_i \in [a_i - \mu_i, b_i - \mu_i]$ for $\mu_i = \mathbb{E}[X_i]$). *Pay attention to the crucial role that independence of $X_1, \ldots, X_n$ plays in the proof! Without independence we would not have been able to exchange the expectation with the product and the proof would break down! And it is not just that the proof would break down, but it is actually possible to construct examples of dependent random variables for which the empirical mean does not converge to its expectation, see Exercise 3.4.* To complete the proof we substitute the bound on the expectation into the previous calculation and obtain:

$$\mathbb{P}\left(\sum_{i=1}^n X_i - \mathbb{E}\left[\sum_{i=1}^n X_i\right] \geq \varepsilon\right) \leq e^{(\lambda^2/8)\left(\sum_{i=1}^n (b_i - a_i)^2\right) - \lambda\varepsilon}.$$

This expression is minimized by

$$\lambda^* = \arg\min_\lambda e^{(\lambda^2/8)\left(\sum_{i=1}^n (b_i - a_i)^2\right) - \lambda\varepsilon} = \arg\min_\lambda \left((\lambda^2/8)\left(\sum_{i=1}^n (b_i - a_i)^2\right) - \lambda\varepsilon\right) = \frac{4\varepsilon}{\sum_{i=1}^n (b_i - a_i)^2}.$$

It is important to note that the best choice of λ does not depend on the sample. In particular, it allows to fix λ before observing the sample. By substituting λ^* into the calculation we obtain the result of the theorem. □

Proof of Lemma 3.6. Note that

$$\mathbb{E}\left[e^{\lambda X}\right] = \mathbb{E}\left[e^{\lambda(X - \mathbb{E}[X]) + \lambda\mathbb{E}[X]}\right] = e^{\lambda\mathbb{E}[X]} \times \mathbb{E}\left[e^{\lambda(X - \mathbb{E}[X])}\right].$$

Hence, it is sufficient to show that for any random variable Z with $\mathbb{E}[Z] = 0$ and $Z \in [a, b]$ we have:

$$\mathbb{E}\left[e^{\lambda Z}\right] \leq e^{\lambda^2 (b-a)^2/8}.$$

By convexity of the exponential function, for $z \in [a, b]$ we have:

$$e^{\lambda z} \leq \frac{z - a}{b - a} e^{\lambda b} + \frac{b - z}{b - a} e^{\lambda a}.$$

Let $p = -a/(b - a)$. Then:

$$\begin{aligned}
\mathbb{E}\left[e^{\lambda Z}\right] &\leq \mathbb{E}\left[\frac{Z - a}{b - a} e^{\lambda b} + \frac{b - Z}{b - a} e^{\lambda a}\right] \\
&= \frac{\mathbb{E}[Z] - a}{b - a} e^{\lambda b} + \frac{b - \mathbb{E}[Z]}{b - a} e^{\lambda a} \\
&= \frac{-a}{b - a} e^{\lambda b} + \frac{b}{b - a} e^{\lambda a} \\
&= \left(1 - p + p e^{\lambda(b-a)}\right) e^{-p\lambda(b-a)} \\
&= e^{\phi(u)},
\end{aligned}$$

where $u = \lambda(b-a)$ and $\phi(u) = -pu + \ln(1-p+pe^u)$ and we used the fact that $\mathbb{E}[Z] = 0$. It is easy to verify that the derivative of ϕ is

$$\phi'(u) = -p + \frac{p}{p + (1-p)e^{-u}}$$

and, therefore, $\phi(0) = \phi'(0) = 0$. Furthermore,

$$\phi''(u) = \frac{p(1-p)e^{-u}}{(p + (1-p)e^{-u})^2} \leq \frac{1}{4}.$$

By Taylor's theorem, $\phi(u) = \phi(0) + u\phi'(0) + \frac{u^2}{2}\phi''(\theta)$ for some $\theta \in [0, u]$. Thus, we have:

$$\phi(u) = \phi(0) + u\phi'(0) + \frac{u^2}{2}\phi''(\theta) = \frac{u^2}{2}\phi''(\theta) \leq \frac{u^2}{8} = \frac{\lambda^2 (b-a)^2}{8}.$$

□

3.3.1 Understanding Hoeffding's Inequality

Hoeffding's inequality involves three interconnected terms: n, ε, and $\delta = 2e^{-2n\varepsilon^2}$, which is the bound on the probability that the event under $\mathbb{P}()$ holds (for the purpose of the discussion we consider two-sided Hoeffding's inequality for random variables bounded in $[0, 1]$). We can fix any two of the three terms n, ε, and δ and then the relation $\delta = e^{-2n\varepsilon^2}$ provides the value of the third. Thus, we have

$$\begin{aligned}
\delta &= 2e^{-2n\varepsilon^2}, \\
\varepsilon &= \sqrt{\frac{\ln\frac{2}{\delta}}{2n}}, \\
n &= \frac{\ln\frac{2}{\delta}}{2\varepsilon^2}.
\end{aligned}$$

Overall, Hoeffding's inequality tells by how much the empirical average $\frac{1}{n}\sum_{i=1}^n X_i$ can deviate from its expectation μ, but the interplay between the three parameters provides several ways of seeing and using Hoeffding's inequality. For example, if the number of samples n is fixed (we have made a fixed number of experiments and now analyze what we can get from them), there is an interplay between the precision ε and confidence δ. We can request higher precision ε, but then we have to compromise on the confidence δ that the desired bound $\left|\frac{1}{n}\sum_{i=1}^n X_i - \mu\right| \leq \varepsilon$ holds. And the other way around: we can request higher confidence δ, but then we have to compromise on precision ε, i.e., we have to increase the allowed range $\pm\varepsilon$ around μ, where we expect to find the empirical average $\frac{1}{n}\sum_{i=1}^n X_i$.

As another example, we may have target precision ε and confidence δ and then the inequality provides us the number of experiments n that we have to perform in order to achieve the target.

It is often convenient to write the inequalities (3.4) and (3.5) with a fixed confidence in mind, thus we have

$$\begin{aligned}
\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^n X_i - \mu \geq \sqrt{\frac{\ln\frac{1}{\delta}}{2n}}\right) &\leq \delta, \\
\mathbb{P}\left(\mu - \frac{1}{n}\sum_{i=1}^n X_i \geq \sqrt{\frac{\ln\frac{1}{\delta}}{2n}}\right) &\leq \delta, \\
\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^n X_i - \mu\right| \geq \sqrt{\frac{\ln\frac{2}{\delta}}{2n}}\right) &\leq \delta.
\end{aligned}$$

(Pay attention that the $\ln 2$ factor in the last inequality comes from the union bound over the first two inequalities: if we want to keep the same confidence we have to compromise on precision.)

In many situations we are interested in the complimentary events. Thus, for example, we have

$$\mathbb{P}\left(\mu - \frac{1}{n}\sum_{i=1}^{n} X_i \leq \sqrt{\frac{\ln\frac{1}{\delta}}{2n}}\right) \geq 1 - \delta.$$

Careful reader may point out that the inequalities above should be strict ("$<$" and "$>$"). This is true, but if it holds for strict inequalities it also holds for non-strict inequalities ("$\leq$" and "$\geq$"). Since strict inequalities provide no practical advantage we will use the non-strict inequalities to avoid the headache of remembering which inequalities should be strict and which should not.

The last inequality essentially says that with probability at least $1 - \delta$ we have

$$\mu \leq \frac{1}{n}\sum_{i=1}^{n} X_i + \sqrt{\frac{\ln\frac{1}{\delta}}{2n}} \tag{3.8}$$

and this is how we will occasionally use it. Note that the random variable is $\frac{1}{n}\sum_{i=1}^{n} X_i$ and the right way of interpreting the above inequality is actually that with probability at least $1 - \delta$

$$\frac{1}{n}\sum_{i=1}^{n} X_i \geq \mu - \sqrt{\frac{\ln\frac{1}{\delta}}{2n}},$$

i.e., the probability is over $\frac{1}{n}\sum_{i=1}^{n} X_i$ and not over μ. However, many generalization bounds that we study in Chapter 4 are written in the first form in the literature and we follow the tradition.

3.4 Sampling Without Replacement

Let $X_1, \ldots, X_n$ be a sequence of random variables *sampled without replacement* from a finite set of values $\mathcal{X} = \{x_1, \ldots, x_N\}$ of size N. The random variables $X_1, \ldots, X_n$ are *dependent*. For example, if $\mathcal{X} = \{-1, +1\}$ and we sample two values then $X_1 = -X_2$. Since $X_1, \ldots, X_n$ are dependent, the concentration results from previous sections do not apply directly. However, the following result by Hoeffding (1963, Theorem 4), which we cite without a proof, allows to extend results for sampling with replacement to sampling without replacement.

Lemma 3.7. *Let $X_1, \ldots, X_n$ denote a random sample without replacement from a finite set $\mathcal{X} = \{x_1, \ldots, x_N\}$ of N real values. Let $Y_1, \ldots, Y_n$ denote a random sample with replacement from $\mathcal{X}$. Then for any continuous and convex function $f : \mathbb{R} \to \mathbb{R}$*

$$\mathbb{E}\left[f\left(\sum_{i=1}^{n} X_i\right)\right] \leq \mathbb{E}\left[f\left(\sum_{i=1}^{n} Y_i\right)\right].$$

In particular, the lemma can be used to prove Hoeffding's inequality for sampling without replacement.

Theorem 3.8 (Hoeffding's inequality for sampling without replacement)**.** *Let $X_1, \ldots, X_n$ denote a random sample without replacement from a finite set $\mathcal{X} = \{x_1, \ldots, x_N\}$ of N values, where each element x_i is in the $[0, 1]$ interval. Let $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$ be the average of the values in $\mathcal{X}$. Then for all $\varepsilon > 0$*

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} X_i - \mu \geq \varepsilon\right) \leq e^{-2n\varepsilon^2},$$
$$\mathbb{P}\left(\mu - \frac{1}{n}\sum_{i=1}^{n} X_i \geq \varepsilon\right) \leq e^{-2n\varepsilon^2}.$$

The proof is a minor adaptation of the proof of Hoeffding's inequality for sampling with replacement using Lemma 3.7 and is left as an exercise. (Note that it requires a small modification inside the proof, because Lemma 3.7 cannot be applied directly to the statement of Hoeffding's inequality.)

While formal proof requires a bit of work, intuitively the result is quite expected. Imagine the process of sampling without replacement. If the average of points sampled so far starts deviating from the mean of the values in $\mathcal{X}$, the average of points that are left in $\mathcal{X}$ deviates in the opposite direction and "applies extra force" to new samples to bring the average back to μ. In the limit when $n = N$ we are guaranteed to have the average of X_i-s being equal to μ.

3.5 Basics of Information Theory: Entropy, Relative Entropy, and the Method of Types

In this section we briefly introduce a number of basic concepts from information theory that are very useful for deriving concentration inequalities. Specifically, we introduce the notions of entropy and relative entropy (Cover and Thomas, 2006, Chapter 2) and some basic tools from the method of types (Cover and Thomas, 2006, Chapter 11).

3.5.1 Entropy

We start with the definition of entropy.

Definition 3.9 (Entropy)**.** *Let $p(x)$ be a distribution of a discrete random variable X taking values in a finite set $\mathcal{X}$. We define the* entropy *of p as:*

$$\mathrm{H}(p) = -\sum_{x \in \mathcal{X}} p(x) \ln p(x).$$

We use the convention that $0 \ln 0 = 0$ (which is justified by continuity of $z \ln z$, since $z \ln z \to 0$ as $z \to 0$).

We have special interest in Bernoulli random variables.

Definition 3.10 (Bernoulli random variable)**.** *X is a Bernoulli random variable with bias p if X accepts values in $\{0, 1\}$ with $\mathbb{P}(X = 0) = 1 - p$ and $\mathbb{P}(X = 1) = p$.*

Note that expectation of a Bernoulli random variable is equal to its bias:

$$\mathbb{E}[X] = 0 \times \mathbb{P}(X = 0) + 1 \times \mathbb{P}(X = 1) = \mathbb{P}(X = 1) = p.$$

With a slight abuse of notation we specialize the definition of entropy to Bernoulli random variables.

Definition 3.11 (Binary entropy)**.** *Let p be a bias of Bernoulli random variable X. We define the entropy of p as*

$$\mathrm{H}(p) = -p \ln p - (1 - p) \ln(1 - p).$$

Note that when we talk about Bernoulli random variables p denotes the bias of the random variable and when we talk about more general random variables p denotes the complete distribution.

Entropy is one of the central quantities in information theory and it has numerous applications. We start by using binary entropy to bound binomial coefficients.

Lemma 3.12.

$$\frac{1}{n+1} e^{n \,\mathrm{H}\left(\frac{k}{n}\right)} \leq \binom{n}{k} \leq e^{n \,\mathrm{H}\left(\frac{k}{n}\right)}.$$

(Note that $\frac{k}{n} \in [0, 1]$ and $\mathrm{H}\left(\frac{k}{n}\right)$ in the lemma is the binary entropy.)

Proof. By the binomial formula we know that for any $p \in [0, 1]$:

$$\sum_{i=0}^{n} \binom{n}{i} p^i (1 - p)^{n-i} = 1. \tag{3.9}$$

We start with the upper bound. Take $p = \frac{k}{n}$. Since the sum is larger than any individual term, for the k-th term of the sum we get:

$$\begin{aligned}
1 &\geq \binom{n}{k} p^k (1-p)^{n-k} \\
&= \binom{n}{k} \left(\frac{k}{n}\right)^k \left(1 - \frac{k}{n}\right)^{n-k} \\
&= \binom{n}{k} \left(\frac{k}{n}\right)^k \left(\frac{n-k}{n}\right)^{n-k} \\
&= \binom{n}{k} e^{k \ln \frac{k}{n} + (n-k) \ln \frac{n-k}{n}} \\
&= \binom{n}{k} e^{n\left(\frac{k}{n} \ln \frac{k}{n} + \frac{n-k}{n} \ln \frac{n-k}{n}\right)} \\
&= \binom{n}{k} e^{-n\,\mathrm{H}\left(\frac{k}{n}\right)}.
\end{aligned}$$

By changing sides of the inequality we obtain the upper bound.

For the lower bound it is possible to show that if we fix $p = \frac{k}{n}$ then $\binom{n}{k} p^k (1-p)^{n-k} \geq \binom{n}{i} p^i (1-p)^{n-i}$ for any $i \in \{0, \ldots, n\}$, see Cover and Thomas (2006, Example 11.1.3) for details. We also note that there are $n+1$ elements in the sum in equation (3.9). Again, take $p = \frac{k}{n}$, then

$$1 \leq (n+1) \max_i \binom{n}{i} \left(\frac{k}{n}\right)^i \left(\frac{n-k}{n}\right)^{n-i} = (n+1) \binom{n}{k} \left(\frac{k}{n}\right)^k \left(\frac{n-k}{n}\right)^{n-k} = (n+1) \binom{n}{k} e^{-n\,\mathrm{H}\left(\frac{k}{n}\right)},$$

where the last step follows the same steps as in the derivation of the upper bound. □

Lemma 3.12 shows that the number of configurations of chosing k out of n objects is directly related to the entropy of the imbalance $\frac{k}{n}$ between the number of objects that are selected (k) and the number of objects that are left out ($n-k$).

3.5.2 The Kullback-Leibler (KL) Divergence (Relative Entropy)

We now introduce an additional quantity, the *Kullback-Leibler (KL) divergence*, also known as *Kullback-Leibler distance* and as *relative entropy*.

Definition 3.13 (Relative entropy or Kullback-Leibler divergence)**.** *Let $p(x)$ and $q(x)$ be two probability distributions of a random variable X (or two probability density functions, if X is a continuous random variable), the* Kullback-Leibler divergence *or* relative entropy *is defined as:*

$$\mathrm{KL}(p\|q) = \mathbb{E}_p\left[\ln \frac{p(X)}{q(X)}\right] = \begin{cases} \sum_{x \in \mathcal{X}} p(x) \ln \frac{p(x)}{q(x)}, & \text{if } \mathcal{X} \text{ is discrete} \\ \int_{x \in \mathcal{X}} p(x) \ln \frac{p(x)}{q(x)} dx, & \text{if } \mathcal{X} \text{ is continuous} \end{cases}.$$

We use the convention that $0 \ln \frac{0}{0} = 0$ and $0 \ln \frac{0}{q} = 0$ and $p \ln \frac{p}{0} = \infty$.

We specialize the definition to Bernoulli distributions.

Definition 3.14 (Binary kl-divergence)**.** *Let p and q be biases of two Bernoulli random variables. The* binary kl divergence *is defined as:*

$$\mathrm{kl}(p\|q) = \mathrm{KL}([1-p, p]\|[1-q, q]) = p \ln \frac{p}{q} + (1-p) \ln \frac{1-p}{1-q}.$$

KL divergence is the central quantity in information theory. Although it is not a distance measure, because it does not satisfy the triangle inequality, it is the right way of measuring distances between probability distributions. This is illustrated by the following example.

Example 3.15. Let $X_1, \ldots, X_n$ be an i.i.d. sample of n Bernoulli random variables with bias p and let $\frac{1}{n}\sum_{i=1}^{n} X_i$ be the empirical bias of the sample. (Note that $\frac{1}{n}\sum_{i=1}^{n} X_i \in \left\{0, \frac{1}{n}, \frac{2}{n}, \ldots, \frac{n}{n}\right\}$.) Then for $p \in (0,1)$

$$\begin{aligned}
\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} X_i = \frac{k}{n}\right) &= \binom{n}{k} p^k (1-p)^{n-k} \\
&= \binom{n}{k} e^{-n\,\mathrm{H}\left(\frac{k}{n}\right)} e^{n\,\mathrm{H}\left(\frac{k}{n}\right)} e^{n\left(\frac{k}{n}\ln p + \frac{n-k}{n}\ln(1-p)\right)} \\
&= \binom{n}{k} e^{-n\,\mathrm{H}\left(\frac{k}{n}\right)} e^{-n\,\mathrm{kl}\left(\frac{k}{n}\|p\right)} \qquad (3.10)
\end{aligned}$$

By Lemma 3.12 we have $\frac{1}{n+1} \leq \binom{n}{k} e^{-n\,\mathrm{H}\left(\frac{k}{n}\right)} \leq 1$, which gives

$$\frac{1}{n+1} e^{-n\,\mathrm{kl}\left(\frac{k}{n}\|p\right)} \leq \mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} X_i = \frac{k}{n}\right) \leq e^{-n\,\mathrm{kl}\left(\frac{k}{n}\|p\right)}. \qquad (3.11)$$

Thus, $\mathrm{kl}\left(\frac{k}{n}\|p\right)$ governs the probability of observing empirical bias $\frac{k}{n}$ when the true bias is p. It is easy to verify that $\mathrm{kl}(p\|p) = 0$, and it is also possible to show that $\mathrm{kl}(\hat{p}\|p)$ is convex in $\hat{p}$, and that $\mathrm{kl}(\hat{p}\|p) \geq 0$ (Cover and Thomas, 2006). And so, the probability of empirical bias is maximized when it coincides with the true bias.

Properties of the KL and kl Divergences

The KL divergence between two probability distributions is always non-negative.

Theorem 3.16 (Nonnegativity of KL (Cover and Thomas, 2006, Theorem 2.3.6))**.** *Let $p(x)$ and $q(x)$ be two probability distributions. Then*

$$\mathrm{KL}(p\|q) \geq 0$$

with equality if and only if $p(x) = q(x)$ for all x.

Corollary 3.17 (Nonnegativity of kl)**.** *For $p, q \in [0,1]$*

$$\mathrm{kl}(p\|q) \geq 0$$

with equality if and only if $p = q$.

The KL divergence is also convex.

Theorem 3.18 (Convexity of KL (Cover and Thomas, 2006, Theorem 2.7.2))**.** $\mathrm{KL}(p\|q)$ *is convex in the pair (p, q); that is, if (p_1, q_1) and (p_2, q_2) are two pairs of probability mass functions, then*

$$\mathrm{KL}(\lambda p_1 + (1-\lambda)p_2 \| \lambda q_1 + (1-\lambda)q_2) \leq \lambda\,\mathrm{KL}(p_1\|q_1) + (1-\lambda)\,\mathrm{KL}(p_2\|q_2)$$

for all $0 \leq \lambda \leq 1$.

Corollary 3.19 (Convexity of kl)**.** *For $p_1, q_1, p_2, q_2 \in [0,1]$*

$$\mathrm{kl}(\lambda p_1 + (1-\lambda)p_2 \| \lambda q_1 + (1-\lambda)q_2) \leq \lambda\,\mathrm{kl}(p_1\|q_1) + (1-\lambda)\,\mathrm{kl}(p_2\|q_2)$$

for all $0 \leq \lambda \leq 1$.

3.6 The kl Inequality

Example 3.15 shows that kl can be used to bound the empirical bias when the true bias is known. But in machine learning we are usually interested in the inverse problem - how to infer the true bias p when the empirical bias $\hat{p}$ is known. Next we demonstrate that this is also possible and that it leads to an inequality, which is tighter than Hoeffding's inequality. We start with a simple version of kl lemma based on one-line derivation. Then we provide a tight version of the kl lemma and a lower bound showing that it cannot be improved any further. We then use the kl lemma to derive a kl inequality, and also provide a tighter version of the kl inequality, which is not based on the kl lemma. And we finish with relaxations of the kl inequality, which provide an intuitive interpretation of its implication.

3.6.1 A Simple Version of the kl Lemma

Lemma 3.20 (Simple kl Lemma)**.** *Let $X_1, \dots, X_n$ be i.i.d. Bernoulli with bias p and let $\hat{p} = \frac{1}{n}\sum_{i=1}^n X_i$ be the empirical bias. Then*

$$\mathbb{E}\left[e^{n\,\mathrm{kl}(\hat{p}\|p)}\right] \leq n+1.$$

Proof. For $p \in (0,1)$

$$\mathbb{E}\left[e^{n\,\mathrm{kl}(\hat{p}\|p)}\right] = \sum_{k=0}^{n} \mathbb{P}\left(\hat{p} = \frac{k}{n}\right) e^{n\,\mathrm{kl}\left(\frac{k}{n}\|p\right)} \leq \sum_{k=0}^{n} e^{-n\,\mathrm{kl}\left(\frac{k}{n}\|p\right)} e^{n\,\mathrm{kl}\left(\frac{k}{n}\|p\right)} = n+1,$$

where the inequality was derived in equation 3.11. For $p \in \{0,1\}$ we have $\mathbb{E}\left[e^{n\,\mathrm{kl}(\hat{p}\|o)}\right] = 1$, so the inequality is satisfied trivially. □

3.6.2 A Tight Version of the kl Lemma

In this section we provide a tight versions of the kl lemma. The improvement is based on a tighter control of the binomial coefficients, which is achieved by using Stirling's approximation of the factorial, $\sqrt{2\pi n}\left(\frac{n}{e}\right)^n \leq n! \leq \sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\frac{1}{12n}}$. The result is due to Wozengraft and Reiffen (1961).

Lemma 3.21. *For $1 \leq k \leq n-1$*

$$\frac{1}{2}\sqrt{\frac{n}{2k(n-k)}}e^{n\,\mathrm{H}\left(\frac{k}{n}\right)} \leq \binom{n}{k} \leq \frac{e^{\frac{1}{12n}}}{\sqrt{2\pi}}\sqrt{\frac{n}{k(n-k)}}e^{n\,\mathrm{H}\left(\frac{k}{n}\right)}.$$

The upper bound can be simplified using $\frac{e^{\frac{1}{12n}}}{\sqrt{2\pi}} < \frac{1}{2}$ for $n \geq 1$.

Proof. By Stirling's approximation

$$\begin{aligned}\binom{n}{k} &\leq \frac{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\frac{1}{12n}}}{\sqrt{2\pi k}\left(\frac{k}{e}\right)^k \sqrt{2\pi(n-k)}\left(\frac{n-k}{e}\right)^{n-k}} \\ &= \frac{e^{\frac{1}{12n}}}{\sqrt{2\pi}}\sqrt{\frac{n}{k(n-k)}}\frac{1}{\left(\frac{k}{n}\right)^k\left(\frac{n-k}{n}\right)^{n-k}} \\ &= \frac{e^{\frac{1}{12n}}}{\sqrt{2\pi}}\sqrt{\frac{n}{k(n-k)}}e^{n\,\mathrm{H}\left(\frac{k}{n}\right)}.\end{aligned}$$

The lower bound is derived in a similar way, see Cover and Thomas (2006, Lemma 17.5.1). □

By combining Theorem 3.21 with Equation (3.10), for $1 \leq k \leq n-1$

$$\frac{1}{2}\sqrt{\frac{n}{2k(n-k)}}e^{-n\,\mathrm{kl}\left(\frac{k}{n}\|p\right)} \leq \mathbb{P}\left(\frac{1}{n}\sum_{i=1}^n X_i = \frac{k}{n}\right) \leq \frac{e^{\frac{1}{12n}}}{\sqrt{2\pi}}\sqrt{\frac{n}{k(n-k)}}e^{-n\,\mathrm{kl}\left(\frac{k}{n}\|p\right)}.$$

The refinement can be used to tighten Theorem 3.20.

Lemma 3.22 (kl Lemma (Maurer, 2004))**.** *Let $X_1, \dots, X_n$ be i.i.d. with $X_1 \in [0,1]$, $\mathbb{E}[X_1] = p$, and $\hat{p} = \frac{1}{n}\sum_{i=1}^n X_i$. Then*

$$\mathbb{E}\left[e^{n\,\mathrm{kl}(\hat{p}\|p)}\right] \leq 2\sqrt{n}.$$

The proof of Theorem 3.22 is based on two auxiliary results. The first is a technical bound on a summation.

Lemma 3.23 ((Maurer, 2004, Lemma 4))**.** *For $n \geq 2$*

$$1 \leq \sum_{k=1}^{n-1} \frac{1}{\sqrt{k(n-k)}} \leq \pi.$$

The second result allows to extend results for Bernoulli random variables to random variables bounded in the $[0,1]$ interval.

Lemma 3.24 ((Maurer, 2004, Lemma 3))**.** *Let $X_1, \dots, X_n$ be i.i.d. with $X_1 \in [0,1]$, and let $Y_1, \dots, Y_n$ be i.i.d. Bernoulli, such that $\mathbb{E}[Y_1] = \mathbb{E}[X_1]$. Then for any convex function $f : [0,1]^n \to \mathbb{R}$*

$$\mathbb{E}[f(X_1, \dots, X_n)] \leq \mathbb{E}[f(Y_1, \dots, Y_n)].$$

The proof of Theorem 3.22 acts the same way as the proof of Theorem 3.20, just using the tighter bound on the binomial coefficients.

Proof of Theorem 3.22. We prove the result for Bernoulli random variables. The extension to general random variables bounded in $[0,1]$ follows by Theorem 3.24. For Bernoulli random variables and $p \in (0,1)$ we have

$$\mathbb{E}\left[e^{n \operatorname{kl}(\hat{p}\|p)}\right] = \sum_{k=0}^{n} \mathbb{P}\left(\hat{p} = \frac{k}{n}\right) e^{n \operatorname{kl}(\frac{k}{n}\|p)} = \sum_{k=0}^{n} \binom{n}{k} e^{-n \operatorname{H}(\frac{k}{n})} e^{-n \operatorname{kl}(\frac{k}{n}\|p)} e^{n \operatorname{kl}(\frac{k}{n}\|p)} = \sum_{k=0}^{n} \binom{n}{k} e^{-n \operatorname{H}(\frac{k}{n})}, \tag{3.12}$$

where the middle equality is by (3.10). By Theorem 3.21, for $n \geq 3$ we have

$$\sum_{k=0}^{n} \binom{n}{k} e^{-n \operatorname{H}(\frac{k}{n})} = 2 + \sum_{k=1}^{n-1} \binom{n}{k} e^{-n \operatorname{H}(\frac{k}{n})} \leq 2 + e^{\frac{1}{12n}} \sqrt{\frac{n}{2\pi}} \sum_{k=1}^{n-1} \frac{1}{\sqrt{k(n-k)}} \leq 2 + e^{\frac{1}{12n}} \sqrt{\frac{\pi n}{2}},$$

where the last inequality is by Theorem 3.23. For $n \geq 8$ we have $2 + e^{\frac{1}{12n}} \sqrt{\frac{\pi n}{2}} \leq 2\sqrt{n}$, whereas for $n \in \{1, \dots, 7\}$ a direct calculation confirms that we also have $\sum_{k=0}^{n} \binom{n}{k} e^{-n \operatorname{H}(\frac{k}{n})} \leq 2\sqrt{n}$. For $p \in \{0,1\}$ we have $\mathbb{E}\left[e^{n \operatorname{kl}(\hat{p}\|p)}\right] = 1$, so the lemma holds trivially. □

The next result shows that the kl Lemma (Theorem 3.22) cannot be improved much further.

Lemma 3.25 (kl Lemma - Lower Bound (Maurer, 2004))**.** *Let $X_1, \dots, X_n$ be i.i.d. Bernoulli with $\mathbb{E}[X_1] = p$ and $\hat{p} = \frac{1}{n} \sum_{i=1}^{n} X_i$. Then for $p \in (0,1)$*

$$\mathbb{E}\left[e^{n \operatorname{kl}(\hat{p}\|p)}\right] \geq \sqrt{n}.$$

Proof. Starting from (3.12) and applying Theorem 3.21, for $p \in (0,1)$ and $n \geq 3$ we have

$$\mathbb{E}\left[e^{n \operatorname{kl}(\hat{p}\|p)}\right] = \sum_{k=0}^{n} \binom{n}{k} e^{-n \operatorname{H}(\frac{k}{n})} = 2 + \sum_{k=1}^{n-1} \binom{n}{k} e^{-n \operatorname{H}(\frac{k}{n})} \geq \frac{1}{2} \sqrt{\frac{n}{2}} \sum_{k=1}^{n-1} \frac{1}{\sqrt{k(n-k)}}.$$

The function $f(n) = \sum_{k=1}^{n-1} \frac{1}{\sqrt{k(n-k)}}$ is monotonically increasing with n. For $n \geq 88$ we have $f(n) > \sqrt{8}$, thus $f(n)\sqrt{\frac{n}{8}} \geq \sqrt{n}$. For $n \in \{1, \dots, 87\}$ a direct calculation confirms that $\sum_{k=0}^{n} \binom{n}{k} e^{-n \operatorname{H}(\frac{k}{n})} \geq \sqrt{n}$. □

3.6.3 kl Inequality

By combining the kl lemma with Markov's inequality, we obtain the kl inequality.

Theorem 3.26 (kl Inequality via kl Lemma)**.** *Let $X_1, \dots, X_n$ be i.i.d. with $X_1 \in [0,1]$, $\mathbb{E}[X_1] = p$, and $\hat{p} = \frac{1}{n} \sum_{i=1}^{n} X_i$. Then*

$$\mathbb{P}\left(\operatorname{kl}(\hat{p}\|p) \geq \frac{\ln \frac{2\sqrt{n}}{\delta}}{n}\right) \leq \delta.$$

Proof.

$$\mathbb{P}\left(\operatorname{kl}(\hat{p}\|p) \geq \frac{\ln \frac{2\sqrt{n}}{\delta}}{n}\right) = \mathbb{P}\left(e^{n \operatorname{kl}(\hat{p}\|p)} \geq \frac{2\sqrt{n}}{\delta}\right) \leq \frac{\delta}{2\sqrt{n}} \mathbb{E}\left[e^{n \operatorname{kl}(\hat{p}\|p)}\right] \leq \delta,$$

where the first inequality is by Markov's inequality, and the second inequality is by the kl lemma (Theorem 3.22). □

Even though Theorem 3.22 cannot be improved much further, it is possible to improve the kl inequality through a direct derivation that does not go through $\mathbb{E}\left[e^{n\,\mathrm{kl}(\hat{p}\|p)}\right]$.

Theorem 3.27 (kl Inequality (Langford, 2005, Foong et al., 2021, 2022)). *Let $X_1, \dots, X_n$ be i.i.d. with $X_1 \in [0,1]$, $\mathbb{E}[X_1] = p$, and $\hat{p} = \frac{1}{n}\sum_{i=1}^{n} X_i$. Then, for any $\delta \in (0,1)$:*

$$\mathbb{P}\left(\mathrm{kl}(\hat{p}\|p) \geq \frac{\ln \frac{1}{\delta}}{n}\right) \leq \delta.$$

We note that the direct derivation of the kl inequality that is behind Theorem 3.27 cannot be combined with PAC-Bayesian analysis that we study in Section 4.8. There we need to use Theorem 3.22 and pay the cost of $\ln 2\sqrt{n}$, as in Theorem 3.26. But in direct applications of the kl inequality, for example, in combination of the kl inequality with the Occam's razor, this cost can be avoided (see Exercise 4.7).

3.6.4 Relaxations of the kl-inequality: Pinsker's and refined Pinsker's inequalities

Theorem 3.27 implies that with probability at least $1-\delta$

$$\mathrm{kl}(\hat{p}\|p) \leq \frac{\ln \frac{1}{\delta}}{n}. \tag{3.13}$$

This leads to an implicit bound on p, which is not very intuitive and not always convenient to work with. In order to understand the behavior of the kl inequality better we use a couple of its relaxations. The first relaxation is known as Pinsker's inequality, see Cover and Thomas (2006, Lemma 11.6.1).

Lemma 3.28 (Pinsker's inequality).

$$\mathrm{KL}(p\|q) \geq \frac{1}{2}\|p-q\|_1^2,$$

where $\|p-q\|_1 = \sum_{x\in\mathcal{X}} |p(x)-q(x)|$ is the L_1-norm.

Corollary 3.29 (Pinsker's inequality for the binary kl divergence).

$$\mathrm{kl}(p\|q) \geq \frac{1}{2}\left(|p-q| + |(1-p)-(1-q)|\right)^2 = 2(p-q)^2. \tag{3.14}$$

By applying Corollary 3.29 to inequality (3.13), we obtain that with probability at least $1-\delta$

$$p \leq \hat{p} + \sqrt{\frac{\mathrm{kl}(\hat{p}\|p)}{2}} \leq \hat{p} + \sqrt{\frac{\ln \frac{1}{\delta}}{2n}}, \tag{3.15}$$

where the first inequality is a deterministic inequality following by (3.14), and the second inequality holds with probability at least $1-\delta$ by (3.13). Note that inequality (3.15) is exactly the same as Hoeffding's inequality in Equation (3.8) (in fact, one way of proving Hoeffding's inequality is by deriving it via the kl divergence). Therefore, the kl inequality is always at least as tight as Hoeffding's inequality. But since (3.15) was achieved by Pinsker's relaxation of the kl inequality, the unrelaxed kl inequality can be tighter than Hoeffding's inequality.

Next we show that for small values of $\hat{p}$ the kl inequality is significantly tighter than Hoeffding's inequality. For this we use refined Pinsker's inequality (Marton, 1996, 1997, Samson, 2000, Boucheron et al., 2013, Lemma 8.4).

Lemma 3.30 (Refined Pinsker's inequality).

$$\mathrm{kl}(p\|q) \geq \frac{(p-q)^2}{2\max\{p,q\}} + \frac{(p-q)^2}{2\max\{(1-p),(1-q)\}}.$$

Corollary 3.31 (Refined Pinsker's inequality). *If $q > p$ then*

$$\mathrm{kl}(p\|q) \geq \frac{(p-q)^2}{2q}.$$

Corollary 3.32 (Refined Pinsker's inequality - upper bound)**.** *If* $\mathrm{kl}(p\|q) \leq \varepsilon$ *then*

$$q \leq p + \sqrt{2p\varepsilon} + 2\varepsilon.$$

Corollary 3.33 (Refined Pinsker's inequality - lower bound)**.** *If* $\mathrm{kl}(p\|q) \leq \varepsilon$ *then*

$$q \geq p - \sqrt{2p\varepsilon}.$$

By applying Corollary 3.32 to inequality (3.13), we obtain that with probability at least $1-\delta$

$$p \leq \hat{p} + \sqrt{\frac{2\hat{p}\ln\frac{1}{\delta}}{n}} + \frac{2\ln\frac{1}{\delta}}{n}. \tag{3.16}$$

When $\hat{p}$ is close to zero, the latter inequality is significantly tighter than Hoeffding's inequality. It exhibits what is know as "fast convergence rate", where for small values of $\hat{p}$ it approaches p at the rate of $\frac{1}{n}$ rather than $\frac{1}{\sqrt{n}}$, as in Hoeffding's inequality. Similarly, Theorem 3.33 clearly illustrates that when $\hat{p}$ is close to zero, the convergence of $\hat{p}$ to p from below also has "fast convergence rate".

We note that the kl inequality is always at least as tight as any of its relaxations, and that although there is no analytic inversion of $\mathrm{kl}(\hat{p}\|p)$, it is possible to invert it numerically to obtain even tighter bounds than the relaxations above. We use $\mathrm{kl}^{-1^+}(\hat{p},\varepsilon) := \max\{p : p \in [0,1] \text{ and } \mathrm{kl}(\hat{p}\|p) \leq \varepsilon\}$ to denote the upper inverse of kl and $\mathrm{kl}^{-1^-}(\hat{p},\varepsilon) := \min\{p : p \in [0,1] \text{ and } \mathrm{kl}(\hat{p}\|p) \leq \varepsilon\}$ to denote the lower inverse of kl. Then under the conditions of Theorem 3.27

$$\mathbb{P}\left(p \geq \mathrm{kl}^{-1^+}\left(\hat{p}, \frac{1}{n}\ln\frac{1}{\delta}\right)\right) \leq \delta, \tag{3.17}$$

$$\mathbb{P}\left(p \leq \mathrm{kl}^{-1^-}\left(\hat{p}, \frac{1}{n}\ln\frac{1}{\delta}\right)\right) \leq \delta. \tag{3.18}$$

Since $\mathrm{kl}(\hat{p}\|p)$ is convex in p, the inverses can be found using binary search.

Finally, we remind the reader that the random variable in the kl inequality is $\hat{p}$. Therefore, the correct way to see all the inequalities above is as inequalities on $\hat{p}$ rather than p. I.e., it is $\hat{p}$ that does not deviate a lot from p with high probability, rather than p staying close to $\hat{p}$ with high probability.

3.7 Split-kl Inequality

The kl inequality in Theorem 3.27 is almost the tightest that can be achieved for sums of Bernoulli random variables. (It is possible to obtain a bit tighter bounds by analyzing the binomial distribution directly, but the extra gains are minor (Langford, 2005).). However, it can potentially be very lose for sums of random variables taking values within the $[0,1]$ interval. The reason is that the kl inequality first maps any random variable taking values in the $[0,1]$ interval to a Bernoulli random variable (a random variable taking values $\{0,1\}$) with identical expectation, and then bounds the concentration of the original random variables by concentration of the Bernoulli random variables, as in Theorem 3.24. Whenever the original random variables have small variance, and the corresponding Bernoulli random variables have large variance, this approach is unable to exploit the small variance. (See Exercise 3.6, where you are asked to prove that if we fix expectation of a random variable, then Bernoulli random variable has the highest variance out of all random variables taking values in the $[0,1]$ interval and having a target expectation.) As an extreme example, imagine random variables $X_1,\ldots,X_n$, which all take value $p=\frac{1}{2}$ with probability 1. Then for any $\varepsilon > 0$ we have $\mathbb{P}\big(\big|p - \frac{1}{n}\sum_{i=1}^n X_i\big| > \varepsilon\big) = 0$, whereas the kl inequality only guarantees convergence of $\hat{p} = \frac{1}{n}\sum_{i=1}^n X_i$ to p at the rate of $\sqrt{\frac{\ln\frac{1}{\delta}}{n}}$.

3.7.1 Split-kl Inequality for Discrete Random Variables

In order to address the issue above, Wu et al. (2024) have proposed a way to represent discrete random variables as a superposition of Bernoulli random variables, and then apply the kl inequality to the Bernoulli elements in the decomposition. This approach preserves the kl tightness for the decomposition

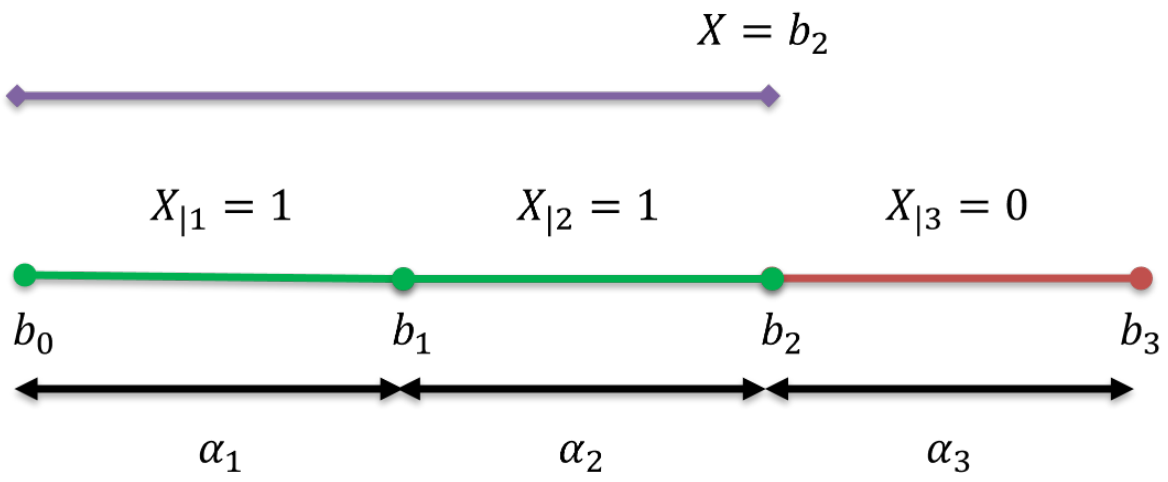


Figure 3.2: **Decomposition of a discrete random variable into a superposition of binary random variables.** The figure illustrates a decomposition of a discrete random variable X with domain of four values $b_0 < b_1 < b_2 < b_3$ into a superposition of three binary random variables, $X = b_0 + \sum_{j=1}^{3} \alpha_j X_{|j}$. A way to think about the decomposition is to compare it to a progress bar. In the illustration X takes value b_2, and so the random variables $X_{|1}$ and $X_{|2}$ corresponding to the first two segments "light up" (take value 1), whereas the random variable $X_{|3}$ corresponding to the last segment remains "turned off" (takes value 0). The value of X equals the sum of the lengths α_j of the "lighted up" segments. (The figure is borrowed from Wu et al. (2024).)

elements, and through it provides the combinatorial tightness of kl for general discrete random variables. The approach builds on an earlier work by Wu and Seldin (2022) for ternary random variables.

We now describe the decomposition. Let $X \in \{b_0, \dots, b_K\}$ be a $(K+1)$-valued random variable, where $b_0 < b_1 < \cdots < b_K$. For $j \in \{1, \dots, K\}$ define $X_{|j} = \mathbb{1}(X \geq b_j)$ and $\alpha_j = b_j - b_{j-1}$. Then $X = b_0 + \sum_{j=1}^{K} \alpha_j X_{|j}$, see Figure 3.2 for an illustration.

For a sequence $X_1, \dots, X_n$ of $(K+1)$-valued random variables with the same support, we let $X_{i|j} = \mathbb{1}(X_i \geq b_j)$ denote the elements of binary decomposition of X_i.

Theorem 3.34 (Split-kl inequality for discrete random variables (Wu et al., 2024)). *Let $X_1, \dots, X_n$ be i.i.d. random variables taking values in $\{b_0, \dots, b_K\}$ with $\mathbb{E}[X_i] = p$ for all i. Let $\hat{p}_{|j} = \frac{1}{n}\sum_{i=1}^{n} X_{i|j}$. Then for any $\delta \in (0,1)$:*

$$\mathbb{P}\left(p \geq b_0 + \sum_{j=1}^{K} \alpha_j \operatorname{kl}^{-1,+}\left(\hat{p}_{|j}, \frac{1}{n}\ln\frac{K}{\delta}\right)\right) \leq \delta.$$

Proof. Let $p_{|j} = \mathbb{E}\left[\hat{p}_{|j}\right]$, then $p = b_0 + \sum_{j=1}^{K} \alpha_j p_{|j}$ and

$$\mathbb{P}\left(p \geq b_0 + \sum_{j=1}^{K} \alpha_j \operatorname{kl}^{-1,+}\left(\hat{p}_{|j}, \frac{1}{n}\ln\frac{K}{\delta}\right)\right) \leq \mathbb{P}\left(\exists j : p_{|j} \geq \operatorname{kl}^{-1,+}\left(\hat{p}_{|j}, \frac{1}{n}\ln\frac{K}{\delta}\right)\right) \leq \delta,$$

where the first inequality is by the decomposition of p and the second inequality is by the union bound and (3.17). □

Since the kl inequalities provide almost the tightest bounds on the deviations of $\hat{p}_{|j}$ from $p_{|j}$ for each j individually, the split-kl inequality is almost the tightest that can be achieved for discrete random variables overall, as long as K and the corresponding $\ln K$ cost in the bound is not too large.

3.7.2 Split-kl Inequality for Bounded Continuous Random Variables

The split-kl inequality can also be applied to continuous random variables. Let $b_0 < b_1 < \cdots < b_K$ be an arbitrary split of an interval $[b_0, b_K]$ into K segments with $\alpha_j = b_j - b_{j-1}$ being the length of segment

j, and let $X \in [b_0, b_K]$ be a continuous random variable. Let

$$X_{|j} = \begin{cases} 0, & \text{if } X < b_{j-1}, \\ \frac{X - b_{j-1}}{\alpha_j}, & \text{if } b_{j-1} \le X \le b_j, \\ 1, & \text{if } X > b_j. \end{cases}$$

Then $X = b_0 + \sum_{j=1}^{K} \alpha_j X_{|j}$, and the split-kl inequality can be applied in exactly the same way as in the discrete case. The tightness of split-kl for continuous random variables depends on whether the probability mass is concentrated on or between the segment boundaries $b_0, \ldots, b_K$ and on the magnitude of the $\ln K$ cost of the union bound.

3.8 Bernstein's Inequality

Bernstein's inequality is one of the most broadly known tools that exploit small variance to obtain tighter concentration. As most concentration of measure inequalities we have seen so far, it is based on a bound on a moment generating function.

Lemma 3.35 (Bernstein's Lemma)**.** *Let Z be a random variable, such that $\mathbb{E}[Z] = 0$, $\mathbb{E}\left[Z^2\right] \le \nu$, and $Z \le b$. Then for any $\lambda \in \left(0, \frac{3}{b}\right)$*

$$\mathbb{E}\left[e^{\lambda Z}\right] \le \exp\left(\frac{\lambda^2 \nu}{2\left(1 - \frac{b\lambda}{3}\right)}\right).$$

Proof. For $x \le 0$ we have $e^x \le 1 + x + \frac{1}{2}x^2$ and, therefore, for $Z \le 0$ we have

$$e^{\lambda Z} \le 1 + \lambda Z + \frac{\lambda^2 Z^2}{2} \le 1 + \lambda Z + \frac{\lambda^2 Z^2}{2\left(1 - \frac{b\lambda}{3}\right)},$$

where the last inequality holds because $\lambda \in \left(0, \frac{3}{b}\right)$, and so $\left(1 - \frac{b\lambda}{3}\right) \in (0, 1)$.

For $Z > 0$ we use Taylor's expansion of the exponent, $e^x = 1 + x + \frac{x^2}{2} + \sum_{i=3}^{\infty} \frac{1}{i!} x^i$, which gives

$$\begin{aligned} e^{\lambda Z} &= 1 + \lambda Z + \frac{\lambda^2 Z^2}{2} + \sum_{i=3}^{\infty} \frac{1}{i!} (\lambda Z)^i \\ &\le 1 + \lambda Z + \frac{\lambda^2 Z^2}{2} + \frac{\lambda^2 Z^2}{2} \sum_{i=3}^{\infty} \left(\frac{1}{3}\lambda Z\right)^{i-2} \\ &= 1 + \lambda Z + \frac{\lambda^2 Z^2}{2} \sum_{i=0}^{\infty} \left(\frac{1}{3}\lambda Z\right)^{i} \\ &\le 1 + \lambda Z + \frac{\lambda^2 Z^2}{2} \sum_{i=0}^{\infty} \left(\frac{1}{3}\lambda b\right)^{i} \\ &= 1 + \lambda Z + \frac{\lambda^2 Z^2}{2\left(1 - \frac{b\lambda}{3}\right)}. \end{aligned}$$

By combining this with the earlier inequality, we obtain that for all Z

$$e^{\lambda Z} \le 1 + \lambda Z + \frac{\lambda^2 Z^2}{2\left(1 - \frac{b\lambda}{3}\right)}.$$

And, therefore,

$$\mathbb{E}\left[e^{\lambda Z}\right] \le 1 + \lambda \mathbb{E}[Z] + \frac{\lambda^2 \mathbb{E}\left[Z^2\right]}{2\left(1 - \frac{b\lambda}{3}\right)} \le 1 + \frac{\lambda^2 \nu}{2\left(1 - \frac{b\lambda}{3}\right)} \le \exp\left(\frac{\lambda^2 \nu}{2\left(1 - \frac{b\lambda}{3}\right)}\right),$$

where in the second step we used the facts that $\mathbb{E}[Z] = 0$ and $\mathbb{E}\left[Z^2\right] \le \nu$, and the last step is based on the inequality $1 + x \le e^x$ that holds for all x. □

Now we need a couple of technical results, which we leave as an exercise.

Lemma 3.36. *For $x > 0$ let $f(x) = 1 + x - \sqrt{1+2x}$, then for any $\varepsilon > 0$*

$$\sup_{\lambda \in \left(0, \frac{1}{c}\right)} \left(\varepsilon\lambda - \frac{\lambda^2 \nu}{2(1-c\lambda)} \right) = \frac{\nu}{c^2} f\left(\frac{c\varepsilon}{\nu}\right).$$

Lemma 3.37. *For $x > 0$ let $f(x) = 1 + x - \sqrt{1+2x}$, then $f^{-1}(x) = x + \sqrt{2x}$.*

And now we are ready to present Bernstein's inequality.

Theorem 3.38 (Bernstein's Inequality)**.** *Let $X_1, \ldots, X_n$ be independent random variables, such that for all i we have $\mathbb{E}[X_i] - X_i \le b$ and $\mathbb{V}[X_i] \le \nu$. Then*

$$\mathbb{P}\left(\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n X_i\right] \ge \frac{1}{n}\sum_{i=1}^n X_i + \sqrt{\frac{2\nu \ln\frac{1}{\delta}}{n}} + \frac{b \ln\frac{1}{\delta}}{3n} \right) \le \delta.$$

Note that if ν is close to zero, Bernstein's inequality provides "fast convergence rate", meaning that $\frac{1}{n}\sum_{i=1}^n X_i$ converges to $\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n X_i\right]$ at the rate of $\frac{1}{n}$ rather than at the rate of $\frac{1}{\sqrt{n}}$.

Proof. The proof is based on Chernoff's bounding technique and follows the same strategy as earlier proofs of Hoeffding's and kl inequalities, just now using Bernstein's lemma instead of Hoeffding's or kl lemma. Let $Z_i = \mathbb{E}[X_i] - X_i$, then $\mathbb{E}[Z_i] = 0$, $\mathbb{E}\left[Z_i^2\right] = \mathbb{V}[X_i] \le \nu$, and $Z_i \le b$. For any $\lambda \in \left(0, \frac{b}{3}\right)$ we have

$$\begin{aligned}
\mathbb{P}\left(\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n X_i\right] \ge \frac{1}{n}\sum_{i=1}^n X_i + \varepsilon \right) &= \mathbb{P}\left(\sum_{i=1}^n Z_i \ge n\varepsilon \right) \\
&= \mathbb{P}\left(e^{\lambda \sum_{i=1}^n Z_i} \ge e^{\lambda n \varepsilon} \right) \\
&\le e^{-\lambda n\varepsilon} \mathbb{E}\left[e^{\lambda \sum_{i=1}^n Z_i} \right] \\
&= e^{-\lambda n\varepsilon} \prod_{i=1}^n \mathbb{E}\left[e^{\lambda Z_i} \right] \\
&\le e^{-\lambda n\varepsilon} \prod_{i=1}^n \exp\left(\frac{\lambda^2 \nu}{2\left(1 - \frac{b\lambda}{3}\right)} \right) \\
&= \exp\left(-n \left(\lambda\varepsilon - \frac{\lambda^2 \nu}{2\left(1 - \frac{b\lambda}{3}\right)} \right) \right),
\end{aligned}$$

where the first inequality is by Markov's inequality and the second inequality is by Bernstein's lemma.

Since the bound holds for any $\lambda \in \left(0, \frac{b}{3}\right)$, we have

$$\mathbb{P}\left(\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n X_i\right] \ge \frac{1}{n}\sum_{i=1}^n X_i + \varepsilon \right) \le \exp\left(-n \sup_{\lambda \in \left(0, \frac{3}{b}\right)} \left(\lambda\varepsilon - \frac{\lambda^2 \nu}{2\left(1 - \frac{b\lambda}{3}\right)} \right) \right) = \exp\left(-naf(u)\right),$$

where $a = \frac{9\nu}{b^2}$, $u = \frac{b\varepsilon}{3\nu}$, $f(u) = 1 + u - \sqrt{1+2u}$, and the inequality follows by Theorem 3.36. Note that the right hand side of the inequality above is deterministic (independent of the random variable $\frac{1}{n}\sum_{i=1}^n X_i$), meaning that the optimal λ can be selected deterministically before observing the sample.

Finally, taking $\exp\left(-naf(u)\right) = \delta$ and using Theorem 3.37 to express ε in terms of δ, we obtain the statement in the theorem. □

3.9 Empirical Bernstein's Inequality

Bernstein's inequality (Theorem 3.38) assumes access to an upper bound ν on the variance. Empirical Bernstein's inequality presented in this section constructs a high-probability upper bound on the variance based on the sample, and then applies it within Bernstein's inequality, i.e., replaces ν with a high-probability upper bound on $\mathbb{V}[X]$.

Theorem 3.39 (Empirical Bernstein's Inequality (Maurer and Pontil, 2009)). *Let $X_1, \dots, X_n$ be independent identically distributed random variables taking values in $[0,1]$. Let $\hat{\nu}_n = \frac{1}{n(n-1)} \sum_{1 \le i < j \le n} (X_i - X_j)^2$. Then for any $\delta \in (0,1]$:*

$$\mathbb{P}\left(\mathbb{E}\left[\frac{1}{n} \sum_{i=1}^{n} X_i \right] \ge \frac{1}{n} \sum_{i=1}^{n} X_i + \sqrt{\frac{2\hat{\nu}_n \ln \frac{2}{\delta}}{n}} + \frac{7 \ln \frac{2}{\delta}}{3(n-1)} \right) \le \delta.$$

The $\ln \frac{2}{\delta}$ term in the bound comes from a union bound over empirical bound on the variance and a bound on expectation of X. Note that the overall cost of replacing the true variance with its estimate is relatively small. Gao and Zhou (2013) offer a slight improvement of the bound, showing that for $n \ge 5$ the denominator in the last term can be increased from $n-1$ to n. We omit a proof of the theorem, but in Exercise 3.12 you are given a chance to prove a slightly weaker version of the theorem yourself.

3.10 Unexpected Bernstein's Inequality

Empirical Bernstein's inequality (Theorem 3.39) proceeds by bounding the variance using empirical variance estimate, and then using Bernstein's inequality to bound the expectation using the variance estimate. Unexpected Bernstein's inequality proceeds by bounding the expectation directly via the first and the second empirical moments. It is based on the following inequality due to Fan et al. (2015, Equation (4.12)): for $z \le 1$ and $\lambda \in [0,1)$

$$e^{-\lambda z + z^2(\lambda + \ln(1-\lambda))} \le 1 - \lambda z. \tag{3.19}$$

The inequality can be used to prove the Unexpected Bernstein's lemma.

Lemma 3.40 (Unexpected Bernstein's lemma (Fan et al., 2015)). *Let X be a random variable bounded from above by $b > 0$. Then for all $\lambda \in \left[0, \frac{1}{b}\right)$*

$$\mathbb{E}\left[e^{\lambda(\mathbb{E}[X] - X) + \frac{b\lambda + \ln(1-b\lambda)}{b^2} X^2} \right] \le 1.$$

A proof is left as Exercise 3.13. Given the Unexpected Bernstein's lemma, we can use already well-established pipeline to prove the Unexpected Bernstein's inequality.

Theorem 3.41 (Unexpected Bernstein's Inequality (Fan et al., 2015, Mhammedi et al., 2019, Wu and Seldin, 2022)). *Let $X_1, \dots, X_n$ be independent identically distributed random variables bounded from above by b for $b > 0$. Let $\mu = \mathbb{E}[X_1]$, $\hat{\mu}_n = \frac{1}{n} \sum_{i=1}^{n} X_i$, and $\hat{s}_n = \frac{1}{n} \sum_{i=1}^{n} X_i^2$. Let $\psi(u) = u - \ln(1+u)$. Then for any $\lambda \in [0, 1/b)$ and $\delta \in (0,1]$:*

$$\mathbb{P}\left(\mu \ge \hat{\mu}_n + \frac{\psi(-\lambda b)}{\lambda b^2} \hat{s}_n + \frac{\ln \frac{1}{\delta}}{\lambda n} \right) \le \delta.$$

A proof is left as Exercise 3.14 and follows exactly the same steps as the proofs of Hoeffding's, kl, and Bernstein's inequalities. As already mentioned, the Unexpected Bernstein's inequality goes in one step from empirical first and second moments, $\hat{\mu}_n$ and $\hat{\nu}_n$, to a bound on μ. Note that in contrast to Hoeffding's and Bernstein's inequalities, the value of λ that minimizes the bound in Theorem 3.41 depends on $\hat{s}_n$, which is a random variable. Therefore, we cannot plug it into the bound. Instead, we can take a grid of λ values, a union bound over the grid, and then pick the best λ from the grid. Mhammedi et al. (2019) proposed to use the grid $\Lambda = \left\{1/2b, \dots, 1/(2^k b)\right\}$ for $k = \left\lceil \log_2 \left(\sqrt{n / \ln(1/\delta)}/2 \right) \right\rceil$, which works reasonably well in practice (Wu and Seldin, 2022). Formally, the bound then becomes:

$$\mathbb{P}\left(\mu \ge \hat{\mu}_n + \min_{\lambda \in \Lambda} \left(\frac{\psi(-\lambda b)}{\lambda b^2} \hat{s}_n + \frac{\ln \frac{k}{\delta}}{\lambda n} \right) \right) \le \sum_{\lambda \in \Lambda} \mathbb{P}\left(\mu \ge \hat{\mu}_n + \frac{\psi(-\lambda b)}{\lambda b^2} \hat{s}_n + \frac{\ln \frac{k}{\delta}}{\lambda n} \right) \le \delta.$$

(Note that $\ln \frac{1}{\delta}$ is replaced by $\ln \frac{k}{\delta}$ due to the union bound.)

3.11 Exercises

Exercise 3.1 (*Tightness of Markov's inequality*)**.** Let ε^* be fixed. Design an example of a random variable X for which

$$\mathbb{P}(X \geq \varepsilon^*) = \frac{\mathbb{E}[X]}{\varepsilon^*}.$$

Prove that the above equality holds for your random variable.

Guidance: "Design a random variable" means design a distribution by which the random variable is distributed. To design a distribution you should say what values the random variable can take and with what probabilities. You should construct an example, where the random variable can take strictly more than one value, because otherwise the example is trivial. But two values are actually sufficient to make a valid non-trivial example.

Exercise 3.2 (*The effect of normalization in Hoeffding's inequality*)**.** Prove that Theorem 3.5 (simplified Hoeffding's inequality for random variables in the $[0, 1]$ interval) follows from Theorem 3.3 (general Hoeffding's inequality). [Showing this for one of the two inequalities is sufficient.]

Remark: Some literature sources present Hoeffding's inequality in the normalized form of Theorem 3.5, whereas other sources present it in the unnormalized form of Theorem 3.3. The exercise aims to illustrate how to go from one to the other and back.

Exercise 3.3 (*Numerical comparison of Markov's, Chebyshev's, and Hoeffding's inequalities*)**.**

A. Make 1,000,000 repetitions of the experiment of drawing 20 i.i.d. Bernoulli random variables $X_1, \ldots, X_{20}$ with mean 0.5 and answer the following questions.

   (a) Plot the empirical frequency of observing $\frac{1}{20}\sum_{i=1}^{20} X_i \geq \alpha$ for $\alpha \in \{0.5, 0.55, 0.6, \ldots, 0.95, 1\}$.

   (b) Explain why the above granularity of α is sufficient. I.e., why, for example, taking $\alpha = 0.51$ will not provide any extra information about the experiment.

   (c) Use Markov's inequality to compute a bound on $\mathbb{P}\Big(\frac{1}{20}\sum_{i=1}^{20} X_i \geq \alpha\Big)$ and plot the bound in the same figure.

   (d) Use Chebyshev's inequality to compute a bound on $\mathbb{P}\Big(\frac{1}{20}\sum_{i=1}^{20} X_i \geq \alpha\Big)$ and plot the bound in the same figure. (You may have a problem calculating the bound for some values of α. If it happens and whenever the bound exceeds 1, replace it with the trivial bound of 1, because we know that probabilities are always bounded by 1.)

   (e) Use Hoeffding's inequality to compute a bound on $\mathbb{P}\Big(\frac{1}{20}\sum_{i=1}^{20} X_i \geq \alpha\Big)$ and plot the bound in the same figure.

   (f) Compare the four plots.

   (g) For $\alpha = 1$ and $\alpha = 0.95$ calculate the exact probability $\mathbb{P}\Big(\frac{1}{20}\sum_{i=1}^{20} X_i \geq \alpha\Big)$ and compare it with the Hoeffding's bound. (No need to add this one to the plot.)

B. Repeat the question with $X_1, \ldots, X_{20}$ with mean 0.1 (i.e., $\mathbb{E}[X_1] = 0.1$) and $\alpha \in \{0.1, 0.15, \ldots, 1\}$.

C. Discuss the results.

Do not forget to put axis labels and a legend in your plots.

Exercise 3.4 (*The role of independence*)**.** Design an example of identically distributed, but *dependent* Bernoulli random variables $X_1, \ldots, X_n$ (i.e., $X_i \in \{0, 1\}$), such that

$$\mathbb{P}\left(\left|\mu - \frac{1}{n}\sum_{i=1}^{n} X_i\right| \geq \frac{1}{2}\right) = 1,$$

where $\mu = \mathbb{E}[X_i]$.

Note that in this case $\frac{1}{n}\sum_{i=1}^{n} X_i$ does not converge to μ as n goes to infinity. The example shows that independence is crucial for convergence of empirical means to the expected values.

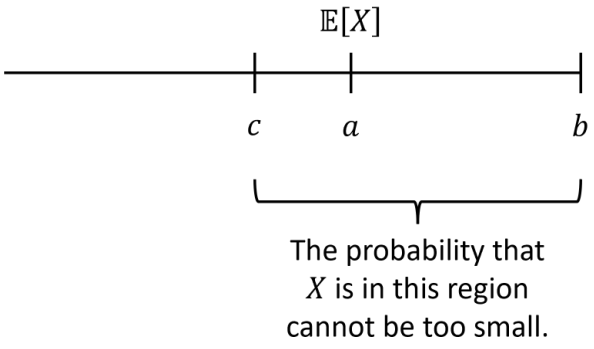


Figure 3.3: Illustration to Exercise 3.5.

Exercise 3.5 (*From a lower bound on the expectation to a lower bound on the probability*)**.** The question shows that if the expectation of a *bounded* random variable X is large, then with high probability X should take large values.

- Let X be a random variable that is always upper bounded by b. Let $c < a < b$. Prove that if $\mathbb{E}[X] \geq a$, then $\mathbb{P}(X \geq c) \geq \frac{a-c}{b-c}$.
- Indicate which part of the proof relies on the assumption that $X \leq b$.

Hint: see the illustration in Figure 3.3. The claim can be proved directly or by using Markov's inequality.

Optional: Let $c < a$. Construct an example of an *unbounded* random variable, such that $\mathbb{E}[X] \geq a$, but $\mathbb{P}(X \geq c)$ is small. The example shows that the boundedness assumption is crucial.

Exercise 3.6 (*Bernoulli Distribution Maximizes the Variance*)**.** Let X be a random variable taking values in the $[0, 1]$ interval, and let Y be a Bernoulli random variable (taking values $\{0, 1\}$), such that $\mathbb{E}[X] = \mathbb{E}[Y]$. Prove that $\mathbb{V}[Y] \geq \mathbb{V}[X]$.

Exercise 3.7 (*Asymmetry of the* kl *divergence*)**.** Prove that kl is asymmetric in its arguments by providing an example of p and q for which $\mathrm{kl}(p\|q) \neq \mathrm{kl}(q\|p)$.

Exercise 3.8 (*Numerical comparison of* kl *inequality with its relaxations and with Hoeffding's inequality*)**.** Let $X_1, \ldots, X_n$ be a sample of n independent Bernoulli random variables with bias $p = \mathbb{P}(X = 1)$. Let $\hat{p}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$ be the empirical average. In this question you make a numerical comparison of the relative power of various bounds on p we have studied. Specifically, we consider the following bounds:

A. **Hoeffding's inequality**: by Hoeffding's inequality, with probability at least $1 - \delta$:

$$p \leq \hat{p}_n + \sqrt{\frac{\ln \frac{1}{\delta}}{2n}}.$$

("The Hoeffding's bound", which you are asked to plot, refers to the right hand side of the inequality above.)

B. **The** kl **inequality**: The bound on p that follows by the kl inequality (Theorem 3.27).

Some guidance: There is no closed-form expression for computing $\mathrm{kl}^{-1^+}(\hat{p}_n, \varepsilon)$, so it has to be computed numerically. The function $\mathrm{kl}(\hat{p}_n\|p)$ is convex in p (you are very welcome to pick some value of $\hat{p}_n$ and plot $\mathrm{kl}(\hat{p}_n\|p)$ as a function of p to get an intuition about its shape). We also have $\mathrm{kl}(\hat{p}_n\|\hat{p}_n) = 0$, which is the minimum, and $\mathrm{kl}(\hat{p}_n\|p)$ monotonically increases in p, as p grows from $\hat{p}_n$ up to 1. So you need to find the point $p \in [\hat{p}_n, 1]$ at which the value of $\mathrm{kl}(\hat{p}_n\|p)$ grows above ε. You could do it inefficiently by linear search or, exponentially more efficiently, by binary search.

A technicality: In the computation of kl we define $0 \ln 0 = 0$. In numerical calculations $0 \ln 0$ is undefined. So you should treat $0 \ln 0$ operations separately, either by directly assigning the zero value or by replacing them with $0 \ln 1 = 0$.

C. **Pinsker's relaxation of the** kl **inequality**: the bound on p that follows from kl-inequality by Pinsker's inequality (Theorem 3.28).

D. **Refined Pinsker's relaxation of the** kl **inequality**: the bound on p that follows from kl-inequality by refined Pinsker's inequality (Theorem 3.32).

In this task you should do the following:

1. Write down explicitly the four bounds on p you are evaluating.
2. Plot the four bounds on p as a function of $\hat{p}_n$ for $\hat{p}_n \in [0,1]$, $n = 1000$, and $\delta = 0.01$. You should plot all the four bounds in one figure, so that you can directly compare them. Clip all the bounds at 1, because otherwise they are anyway meaningless and will only destroy the scale of the figure.
3. Generate a "zoom in" plot for $\hat{p}_n \in [0, 0.1]$.
4. Compare Hoeffding's lower bound on p with kl lower bound on p for the same values of $\hat{p}_n, n, \delta$ in a separate figure (no need to consider the relaxations of the kl).

   Some guidance: For computing the "lower inverse" $\mathrm{kl}^{-1^-}(\hat{p}_n, \varepsilon)$ you can either adapt the function for computing the "upper inverse" you wrote earlier (and we leave it to you to think how to do this), or implement a dedicated function for computing the "lower inverse". Direct computation of the "lower inverse" works the same way as the computation of the "upper inverse". The function $\mathrm{kl}(\hat{p}_n \| p)$ is convex in p with minimum $\mathrm{kl}(\hat{p}_n \| \hat{p}_n) = 0$ achieved at $p = \hat{p}_n$, and monotonically decreasing in p, as p increases from 0 to $\hat{p}_n$. So you need to find the point $p \in [0, \hat{p}_n]$ at which the value of $\mathrm{kl}(\hat{p}_n \| p)$ decreases below ε. You can do it by linear search or, more efficiently, by binary search. And, as mentioned earlier, you can save all the code writing if you find a smart way to reuse the function for computing the "upper inverse" to compute the "lower inverse". Whatever way you chose you should explain in your main `.pdf` submission file how you computed the upper and the lower bound.
5. Write down your conclusions from the experiment. For what values of $\hat{p}_n$ which bounds are tighter and is the difference significant?
6. [Optional, not for submission.] You are welcome to experiment with other values of n and δ.

Exercise 3.9 (*Refined Pinsker's Upper Bound*)**.** Prove Theorem 3.32. You are allowed to base the proof on Theorem 3.30.

Exercise 3.10 (*Refined Pinsker's Lower Bound*)**.** Prove Theorem 3.33. You are allowed to base the proof on Theorem 3.30.

Exercise 3.11 (*Numerical comparison of the* kl *and split-*kl *inequalities*)**.** Compare the kl and split-kl inequalities. Take a ternary random variable (a random variable taking three values) $X \in \left\{0, \frac{1}{2}, 1\right\}$. Let $p_0 = \mathbb{P}(X = 0)$, $p_{\frac{1}{2}} = \mathbb{P}\left(X = \frac{1}{2}\right)$, and $p_1 = \mathbb{P}(X = 1)$. Set $p_0 = p_1 = (1 - p_{\frac{1}{2}})/2$, i.e., the probabilities of $X = 0$ and $X = 1$ are equal, and there is just one parameter $p_{\frac{1}{2}}$, which controls the probability mass of the central value. Compare the two bounds as a function of $p_{\frac{1}{2}} \in [0,1]$. Let $p = \mathbb{E}[X]$ (in the constructed example, for any value of $p_{\frac{1}{2}}$ we have $p = \frac{1}{2}$, because $p_0 = p_1$). For each value of $p_{\frac{1}{2}}$ in a grid covering the $[0,1]$ interval draw a random sample $X_1, \ldots, X_n$ from the distribution we have constructed and let $\hat{p}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$. Generate a figure, where you plot the kl and the split-kl bounds on $p - \hat{p}_n$ as a function of $p_{\frac{1}{2}}$ for $p_{\frac{1}{2}} \in [0,1]$. For the kl bound, the bound on $p - \hat{p}_n$ is $\mathrm{kl}^{-1^+}\left(\hat{p}_n, \frac{\ln \frac{1}{\delta}}{n}\right) - \hat{p}_n$; pay attention that in contrast to Exercise 3.8 we subtract the value of $\hat{p}_n$ after inversion of kl to get a bound on the difference $p - \hat{p}_n$ rather than on p. For the split-kl bound you subtract $\hat{p}_n$ from the right hand side of the expression inside the probability in Theorem 3.34. Take $n = 100$ and $\delta = 0.05$. Briefly reflect on the outcome of the comparison.

Exercise 3.12 (*A Simple Version of Empirical Bernstein's Inequality*)**.** In this exercise you will derive a bit weaker form of Empirical Bernstein's inequality through a relatively straightforward derivation.

1. Let X and X' be two independent identically distributed random variables. Prove that $\mathbb{E}\left[(X - X')^2\right] = 2\mathbb{V}[X]$.

2. Let $X_1, \ldots, X_n$ be independent identically distributed random variables taking values in the $[0,1]$ interval, and assume that n is even. Let $\hat{\nu}_n = \frac{1}{n}\sum_{i=1}^{n/2}(X_{2i} - X_{2i-1})^2$ and let $\nu = \mathbb{V}[X_1]$. Prove that

$$\mathbb{P}\left(\nu \geq \hat{\nu}_n + \sqrt{\frac{\ln\frac{1}{\delta}}{n}}\right) \leq \delta.$$

3. Let $X_1, \ldots, X_n$, n, ν, and $\hat{\nu}_n$ as before, and let $\mu = \mathbb{E}[X_1]$. Prove that

$$\mathbb{P}\left(\mu \geq \frac{1}{n}\sum_{i=1}^{n} X_i + \sqrt{\frac{2\hat{\nu}_n \ln\frac{2}{\delta}}{n}} + \sqrt{2}\left(\frac{\ln\frac{2}{\delta}}{n}\right)^{\frac{3}{4}} + \frac{\ln\frac{2}{\delta}}{3n}\right) \leq \delta. \tag{3.20}$$

Hint: The proof uses the inequalities $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ *for* $a, b \geq 0$*, and* $\mathbb{P}(A) \leq \mathbb{P}(A|B) + \mathbb{P}(\bar{B})$ *for any pair of events* A *and* B*, where* $\bar{B}$ *is the complement of* B*.*

Discussion: since for $x \leq 1$ *we have* $x^{\frac{3}{4}} \geq x$*, the bound in* (3.20) *is slightly weaker than the bound in Theorem 3.39, but the proof is much simpler. The work of Maurer and Pontil (2009) and Gao and Zhou (2013) removes the* $\left(\frac{\ln\frac{2}{\delta}}{n}\right)^{\frac{3}{4}}$ *term.*

Exercise 3.13. Prove Theorem 3.40.

Exercise 3.14. Prove Theorem 3.41.

Chapter 4

Generalization Bounds for Classification

One of the most central questions in machine learning is: "How much can we trust the predictions of a learning algorithm?". A way of answering this question is by providing generalization bounds on the expected performance of the algorithm on new data points. In this chapter we derive a number of generalization bounds for supervised classification.

4.1 Overview: Learning by Selection

The classical process of learning can be seen as a selection process (see Figure 4.1):

1. We start with a hypothesis set $\mathcal{H}$, which is a set of plausible prediction rules (for example, linear separators).

2. We observe a sample S sampled i.i.d. according to a fixed, but unknown distribution $p(X, Y)$.

3. Based on the empirical performances $\hat{L}(h, S)$ of the hypotheses in $\mathcal{H}$, we *select* a prediction rule $\hat{h}_S^*$, which we consider to be the "best" in $\mathcal{H}$ in some sense. Typically, $\hat{h}_S^*$ is either the *empirical risk minimizer* (ERM), $\hat{h}_S^* = \arg\min_h \hat{L}(h, S)$, or a regularized empirical risk minimizer.

4. $\hat{h}_S^*$ is then applied to predict labels for new samples X.

In this chapter we are concerned with the question of what can be said about the expected loss $L(\hat{h}_S^*)$, which is the error we are expected to make on new samples. More precisely, we provide tools for bounding the probability that $\hat{L}(\hat{h}_S^*, S)$ is significantly smaller than $L(\hat{h}_S^*)$. Recall that $\hat{L}(\hat{h}_S^*, S)$ is observed and $L(\hat{h}_S^*)$ is unobserved. Having small $\hat{L}(\hat{h}_S^*, S)$ and large $L(\hat{h}_S^*)$ is undesired, because it means that based on $\hat{L}(\hat{h}_S^*, S)$ we believe that $\hat{h}_S^*$ performs well, but in reality it does not.

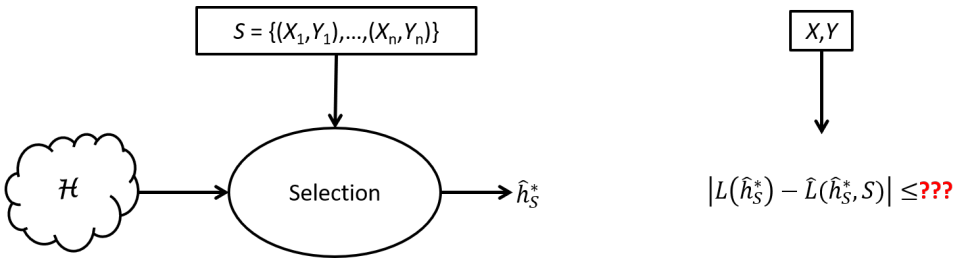


Figure 4.1: Learning by Selection.

Assumtions ***There are two key assumptions we make throughout the chapter:***

1. ***The samples in S are i.i.d..***
2. ***The new samples (X, Y) come from the same distribution as the samples in S.***

These are the assumptions behind concentration of measure inequalities developed in Chapter 3 and it is important to remember that if they are not satisfied the results derived in this chapter are not valid.

In a sense, it is intuitive why we have to make these assumptions. For example, if we train a language model using data from The Wall Street Journal and then apply it to Twitter the change in prediction accuracy can be very dramatic. Even though both are written in English and comprehensible by humans, the language used by professional journalists writing for The Wall Street Journal is very different from the language used in the short tweets.

The two assumptions are behind most supervised learning algorithms that you can meet in practice and, therefore, it is important to keep them in mind. In Chapter 7 we discuss how to depart from them, but for now we stick with them.

Given the assumptions above, for any fixed prediction rule that is independent of S, the empirical loss is an unbiased estimate of the true loss, $\mathbb{E}\left[\hat{L}(h, S)\right] = L(h)$. An intuitive way to see it is that under the assumptions that the samples in S are i.i.d. and coming from the same distribution as new samples (X, Y), from the perspective of h the new samples (X, Y) are in no way different from the samples in S: any new sample (X, Y) could have happened to be in S instead of some other sample (X_i, Y_i) (they are "exchangeable"). Formally,

$$\begin{aligned}
\mathbb{E}_{(X_1,Y_1),\ldots,(X_n,Y_n)}\left[\hat{L}(h, S)\right] &= \mathbb{E}_{(X_1,Y_1),\ldots,(X_n,Y_n)}\left[\frac{1}{n}\sum_{i=1}^{n}\ell(h(X_i), Y_i)\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{(X_1,Y_1),\ldots,(X_n,Y_n)}\left[\ell(h(X_i), Y_i)\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{(X_i,Y_i)}\left[\ell(h(X_i), Y_i)\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}L(h) \\
&= L(h).
\end{aligned}$$

However, when we make the selection of $\hat{h}^*_S$ based on S the "exchangeability" argument no longer applies and $\mathbb{E}\left[\hat{L}(\hat{h}^*_S, S)\right] \neq \mathbb{E}\left[L(\hat{h}^*_S)\right]$ (note that $\hat{h}^*_S$ is a random variable depending on S and we take expectation with respect to this randomness). This is because $\hat{h}^*_S$ is tailored to S (for example, it minimizes $\hat{L}(h, S)$) and from the perspective of selection process the samples in S are not exchangeable with new samples (X, Y). If we exchange the samples we may end up with a different $\hat{h}^*_S$. In the extreme case when the hypothesis space $\mathcal{H}$ is so rich that it can fit any possible labeling of the data (for example, the hypothesis space corresponding to 1-nearest-neighbor prediction rule) we may end up in a situation, where $\hat{L}(\hat{h}^*_S, S)$ is always zero, but $\mathbb{E}\left[L(\hat{h}^*_S)\right] \geq \frac{1}{4}$, as in the following informal example.

Informal Lower Bound Imagine that we want to learn a classifier that predicts whether a student's birthday is on an even or odd day based on student's id. Assume that the total number of students is $2n$, that the hypothesis class $\mathcal{H}$ includes all possible mappings from student id to even/odd, so that $|\mathcal{H}| = 2^{2n}$, and that we observe a sample of n uniformly sampled students (potentially with repetitions). Since all possible mappings are within $\mathcal{H}$, we have $\hat{h}^*_S \in \mathcal{H}$ for which $\hat{L}(\hat{h}^*_S, S) = 0$. However, $\hat{h}^*_S$ is guaranteed to make zero error only on the samples that were observed, which constitute at most half of the total number of students. For the remaining students $\hat{h}^*_S$ can, at the best, make a random guess which will succeed with probability $\frac{1}{2}$. Therefore, the expected loss of $\hat{h}^*_S$ is $L(\hat{h}^*_S) \geq \frac{1}{2} \cdot 0 + \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$, where the first term is an upper bound on the probability of observing an already seen student times the expected error $\hat{h}^*_S$ makes in this case and the second term is a lower bound on the probability of

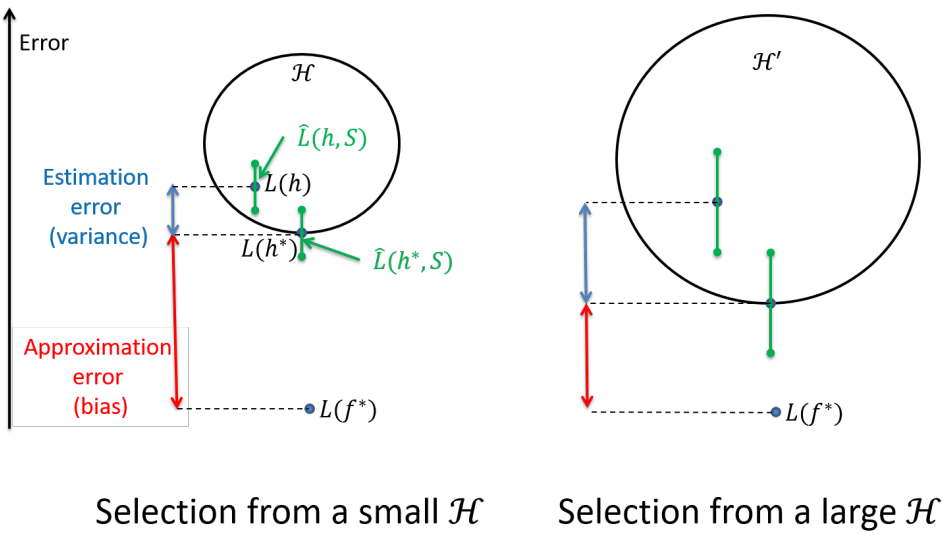


Figure 4.2: Learning by Selection.

observing a new student times the expected error $\hat{h}_S^*$ makes in this case. For a more formal treatment see the lower bounds in Chapter 4.7.

Considering it from the perspective of expectations, we have:

$$\begin{aligned}
\mathbb{E}_{(X_1,Y_1),\ldots,(X_n,Y_n)}\left[\hat{L}(\hat{h}_S^*,S)\right] &= \mathbb{E}_{(X_1,Y_1),\ldots,(X_n,Y_n)}\left[\frac{1}{n}\sum_{i=1}^{n}\ell(\hat{h}_S^*(X_i),Y_i)\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{(X_1,Y_1),\ldots,(X_n,Y_n)}\left[\ell(\hat{h}_S^*(X_i),Y_i)\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{(X_1,Y_1),\ldots,(X_n,Y_n)}\left[\ell(\hat{h}_S^*(X_1),Y_1)\right] \\
&= \mathbb{E}_{(X_1,Y_1),\ldots,(X_n,Y_n)}\left[\ell(\hat{h}_S^*(X_1),Y_1)\right] \\
&\neq \mathbb{E}_{(X,Y)}\left[\mathbb{E}_{(X_1,Y_1),\ldots,(X_n,Y_n)}\left[\ell(\hat{h}_S^*(X),Y)\right]\right] \\
&= \mathbb{E}_{(X_1,Y_1),\ldots,(X_n,Y_n)}\left[\mathbb{E}_{(X,Y)}\left[\ell(\hat{h}_S^*(X),Y)\right]\right] \\
&= \mathbb{E}_{(X_1,Y_1),\ldots,(X_n,Y_n)}\left[L(\hat{h}_S^*)\right].
\end{aligned}$$

The selection leads to the approximation-estimation trade-off (a.k.a. bias-variance trade-off), see Figure 4.2. If the hypothesis class $\mathcal{H}$ is small it is easy to identify a good hypothesis h in $\mathcal{H}$, but since $\mathcal{H}$ is small it is likely that all the hypotheses in $\mathcal{H}$ are weak. On the other hand, if $\mathcal{H}$ is large it is more likely to contain stronger hypotheses, but at the same time the probability of confusion with a poor hypothesis grows. This is because there is always a small chance that the empirical loss $\hat{L}(h,S)$ does not represent the true loss $L(h)$ faithfully. The more hypotheses we take, the higher is the chance that $\hat{L}(h,S)$ is misleading for some of them, which increases the chance of confusion.

Finding a good balance between approximation and estimation errors is one of the central questions in machine learning. The main tool for analyzing the trade-off from the theoretical perspective are concentration of measure inequalities. Since concentration of measure inequalities do not apply when the prediction rule $\hat{h}_S^*$ depends on S, the main approach to analyzing the prediction power of $\hat{h}_S^*$ is to consider cases with no dependency and then take a union bound over selection from these cases. In this chapter we study three different ways of implementing this idea, see Figure 4.3 for an overview. We distinguish between *hard selection*, where the learning procedure returns a single hypothesis h and *soft selection*, where the learning procedure returns a distribution over $\mathcal{H}$.

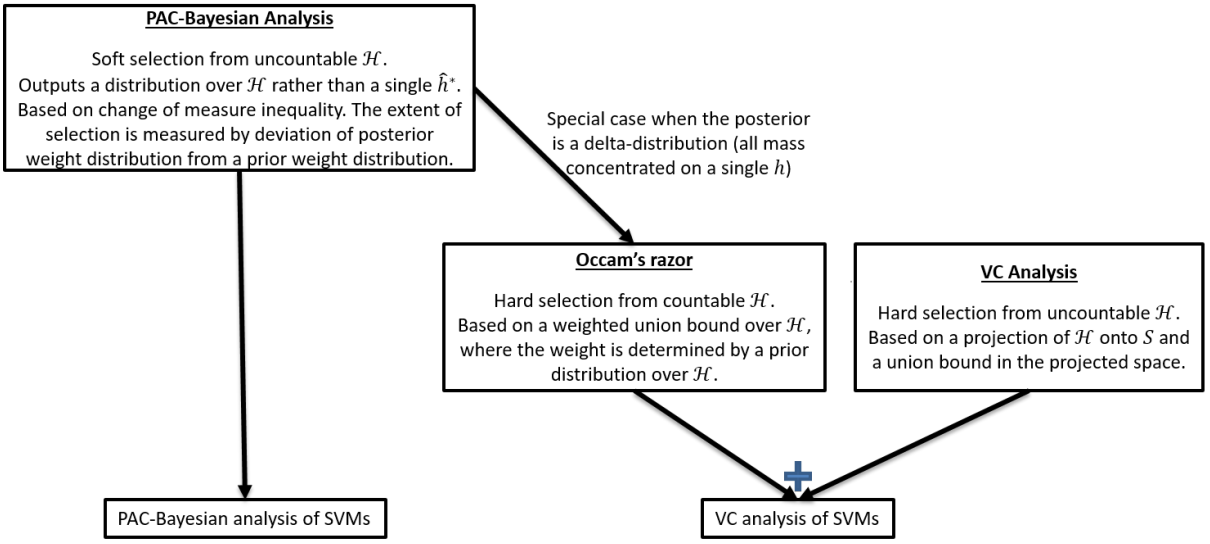


Figure 4.3: Overview of the major approaches to derivation of generalization bounds considered in this chapter.

1. *Occam's razor* applies to *hard selection* from a *countable* hypothesis space $\mathcal{H}$ and it is based on a weighted union bound over $\mathcal{H}$. We know that for every fixed h the expected loss is close to the empirical loss, meaning that $\left|L(h) - \hat{L}(h, S)\right|$ is small. When $\mathcal{H}$ is countable we can take a weighted union bound and obtain that $\left|L(h) - \hat{L}(h, S)\right|$ is "small" for all $h \in \mathcal{H}$ (where the magnitude of "small" is inversely proportional to the weight of h in the union bound) and thus it is "small" for $\hat{h}^*_S$.

2. *Vapnik-Chervonenkis (VC) analysis* applies to *hard selection* from an *uncountable* hypothesis space $\mathcal{H}$ and it is based on projection of $\mathcal{H}$ onto S and a union bound over what we obtain after the projection. The idea is that even when $\mathcal{H}$ is uncountably infinite, there is only a finite number of "behaviors" (ways to label S) we can observe on a finite sample S. In other words, when we look at $\mathcal{H}$ through the lens of S, we can only distinguish a finite number of subsets of $\mathcal{H}$, whereas everything that falls within the subsets is equivalent in terms of $\hat{L}(h, S)$. Therefore, S only serves for a (finite) selection of a subset of $\mathcal{H}$ out of a finite number of subsets, whereas the (infinite) selection from within the subset is independent of S. Selection that is independent of S introduces no bias. As before, the VC analysis exploits the fact that for any fixed h the distance $\left|L(h) - \hat{L}(h, S)\right|$ is small and then takes a union bound over the potential dependencies, which are the dependencies between the subsets (the projections) and S.

3. *PAC-Bayesian analysis* applies to *soft selection* from an *uncountable* hypothesis space $\mathcal{H}$ and it is based on *change of measure inequality*, which can be seen as a refinement of the union bound. Unlike the preceding two approaches, which return a single classifier $\hat{h}^*_S$, PAC-Bayesian analysis returns a *randomized classifier* defined by a distribution ρ over $\mathcal{H}$. The actual classification then happens by drawing a new classifier h from $\mathcal{H}$ according to ρ at each prediction round and applying it to make a prediction. When $\mathcal{H}$ is countable, ρ can (but does not have to) be a delta-distribution allocating all the mass to a single hypothesis $\hat{h}^*_S$, and in this case the generalization guarantees recover those obtained by the Occam's razor approach. The amount of selection is measured by deviation of ρ from a prior distribution π, where π is selected independently of S. It is natural to put more of ρ-mass on hypotheses that perform well on S, but the more we skew ρ toward well-performing hypotheses the more it deviates from π. This provides a more refined way of measuring the amount of selection compared to the other two approaches. Furthermore, randomization allows to avoid selection when it is not necessary. The avoidance of selection reduces the variance without impairing the bias. For example, when two hypotheses have similar empirical performance we do not have to commit to one of them, but can instead distribute ρ equally among them. The analysis

then provides a certain "bonus" for avoiding commitment.

4.2 Generalization Bound for a Single Hypothesis

We start with the simplest case, where $\mathcal{H}$ consists of a single prediction rule h. We are interested in the quality of h, measured by $L(h)$, but all we can measure is $\hat{L}(h,S)$. What can we say about $L(h)$ based on $\hat{L}(h,S)$? Note that the samples $(X_i,Y_i) \in S$ come from the same distribution as any future samples (X,Y) we will observe. Therefore, $\ell(h(X_i),Y_i)$ has the same distribution as $\ell(h(X),Y)$ for any future sample (X,Y). Let $Z_i = \ell(h(X_i),Y_i)$ be the loss of h on (X_i,Y_i). Then $\hat{L}(h,S) = \frac{1}{n}\sum_{i=1}^{n} Z_i$ is an average of n i.i.d. random variables with $\mathbb{E}[Z_i] = \mathbb{E}[\ell(h(X),Y)] = L(h)$. The distance between $\hat{L}(h,S)$ and $L(h)$ can thus be bounded by application of Hoeffding's inequality.

Theorem 4.1. *Assume that ℓ is bounded in the $[0,1]$ interval (i.e., $\ell(Y',Y) \in [0,1]$ for all Y', Y), then for a single h and any $\delta \in (0,1)$ we have:*

$$\mathbb{P}\left(L(h) \geq \hat{L}(h,S) + \sqrt{\frac{\ln \frac{1}{\delta}}{2n}} \right) \leq \delta \tag{4.1}$$

and

$$\mathbb{P}\left(\left| L(h) - \hat{L}(h,S) \right| \geq \sqrt{\frac{\ln \frac{2}{\delta}}{2n}} \right) \leq \delta. \tag{4.2}$$

Proof. For (4.1) take $\varepsilon = \sqrt{\frac{\ln \frac{1}{\delta}}{2n}}$ in (3.5) and rearrange the terms. Equation (4.2) follows in a similar way from the two-sided Hoeffding's inequality. Note that in (4.1) we have $\frac{1}{\delta}$ and in (4.2) we have $\frac{2}{\delta}$. □

There is an alternative way to read equation (4.1): with probability at least $1-\delta$ we have

$$L(h) \leq \hat{L}(h,S) + \sqrt{\frac{\ln \frac{1}{\delta}}{2n}}.$$

We remind the reader that the above inequality should actually be interpreted as

$$\hat{L}(h,S) \geq L(h) - \sqrt{\frac{\ln \frac{1}{\delta}}{2n}}$$

and it means that with probability at least $1-\delta$ the empirical loss $\hat{L}(h,S)$ does not underestimate the expected loss $L(h)$ by more than $\sqrt{\ln(1/\delta)/2n}$. However, it is customary to write the inequality in the first form (as an upper bound on $L(h)$ and we follow the tradition (see the discussion at the end of Section 3.3.1).

Theorem 4.1 is analogous to the problem of estimating a bias of a coin based on coin flip outcomes. There is always a small probability that the flip outcomes will not be representative of the coin bias. For example, it may happen that we flip a fair coin 1000 times (without knowing that it is a fair coin!) and observe "all heads" or some other misleading outcome. And if this happens we are doomed - there is nothing we can do when the sample does not represent the reality faithfully. Fortunately for us, this happens with a small probability that decreases exponentially with the sample size n.

Whether we use the one-sided bound (4.1) or the two-sided bound (4.2) depends on the situation. In most cases we are interested in the upper bound on the expected performance of the prediction rule given by (4.1).

4.3 Generalization Bound for Finite Hypothesis Classes

A hypothesis set $\mathcal{H}$ containing a single hypothesis is a very boring set. In fact, we cannot learn in this case, because we end up with the same single hypothesis no matter what the sample S is. Learning becomes interesting when training sample S helps to improve future predictions or, equivalently, decrease

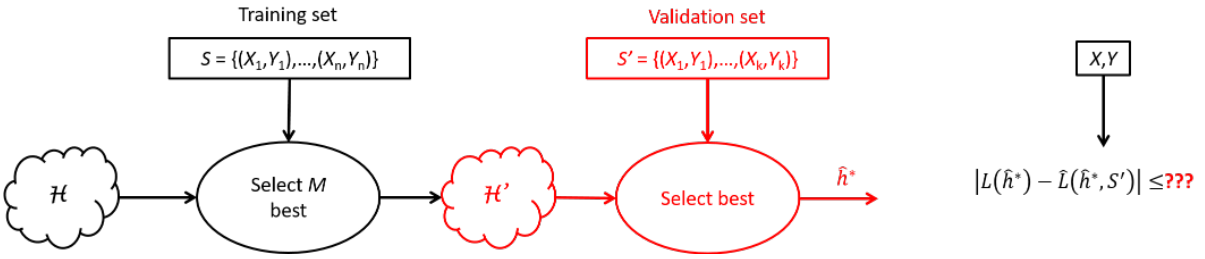


Figure 4.4: Validation (the red part in the figure) is identical to learning with a reduced hypothesis set $\mathcal{H}'$ (most often $\mathcal{H}'$ is finite).

the expected loss $L(h)$. In this section we consider the simplest non-trivial case, where $\mathcal{H}$ consists of a finite number of hypotheses M. There are at least two cases, where we meet a finite $\mathcal{H}$ in real life. The first is when the input space $\mathcal{X}$ is finite. This case is relatively rare. The second and much more frequent case is when $\mathcal{H}$ itself is an outcome of a learning process. For example, this is what happens in a validation procedure, see Figure 4.4. In validation we are using a validation set in order to select the best hypothesis out of a finite number of candidates corresponding to different parameter values and/or different algorithms.

And now comes the delicate point. Let $\hat{h}_S^*$ be a hypothesis with minimal empirical risk, $\hat{h}_S^* = \arg\min_h \hat{L}(h, S)$ (it is natural to pick the empirical risk minimizer $\hat{h}_S^*$ to make predictions on new samples, but the following discussion equally applies to any other selection rule that takes sample S into account; note that there may be multiple hypotheses that achieve the minimal empirical error and in this case we can pick one arbitrarily). While for each h individually $\mathbb{E}\left[\hat{L}(h, S)\right] = L(h)$, this is not true for $\mathbb{E}\left[\hat{L}(\hat{h}_S^*, S)\right]$. In other words, $\mathbb{E}\left[\hat{L}(\hat{h}_S^*, S)\right] \neq \mathbb{E}\left[L(\hat{h}_S^*)\right]$ (we have to put expectation on the right hand side, because $\hat{h}_S^*$ depends on the sample). The reason is that when we pick $\hat{h}_S^*$ that minimizes the empirical error on S, from the perspective of $\hat{h}_S^*$ the samples in S no longer look identical to future samples (X, Y). This is because $\hat{h}_S^*$ is selected in a very special way - it is selected to minimize the empirical error on S and, thus, it is tailored to S and most likely does better on S than on new random samples (X, Y). One way to handle this issue is to apply a union bound.

Theorem 4.2. *Assume that ℓ is bounded in the $[0, 1]$ interval and that $|\mathcal{H}| = M$. Then for any $\delta \in (0, 1)$ we have:*

$$\mathbb{P}\left(\exists h \in \mathcal{H} : L(h) \geq \hat{L}(h, S) + \sqrt{\frac{\ln \frac{M}{\delta}}{2n}}\right) \leq \delta. \tag{4.3}$$

Proof.

$$\mathbb{P}\left(\exists h \in \mathcal{H} : L(h) \geq \hat{L}(h, S) + \sqrt{\frac{\ln \frac{M}{\delta}}{2n}}\right) \leq \sum_{h \in \mathcal{H}} \mathbb{P}\left(L(h) \geq \hat{L}(h, S) + \sqrt{\frac{\ln \frac{M}{\delta}}{2n}}\right) \leq \sum_{h \in \mathcal{H}} \frac{\delta}{M} = \delta,$$

where the first inequality is by the union bound and the second is by Hoeffding's inequality. □

Another way of reading Theorem 4.2 is: with probability at least $1 - \delta$ for all $h \in \mathcal{H}$

$$L(h) \leq \hat{L}(h, S) + \sqrt{\frac{\ln \frac{M}{\delta}}{2n}}. \tag{4.4}$$

It means that no matter which h from $\mathcal{H}$ is returned by the algorithm, with high probability we have the guarantee (4.4). In particular, it holds for $\hat{h}_S^*$. Again, remember that the random quantity is actually $\hat{L}(h, S)$ and the right way to read the bound is $\hat{L}(h, S) \geq L(h) - \sqrt{\ln(M/\delta)/2n}$, see the discussion in the previous section.

The price for considering M hypotheses instead of a single one is $\ln M$. Note that it grows only logarithmically with M! Also note that there is no contradiction between the upper bound and the lower

bound we have discussed in Section 4.1. In the construction of the lower bound we took $M = |\mathcal{H}| = 2^{2n}$. If we substitute this value of M into (4.4) we obtain $\sqrt{\ln(M/\delta)/2n} \geq \sqrt{\ln(2)} \geq 0.8$, which has no contradiction with $L(h) \geq 0.25$.

Similar to theorem 4.1 it is possible to derive a two-sided bound on the error. It is also possible to derive a lower bound by using the other side of Hoeffding's inequality (3.4): with probability at least $1 - \delta$, for all $h \in \mathcal{H}$ we have $L(h) \geq \hat{L}(h, S) - \sqrt{\ln(M/\delta)/2n}$. Typically we want the upper bound on $L(h)$, but if we want to compare two prediction rules, h and h', we need an upper bound for one and a lower bound for the other. The "lazy" approach is to take the two-sided bound for everything, but sometimes it is possible to save the factor of $\ln(2)$ by carefully considering which hypotheses require the lower bound and which require the upper bound and applying the union bound correspondingly (we are not getting into the details).

4.4 Occam's Razor Bound

Now we take a closer look at Hoeffding's inequality. It says that

$$\mathbb{P}\left(L(h) \geq \hat{L}(h, S) + \sqrt{\frac{\ln\left(\frac{1}{\delta}\right)}{2n}} \right) \leq \delta,$$

where δ is the probability that things go wrong and $\hat{L}(h, S)$ happens to be far away from $L(h)$ because S is not representative for the performance of h. There is a dependence between the probability that things go wrong and the requirement on the closeness between $L(h)$ and $\hat{L}(h, S)$. If we want them to be very close (meaning that $\ln\left(\frac{1}{\delta}\right)$ is small) then δ has to be large, but if we can allow larger distance then δ can be smaller.

So, δ can be seen as our "confidence budget" (or, more precisely, "uncertainty budget") - the probability that we allow things to go wrong. The idea behind Occam's Razor bound is to distribute this budget unevenly among the hypotheses in $\mathcal{H}$. We use $\pi(h) \geq 0$, such that $\sum_{h \in \mathcal{H}} \pi(h) \leq 1$ as our distribution of the confidence budget δ, where each hypothesis h is assigned $\pi(h)$ fraction of the budget. This means that for every hypothesis $h \in \mathcal{H}$ the sample S is allowed to be "non representative" with probability at most $\pi(h)\delta$, so that the probability that there exists any $h \in \mathcal{H}$ for which S is not representative is at most δ (by the union bound). The price that we pay is that the precision (the closeness of $\hat{L}(h, S)$ to $L(h)$) now differs from one hypothesis to another and depends on the confidence budget $\pi(h)\delta$ that was assigned to it. More precisely, $\hat{L}(h, S)$ is allowed to underestimate $L(h)$ by up to $\sqrt{\ln(1/(\pi(h)\delta))/2n}$. The precision increases when $\pi(h)$ increases, but since $\sum_{h \in \mathcal{H}} \pi(h) \leq 1$ we cannot afford high precision for every h and have to compromise. More on this in the next theorem and its applications that follow.

Theorem 4.3 (Occam's razor)**.** *Let ℓ be bounded in $[0, 1]$, let $\mathcal{H}$ be a countable hypothesis set and let $\pi(h)$ be independent of the sample and satisfying $\pi(h) \geq 0$ for all h and $\sum_{h \in \mathcal{H}} \pi(h) \leq 1$. Then:*

$$\mathbb{P}\left(\exists h \in \mathcal{H} : L(h) \geq \hat{L}(h, S) + \sqrt{\frac{\ln\left(\frac{1}{\pi(h)\delta}\right)}{2n}} \right) \leq \delta.$$

Proof.

$$\begin{aligned}
\mathbb{P}\left(\exists h \in \mathcal{H} : L(h) \geq \hat{L}(h, S) + \sqrt{\frac{\ln\left(\frac{1}{\pi(h)\delta}\right)}{2n}} \right) &\leq \sum_{h \in \mathcal{H}} \mathbb{P}\left(L(h) \geq \hat{L}(h, S) + \sqrt{\frac{\ln\left(\frac{1}{\pi(h)\delta}\right)}{2n}} \right) \\
&\leq \sum_{h \in \mathcal{H}} \pi(h)\delta \\
&\leq \delta,
\end{aligned}$$

where the first inequality is by the union bound, the second inequality is by Hoeffding's inequality, and the last inequality is by the assumption on $\pi(h)$. Note that $\pi(h)$ has to be selected before we observe the sample (or, in other words, independently of the sample), otherwise the second inequality does not hold. More explicitly, in Hoeffding's inequality $\mathbb{P}\Big(\mathbb{E}\left[Z_1\right] - \frac{1}{n}\sum_{i=1}^{n} Z_i \geq \sqrt{\ln(1/\delta')/2n}\Big) \leq \delta'$ the parameter δ' has to be independent of $Z_1, \ldots, Z_n$. For $\pi(h)$ independent of S we take $\delta' = \pi(h)\delta$ and apply the inequality. But if $\pi(h)$ would be dependent on S we would not be able to apply it. □

Another way of reading Theorem 4.3 is that with probability at least $1-\delta$, for all $h \in \mathcal{H}$:

$$L(h) \leq \hat{L}(h, S) + \sqrt{\frac{\ln\left(\frac{1}{\pi(h)\delta}\right)}{2n}}.$$

Again, refer back to the discussion in Section 4.2 regarding the correct interpretation of the inequality. Note that the bound on $L(h)$ depends both on $\hat{L}(h, S)$ and on $\pi(h)$. Therefore, according to the bound, the best generalization is achieved by h that optimizes the trade-off between empirical performance $\hat{L}(h, S)$ and $\pi(h)$, where $\pi(h)$ can be interpreted as a complexity measure or a prior belief. Also, note that $\pi(h)$ can be designed arbitrarily, but it should be independent of the sample S. If $\pi(h)$ happens to put more mass on h-s with low $\hat{L}(h, S)$ the bound will be tighter, otherwise the bound will be looser, but it will still be a valid bound. But we cannot readjust $\pi(h)$ after observing S! Some considerations behind the choice of $\pi(h)$ are provided in Section 4.4.1.

Also note that while we can select $\pi(h)$ such that $\sum_{h\in\mathcal{H}} \pi(h) = 1$ and interpret π as a probability distribution over $\mathcal{H}$, it is not a requirement (we may have $\sum_{h\in\mathcal{H}} \pi(h) < 1$) and π is used as an auxiliary construction for derivation of the bound rather than the prior distribution in the Bayesian sense (for readers who are familiar with Bayesian learning). However, we can use π to incorporate prior knowledge into the learning procedure.

4.4.1 Applications of Occam's Razor bound

We consider two applications of Occam's Razor bound.

Generalization bound for finite hypotheses spaces

An immediate corollary of Occam's razor bound is the generalization bound for finite hypotheses classes that we have already seen in Section 4.3.

Corollary 4.4. *Let $\mathcal{H}$ be a finite hypotheses class of size M, then*

$$\mathbb{P}\left(\exists h \in \mathcal{H} : L(h) \geq \hat{L}(h, S) + \sqrt{\frac{\ln\left(M/\delta\right)}{2n}}\right) \leq \delta.$$

Proof. We set $\pi(h) = \frac{1}{M}$ (which means that we distribute the confidence budget δ uniformly among the hypotheses in $\mathcal{H}$) and apply Theorem 4.3. □

Generalization bound for binary decision trees

Theorem 4.5. *Let $\mathcal{H}_d$ be the set of binary decision trees of depth d and let $\mathcal{H} = \bigcup_{d=0}^{\infty} \mathcal{H}_d$ be the set of binary decision trees of unlimited depth. Let $d(h)$ be the depth of tree (hypothesis) h. Then*

$$\mathbb{P}\left(\exists h \in \mathcal{H} : L(h) \geq \hat{L}(h, S) + \sqrt{\frac{\ln\left(2^{2^{d(h)}} 2^{d(h)+1}/\delta\right)}{2n}}\right) \leq \delta.$$

Proof. We first note that $|\mathcal{H}_d| = 2^{2^d}$. We define $\pi(h) = \frac{1}{2^{d(h)+1}} \frac{1}{2^{2^{d(h)}}}$. The first part of $\pi(h)$ distributes the confidence budget δ among $\mathcal{H}_d$-s (we can see it as $\pi(\mathcal{H}_d) = \frac{1}{2^{d+1}}$, the share of confidence budget that goes to $\mathcal{H}_d$) and the second part of $\pi(h)$ distributes the confidence budget uniformly within $\mathcal{H}_d$. Since $\sum_{d=0}^{\infty} \frac{1}{2^{d+1}} = 1$, the assumption $\sum_{h\in\mathcal{H}} \pi(h) \leq 1$ is satisfied. The result follows by application of Theorem 4.3. □

Note that the bound depends on $\ln\left(\frac{1}{\pi(h)\delta}\right)$ and the dominating term in $\frac{1}{\pi(h)}$ is $2^{2^{d(h)}}$. It comes from the uniform distribution of confidence within $\mathcal{H}_d$, which makes sense unless we have some prior information about the problem. In absence of such information there is no reason to give preference to any of the trees within $\mathcal{H}_d$, because $\mathcal{H}_d$ is symmetric.

Thus, the prior in the proof of Theorem 4.5 exploits structural symmetries within the hypothesis subclasses $\mathcal{H}_d$ and assigns equal weight to hypotheses that are symmetric under permutation of names of the input variables. While we want $\pi(h)$ to be as large as possible for every h, the number of such permutation symmetric hypotheses is the major barrier dictating how large $\pi(h)$ can be (because π has to satisfy $\sum_{h\in\mathcal{H}}\pi(h)\leq 1$). Deeper trees have more symmetric permutations and, therefore, get smaller $\pi(h)$ compared to shallower trees. If there is prior information that breaks the permutation symmetry, it can be used to assign higher prior to the corresponding trees, and if it correctly reflects the true data distribution it will also lead to tighter bounds. If the prior information does not match the true data distribution such adjustments may have the opposite effect.

Concerning the top-level distribution of confidence budget over $\mathcal{H}_d$-s, the $\pi(\mathcal{H}_d)=\frac{1}{2^{d+1}}$ part, we could have selected a different series to work with. For example, we could have used $\pi(\mathcal{H}_d)=\frac{1}{(d+1)(d+2)}$ (for which we have $\sum_{d=0}^{\infty}\frac{1}{(d+1)(d+2)}=\sum_{d=1}^{\infty}\frac{1}{d(d+1)}=\sum_{d=1}^{\infty}\left(\frac{1}{d}-\frac{1}{d+1}\right)=1$) or any other series that sums up to 1. In the case of binary decision trees the dominating complexity term is $\ln\left(2^{2^{d(h)}}\right)$, and the choice of the top-level prior has a small impact. However, more generally in absence of prior knowledge "flatter" priors, like the one based on $\frac{1}{(d+1)(d+2)}$ series, make more sense, and for some problems it makes a big difference, see Exercise 4.6 for an example.

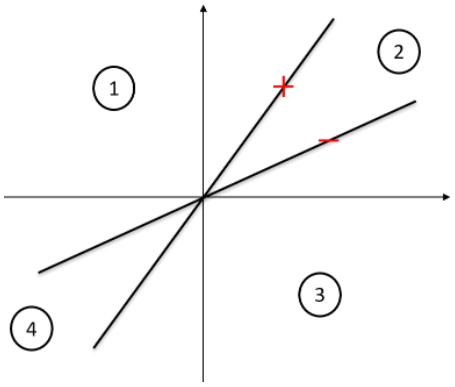
Note that for a countably infinite set $\{1,2,\dots\}$ assigning $\pi(i)=\frac{1}{i(i+1)}$ or $\pi(i)=\frac{6}{\pi^2 i^2}$ (we have $\sum_{i=1}^{\infty}\frac{1}{i^2}=\frac{\pi^2}{6}$) is "almost as flat as it can get" in the following sense. If we would have had just i items, $\{1,\dots,i\}$, we could have assigned $\pi(j)=\frac{1}{j}$ for all $1\leq j\leq i$, which would have been a "flat" prior. However, for a countably infinite set of items the series $\sum_{i=1}^{\infty}\frac{1}{i}$ diverges, and so $\pi(i)$ has to be smaller than $\frac{1}{i}$. Since $\pi(i)$ enters the bound as $\ln\frac{1}{\pi(i)}$ and $\ln(i(i+1))\approx 2\ln i$, the difference between using $\pi(i)=\frac{1}{i}$ (which we cannot do!) and $\pi(i)=\frac{1}{i(i+1)}$ is relatively small, and in this sense $\pi(i)=\frac{1}{i(i+1)}$ is "almost as flat as it can be".

4.5 Vapnik-Chervonenkis (VC) Analysis

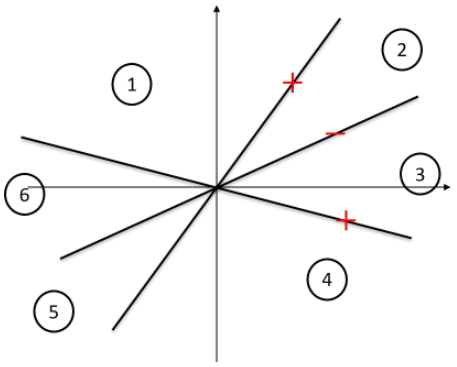
Now we present Vapnik-Chervonenkis (VC) analysis of generalization when a hypothesis is selected from an uncountably infinite hypothesis class $\mathcal{H}$. The reason that we are able to derive a generalization bound even though we are selecting from an uncountably large set is that only a finite part of this selection is based on the sample S, whereas the remaining uncountable selection is not based on the sample and, therefore, introduces no bias. Since the sample is finite, the number of distinct labeling patterns, also called dichotomies, $(h(X_1),\dots,h(X_n))$ is also finite. When two hypotheses, h and h', produce the same labeling pattern, $(h(X_1),\dots,h(X_n))=(h'(X_1),\dots,h'(X_n))$, the sample does not discriminate between them and the selection between h and h' is based on some other considerations rather than the sample. Therefore, the sample defines a finite number of (typically uncountably infinite) subsets of the hypothesis space $\mathcal{H}$, where hypotheses within the same subset produce the same labeling pattern $(h(X_1),\dots,h(X_n))$. The sample then allows selection of the "best" subset, for example, the subset that minimizes the empirical error. All prediction rules within the same subset have the same empirical error $\hat{L}(h,S)$ and selection among them is independent of S. See Figure 4.5 for an illustration.

The *effective selection* based on the sample S depends on the number of subsets of $\mathcal{H}$ with distinct labeling patterns on S. When the number of such subsets is exponential in the size of the sample n, the selection is too large and leads to overfitting, as we have already seen for selection from large finite hypothesis spaces in the earlier sections. I.e., we cannot guarantee closeness of $\hat{L}(\hat{h}^*_S,S)$ to $L(\hat{h}^*_S)$. However, if the number of subsets is subexponential in n, we can provide generalization guarantees for $L(\hat{h}^*_S)$. In Figure 4.5 we illustrate (informally) that at a certain point the number of subsets of the class of homogeneous linear separators in $\mathbb{R}^2$ stops growing exponentially with n.[1] For $n=2$ the sample defines $4=2^n$ subsets, but for $n=3$ the sample defines $6<2^n$ subsets. It can be formally shown that

[1] Homogeneous linear separators are linear separators passing through the origin.



(a) Subsets of linear homogeneous separators defined by two sample points.



(b) Subsets of linear homogeneous separators defined by three sample points.

Figure 4.5: **Subsets of homogeneous linear separators in $\mathbb{R}^2$ formed by 4.5a two and 4.5b three sample points.** A homogeneous linear separator in $\mathbb{R}^2$ is defined by a vector $w \in \mathbb{R}^2$. The sample points define a number of regions in $\mathbb{R}^2$ that are shown by the numbers in circles. We say that a linear separator falls within a certain region when the vector w defining it falls within that region. All homogeneous linear separators falling within the same region have the same empirical loss $\hat{L}(h, S)$ and, therefore, any selection among them is not based on the sample S and introduces no bias. The sample only discriminates between the subsets.

no 3 sample points can define more than 6 subsets of the space of homogeneous linear separators in $\mathbb{R}^2$ (some may define less, but that is even better for us) and that for $n > 2$ the number of subsets grows polynomially rather than exponentially with n.

In what follows we first bound the distance between $\hat{L}(h, S)$ and $L(h)$ for all $h \in \mathcal{H}$ in terms of the number of subsets using symmetrization (Section 4.5.1) and then bound the number of subsets (Section 4.5.2).

4.5.1 The VC Analysis: Symmetrization

We start with a couple of definitions.

Definition 4.6 (Dichotomies)**.** *Let $x_1, \ldots, x_n \in \mathcal{X}$. The set of* dichotomies *(the labeling patterns) generated by $\mathcal{H}$ on $x_1, \ldots, x_n$ is defined by*

$$\mathcal{H}(x_1, \ldots, x_n) = \{h(x_1), \ldots, h(x_n) : h \in \mathcal{H}\}.$$

Definition 4.7 (The Growth Function)**.** *The* growth function *of $\mathcal{H}$ is the maximal number of dichotomies it can generate on n points:*

$$m_{\mathcal{H}}(n) = \max_{x_1, \ldots, x_n} |\mathcal{H}(x_1, \ldots, x_n)|.$$

Pay attention that $m_{\mathcal{H}}(n)$ is defined by the "worst-case" configuration of points $x_1, \ldots, x_n$, for which $|\mathcal{H}(x_1, \ldots, x_n)|$ is maximized. Thus, for lower bounding $m_{\mathcal{H}}(n)$ (i.e., for showing that $m_{\mathcal{H}}(n) \geq v$ for some value v) we have to find a configuration of points $x_1, \ldots, x_n$ for which $|\mathcal{H}(x_1, \ldots, x_n)| \geq v$ or, at least, prove that such configuration exists. For upper bounding $m_{\mathcal{H}}(n)$ (i.e., for showing that $m_{\mathcal{H}}(n) \leq v$) we have to show that for any possible configuration of points $x_1, \ldots, x_n$ we have $|\mathcal{H}(x_1, \ldots, x_n)| \leq v$. In other words, coming up with an example of a particular configuration $x_1, \ldots, x_n$ for which $|\mathcal{H}(x_1, \ldots, x_n)| \leq v$ is insufficient for proving that $m_{\mathcal{H}}(n) \leq v$, because there may potentially be an alternative configuration of points achieving a larger number of labeling configurations. To be concrete, the illustration in Figure 4.5b shows that for the hypothesis space $\mathcal{H}$ of homogeneous linear separators in $\mathbb{R}^2$ we have $m_{\mathcal{H}}(3) \geq 6$, but it does not show that $m_{\mathcal{H}}(3) \leq 6$. If we want to prove that $m_{\mathcal{H}}(3) \leq 6$ we have to show that no configuration of 3 sample points can differentiate between more than 6 distinct subsets of the hypothesis space. More generally, if we want to show that $m_{\mathcal{H}}(n) = v$ we have to show that $m_{\mathcal{H}}(n) \geq v$ and $m_{\mathcal{H}}(n) \leq v$. I.e., the only way to show equality is by proving a lower and an upper bound.

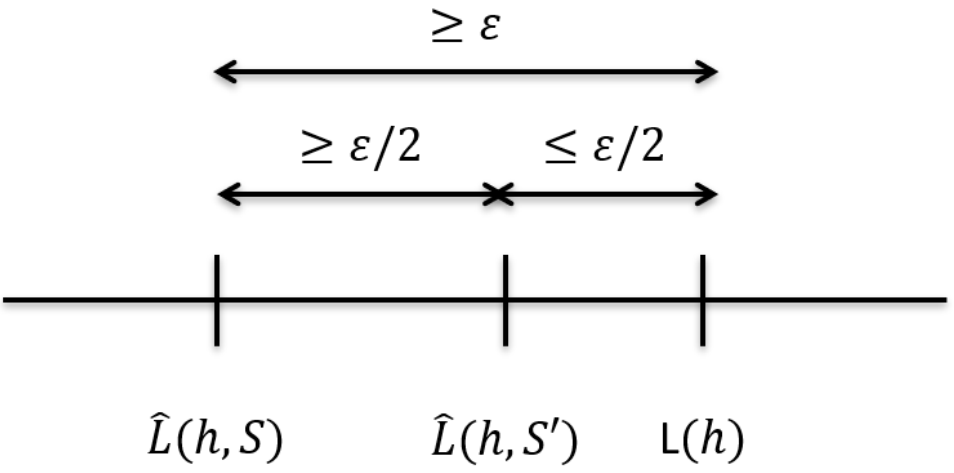


Figure 4.6: **Illustration for Step 2 of the proof of Theorem 4.8.**

The following theorem uses the growth function to bound the distance between empirical and expected loss for all $h \in \mathcal{H}$.

Theorem 4.8. *Assume that ℓ is bounded in the $[0,1]$ interval. Then for any $\delta \in (0,1)$*

$$\mathbb{P}\left(\exists h \in \mathcal{H} : L(h) \geq \hat{L}(h,S) + \sqrt{\frac{8 \ln \frac{2m_{\mathcal{H}}(2n)}{\delta}}{n}}\right) \leq \delta.$$

The result is useful when $m_{\mathcal{H}}(2n) \ll e^n$. In Section 4.5.2 we discuss when we can and cannot expect to have it, but for now we concentrate on the proof of the theorem.

The proof of the theorem is based on three ingredients. First we introduce a "ghost sample" S', which is an imaginary sample of the same size as S (i.e., of size n). We do not need to have this sample at hand, but we ask what would have happened if we had such sample. Then we apply symmetrization: we show that the probability that for any h the empirical loss $\hat{L}(h,S)$ is far from $L(h)$ by more than ε is bounded by twice the probability that $\hat{L}(h,S)$ is far from $\hat{L}(h,S')$ by more than $\varepsilon/2$. This allows us to consider the behavior of $\mathcal{H}$ on the two samples, S and S', instead of studying it over all $\mathcal{X}$ (because the definition of $L(h)$ involves all $\mathcal{X}$, whereas the definition of $\hat{L}(h,S')$ involves only S'). In the third step we project $\mathcal{H}$ onto the two samples, S and S'. Even though $\mathcal{H}$ is uncountably infinite, when we look at it through the prism of $S \cup S'$ we can only observe a finite number of distinct behaviors. More precisely, the number of different ways $\mathcal{H}$ can label $S \cup S'$ is at most $m_{\mathcal{H}}(2n)$. We show that the probability that for any of the possible ways to label $S \cup S'$ the empirical losses $\hat{L}(h,S)$ and $\hat{L}(h,S')$ diverge by more than $\varepsilon/2$ decreases exponentially with n.

Now we do this formally.

Step 1 We introduce a ghost sample $S' = \{(X'_1, Y'_1), \ldots, (X'_n, Y'_n)\}$ of size n.

Step 2 [Symmetrization] We prove the following result.

Lemma 4.9. *Assuming that $e^{-n\varepsilon^2/2} \leq \frac{1}{2}$ we have*

$$\mathbb{P}\Big(\exists h \in \mathcal{H} : L(h) - \hat{L}(h,S) \geq \varepsilon\Big) \leq 2\mathbb{P}\Big(\exists h \in \mathcal{H} : \hat{L}(h,S') - \hat{L}(h,S) \geq \frac{\varepsilon}{2}\Big). \tag{4.5}$$

The illustration in Figure 4.6 should be helpful for understanding the proof. The distance $L(h) - \hat{L}(h,S)$ can be expressed as $L(h) - \hat{L}(h,S) = (L(h) - \hat{L}(h,S')) + (\hat{L}(h,S') - \hat{L}(h,S))$. We remind that in general empirical losses are likely to be close to their expected values. More explicitly, under the mild assumption that $e^{-n\varepsilon^2/2} \leq 1/2$ we have that $L(h) - \hat{L}(h,S') \leq \varepsilon/2$ with probability greater than $1/2$. If $L(h) - \hat{L}(h,S) \geq \varepsilon$ and $L(h) - \hat{L}(h,S') \leq \varepsilon/2$ we must have $\hat{L}(h,S') - \hat{L}(h,S) \geq \varepsilon/2$ (see the illustration). The proof is based on a careful exploitation of this observation.

Proof of Lemma 4.9. We start from the right hand side of (4.5).

$$\begin{aligned}
&\mathbb{P}\Big(\exists h \in \mathcal{H} : \hat{L}(h,S') - \hat{L}(h,S) \geq \frac{\varepsilon}{2}\Big) \\
&\quad \geq \mathbb{P}\Big(\Big(\exists h \in \mathcal{H} : \hat{L}(h,S') - \hat{L}(h,S) \geq \frac{\varepsilon}{2}\Big) \text{ AND } \Big(\exists h \in \mathcal{H} : L(h) - \hat{L}(h,S) \geq \varepsilon\Big)\Big) \\
&\quad = \mathbb{P}\Big(\exists h \in \mathcal{H} : L(h) - \hat{L}(h,S) \geq \varepsilon\Big)\mathbb{P}\Big(\exists h \in \mathcal{H} : \hat{L}(h,S') - \hat{L}(h,S) \geq \frac{\varepsilon}{2}\Big|\exists h \in \mathcal{H} : L(h) - \hat{L}(h,S) \geq \varepsilon\Big).
\end{aligned} \tag{4.6}$$

The inequality follows by the fact that for any two events A and B we have $\mathbb{P}(A) \geq \mathbb{P}(A \text{ AND } B)$ and the equality by $\mathbb{P}(A \text{ AND } B) = \mathbb{P}(B)\mathbb{P}(A|B)$. The first term in (4.6) is the term we want and we need to lower bound the second term. We let h^* be any h for which, by conditioning, we have $L(h^*) - \hat{L}(h^*,S) \geq \varepsilon$. With high probability we have that $\hat{L}(h^*,S')$ is close to $L(h^*)$ up to $\varepsilon/2$. And since we are given that $\hat{L}(h,S)$ is far from $L(h^*)$ by more than ε it must also be far from $\hat{L}(h^*,S')$ by more than $\varepsilon/2$ with high probability, see the illustration in Figure 4.6. Formally, we have:

$$\begin{aligned}
&\mathbb{P}\Big(\exists h \in \mathcal{H} : \hat{L}(h,S') - \hat{L}(h,S) \geq \frac{\varepsilon}{2}\Big|\exists h \in \mathcal{H} : L(h) - \hat{L}(h,S) \geq \varepsilon\Big) \\
&\qquad \geq \mathbb{P}\Big(\hat{L}(h^*,S') - \hat{L}(h^*,S) \geq \frac{\varepsilon}{2}\Big|L(h^*) - \hat{L}(h^*,S) \geq \varepsilon\Big) && (4.7)\\
&\qquad \geq \mathbb{P}\Big(L(h^*) - \hat{L}(h^*,S') \leq \frac{\varepsilon}{2}\Big|L(h^*) - \hat{L}(h^*,S) \geq \varepsilon\Big) && (4.8)\\
&\qquad = \mathbb{P}\Big(L(h^*) - \hat{L}(h^*,S') \leq \frac{\varepsilon}{2}\Big) && (4.9)\\
&\qquad \geq 1 - \mathbb{P}\Big(L(h^*) - \hat{L}(h^*,S') \geq \frac{\varepsilon}{2}\Big) \\
&\qquad \geq 1 - e^{-2n(\varepsilon/2)^2} && (4.10)\\
&\qquad \geq \frac{1}{2}. && (4.11)
\end{aligned}$$

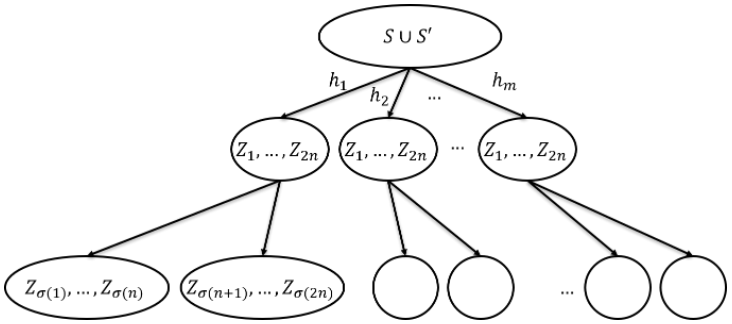
Explanation of the steps: in (4.7) the event on the left hand side includes the event on the right hand side; in (4.8) we have $\hat{L}(h,S') - \hat{L}(h,S) = \Big(L(h) - \hat{L}(h,S)\Big) - \Big(L(h) - \hat{L}(h,S')\Big)$ and since we are given that $L(h) - \hat{L}(h,S) \geq \varepsilon$ the event $\hat{L}(h,S') - \hat{L}(h,S) \geq \varepsilon/2$ follows from $L(h) - \hat{L}(h,S') \leq \varepsilon/2$, see Figure 4.6; in (4.9) we can remove the conditioning on S, because the event of interest concerns S', which is independent of S; (4.10) follows by Hoeffding's inequality; and (4.11) follows by the lemma's assumption on $e^{-n\varepsilon^2/2}$.

By plugging the result back into (4.6) and multiplying by 2 we obtain the statement of the lemma. □

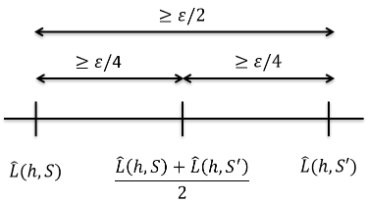
Step 3 [Projection] Now we focus on $\mathbb{P}\Big(\exists h \in \mathcal{H} : \hat{L}(h,S') - \hat{L}(h,S) \geq \frac{\varepsilon}{2}\Big)$, which concerns the behavior of $\mathcal{H}$ on two finite samples, S and S'. There are two possible ways to sample S and S'. The first is to sample S and then S'. An alternative way is to sample a joint sample $S_{2n} = S \cup S'$ and then split it into S and S' by randomly assigning half of the samples into S and half into S'. The two procedures are equivalent and lead to the same distribution over S and S'. We focus on the second procedure. Its advantage is that once we have sampled $S \cup S'$ the number of ways to label it with hypotheses from $\mathcal{H}$ is finite, even though $\mathcal{H}$ is uncountably infinite. This way we turn an uncountably infinite problem into a finite problem. The number of different sequences of losses on $S \cup S'$ is at most the number of different ways to label it, which is at most the growth function $m_{\mathcal{H}}(2n)$ by definition. The probability of having $\hat{L}(h,S') - \hat{L}(h,S) \geq \varepsilon/2$ for a fixed h reduces to the probability of splitting a sequence of $2n$ losses into n and n losses and having more than $\varepsilon/2$ difference between the average of the two. The latter reduces to the problem of sampling n losses without replacement from a bag of $2n$ losses and obtaining an average which deviates from the bag's average by more than $\varepsilon/4$, see Figure 4.7. This probability can be bounded by Hoeffding's inequality for sampling without replacement and decreases as $e^{-n\varepsilon^2/8}$. Putting this together we obtain the following result.

Lemma 4.10.

$$\mathbb{P}\Big(\exists h \in \mathcal{H} : \hat{L}(h,S') - \hat{L}(h,S) \geq \frac{\varepsilon}{2}\Big) \leq m_{\mathcal{H}}(2n)e^{-n\varepsilon^2/8}.$$



(a) Illustration of the split.



(b) Illustration of the distances.

Figure 4.7: **Illustration of the split of $S \cup S'$ into S and S'.** On the left: First we sample the joint sample $S \cup S'$. Then each hypothesis h_j produces a "big bag" of losses $\{Z_1, \dots, Z_{2n}\}$, where $Z_i = \ell(h_j(X_i), Y_i)$. Even though $\mathcal{H}$ is uncountably infinite, the number of different ways to label $S \cup S'$ is at most $m_{\mathcal{H}}(2n)$ by the definition of the growth function and thus the number of different "big bags" of losses is at most $m_{\mathcal{H}}(2n)$ (in the illustration we have $m \leq m_{\mathcal{H}}(2n)$). Finally, we split $S \cup S'$ into S and S', which corresponds to splitting the "big bags" of $2n$ losses into pairs of "small bags" of n losses, corresponding to $\hat{L}(h_j, S)$ and $\hat{L}(h_j, S')$. On the right: we illustrate the distances between the average losses in a pair of "small bags" and the corresponding "big bag", which is the average of the two "small bags".

As you may guess, $m_{\mathcal{H}}(2n)$ comes from a union bound over the number of possible sequences of losses we may obtain with hypotheses from $\mathcal{H}$ on $S \cup S'$. We now prove the lemma formally.

Proof of Lemma 4.10.

$$\begin{aligned}\mathbb{P}\Big(\exists h \in \mathcal{H} : \hat{L}(h, S') - \hat{L}(h, S) \geq \frac{\varepsilon}{2}\Big) &= \sum_{S \cup S'} \mathbb{P}(S \cup S')\mathbb{P}\Big(\exists h \in \mathcal{H} : \hat{L}(h, S') - \hat{L}(h, S) \geq \frac{\varepsilon}{2}\Big| S \cup S'\Big) \\ &\leq \sup_{S \cup S'} \mathbb{P}\Big(\exists h \in \mathcal{H} : \hat{L}(h, S') - \hat{L}(h, S) \geq \frac{\varepsilon}{2}\Big| S \cup S'\Big).\end{aligned}$$

Pay attention that the conditional probabilities are with respect to the splitting of $S \cup S'$ into S and S'.

Let $\mathcal{Z}(S \cup S') = \{Z_1, \dots, Z_{2n} : Z_i = \ell(h(X_i), Y_i), h \in \mathcal{H}\}$ be the set of all possible sequences of losses that can be obtained by applying $h \in \mathcal{H}$ to $S \cup S'$. Since there are at most $m_{\mathcal{H}}(2n)$ distinct ways to label $S \cup S'$ we have $|\mathcal{Z}(S \cup S')| \leq m_{\mathcal{H}}(2n)$. Let $\sigma : \{1, \dots, 2n\} \to \{1, \dots, 2n\}$ denote a permutation of indexes. We have

$$\begin{aligned}&\sup_{S \cup S'} \mathbb{P}\Big(\exists h \in \mathcal{H} : \hat{L}(h, S') - \hat{L}(h, S) \geq \frac{\varepsilon}{2}\Big| S \cup S'\Big) \\ &\quad= \sup_{S \cup S'} \mathbb{P}\left(\exists \{Z_1, \dots, Z_{2n}\} \in \mathcal{Z}(S \cup S') : \frac{1}{n}\sum_{i=1}^{n} Z_{\sigma(i)} - \frac{1}{n}\sum_{i=n+1}^{2n} Z_{\sigma(i)} \geq \frac{\varepsilon}{2}\right) && (4.12) \\ &\quad\leq \sup_{S \cup S'} \sum_{\{Z_1, \dots, Z_{2n}\} \in \mathcal{Z}(S \cup S')} \mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} Z_{\sigma(i)} - \frac{1}{n}\sum_{i=n+1}^{2n} Z_{\sigma(i)} \geq \frac{\varepsilon}{2}\right) && (4.13) \\ &\quad= \sup_{S \cup S'} \sum_{\{Z_1, \dots, Z_{2n}\} \in \mathcal{Z}(S \cup S')} \mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} Z_{\sigma(i)} - \frac{1}{2n}\sum_{i=1}^{2n} Z_i \geq \frac{\varepsilon}{4}\right) && (4.14) \\ &\quad\leq \sup_{S \cup S'} \sum_{\{Z_1, \dots, Z_{2n}\} \in \mathcal{Z}(S \cup S')} e^{-n\varepsilon^2/8} && (4.15) \\ &\quad\leq \sup_{S \cup S'} m_{\mathcal{H}}(2n) e^{-n\varepsilon^2/8} && (4.16) \\ &\quad= m_{\mathcal{H}}(2n) e^{-n\varepsilon^2/8},\end{aligned}$$

where (4.12) follows by the fact that $\mathcal{Z}(S \cup S')$ is the set of all possible losses on $S \cup S'$ and in the step of splitting $S \cup S'$ into S and S' and computing $\hat{L}(h, S')$ and $\hat{L}(h, S)$ we are splitting a "big bag" of $2n$

losses into two "small bags" of n and n; all that is left from $\mathcal{H}$ in the splitting process is $\mathcal{Z}(S \cup S')$; the probability in (4.12) is over the split of $S \cup S'$ into S and S', which is expressed by taking the first n elements of a random permutation σ of indexes into S' and the last n elements into S and the probability is over σ; in (4.13) we apply the union bound; for (4.14) see the illustration in Figure 4.7b; in (4.15) we apply Hoeffding's inequality for sampling without replacement (Theorem 3.8) to the process of randomly sampling n losses out of $2n$ and observing $\varepsilon/4$ deviation from the average; in (4.16) we apply the bound on $|\mathcal{Z}(S \cup S')|$. □

Step 4 [Putting Everything Together] All that is left for the proof of Theorem 4.8 is to put Lemmas 4.9 and 4.10 together.

Proof of Theorem 4.8. Assuming that $e^{-n\varepsilon^2/2} \leq 1/2$ we have by Lemmas 4.9 and 4.10:

$$\begin{aligned}\mathbb{P}\Big(\exists h \in \mathcal{H} : L(h) - \hat{L}(h, S) \geq \varepsilon\Big) &\leq 2\mathbb{P}\Big(\exists h \in \mathcal{H} : \hat{L}(h, S') - \hat{L}(h, S) \geq \frac{\varepsilon}{2}\Big) \\ &\leq 2m_{\mathcal{H}}(2n)e^{-n\varepsilon^2/8}.\end{aligned}$$

Note that if $e^{-n\varepsilon^2/2} > 1/2$ then $2m_{\mathcal{H}}(2n)e^{-n\varepsilon^2/8} > 1$ and the inequality is satisfied trivially (because probabilities are always upper bounded by 1).

By denoting the right hand side of the inequality by δ and solving for ε we obtain the result. □

4.5.2 Bounding the Growth Function: The VC-dimension

In Theorem 4.8 we relate the distance between the expected and empirical losses to the growth function of $\mathcal{H}$. Our next goal is to bound the growth function. In order to do so we introduce the concept of shattering and the VC dimension.

Definition 4.11. *A set of points $x_1, \ldots, x_n$ is* shattered *by $\mathcal{H}$ if functions from $\mathcal{H}$ can produce all possible binary labellings of $x_1, \ldots, x_n$ or, in other words, if*

$$\|\mathcal{H}(x_1, \ldots, x_n)\| = 2^n.$$

For example, the set of homogeneous linear separators in $\mathbb{R}^2$ shatters the two points in Figure 4.5a, but it does not shatter the three points in Figure 4.5b. Note that if two points lie on one line passing through the origin, they are not shattered by the set of homogeneous linear separators, because they always get the same label. Thus, we may have two sets of points of the same size, where one is shattered and the other is not.

Definition 4.12. *The* Vapnik-Chervonenkis *(*VC*)* dimension *of $\mathcal{H}$, denoted by $d_{\mathrm{VC}}(\mathcal{H})$ is the maximal number of points that can be shattered by $\mathcal{H}$. In other words,*

$$d_{\mathrm{VC}}(\mathcal{H}) = \max\{n | m_{\mathcal{H}}(n) = 2^n\}.$$

If $m_{\mathcal{H}}(n) = 2^n$ for all n, then $d_{\mathrm{VC}}(\mathcal{H}) = \infty$.

Similar to the growth function, if we want to show that $d_{\mathrm{VC}}(\mathcal{H}) = d$ we have to show that $d_{\mathrm{VC}}(\mathcal{H}) \geq d$ and $d_{\mathrm{VC}}(\mathcal{H}) \leq d$. For example, the illustration in Figure 4.5a provides a configuration of points that are shattered by homogeneous separating hyperplanes in $\mathbb{R}^2$ and thus shows that the VC-dimension of homogeneous separating hyperplanes in $\mathbb{R}^2$ is at least 2. However, the illustration in Figure 4.5b *does not* demonstrate that the VC-dimension of homogeneous separating hyperplanes in R^2 is smaller than 3. If we want to show that the VC-dimension of homogeneous separating hyperplanes in $\mathbb{R}^2$ is smaller than 3 we have to prove that no configuration of 3 points can be shattered. It is not sufficient to show that one particular configuration of points cannot be shattered. In the same spirit, two points lying on the same line passing through the origin cannot be shattered by homogeneous linear separators, but this does not tell anything about the VC-dimension, because we have another configuration of two points in Figure 4.5a that can be shattered. It is possible to show that the VC-dimension of homogeneous separating hyperplanes in $\mathbb{R}^d$ is d and the VC-dimension of general separating hyperplanes in $\mathbb{R}^d$ (not necessarily passing through the origin) is $d + 1$, see Abu-Mostafa et al. (2012, Exercise 2.4).

The next theorem bounds the growth function in terms of the VC-dimension.

Theorem 4.13 (Sauer's Lemma)**.**

$$m_{\mathcal{H}}(n) \leq \sum_{i=0}^{d_{\mathrm{VC}}(\mathcal{H})} \binom{n}{i}. \tag{4.17}$$

We remind that the binomial coefficient $\binom{n}{k}$ counts the number of ways to pick k elements out of n and that for $n < k$ it is defined as $\binom{n}{k} = 0$. Thus, equation (4.17) is well-defined even when $n < d_{\mathrm{VC}}(\mathcal{H})$. We also remind that $\sum_{i=0}^{n} \binom{n}{i} = 2^n$, where 2^n is the number of all possible subsets of n elements, which is equal to the sum over i going from 0 to n to select i elements out of n. For $n \leq d_{\mathrm{VC}}(\mathcal{H})$ we have $m_{\mathcal{H}}(n) = 2^n$ and the inequality is satisfied trivially.

The proof of Theorem 4.13 slightly reminds the combinatorial proof of the binomial identity

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}.$$

One way to count the number of ways to select k elements out of n on the right hand side is to take one element aside. If that element is selected, then we have $\binom{n-1}{k-1}$ possibilities to select $k-1$ additional elements out of the remaining $n-1$. If the element is not selected, then we have $\binom{n-1}{k}$ possibilities to select all k elements out of remaining $n-1$. The sets including the first element are disjoint from the sets excluding it, leading to the identity above.

We need one more definition for the proof of Theorem 4.13.

Definition 4.14. *Let $B(n, d)$ be the maximal number of possible ways to label n points, so that no $d+1$ points are shattered.*

By the definition, we have $m_{\mathcal{H}}(n) \leq B(n, d_{\mathrm{VC}}(\mathcal{H}))$.

Proof of Theorem 4.13. We prove by induction that

$$B(n, d) \leq \sum_{i=0}^{d} \binom{n}{i}. \tag{4.18}$$

For the induction base we have $B(n, 0) = 1 = \binom{n}{0}$: if no points are shattered there is just one way to label the points. If there would be more than one way, they would differ in at least one point and that point would be shattered. By the definition of binomial coefficients, which says that for $k > n$ we have $\binom{n}{k} = 0$, we also know that for $n < d$ we have $B(n, d) = B(n, n)$. In particular, $B(0, d) = B(0, 0) = 1$.

Now we proceed with induction on d and for each d we do an induction on n. We show that

$$B(n, d) \leq B(n-1, d) + B(n-1, d-1).$$

Let $\mathcal{S}$ be a maximal set of dichotomies (labeling patterns) on n points $x_1, \ldots, x_n$. We take one point aside, x_n, and split $\mathcal{S}$ into three disjoint subsets: $\mathcal{S} = \mathcal{S}^* \cup \mathcal{S}^+ \cup \mathcal{S}^-$. The set $\mathcal{S}^*$ contains dichotomies on n points that appear with just one sign on x_n, either positive or negative. The sets $\mathcal{S}^+$ and $\mathcal{S}^-$ contain all dichotomies that appear with both positive and negative sign on x_n, where the positive ones are collected in $\mathcal{S}^+$ and the negative ones are collected in $\mathcal{S}^-$. Thus, the sets $\mathcal{S}^+$ and $\mathcal{S}^-$ are identical except in their labeling of x_n, where in $\mathcal{S}^+$ it is always labeled as $+$ and in $\mathcal{S}^-$ always as $-$. By contradiction, the number of points $x_1, \ldots, x_{n-1}$ that are shattered by $\mathcal{S}^-$ cannot be larger than $d-1$, because otherwise the number of points that are shattered by $\mathcal{S}$, which includes $\mathcal{S}^+$ and $\mathcal{S}^-$, would be larger than d, since we can use $\mathcal{S}^+$ and $\mathcal{S}^-$ to add x_n to the set of shattered points. Therefore, $|\mathcal{S}^-| \leq B(n-1, d-1)$. At the same time, the number of points $x_1, \ldots, x_{n-1}$ that are shattered by $\mathcal{S}^* \cup \mathcal{S}^+$ cannot be larger than d, because the total number of points shattered by $\mathcal{S}$ is at most d. Thus, we have $|\mathcal{S}^* \cup \mathcal{S}^+| \leq B(n-1, d)$. And overall

$$B(n, d) = |\mathcal{S}| = |\mathcal{S}^* \cup \mathcal{S}^+| + |\mathcal{S}^-| \leq B(n-1, d) + B(n-1, d-1),$$

as desired. By the induction assumption equation (4.18) is satisfied for $B(n-1, d)$ and $B(n-1, d-1)$,

and we have

$$\begin{aligned} B(n,d) &\leq \sum_{i=0}^{d} \binom{n-1}{i} + \sum_{i=0}^{d-1} \binom{n-1}{i} \\ &= 1 + \sum_{i=0}^{d-1} \left(\binom{n-1}{i+1} + \binom{n-1}{i} \right) \\ &= \sum_{i=0}^{d} \binom{n}{i}, \end{aligned}$$

as desired. Finally, as we have already observed, $m_{\mathcal{H}}(n) \leq B(n, d_{\mathrm{VC}}(\mathcal{H}))$, completing the proof. □

The following lemma provides a more explicit bound on the growth function.

Lemma 4.15.

$$\sum_{i=0}^{d} \binom{n}{i} \leq n^d + 1.$$

The proof is based on induction, see Exercise 4.11.

By plugging the results of Theorem 4.13 and Lemma 4.15 into Theorem 4.8 we obtain the VC generalization bound.

Theorem 4.16 (VC generalization bound)**.** *Let $\mathcal{H}$ be a hypotheses class with VC-dimension $d_{\mathrm{VC}}(\mathcal{H}) = d_{\mathrm{VC}}$. Then:*

$$\mathbb{P}\left(\exists h \in \mathcal{H} : L(h) \geq \hat{L}(h,S) + \sqrt{\frac{8 \ln \left(2 \left((2n)^{d_{\mathrm{VC}}} + 1 \right) / \delta \right)}{n}} \right) \leq \delta.$$

For example, the VC-dimension of linear separators in $\mathbb{R}^d$ is $d+1$ and theorem 4.16 provides generalization guarantees for learning with linear separators in finite-dimensional spaces, as long as the dimension of the space d is small in relation to the number of points n.

4.6 VC Analysis of SVMs

Kernel Support Vector Machines (SVMs) can map the data into high and potentially infinite-dimensional spaces. For example, Radial Basis Function (RBF) kernels map the data into an infinite-dimensional space. In the following we provide a more refined analysis of generalization in learning with linear separators in high-dimensional spaces. The analysis is based on the notion of *separation with a margin*. We use the following definitions.

Definition 4.17 (Fat Shattering)**.** *Let $\mathcal{H}_\gamma = \{(\mathbf{w}, b) : \|\mathbf{w}\| \leq 1/\gamma\}$ be the space of hyperplanes described by $\mathbf{w}$ and b, where $\mathbf{w}$ is a vector in $\mathbb{R}^d$ (with potentially infinite dimension d) with $\|\mathbf{w}\| \leq 1/\gamma$ and $b \in \mathbb{R}$. We say that a set of points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ is* fat-shattered *by $\mathcal{H}_\gamma$ if for any set of labels $\{y_1, \ldots, y_n\} \in \{\pm 1\}^n$ we have a hyperplane $(\mathbf{w}, b) \in \mathcal{H}_\gamma$ that satisfies $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1$ for all $i \in \{1, \ldots, n\}$.*

Note that when $y = \operatorname{sign}(\langle \mathbf{w}, \mathbf{x} \rangle + b)$ the distance of a point $\mathbf{x}$ to a hyperplane h defined by $(\mathbf{w}, b)$ is given by $\operatorname{dist}(h, \mathbf{x}) = \frac{y(\langle \mathbf{w}, \mathbf{x} \rangle + b)}{\|\mathbf{w}\|}$ (Abu-Mostafa et al., 2015, Page 5, Chapter 8) and for $h = (\mathbf{w}, b) \in \mathcal{H}_\gamma$ and $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ fat-shattered by $\mathcal{H}_\gamma$ we obtain $\operatorname{dist}(h, \mathbf{x}_i) \geq \gamma$ for all $i \in \{1, \ldots, n\}$. It means that any possible labeling of $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ can be achieved with margin at least γ.

Definition 4.18 (Fat Shattering Dimension)**.** *We say that* fat shattering dimension $d_{\mathrm{FAT}}(\mathcal{H}_\gamma) = d$ *if d is the maximal number of points that can be fat shattered by $\mathcal{H}_\gamma$. (I.e., there exist d points that can be fat shattered by $\mathcal{H}_\gamma$ and no $d+1$ points can be fat shattered by $\mathcal{H}_\gamma$.)*

Note that $d_{\mathrm{FAT}}(\mathcal{H}_\gamma) \leq d_{\mathrm{VC}}(\mathcal{H}_\gamma) \leq d + 1$, where d is the dimension of $\mathbf{w}$. (If we can shatter n points with margin γ we can also shatter them without the margin.)

The following theorem bounds the fat shattering dimension of $\mathcal{H}_\gamma$, see Abu-Mostafa et al. (2015) for a proof.

Theorem 4.19 ((Abu-Mostafa et al., 2015, Theorem 8.5)). *Assume that the input space $\mathcal{X}$ is a ball of radius R in $\mathbb{R}^d$ (i.e., $\|x\| \leq R$ for all $x \in \mathcal{X}$), where d may potentially be infinite. Then:*

$$d_{\text{FAT}}(\mathcal{H}_\gamma) \leq \lceil R^2/\gamma^2 \rceil + 1,$$

where $\lceil R^2/\gamma^2 \rceil$ is the smallest integer that is greater or equal to R^2/γ^2.

The important point is that the bound on fat shattering dimension is independent of the dimension of the space $\mathbb{R}^d$ that $\mathbf{w}$ comes from.

We define fat losses that count as error everything that falls too close to the separating hyperplane or on the wrong side of it.

Definition 4.20 (Fat Losses). *For $h = (\mathbf{w}, b)$ we define the fat losses*

$$\ell_{\text{FAT}}(h(\mathbf{x}), y) = \begin{cases} 0, & \text{if } y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 \\ 1, & \text{otherwise,} \end{cases}$$
$$L_{\text{FAT}}(h) = \mathbb{E}\left[\ell_{\text{FAT}}(h(\mathbf{X}), Y)\right],$$
$$\hat{L}_{\text{FAT}}(h, S) = \frac{1}{n}\sum_{i=1}^{n} \ell_{\text{FAT}}(h(\mathbf{X}_i), Y_i).$$

In relation to the fat losses the fat shattering dimension acts in the same way as the VC-dimension in relation to the zero-one loss. In particular, we have the following result that relates $L_{\text{FAT}}(h)$ to $\hat{L}_{\text{FAT}}(h, S)$ via $d_{\text{FAT}}(\mathcal{H}_\gamma)$ (the proof is left as an exercise).

Theorem 4.21.

$$\mathbb{P}\left(\exists h \in \mathcal{H}_\gamma : L_{\text{FAT}}(h) \geq \hat{L}_{\text{FAT}}(h, S) + \sqrt{\frac{8 \ln\left(2\left((2n)^{d_{\text{FAT}}(\mathcal{H}_\gamma)} + 1\right)/\delta\right)}{n}} \right) \leq \delta.$$

Now we are ready to analyze generalization in learning with fat linear separation. For the analysis we make a simplifying assumption that the data are contained within a ball of radius $R = 1$. The analysis for general R is left as an exercise. Note that R refers to the radius of the ball *after* potential transformation of the data through a feature mapping / kernel function. For example, the RBF kernel maps the data into an infinite dimensional space and we consider the radius of the ball containing the transformed data in the infinite dimensional space.

Theorem 4.22. *Assume that the input space $\mathcal{X}$ is a ball of radius $R = 1$ in $\mathbb{R}^d$, where d is potentially infinite. Let $\mathcal{H}$ be the space of linear separators $h = (\mathbf{w}, b)$. Then*

$$\mathbb{P}\left(\exists h \in \mathcal{H} : L_{\text{FAT}}(h) \geq \hat{L}_{\text{FAT}}(h, S) + \sqrt{\frac{8 \ln\left(2\left((2n)^{1+\lceil \|\mathbf{w}\|^2 \rceil} + 1\right)(1 + \lceil \|\mathbf{w}\|^2 \rceil)\lceil \|\mathbf{w}\|^2 \rceil/\delta\right)}{n}} \right) \leq \delta.$$

Observe that $L(h) \leq L_{\text{FAT}}(h)$ and, therefore, the theorem provides a generalization bound for $L(h)$. (If we count correct classifications within the margin as errors we only increase the loss.)

Proof. The proof is based on combination of VC and Occam's razor bounding techniques, see the illustration in Figure 4.8. We start by noting that Theorem 4.19 is interesting when $\lceil R^2/\gamma^2 \rceil < d + 1$, because as we have already noted $d_{\text{FAT}}(\mathcal{H}_\gamma) \leq d_{\text{VC}}(\mathcal{H}_\gamma) \leq d + 1$. We slice the hypotheses space $\mathcal{H}$ into a nested sequence of subspaces $\mathcal{H}_1 \subset \mathcal{H}_2 \subset \cdots \subset \mathcal{H}_{d-1} \subset \mathcal{H}_d = \mathcal{H}$, where for all $i < d$ we define $\mathcal{H}_i$ to be the hypothesis space $\mathcal{H}_\gamma$ with $1/\gamma^2 = i$. In other words, $\mathcal{H}_i = \mathcal{H}_{\left\{\gamma = \frac{1}{\sqrt{i}}\right\}}$ (do not let the notation to confuse you, by $\mathcal{H}_i$ we denote the i-th hypothesis space in the nested sequence of hypothesis spaces and

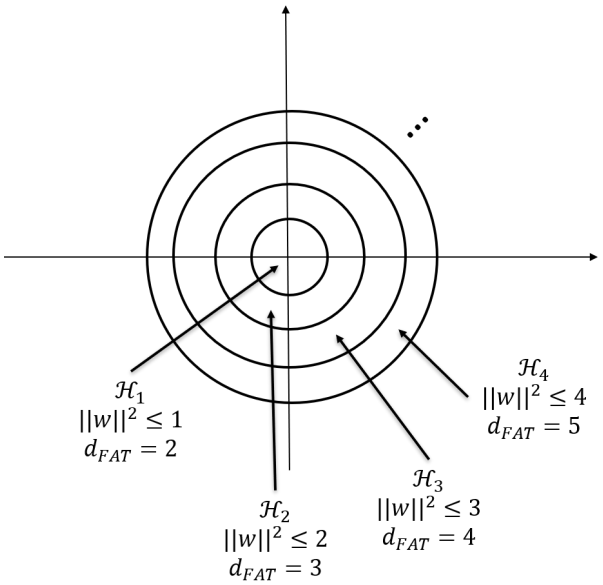


Figure 4.8: **Illustration for the proof of Theorem 4.22**

by $\mathcal{H}_\gamma$ we denote the hypothesis space with $\|\mathbf{w}\|$ upper bounded by $1/\gamma$). By Theorem 4.19 we have $d_{\text{FAT}}(\mathcal{H}_i) = i + 1$ and then by Theorem 4.21:

$$\mathbb{P}\left(\exists h \in \mathcal{H}_i : L_{\text{FAT}}(h) \geq \hat{L}_{\text{FAT}}(h, S) + \sqrt{\frac{8 \ln \left(2 \left((2n)^{1+i} + 1 \right) / \delta_i \right)}{n}} \right) \leq \delta_i.$$

We take $\delta_i = \frac{1}{i(i+1)}\delta$ and note that $\sum_{i=1}^{\infty} \frac{1}{i(i+1)} = \sum_{i=1}^{\infty} \left(\frac{1}{i} - \frac{1}{i+1} \right) = \left(1 - \frac{1}{2}\right) + \left(\frac{1}{2} - \frac{1}{3}\right) + \left(\frac{1}{3} - \frac{1}{4}\right) + \cdots = 1$. We also note that $\mathcal{H} = \bigcup_{i=1}^{d} (\mathcal{H}_i \setminus \mathcal{H}_{i-1})$, where $\mathcal{H}_0$ is defined as the empty set and $\mathcal{H}_i \setminus \mathcal{H}_{i-1}$ is the difference between sets $\mathcal{H}_i$ and $\mathcal{H}_{i-1}$ (everything that is in $\mathcal{H}_i$, but not in $\mathcal{H}_{i-1}$). Note that the sets $\mathcal{H}_i \setminus \mathcal{H}_{i-1}$ and $\mathcal{H}_j \setminus \mathcal{H}_{j-1}$ are disjoint for $i \neq j$. Also note that δ_i is a distribution of our confidence budget δ among $\mathcal{H}_i \setminus \mathcal{H}_{i-1}$-s. Finally, note that if $h = (\mathbf{w}, b) \in \mathcal{H}_i \setminus \mathcal{H}_{i-1}$ then $\lceil \|\mathbf{w}\|^2 \rceil = i$. The remainder of

the proof follows the same lines as the proof of Occam's razor bound:

$$
\begin{aligned}
&\mathbb{P}\left(\exists h \in \mathcal{H} : L_{\mathrm{FAT}}(h) \geq \hat{L}_{\mathrm{FAT}}(h,S) + \sqrt{\frac{8\ln\left(2\left((2n)^{1+\lceil\|\mathbf{w}\|^2\rceil}+1\right)(1+\lceil\|\mathbf{w}\|^2\rceil)\lceil\|\mathbf{w}\|^2\rceil/\delta\right)}{n}}\right)\\
&\quad= \mathbb{P}\left(\exists h \in \bigcup_{i=1}^{d}\mathcal{H}_i \setminus \mathcal{H}_{i-1} : L_{\mathrm{FAT}}(h) \geq \hat{L}_{\mathrm{FAT}}(h,S) + \sqrt{\frac{8\ln\left(2\left((2n)^{1+\lceil\|\mathbf{w}\|^2\rceil}+1\right)(1+\lceil\|\mathbf{w}\|^2\rceil)\lceil\|\mathbf{w}\|^2\rceil/\delta\right)}{n}}\right)\\
&\quad= \sum_{i=1}^{d}\mathbb{P}\left(\exists h \in \mathcal{H}_i \setminus \mathcal{H}_{i-1} : L_{\mathrm{FAT}}(h) \geq \hat{L}_{\mathrm{FAT}}(h,S) + \sqrt{\frac{8\ln\left(2\left((2n)^{1+\lceil\|\mathbf{w}\|^2\rceil}+1\right)(1+\lceil\|\mathbf{w}\|^2\rceil)\lceil\|\mathbf{w}\|^2\rceil/\delta\right)}{n}}\right)\\
&\quad= \sum_{i=1}^{d}\mathbb{P}\left(\exists h \in \mathcal{H}_i \setminus \mathcal{H}_{i-1} : L_{\mathrm{FAT}}(h) \geq \hat{L}_{\mathrm{FAT}}(h,S) + \sqrt{\frac{8\ln\left(2\left((2n)^{1+i}+1\right)(1+i)\,i/\delta\right)}{n}}\right)\\
&\quad= \sum_{i=1}^{d}\mathbb{P}\left(\exists h \in \mathcal{H}_i \setminus \mathcal{H}_{i-1} : L_{\mathrm{FAT}}(h) \geq \hat{L}_{\mathrm{FAT}}(h,S) + \sqrt{\frac{8\ln\left(2\left((2n)^{1+i}+1\right)/\delta_i\right)}{n}}\right)\\
&\quad\leq \sum_{i=1}^{d}\mathbb{P}\left(\exists h \in \mathcal{H}_i : L_{\mathrm{FAT}}(h) \geq \hat{L}_{\mathrm{FAT}}(h,S) + \sqrt{\frac{8\ln\left(2\left((2n)^{1+i}+1\right)/\delta_i\right)}{n}}\right)\\
&\quad\leq \sum_{i=1}^{d}\delta_i = \sum_{i=1}^{d}\frac{1}{i(i+1)}\delta = \delta\sum_{i=1}^{d}\frac{1}{i(i+1)} \leq \delta\sum_{i=1}^{\infty}\frac{1}{i(i+1)} = \delta.
\end{aligned}
$$

□

4.7 VC Lower Bound

In this section we show that when the VC-dimension is unbounded, it is impossible to bound the distance between $L(h)$ and $\hat{L}(h,S)$.

Theorem 4.23. *Let $\mathcal{H}$ be a hypothesis class with $d_{VC}(\mathcal{H}) = \infty$. Then for any n there exists a distribution over $\mathcal{X}$ and a class of target functions $\mathcal{F}$, such that*

$$\mathbb{E}\left[\sup_h\left(L(h) - \hat{L}(h,S)\right)\right] \geq 0.25,$$

where the expectation is over selection of a sample of size n and a target function.

Proof. Pick n. Since $d_{VC}(\mathcal{H}) = \infty$ we know that there exist $2n$ points that are shattered by $\mathcal{H}$. Let the sample space $\mathcal{X}_{2n} = \{x_1, \ldots, x_{2n}\}$ be these points and let $p(x)$ be uniform on $\mathcal{X}_{2n}$. Let $\mathcal{F}$ be the set of all possible functions from $\mathcal{X}_{2n}$ to $\{0,1\}$ and let $p(f)$ be uniform over $\mathcal{F}$. Let S be a sample of n points. Let $\{\mathcal{F}_k(S)\}_k$ be maximal subsets of $\mathcal{F}$, such that $\mathcal{F} = \bigcup_k \mathcal{F}_k(S)$ and any $f_i, f_j \in \mathcal{F}_k(S)$ agree on S. Note that since $\mathcal{X}_{2n}$ is shattered by $\mathcal{H}$, for any S, any $\mathcal{F}_k$, and any $f_i \in \mathcal{F}_k$ that was used to label S there exists $h^*(\mathcal{F}_k(S), S) \in \mathcal{H}$, such that for any $f_i \in \mathcal{F}_k(S)$ the empirical error $\hat{L}(h^*(f_i,S),S) = 0$. Let

$p(k)$ and $p(i)$ be uniform. Then:

$$\begin{aligned}
\mathbb{E}\left[\sup_h \left(L(h) - \hat{L}(h,S)\right)\right] &= \mathbb{E}_{f\sim p(f)}\left[\mathbb{E}_{S\sim p(X)^n}\left[\sup_h \left(L(h) - \hat{L}(h,S)\right)\right]\Big| f\right] \\
&= \mathbb{E}_{S\sim p(X)^n}\left[\mathbb{E}_{f\sim p(f)}\left[\sup_h \left(L(h) - \hat{L}(h,S)\right)\right]\Big| S\right] \\
&= \mathbb{E}_{S\sim p(X)^n}\left[\mathbb{E}_{k\sim p(k)}\left[\mathbb{E}_{i\sim p(i)}\left[\sup_h \left(L(h) - \hat{L}(h,S)\right)\right]\Big|\mathcal{F}_k\right]\Big| S\right] \\
&\geq \mathbb{E}_{S\sim p(X)^n}\left[\mathbb{E}_{k\sim p(k)}\left[\mathbb{E}_{i\sim p(i)}\left[L(h^*(\mathcal{F}_k,S)) - \hat{L}(h^*(\mathcal{F}_k,S),S)\right]\Big|\mathcal{F}_k\right]\Big| S\right] \\
&= \mathbb{E}_{S\sim p(X)^n}\left[\mathbb{E}_{k\sim p(k)}\left[\mathbb{E}_{i\sim p(i)}\left[L(h^*(\mathcal{F}_k,S))\right]\Big|\mathcal{F}_k\right]\Big| S\right] \\
&= \mathbb{E}_{S\sim p(X)^n}\left[\mathbb{E}_{k\sim p(k)}\left[0.25\right]\Big| S\right] \\
&= 0.25.
\end{aligned}$$

□

Corollary 4.24. *Under the assumptions of Theorem 4.23, with probability at least 0.125,* $\sup_h(L(h) - \hat{L}(h,S)) \geq 0.125$. *Thus, it is impossible to have high-probability bounds on* $\sup_h(L(h) - \hat{L}(h,S))$ *that converge to zero as n goes to infinity.*

Proof. Note that $\sup_h(L(h) - \hat{L}(h,S)) \leq 1$, since ℓ is bounded in $[0,1]$. Assume by contradiction that $\mathbb{P}\Big(\sup_h \left(L(h) - \hat{L}(h,S)\right) \geq 0.125\Big) < 0.125$. Then

$$\mathbb{E}\left[\sup_h \left(L(h) - \hat{L}(h,S)\right)\right] \leq 0.125 \times 1 + (1 - 0.125) \times 0.125 < 2 \times 0.125 = 0.25,$$

which is in contradiction with Theorem 4.23. □

4.8 PAC-Bayesian Analysis

Occam's razor and VC analysis consider hard selection of a single hypothesis from a hypothesis class. In PAC-Bayesian analysis hard selection is replaced by a soft selection: instead of selecting a single hypothesis, it is allowed to select a distribution over the hypothesis space. When the distribution is a delta-distribution allocating all the mass to a single hypothesis, PAC-Bayes recovers the Occam's razor bound. However, the possibility of soft selection provides much more freedom and control over the approximation-estimation trade-off.

PAC-Bayes achieves generalization by employing *active avoidance of selection* when it is not necessary. For example, if two classifiers achieve the same empirical error on a sample, there is no reason to select among them. PAC-Bayes achieves this by spreading the probability of selecting classifiers uniformly over classifiers that are indistinguishable based on the sample. It leads to reduction of estimation error (because there is less selection) without affecting the approximation error.

PAC-Bayesian generalization bounds are based on *change of measure* inequality, which acts as a replacement for the union bound. Change of measure inequality has two important advantages over the union bound: (1) it is tighter (see Exercise 4.7) and (2) it can be applied to uncountably infinite hypothesis classes. Additionally, soft selection allows application of gradient-descent type methods to optimize the distribution over $\mathcal{H}$, which in some cases leads to efficient algorithms for direct minimization of the PAC-Bayesian bounds.

Soft selection is implemented by *randomized classifiers*, which are formally defined below.

Definition 4.25 (Randomized Classifier)**.** *Let ρ be a distribution over $\mathcal{H}$. A* randomized classifier *associated with ρ (and named ρ) acts according to the following scheme. At each prediction round it:*

1. *Picks $h \in \mathcal{H}$ according to $\rho(h)$*
2. *Observes x*
3. *Returns $h(x)$*

The expected loss of ρ is $\mathbb{E}_{h\sim\rho}[L(h)]$ and the empirical loss is $\mathbb{E}_{h\sim\rho}\left[\hat{L}(h,S)\right]$. Whenever it does not lead to confusion, we will shorten the notation to $\mathbb{E}_\rho[L(h)]$ and $\mathbb{E}_\rho\left[\hat{L}(h,S)\right]$.

There is a large number of different PAC-Bayesian inequalities. We start with the classical one due to Seeger (2002).

Theorem 4.26 (PAC-Bayes-kl inequality)**.** *For any "prior" distribution π over $\mathcal{H}$ that is independent of S and $\delta \in (0,1]$:*

$$\mathbb{P}\left(\exists \rho : \mathrm{kl}\left(\mathbb{E}_\rho\left[\hat{L}(h,S)\right]\middle\|\mathbb{E}_\rho[L(h)]\right) \geq \frac{\mathrm{KL}(\rho\|\pi) + \ln\frac{2\sqrt{n}}{\delta}}{n}\right) \leq \delta. \tag{4.19}$$

Another way of reading the theorem is that with probability at least $1-\delta$, for all "posterior" distributions ρ over $\mathcal{H}$:

$$\mathrm{kl}\left(\mathbb{E}_\rho\left[\hat{L}(h,S)\right]\middle\|\mathbb{E}_\rho[L(h)]\right) \leq \frac{\mathrm{KL}(\rho\|\pi) + \ln\frac{2\sqrt{n}}{\delta}}{n}.$$

The meaning of "prior" should be interpreted in exactly the same way as the "prior" in Occam's razor bound: it is any distribution over $\mathcal{H}$ that sums up to one and does not depend on the sample S. The prior is an auxiliary construction for deriving the bound, and in contrast to Bayesian learning there is no assumption that it reflects any real-world distribution over $\mathcal{H}$. Distribution ρ is called a "posterior" distribution, because it is allowed to depend on the sample. The bound holds for all posterior distributions, including the Bayes posterior, and, therefore, it can be used to provide generalization guarantees for Bayesian learning. The posterior distribution minimizing the bound is typically *not* the Bayes posterior.

Before proceeding to the proof of the theorem (given in Section 4.8.2), we provide some intuition on what the theorem tells us. Since the kl inequality is not the most intuitive divergence measure, we start by applying Pinsker's relaxation of the kl (Theorem 3.29) to obtain a more digestible (although weaker) form of the bound: with probability at least $1-\delta$, for all distributions ρ

$$\mathbb{E}_\rho[L(h)] \leq \mathbb{E}_\rho\left[\hat{L}(h,S)\right] + \sqrt{\frac{\mathrm{KL}(\rho\|\pi) + \ln\frac{2\sqrt{n}}{\delta}}{2n}}. \tag{4.20}$$

Note that for $\rho = \pi$ the KL term is zero and we recover the generalization bound for a single hypothesis in Theorem 4.1 (except the minor difference of $\ln\frac{1}{\delta}$ being replaced by $\ln\frac{2\sqrt{n}}{\delta}$, which we ignore for now). Taking $\rho = \pi$ amounts to making no selection. Thus, if we start with a prior distribution π and stay with it after observing the sample, we take no information from the sample (make no selection), and we retain the usual Hoeffding's or kl guarantee on generalization of a single prediction rule (a prediction rule that is independent of the sample).

More generally, the amount of selection based on the sample is measured by $\mathrm{KL}(\rho\|\pi)$. The more the posterior distribution ρ deviates from the prior distribution π in the $\mathrm{KL}(\rho\|\pi)$ sense, the more information is taken from the sample, and the complexity term $\mathrm{KL}(\rho\|\pi)$ goes up. On the other hand, allocating higher mass ρ to prediction rules with small empirical error $\hat{L}(h,S)$ reduces the first term of the bound, $\mathbb{E}_\rho[\hat{L}(h,S)]$. Therefore, there is a trade-off between giving preference to prediction rules with good empirical performance $\hat{L}(h,S)$ and keeping ρ close to the prior π.

There is an important difference between the PAC-Bayes bounds and the VC bounds. In the VC bounds the complexity is measured by the VC dimension *of the hypothesis class*, meaning that the complexity cost d_{VC} is paid upfront (before observing the sample), and then the algorithm is allowed to select any prediction rule in $\mathcal{H}$. It can be compared to an all-you-can-eat buffet restaurant, where the customers pay at the entrance, and then get the freedom to select anything they want. Note that in the VC analysis there is no way to define complexity of individual prediction rules, because the complexity is of the hypothesis class. (It is possible to slice a hypothesis space into subclasses, and combine the VC analysis with the Occam's razor to measure the complexity at a subclass level, as we did in the analysis of SVMs, but it is much less flexible than what can be done with PAC-Bayes.) In PAC-Bayes the complexity is measured by $\mathrm{KL}(\rho\|\pi)$, where ρ can be tuned based on the sample. Thus, it measures the actual amount of selection done based on the sample, and if there was no selection ($\rho = \pi$), the cost is

zero. This can be compared to a restaurant, where the customers pay at the exit based on the dishes they have actually taken *after* seeing the menu (the sample). And if they did not select any of the dishes, they exit without paying anything. Moreover, PAC-Bayes provides a possibility to define complexity $\pi(h)$ for each prediction rule individually (same as in Occam's razor, but now also for uncountably infinite sets of prediction rules) and this way incorporate prior information into learning.

For further intuition on the bound we decompose the KL-divergence:

$$\mathrm{KL}(\rho\|\pi) = \mathbb{E}_\rho\left[\ln\frac{\rho(h)}{\pi(h)}\right] = \underbrace{\mathbb{E}_\rho\left[\ln\frac{1}{\pi(h)}\right]}_{\substack{\text{Average}\\ \text{complexity}}} - \underbrace{\mathrm{H}(\rho)}_{\text{Entropy}}.$$

The first term in the decomposition, $\mathbb{E}_\rho\left[\ln\frac{1}{\pi(h)}\right]$, is the average complexity $\ln\frac{1}{\pi(h)}$ of prediction rules when they are selected according to ρ, where the complexity is defined by the prior distribution π. The second term, $\mathrm{H}(\rho)$, is the entropy of the posterior distribution ρ. The entropy is highest when ρ is close to a uniform distribution, and it is lowest when ρ is close to a delta-distribution. Thus, $\mathrm{H}(\rho)$ can be seen as a "bonus" for avoidance of selection we have mentioned earlier. If $\mathcal{H}$ is countable and ρ places all the probability mass on a single prediction rule (implying hard selection), then $\mathrm{H}(\rho) = 0$ and there is no "bonus", whereas is ρ spreads the probability mass over multiple prediction rules, then $-\mathrm{H}(\rho) < 0$ decreases the complexity term.

For hard selection from a countable set of prediction rules, the bound in Equation (4.20) recovers the Occam's razor bound in Theorem 4.3 (ignoring the $\ln 2\sqrt{n}$ term). Note that if there are several prediction rules with identical $\hat{L}(h,S)$ and $\pi(h)$, then spreading ρ uniformly over them yields $-\mathrm{H}(\rho)$ bonus in the complexity term without affecting any other terms in the bound, which once again demonstrates that avoiding selection when it is not necessary improves generalization.

A tighter relaxation of the kl divergence based on Refined Pinsker's Inequality in Theorem 3.32 illustrates additional properties of the PAC-Bayes-kl inequality. By applying the relaxation to Theorem 4.26 we obtain that with probability at least $1-\delta$, for all distributions ρ

$$\mathbb{E}_\rho[L(h)] \le \mathbb{E}_\rho[\hat{L}(h,S)] + \sqrt{\frac{2\mathbb{E}_\rho[\hat{L}(h,S)]\left(\mathrm{KL}(\rho\|\pi) + \ln\frac{2\sqrt{n}}{\delta}\right)}{n}} + \frac{2\left(\mathrm{KL}(\rho\|\pi) + \ln\frac{2\sqrt{n}}{\delta}\right)}{n}. \tag{4.21}$$

The relaxation shows that PAC-Bayes-kl inequality exhibits "fast convergence rate". Namely, if $\mathbb{E}_\rho[\hat{L}(h,S)]$ is close to zero, then the distance between $\mathbb{E}_\rho[\hat{L}(h,S)]$ and $\mathbb{E}_\rho[L(h)]$ decreases at the rate of $\frac{1}{n}$ rather than $\frac{1}{\sqrt{n}}$. The "fast convergence rate" provides an extra advantage for placing higher mass on prediction rules with empirical loss is especially close to zero.

We note that the "fast convergence rate" is a property of the kl divergence. It is possible to derive a similar "fast" version of the Occam's razor bound by basing it on the kl inequality, see Exercise 4.7.

Finally, we note that Equations (4.20) and (4.21) are deterministic relaxations of the bound in Theorem 4.26, and that the PAC-Bayes-kl inequality is always at least as tight as the best of its relaxations. And even though the kl has no analytic inverse, it is possible to invert it numerically to obtain the tightest bound.

4.8.1 Relation and Differences with other Learning Approaches

Even though it has "Bayesian" in its name, PAC-Bayesian analysis is a frequentist (PAC) learning framework. It has the following relation and differences with Bayesian learning and with VC analysis / Radamacher complexities.

Relation with Bayesian learning

1. Explicit way to incorporate prior information (via $\pi(h)$).

2. Possibility of sequential updates of the prior π.

Differences with Bayesian learning

1. Explicit high-probability guarantee on the expected performance.
2. No belief in prior correctness (a frequentist bound).
3. Explicit dependence on the loss function. (Bayesian posterior does not depend on the choice of the loss function, whereas PAC-Bayesian posterior does.)
4. Different weighting of prior belief $\pi(h)$ vs. evidence $\hat{L}(h, S)$.
5. Holds for *any* distribution ρ (including the Bayes posterior).

Relation with VC analysis / Radamacher complexities

1. Explicit high-probability guarantee on the expected performance.
2. Explicit dependence on the loss function.

Difference with VC analysis / Radamacher complexities

1. Complexity is defined individually for each h via $\pi(h)$ (rather than "complexity of a hypothesis class").
2. Explicit way to incorporate prior knowledge.
3. The bound is defined for randomized classifiers ρ (not individual h). For uncountably infinite sets of prediction rules it is usually necessary to spread the probability mass of ρ over a measurable subset of $\mathcal{H}$, because for continuous probability distributions concentrating ρ on a single h leads to explosion on $\mathrm{KL}(\rho\|\pi)$. It means that the PAC-Bayes classifiers are inherently randomized (rather than deterministic). However, it is possible to apply derandomization techniques, for example, ρ-weighted majority votes.

In a sense, PAC-Bayesian analysis takes the best out of Bayesian learning and VC analysis and puts it together. Specifically, it provides a possibility to incorporate prior information through π, and it provides frequentist generalization guarantees. Additionally, it provides efficient learning algorithms, since $\mathrm{KL}(\rho\|\pi)$ is convex in ρ and $\mathbb{E}_\rho[\hat{L}(h, S)]$ is linear in ρ, allowing efficient optimization of the bounds with respect to ρ.

4.8.2 A Proof of PAC-Bayes-kl Inequality

At the basis of most of PAC-Bayesian bounds lies the change of measure inequality, which acts as replacement of the union bound for uncountably infinite sets.

Theorem 4.27 (Change of measure inequality)**.** *For any measurable function $f(h)$ on $\mathcal{H}$ and any distributions ρ and π:*

$$\mathbb{E}_{h\sim\rho(h)}\left[f(h)\right] \leq \mathrm{KL}(\rho\|\pi) + \ln \mathbb{E}_{h\sim\pi(h)}\left[e^{f(h)}\right].$$

Proof.

$$\begin{aligned}
\mathbb{E}_{\rho(h)}\left[f(h)\right] &= \mathbb{E}_{\rho(h)}\left[\ln\left(\frac{\rho(h)}{\pi(h)} \times e^{f(h)} \times \frac{\pi(h)}{\rho(h)}\right)\right] \\
&= \mathrm{KL}(\rho\|\pi) + \mathbb{E}_{\rho(h)}\left[\ln\left(e^{f(h)} \times \frac{\pi(h)}{\rho(h)}\right)\right] \\
&\leq \mathrm{KL}(\rho\|\pi) + \ln \mathbb{E}_{\rho(h)}\left[e^{f(h)} \times \frac{\pi(h)}{\rho(h)}\right] \\
&= \mathrm{KL}(\rho\|\pi) + \ln \mathbb{E}_{\pi(h)}\left[e^{f(h)}\right],
\end{aligned}$$

where the inequality in the third step is justified by Jensen's inequality (Theorem B.30). Note that there is nothing probabilistic in the statement of the theorem - it is a deterministic result. □

In the next lemma we extend f to be a function of h and a sample S and apply a probabilistic argument to the last term of change-of-measure inequality. The lemma is the foundation for most PAC-Bayesian bounds.

Lemma 4.28 (PAC-Bayes lemma)**.** *For any measurable function $f : \mathcal{H} \times (\mathcal{X} \times \mathcal{Y})^n \to \mathbb{R}$ and any distribution π on $\mathcal{H}$ that is independent of the sample S*

$$\mathbb{P}\left(\exists \rho : \mathbb{E}_{h\sim\rho}\left[f(h,S)\right] \geq \mathrm{KL}(\rho\|\pi) + \ln \frac{\mathbb{E}_{h\sim\pi}\left[\mathbb{E}_S\left[e^{f(h,S)}\right]\right]}{\delta}\right) \leq \delta,$$

where the probability is with respect to the draw of the sample S and $\mathbb{E}_S$ is the expectation with respect to the draw of S.

An equivalent way of writing the above statement is

$$\mathbb{P}\left(\forall \rho : \mathbb{E}_{h\sim\rho}\left[f(h,S)\right] \leq \mathrm{KL}(\rho\|\pi) + \ln \frac{\mathbb{E}_{h\sim\pi}\left[\mathbb{E}_S\left[e^{f(h,S)}\right]\right]}{\delta}\right) \geq 1-\delta$$

or, in words, with probability at least $1-\delta$ over the draw of S, for all ρ simultaneously

$$\mathbb{E}_\rho\left[f(h,S)\right] \leq \mathrm{KL}(\rho\|\pi) + \ln \frac{\mathbb{E}_\pi\left[\mathbb{E}_S\left[e^{f(h,S)}\right]\right]}{\delta}.$$

We first present a slightly less formal, but more intuitive proof and then provide a formal one. By change of measure inequality we have

$$\begin{aligned}
\mathbb{E}_\rho\left[f(h,S)\right] &\leq \mathrm{KL}(\rho\|\pi) + \ln \mathbb{E}_\pi\left[e^{f(h,S)}\right] \\
&\underset{w.p.\geq 1-\delta}{\leq} \mathrm{KL}(\rho\|\pi) + \ln \frac{\mathbb{E}_S\left[\mathbb{E}_\pi\left[e^{f(h,S)}\right]\right]}{\delta} \\
&= \mathrm{KL}(\rho\|\pi) + \ln \frac{\mathbb{E}_\pi\left[\mathbb{E}_S\left[e^{f(h,S)}\right]\right]}{\delta},
\end{aligned}$$

where in the second line we apply Markov's inequality to the random variable $Z = \mathbb{E}_\pi\left[e^{f(h,S)}\right]$ (and the inequality holds with probability at least $1-\delta$) and in the last line we can exchange the order of expectations, because π is independent of S. The key observation is that the change-of-measure inequality relates all posterior distributions ρ to a single prior distribution π in a deterministic way and the probabilistic argument (the Markov's inequality) is applied to a single random quantity $\mathbb{E}_\pi\left[e^{f(h,S)}\right]$. This way change-of-measure inequality replaces the union bound, and it holds even when $\mathcal{H}$ is uncountably infinite.

Now we provide a formal proof.

Proof.

$$\begin{aligned}
\mathbb{P}\left(\exists \rho : \mathbb{E}_\rho\left[f(h,S)\right] \geq \mathrm{KL}(\rho\|\pi) + \ln \frac{\mathbb{E}_\pi\left[\mathbb{E}_S\left[e^{f(h,S)}\right]\right]}{\delta}\right) &\leq \mathbb{P}\left(\mathbb{E}_\pi\left[e^{f(h,S)}\right] \geq \frac{\mathbb{E}_\pi\left[\mathbb{E}_S\left[e^{f(h,S)}\right]\right]}{\delta}\right) \quad (4.22) \\
&= \mathbb{P}\left(\mathbb{E}_\pi\left[e^{f(h,S)}\right] \geq \frac{\mathbb{E}_S\left[\mathbb{E}_\pi\left[e^{f(h,S)}\right]\right]}{\delta}\right) \quad (4.23) \\
&\leq \delta,
\end{aligned}$$

where (4.22) follows by change-of-measure inequality (elaborated below), in (4.23) we can exchange the order of expectations, because π is independent of S, and in the last step we apply Markov's inequality to the random variable $Z = \mathbb{E}_\pi\left[e^{f(h,S)}\right]$.

An elaboration concerning Step (4.22). By change of measure inequality, we have that $\forall \rho : \mathbb{E}_\rho\left[f(h,S)\right] \leq \mathrm{KL}(\rho\|\pi) + \ln \mathbb{E}_\pi\left[e^{f(h,S)}\right]$. Therefore, *if* $\mathbb{E}_\pi\left[e^{f(h,S)}\right] \leq \frac{\mathbb{E}_S\left[\mathbb{E}_\pi\left[e^{f(h,S)}\right]\right]}{\delta}$, *then* $\forall \rho : \mathbb{E}_\rho\left[f(h,S)\right] \leq \mathrm{KL}(\rho\|\pi) + \ln \frac{\mathbb{E}_S\left[\mathbb{E}_\pi\left[e^{f(h,S)}\right]\right]}{\delta}$. Let A denote the event in the if-statement and B denote the event in the then-statement. Then $\mathbb{P}(A) \leq \mathbb{P}(B)$ and, therefore, $\mathbb{P}\big(\bar{A}\big) \geq \mathbb{P}\big(\bar{B}\big)$, where $\bar{A}$ denotes the complement of event

A. The complement of A is $\mathbb{E}_\pi\left[e^{f(h,S)}\right] > \frac{\mathbb{E}_S\left[\mathbb{E}_\pi\left[e^{f(h,S)}\right]\right]}{\delta}$ and the complement of B is $\exists \rho : \mathbb{E}_\rho\left[f(h,S)\right] > \mathrm{KL}(\rho\|\pi) + \ln\frac{\mathbb{E}_S\left[\mathbb{E}_\pi\left[e^{f(h,S)}\right]\right]}{\delta}$, which gives us the inequality in Step (4.22) (as usually, we are being a tiny bit sloppy and do not trace which inequalities are strict and which are weak; with a slight extra effort this could be done, but it does not matter in practice, so we save the effort). The important point is that the change-of-measure inequality relates all posterior distributions ρ to a single prior distribution π in a deterministic way, and the probabilistic argument is applied to a single random variable $\mathbb{E}_\pi\left[e^{f(h,S)}\right]$, avoiding the need in taking a union bound. This way the change of measure inequality acts as a replacement of the union bound. □

Different PAC-Bayesian inequalities are obtained by different choices of the function $f(h,S)$. A key consideration in the choice of $f(h,S)$ is the possibility to bound $\mathbb{E}_S\left[e^{f(h,S)}\right]$. For example, we have done it for $f(h,S) = n\,\mathrm{kl}(\hat{L}(h,S)\|L(h))$ in Theorem 3.22, and this is the choice of f in the proof of PAC-Bayes-kl inequality. Other choices of f are possible. For example, Hoeffding's Lemma 3.6 provides a bound on $\mathbb{E}_S\left[e^{f(h,S)}\right]$ for $f(h,S) = \lambda\left(L(h) - \hat{L}(h,S)\right)$, which can be used to derive PAC-Bayes-Hoeffding inequality. We refer to Seldin et al. (2012) for more details.

Proof of Theorem 4.26. We provide an intuitive derivation and leave the formal one (as in the proof of Lemma 4.28) as an exercise.

We take $f(h,S) = n\,\mathrm{kl}(\hat{L}(h,S)\|L(h))$. Then we have

$$\begin{aligned}
n\,\mathrm{kl}\left(\mathbb{E}_\rho\left[\hat{L}(h,S)\right]\middle\|\mathbb{E}_\rho\left[L(h)\right]\right) &\leq \mathbb{E}_\rho\left[n\,\mathrm{kl}(\hat{L}(h,S)\|L(h))\right] \\
&\underset{w.p.\geq 1-\delta}{\leq} \mathrm{KL}(\rho\|\pi) + \ln\frac{\mathbb{E}_\pi\left[\mathbb{E}_S\left[e^{n\,\mathrm{kl}(\hat{L}(h,S)\|L(h))}\right]\right]}{\delta} \\
&\leq \mathrm{KL}(\rho\|\pi) + \ln\frac{\mathbb{E}_\pi\left[2\sqrt{n}\right]}{\delta} \\
&= \mathrm{KL}(\rho\|\pi) + \ln\frac{2\sqrt{n}}{\delta},
\end{aligned}$$

where the first inequality is by convexity of kl (Theorem 3.19), the second inequality is by the PAC-Bayes Lemma (and it holds with probability at least $1-\delta$ over the draw of S), and the third inequality is by Lemma 3.22. □

4.8.3 Application to SVMs

In order to apply PAC-Bayesian bound to a given problem we have to design a prior distribution π and then bound the KL-divergence $\mathrm{KL}(\rho\|\pi)$ for the posterior distributions of interest. Sometimes we resort to a restricted class of ρ-s, for which we are able to bound $\mathrm{KL}(\rho\|\pi)$. You can see how this is done for SVMs in Langford (2005, Section 5.3).

4.8.4 Relaxation of PAC-Bayes-kl: PAC-Bayes-λ Inequality

Due to its implicit form, PAC-Bayes-kl inequality is not very convenient for optimization. One way around is to replace the bound with a linear trade-off $\beta n\mathbb{E}_\rho\left[\hat{L}(h,S)\right] + \mathrm{KL}(\rho\|\pi)$. Since $\mathrm{KL}(\rho\|\pi)$ is convex in ρ and $\mathbb{E}_\rho\left[\hat{L}(h,S)\right]$ is linear in ρ, for a fixed β the trade-off is convex in ρ and can be minimized. (We note that parametrization of ρ, for example the popular restriction of ρ to a Gaussian posterior (Langford, 2005), may easily break the convexity (Germain et al., 2009). We get back to this point in Section 4.8.6.) The value of β can then be tuned by cross-validation or substitution of $\rho(\beta)$ into the bound (the former usually works better).

Below we present a more rigorous approach. We prove the following relaxation of PAC-Bayes-kl inequality, which leads to a bound that can be optimized by alternating minimization.

Theorem 4.29 (PAC-Bayes-λ Inequality)**.** *For any probability distribution π over $\mathcal{H}$ that is independent of S and any $\delta \in (0,1)$, with probability greater than $1-\delta$ over a random draw of a sample S, for all*

distributions ρ over $\mathcal{H}$ and all $\lambda \in (0,2)$ and $\gamma > 0$ simultaneously:

$$\mathbb{E}_\rho\left[L(h)\right] \leq \frac{\mathbb{E}_\rho\left[\hat{L}(h,S)\right]}{1-\frac{\lambda}{2}} + \frac{\mathrm{KL}(\rho\|\pi) + \ln\frac{2\sqrt{n}}{\delta}}{\lambda\left(1-\frac{\lambda}{2}\right)n}, \tag{4.24}$$

$$\mathbb{E}_\rho\left[L(h)\right] \geq \left(1-\frac{\gamma}{2}\right)\mathbb{E}_\rho[\hat{L}(h,S)] - \frac{\mathrm{KL}(\rho\|\pi) + \ln\frac{2\sqrt{n}}{\delta}}{\gamma n}. \tag{4.25}$$

At the moment we focus on the upper bound in equation (4.24). Note that the theorem holds for *all* values of $\lambda \in (0,2)$ simultaneously. Therefore, we can optimize the bound with respect to λ and pick the best one.

Proof. We prove the upper bound in equation (4.24). Proof of the lower bound (4.25) is analogous and left as an exercise. Proof of the statement that the upper and lower bounds hold simultaneously (require no union bound) is also left as an exercise.

By refined Pinsker's inequality in Corollary 3.31, for $p < q$

$$\mathrm{kl}(p\|q) \geq (q-p)^2/(2q). \tag{4.26}$$

By PAC-Bayes-kl inequality, Theorem 4.26, with probability greater than $1-\delta$ for all ρ simultaneously

$$\mathrm{kl}\left(\mathbb{E}_\rho\left[\hat{L}(h,S)\right]\middle\|\mathbb{E}_\rho\left[L(h)\right]\right) \leq \frac{\mathrm{KL}(\rho\|\pi) + \ln\frac{2\sqrt{n}}{\delta}}{n}.$$

By application of inequality (4.26), the above inequality can be relaxed to

$$\mathbb{E}_\rho\left[L(h)\right] - \mathbb{E}_\rho\left[\hat{L}(h,S)\right] \leq \sqrt{2\mathbb{E}_\rho\left[L(h)\right]\frac{\mathrm{KL}(\rho\|\pi) + \ln\frac{2\sqrt{n}}{\delta}}{n}}. \tag{4.27}$$

We have that

$$\min_{\lambda:\lambda>0}\left(\lambda x + \frac{y}{\lambda}\right) = 2\sqrt{xy}$$

(we leave this statement as a simple exercise). Thus, $\sqrt{xy} \leq \frac{1}{2}\left(\lambda x + \frac{y}{\lambda}\right)$ for all $\lambda > 0$ and by applying this inequality to (4.27) we have that with probability at least $1-\delta$ for all ρ and $\lambda > 0$

$$\mathbb{E}_\rho\left[L(h)\right] - \mathbb{E}_\rho\left[\hat{L}(h,S)\right] \leq \frac{\lambda}{2}\mathbb{E}_\rho\left[L(h)\right] + \frac{\mathrm{KL}(\rho\|\pi) + \ln\frac{2\sqrt{n}}{\delta}}{\lambda n}.$$

By changing sides

$$\left(1-\frac{\lambda}{2}\right)\mathbb{E}_\rho\left[L(h)\right] \leq \mathbb{E}_\rho\left[\hat{L}(h,S)\right] + \frac{\mathrm{KL}(\rho\|\pi) + \ln\frac{2\sqrt{n}}{\delta}}{\lambda n}.$$

For $\lambda < 2$ we can divide both sides by $\left(1-\frac{\lambda}{2}\right)$ and obtain the theorem statement. □

4.8.5 Alternating Minimization of the PAC-Bayes-λ Bound

We use the term *PAC-Bayes-λ bound* to refer to the right hand side of PAC-Bayes-λ inequality. A great advantage of the PAC-Bayes-λ bound is that it can be conveniently minimized by alternating minimization with respect to ρ and λ. Since $\mathbb{E}_\rho\left[\hat{L}(h,S)\right]$ is linear in ρ and $\mathrm{KL}(\rho\|\pi)$ is convex in ρ (Cover and Thomas, 2006), for a fixed λ the bound is convex in ρ and the minimum is achieved by

$$\rho(h) = \frac{\pi(h)e^{-\lambda n\hat{L}(h,S)}}{\mathbb{E}_\pi\left[e^{-\lambda n\hat{L}(h',S)}\right]}, \tag{4.28}$$

where $\mathbb{E}_\pi\left[e^{-\lambda n\hat{L}(h',S)}\right]$ is a convenient way of writing the normalization factor, which covers continuous and discrete hypothesis spaces in a unified notation. In the discrete case, which will be of main interest

for us, $\mathbb{E}_\pi\left[e^{-\lambda n\hat{L}(h',S)}\right] = \sum_{h'\in\mathcal{H}}\pi(h')e^{-\lambda n\hat{L}(h',S)}$. We leave a proof of the statement that (4.28) defines ρ which achieves the minimum of the bound as an exercise to the reader. Furthermore, for $t \in (0,1)$ and $a, b \geq 0$ the function $\frac{a}{1-t} + \frac{b}{t(1-t)}$ is convex in t (Tolstikhin and Seldin, 2013) and, therefore, for a fixed ρ the right hand side of inequality (4.24) is convex in λ for $\lambda \in (0,2)$ and the minimum is achieved by

$$\lambda = \frac{2}{\sqrt{\frac{2n\mathbb{E}_\rho\left[\hat{L}(h,S)\right]}{\left(\mathrm{KL}(\rho\|\pi)+\ln\frac{2\sqrt{n}}{\delta}\right)}+1}+1}. \tag{4.29}$$

Note that the optimal value of λ is smaller than 1. Alternating application of update rules (4.28) and (4.29) monotonically decreases the bound, and thus converges.

We note that while the right hand side of inequality (4.24) is convex in ρ for a fixed λ and convex in λ for a fixed ρ, it is not simultaneously convex in ρ and λ. Joint convexity would have been a sufficient, but it is not a necessary condition for convergence of alternating minimization to the global minimum of the bound. Thiemann et al. (2017) provide sufficient conditions under which the procedure converges to the global minimum, as well as examples of situations where this does not happen.

4.8.6 Construction of a Hypothesis Space for PAC-Bayes-λ

If $\mathcal{H}$ is infinite, computation of the partition function (the denominator in (4.28)) is intractable. This could be resolved by parametrization of ρ (for example, restriction of ρ to a Gaussian posterior), but, as we have already mentioned, this may break the convexity of the bound in ρ. Fortunately, things get easy when $\mathcal{H}$ is finite. The crucial step is to construct a sufficiently powerful finite hypothesis space $\mathcal{H}$. One possibility that we consider here is to construct $\mathcal{H}$ by training m hypotheses, where each hypothesis is trained on r random points from S and validated on the remaining $n-r$ points. This construction resembles a cross-validation split of the data. However, in cross-validation r is typically large (close to n) and validation sets are non-overlapping. The approach considered here works for any r and has additional computational advantages when r is small. We do not require validation sets to be non-overlapping and overlaps between training sets are allowed. Below we describe the construction more formally.

Let $h \in \{1,\ldots,m\}$ index the hypotheses in $\mathcal{H}$. Let S_h denote the training set of h and $S \setminus S_h$ the validation set. S_h is a subset of r points from S, which are selected independently of their values (for example, subsampled randomly or picked according to a predefined partition of the data). We define the validation error of h by $\hat{L}^{\mathrm{val}}(h,S) = \frac{1}{n-r}\sum_{(X,Y)\in S\setminus S_h}\ell(h(X),Y)$. Note that the validation errors are $(n-r)$ i.i.d. random variables with bias $L(h)$ and, therefore, for $f(h,S) = (n-r)\,\mathrm{kl}(\hat{L}^{\mathrm{val}}(h,S)\|L(h))$ we have $\mathbb{E}_S\left[e^{f(h,S)}\right] \leq 2\sqrt{n-r}$. The following result is a straightforward adaptation of Theorem 4.29 to this setting (we leave the proof as an exercise to the reader).

Theorem 4.30 (PAC-Bayesian Aggregation)**.** *Let S be a sample of size n. Let $\mathcal{H}$ be a set of m hypotheses, where each $h \in \mathcal{H}$ is trained on r points from S selected independently of the composition of S. For any probability distribution π over $\mathcal{H}$ that is independent of S and any $\delta \in (0,1)$, with probability greater than $1-\delta$ over a random draw of a sample S, for all distributions ρ over $\mathcal{H}$ and $\lambda \in (0,2)$ simultaneously:*

$$\mathbb{E}_\rho\left[L(h)\right] \leq \frac{\mathbb{E}_\rho\left[\hat{L}^{\mathrm{val}}(h,S)\right]}{1-\frac{\lambda}{2}} + \frac{\mathrm{KL}(\rho\|\pi)+\ln\frac{2\sqrt{n-r}}{\delta}}{\lambda\left(1-\frac{\lambda}{2}\right)(n-r)}. \tag{4.30}$$

It is natural, but not mandatory to select a uniform prior $\pi(h) = 1/m$. The bound in equation (4.30) can be minimized by alternating application of the update rules in equations (4.28) and (4.29) with n being replaced by $n-r$ and $\hat{L}$ by $\hat{L}^{\mathrm{val}}$. For evaluation of the empirical performance of this learning approach see Thiemann et al. (2017).

4.9 PAC-Bayesian Analysis of Ensemble Classifiers

So far in this chapter we have discussed various methods of selection of classifiers from a hypothesis set $\mathcal{H}$. We now turn to *aggregation* of predictions by multiple classifiers through a *weighted majority*

vote. The power of the majority vote is in the "cancellation of errors" effect: *if* predictions of different classifiers are uncorrelated and they all predict better than a random guess (meaning that $L(h) < 1/2$), the errors tend to cancel out. This can be compared to a consultation of medical experts, which tends to predict better than the best expert in the set. Most machine learning competitions are won by strategies that aggregate predictions of multiple classifiers. The assumptions that the errors are uncorrelated and the predictions are better than random are important. For example, if we have three hypotheses with $L(h) = p$ and independent errors, the probability that a uniform majority vote MV_u makes an error equals the probability that at least two out of the three hypotheses make an error. You are welcome to verify that in this case for $p \leq 1/2$ we have $L(\mathrm{MV}_u) \leq \mathbb{E}_u\left[L(h)\right]$, where u is the uniform distribution. If the errors are correlated, it can be shown that $L(\mathrm{MV}_\rho)$ can be larger than $\mathbb{E}_\rho\left[L(h)\right]$, but as we show below it is never larger than $2\mathbb{E}_\rho\left[L(h)\right]$. The reader is welcome to construct an example, where $L(\mathrm{MV}_u) > \mathbb{E}_u\left[L(h)\right]$.

4.9.1 Ensemble Classifiers and Weighted Majority Vote

We now turn to some formal definitions. Ensemble classifiers predict by taking a weighted aggregation of predictions by hypotheses from $\mathcal{H}$. In multi-class prediction (the label space $\mathcal{Y}$ is finite) ρ-weighted majority vote MV_ρ predicts

$$\mathrm{MV}_\rho(X) = \arg\max_{Y \in \mathcal{Y}} \sum_{(h \in \mathcal{H}) \wedge (h(X) = Y)} \rho(h),$$

where $\wedge$ represents the logical "and" operation and ties can be resolved arbitrarily.

In binary prediction with prediction space $h(X) \in \{\pm 1\}$ weighted majority vote can be written as

$$\mathrm{MV}_\rho(X) = \mathrm{sign}\left(\mathbb{E}_\rho\left[h(X)\right]\right),$$

where $\mathrm{sign}(x) = 1$ if $x > 0$ and -1 otherwise (the value of $\mathrm{sign}(0)$ can be defined arbitrarily). For a countable hypothesis space this becomes

$$\mathrm{MV}_\rho(X) = \mathrm{sign}\left(\sum_{h \in \mathcal{H}} \rho(h) h(X)\right). \tag{4.31}$$

4.9.2 First Order Oracle Bound for the Weighted Majority Vote

If majority vote makes an error, we know that at least a ρ-weighted half of the classifiers have made an error and, therefore, $\ell(\mathrm{MV}_\rho(X), Y) \leq \mathbb{1}(\mathbb{E}_\rho[\mathbb{1}(h(X) \neq Y)] \geq 0.5)$. This observation leads to the well-known first order oracle bound for the loss of weighted majority vote.

Theorem 4.31 (First Order Oracle Bound)**.**

$$L(\mathrm{MV}_\rho) \leq 2\mathbb{E}_\rho[L(h)].$$

Proof. We have $L(\mathrm{MV}_\rho) = \mathbb{E}_D[\ell(\mathrm{MV}_\rho(X), Y)] \leq \mathbb{P}(\mathbb{E}_\rho[\mathbb{1}(h(X) \neq Y)] \geq 0.5)$. By applying Markov's inequality to random variable $Z = \mathbb{E}_\rho[\mathbb{1}(h(X) \neq Y)]$ we have:

$$L(\mathrm{MV}_\rho) \leq \mathbb{P}(\mathbb{E}_\rho[\mathbb{1}(h(X) \neq Y)] \geq 0.5) \leq 2\mathbb{E}_D[\mathbb{E}_\rho[\mathbb{1}(h(X) \neq Y)]] = 2\mathbb{E}_\rho[L(h)].$$

□

PAC-Bayesian analysis can be used to bound $\mathbb{E}_\rho[L(h)]$ in Theorem 4.31 in terms of $\mathbb{E}_\rho[\hat{L}(h, S)]$, thus turning the oracle bound into an empirical one. The disadvantage of the first order approach is that $\mathbb{E}_\rho[L(h)]$ ignores correlations of predictions, which is the main power of the majority vote.

4.9.3 Second Order Oracle Bound for the Weighted Majority Vote

Now we present a second order bound for the weighted majority vote, which is based on a second order Markov's inequality: for a non-negative random variable Z and $\varepsilon > 0$, we have $\mathbb{P}(Z \geq \varepsilon) = \mathbb{P}\big(Z^2 \geq \varepsilon^2\big) \leq \varepsilon^{-2}\mathbb{E}\left[Z^2\right]$. We define *tandem loss* of two hypotheses h and h' by

$$\ell(h(X), h'(X), Y) = \mathbb{1}(h(X) \neq Y \wedge h'(X) \neq Y).$$

The tandem loss counts an error on a sample (X, Y) only if both h and h' err on (X, Y). We define the expected tandem loss by

$$L(h, h') = \mathbb{E}_D[\mathbb{1}(h(X) \neq Y \wedge h'(X) \neq Y)].$$

The following lemma relates the expectation of the second moment of the standard loss to the expected tandem loss. We use the shorthand $\mathbb{E}_{\rho^2}[L(h, h')] = \mathbb{E}_{h\sim\rho, h'\sim\rho}[L(h, h')]$.

Lemma 4.32. *In multiclass classification*

$$\mathbb{E}_D[\mathbb{E}_\rho[\mathbb{1}(h(X) \neq Y)]^2] = \mathbb{E}_{\rho^2}[L(h, h')].$$

Proof.

$$\begin{aligned}
\mathbb{E}_D[\mathbb{E}_\rho[\mathbb{1}(h(X) \neq Y)]^2] &= \mathbb{E}_D[\mathbb{E}_\rho[\mathbb{1}(h(X) \neq Y)]\mathbb{E}_\rho[\mathbb{1}(h(X) \neq Y)]] \\
&= \mathbb{E}_D[\mathbb{E}_{\rho^2}[\mathbb{1}(h(X) \neq Y)\mathbb{1}(h'(X) \neq Y)]] \\
&= \mathbb{E}_D[\mathbb{E}_{\rho^2}[\mathbb{1}(h(X) \neq Y \wedge h'(X) \neq Y)]] \\
&= \mathbb{E}_{\rho^2}[\mathbb{E}_D[\mathbb{1}(h(X) \neq Y \wedge h'(X) \neq Y)]] \\
&= \mathbb{E}_{\rho^2}[L(h, h')].
\end{aligned} \tag{4.32}$$

□

A combination of second order Markov's inequality with Lemma 4.32 leads to the following result.

Theorem 4.33 (Second Order Oracle Bound)**.** *In multiclass classification*

$$L(\mathrm{MV}_\rho) \leq 4\mathbb{E}_{\rho^2}[L(h, h')]. \tag{4.33}$$

Proof. By second order Markov's inequality applied to $Z = \mathbb{E}_\rho[\mathbb{1}(h(X) \neq Y)]$ and Lemma 4.32:

$$L(\mathrm{MV}_\rho) \leq \mathbb{P}(\mathbb{E}_\rho[\mathbb{1}(h(X) \neq Y)] \geq 0.5) \leq 4\mathbb{E}_D[\mathbb{E}_\rho[\mathbb{1}(h(X) \neq Y)]^2] = 4\mathbb{E}_{\rho^2}[L(h, h')].$$

□

A Specialized Bound for Binary Classification

We provide an alternative form of Theorem 4.33, which can be used to exploit unlabeled data in binary classification. We denote the *expected disagreement* between hypotheses h and h' by $\mathbb{D}(h, h') = \mathbb{E}_D[\mathbb{1}(h(X) \neq h'(X))]$ and express the tandem loss in terms of standard loss and disagreement.

Lemma 4.34. *In binary classification*

$$\mathbb{E}_{\rho^2}[L(h, h')] = \mathbb{E}_\rho[L(h)] - \frac{1}{2}\mathbb{E}_{\rho^2}[\mathbb{D}(h, h')].$$

Proof of Lemma 4.34. Picking from (4.32), we have

$$\begin{aligned}
\mathbb{E}_\rho[\mathbb{1}(h(X) \neq Y)]\mathbb{E}_\rho[\mathbb{1}(h(X) \neq Y)] &= \mathbb{E}_\rho[\mathbb{1}(h(X) \neq Y)](1 - \mathbb{E}_\rho[(1 - \mathbb{1}(h(X) \neq Y))] \\
&= \mathbb{E}_\rho[\mathbb{1}(h(X) \neq Y)] - \mathbb{E}_\rho[\mathbb{1}(h(X) \neq Y)]\mathbb{E}_\rho[\mathbb{1}(h(X) = Y)] \\
&= \mathbb{E}_\rho[\mathbb{1}(h(X) \neq Y)] - \mathbb{E}_{\rho^2}[\mathbb{1}(h(X) \neq Y \wedge h'(X) = Y)] \\
&= \mathbb{E}_\rho[\mathbb{1}(h(X) \neq Y)] - \frac{1}{2}\mathbb{E}_{\rho^2}[\mathbb{1}(h(X) \neq h'(X))].
\end{aligned}$$

By taking expectation with respect to D on both sides and applying Lemma 4.32 to the left hand side, we obtain:

$$\mathbb{E}_{\rho^2}[L(h, h')] = \mathbb{E}_D[\mathbb{E}_\rho[\mathbb{1}(h(X) \neq Y)] - \frac{1}{2}\mathbb{E}_{\rho^2}[\mathbb{1}(h(X) \neq h'(X))]] = \mathbb{E}_\rho[L(h)] - \frac{1}{2}\mathbb{E}_{\rho^2}[\mathbb{D}(h, h')].$$

□

The lemma leads to the following result.

Theorem 4.35 (Second Order Oracle Bound for Binary Classification)**.** *In binary classification*

$$L(\mathrm{MV}_\rho) \leq 4\mathbb{E}_\rho[L(h)] - 2\mathbb{E}_{\rho^2}[\mathbb{D}(h,h')]. \tag{4.34}$$

Proof. The theorem follows by plugging the result of Lemma 4.34 into Theorem 4.33. □

The advantage of the alternative way of writing the bound is the possibility of using unlabeled data for estimation of $\mathbb{D}(h,h')$ in binary prediction (see also Germain et al., 2015). We note, however, that estimation of $\mathbb{E}_{\rho^2}[\mathbb{D}(h,h')]$ has a slow convergence rate, as opposed to $\mathbb{E}_{\rho^2}[L(h,h')]$, which has a fast convergence rate. We discuss this point in Section 4.9.7.

4.9.4 Comparison of the First and Second Order Oracle Bounds

From Theorems 4.31 and 4.35 we see that in binary classification the second order bound is tighter when $\mathbb{E}_{\rho^2}[\mathbb{D}(h,h')] > \mathbb{E}_\rho[L(h)]$. Below we provide a more detailed comparison of Theorems 4.31 and 4.33 in the worst, the best, and the independent cases. The comparison only concerns the oracle bounds, whereas estimation of the oracle quantities, $\mathbb{E}_\rho[L(h)]$ and $\mathbb{E}_{\rho^2}[L(h,h')]$, is discussed in Section 4.9.7.

The worst case Since $\mathbb{E}_{\rho^2}[L(h,h')] \leq \mathbb{E}_\rho[L(h)]$ the second order bound is at most twice worse than the first order bound. The worst case happens, for example, if all hypotheses in $\mathcal{H}$ give identical predictions. Then $\mathbb{E}_{\rho^2}[L(h,h')] = \mathbb{E}_\rho[L(h)] = L(\mathrm{MV}_\rho)$ for all ρ.

The best case Imagine that $\mathcal{H}$ consists of $M \geq 3$ hypotheses, such that each hypothesis errs on $1/M$ of the sample space (according to the distribution D) and that the error regions are disjoint. Then $L(h) = 1/M$ for all h and $L(h,h') = 0$ for all $h \neq h'$ and $L(h,h) = 1/M$. For a uniform distribution ρ on $\mathcal{H}$ the first order bound is $2\mathbb{E}_\rho[L(h)] = 2/M$ and the second order bound is $4\mathbb{E}_{\rho^2}[L(h,h')] = 4/M^2$ and $L(\mathrm{MV}_\rho) = 0$. In this case the second order bound is an order of magnitude tighter than the first order.

The independent case Assume that all hypotheses in $\mathcal{H}$ make independent errors and have the same error rate, $L(h) = L(h')$ for all h and h'. Then for $h \neq h'$ we have $L(h,h') = \mathbb{E}_D[\mathbb{1}(h(X) \neq Y \wedge h'(X) \neq Y)] = \mathbb{E}_D[\mathbb{1}(h(X) \neq Y)\mathbb{1}(h'(X) \neq Y)] = \mathbb{E}_D[\mathbb{1}(h(X) \neq Y)]\mathbb{E}_D[\mathbb{1}(h'(X) \neq Y)] = L(h)^2$ and $L(h,h) = L(h)$. For a uniform distribution ρ the second order bound is $4\mathbb{E}_{\rho^2}[L(h,h')] = 4(L(h)^2 + \frac{1}{M}L(h)(1-L(h)))$ and the first order bound is $2\mathbb{E}_\rho[L(h)] = 2L(h)$. Assuming that M is large, so that we can ignore the second term in the second order bound, we obtain that it is tighter for $L(h) < 1/2$ and looser otherwise. The former is the interesting regime, especially in binary classification.

4.9.5 Second Order PAC-Bayesian Bounds for the Weighted Majority Vote

Now we provide an empirical bound for the weighted majority vote. We define the *empirical tandem loss*

$$\hat{L}(h,h',S) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(h(X_i) \neq Y_i \wedge h'(X_i) \neq Y_i)$$

and provide a bound on the expected loss of ρ-weighted majority vote in terms of the empirical tandem losses.

Theorem 4.36. *For any probability distribution π on $\mathcal{H}$ that is independent of S and any $\delta \in (0,1)$, with probability at least $1-\delta$ over a random draw of S, for all distributions ρ on $\mathcal{H}$ and all $\lambda \in (0,2)$ simultaneously:*

$$L(\mathrm{MV}_\rho) \leq 4\left(\frac{\mathbb{E}_{\rho^2}[\hat{L}(h,h',S)]}{1-\lambda/2} + \frac{2\,\mathrm{KL}(\rho\|\pi) + \ln(2\sqrt{n}/\delta)}{\lambda(1-\lambda/2)n}\right).$$

Proof. The theorem follows by using the bound in equation (4.24) to bound $\mathbb{E}_{\rho^2}[L(h,h')]$ in Theorem 4.33. We note that $\mathrm{KL}(\rho^2\|\pi^2) = 2\,\mathrm{KL}(\rho\|\pi)$ (Germain et al., 2015, Page 814). □

It is also possible to use PAC-Bayes-kl to bound $\mathbb{E}_{\rho^2}[L(h,h')]$ in Theorem 4.33, which actually gives a tighter bound, but the bound in Theorem 4.36 is more convenient for minimization. We refer the reader to Masegosa et al. (2020) for a procedure for bound minimization.

A specialized bound for binary classification

We define the *empirical disagreement*

$$\hat{\mathbb{D}}(h, h', S') = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}(h(X_i) \neq h'(X_i)),$$

where $S' = \{X_1, \ldots, X_m\}$. The set S' may overlap with the labeled set S, however, S' may include additional unlabeled data. The following theorem bounds the loss of weighted majority vote in terms of empirical disagreements. Due to possibility of using unlabeled data for estimation of disagreements in the binary case, the theorem has the potential of yielding a tighter bound when a considerable amount of unlabeled data is available.

Theorem 4.37. *In binary classification, for any probability distribution π on $\mathcal{H}$ that is independent of S and S' and any $\delta \in (0, 1)$, with probability at least $1 - \delta$ over a random draw of S and S', for all distributions ρ on $\mathcal{H}$ and all $\lambda \in (0, 2)$ and $\gamma > 0$ simultaneously:*

$$L(\mathrm{MV}_\rho) \leq 4 \left(\frac{\mathbb{E}_\rho[\hat{L}(h, S)]}{1 - \lambda/2} + \frac{\mathrm{KL}(\rho\|\pi) + \ln(4\sqrt{n}/\delta)}{\lambda(1 - \lambda/2)n} \right) - 2 \left((1 - \gamma/2)\mathbb{E}_{\rho^2}[\hat{\mathbb{D}}(h, h', S')] - \frac{2\,\mathrm{KL}(\rho\|\pi) + \ln(4\sqrt{m}/\delta)}{\gamma m} \right).$$

Proof. The theorem follows by using the upper bound in equation (4.24) to bound $\mathbb{E}_\rho[L(h)]$ and the lower bound in equation (4.25) to bound $\mathbb{E}_{\rho^2}[\mathbb{D}(h, h')]$ in Theorem 4.35. We replace δ by $\delta/2$ in the upper and lower bound and take a union bound over them. □

Using PAC-Bayes-kl to bound $\mathbb{E}_\rho[L(h)]$ and $\mathbb{E}_{\rho^2}[\mathbb{D}(h, h')]$ in Theorem 4.35 gives a tighter bound, but the bound in Theorem 4.37 is more convenient for minimisation. We refer to Masegosa et al. (2020) for a procedure for bound minimization.

4.9.6 Ensemble Construction

It is possible to use the same procedure as in Section 4.8.6 to construct an ensemble. Tandem losses can then be estimated on overlaps of validation sets, $(S \setminus S_h) \cap (S \setminus S_{h'})$. The sample size in Theorem 4.36 should then be replaced by $\min_{h,h'} |(S \setminus S_h) \cap (S \setminus S_{h'})|$.

4.9.7 Comparison of the Empirical Bounds

We provide a high-level comparison of the empirical first order bound (FO), the empirical second order bound based on the tandem loss (TND, Theorem 4.36), and the new empirical second order bound based on disagreements (DIS, Theorem 4.37). The two key quantities in the comparison are the sample size n in the denominator of the bounds and fast and slow convergence rates for the standard (first order) loss, the tandem loss, and the disagreements. Tolstikhin and Seldin (2013) have shown that if we optimize λ for a given ρ, the PAC-Bayes-λ bound in equation (4.24) can be written as

$$\mathbb{E}_\rho[L(h)] \leq \mathbb{E}_\rho[\hat{L}(h, S)] + \sqrt{\frac{2\mathbb{E}_\rho[\hat{L}(h, S)]\,(\mathrm{KL}(\rho\|\pi) + \ln(2\sqrt{n}/\delta))}{n}} + \frac{2\,(\mathrm{KL}(\rho\|\pi) + \ln(2\sqrt{n}/\delta))}{n}.$$

This form of the bound, also used by McAllester (2003), is convenient for explanation of fast and slow rates. If $\mathbb{E}_\rho[\hat{L}(h, S)]$ is large, then the middle term on the right hand side dominates the complexity and the bound decreases at the rate of $1/\sqrt{n}$, which is known as a *slow rate*. If $\mathbb{E}_\rho[\hat{L}(h, S)]$ is small, then the last term dominates and the bound decreases at the rate of $1/n$, which is known as a *fast rate*.

FO **vs.** TND The advantage of the FO bound is that the validation sets $S \setminus S_h$ available for estimation of the first order losses $\hat{L}(h, S_h)$ are larger than the validation sets $(S \setminus S_h) \cap (S \setminus S_{h'})$ available for estimation of the tandem losses. Therefore, the denominator $n_{\texttt{min}} = \min_h |S \setminus S_h|$ in the FO bound is larger than the denominator $n_{\texttt{min}} = \min_{h,h'} |(S \setminus S_h) \cap (S \setminus S_{h'})|$ in the TND bound. The TND disadvantage can

be reduced by using data splits with large validation sets $S \setminus S_h$ and small training sets S_h, as long as small training sets do not overly impact the quality of base classifiers h. Another advantage of the FO bound is that its complexity term has $\mathrm{KL}(\rho\|\pi)$, whereas the TND bound has $2\,\mathrm{KL}(\rho\|\pi)$. The advantage of the TND bound is that $\mathbb{E}_{\rho^2}[L(h,h')] \leq E_\rho[L(h)]$ and, therefore, the convergence rate of the tandem loss is typically faster than the convergence rate of the first order loss. The interplay of the estimation advantages and disadvantages, combined with the advantages and disadvantages of the underlying oracle bounds discussed in Section 4.9.4, depends on the data and the hypothesis space.

TND **vs.** DIS The advantage of the DIS bound relative to the TND bound is that in presence of a large amount of unlabeled data the disagreements $\mathbb{D}(h,h')$ can be tightly estimated (the denominator m is large) and the estimation complexity is governed by the first order term, $\mathbb{E}_\rho[L(h)]$, which is "easy" to estimate, as discussed above. However, the DIS bound has two disadvantages. A minor one is its reliance on estimation of two quantities, $\mathbb{E}_\rho[L(h)]$ and $\mathbb{E}_{\rho^2}[\mathbb{D}(h,h')]$, which requires a union bound, e.g., replacement of δ by $\delta/2$. A more substantial one is that the disagreement term is desired to be large, and thus has a slow convergence rate. Since slow convergence rate relates to fast convergence rate as $1/\sqrt{n}$ to $1/n$, as a rule of thumb the DIS bound is expected to outperform TND only when the amount of unlabeled data is at least quadratic in the amount of labeled data, $m > n^2$.

For experimental comparison of the bounds and further details we refer the reader to Masegosa et al. (2020). A follow-up work by Wu et al. (2021) improves the analysis by introducing a second order oracle bound for the weighted majority vote based on a parametric form of the Chebyshev-Cantelli inequality.

4.10 PAC-Bayes-split-kl Inequality

The PAC-Bayes-kl inequality in Theorem 4.26 is a good choice for binary losses, because the kl Lemma (Theorem 3.22) is tight for Bernoulli random variables. But if the loss function happens to take more than two values, then PAC-Bayes-kl might not necessarily be the best choice. In this section we focus on losses taking a finite set of values. Examples of such losses include the *excess loss*, which is a difference of (potentially weighted) losses of two prediction rules, $f(h,h',X,Y) = \ell(h(X),Y) - \gamma\ell(h'(X),Y)$, where γ is a weighting parameter. When $\ell(h(X),Y)$ is the zero-one loss, the excess loss $f(h,h',X,Y) \in \{-\gamma, 0, 1-\gamma, 1\}$. We will work with such losses in Recursive PAC-Bayes in Section 4.11. Another example is a tandem loss with an offset, $\ell_\alpha(h(X),h'(X),Y) = (\ell(h(X),Y)-\alpha)(\ell(h'(X),Y)-\alpha) \in \left\{-\alpha(1-\alpha), \alpha^2, (1-\alpha)^2\right\}$ introduced by Wu et al. (2021) in their refined analysis of the weighted majority vote. And a third example is prediction with abstention, where a learner is allowed to abstain at a fixed cost $\gamma < 0.5$, and otherwise pay the zero-one loss on the prediction.

We use the same approach as we used in split-kl inequality in Section 3.7. Namely, we represent discrete random variables as a superposition of Bernoulli random variables, and then apply PAC-Bayes-kl to the decomposition.

Let $f : \mathcal{H} \times \mathcal{Z} \to \{b_0, \ldots, b_K\}$ be a $(K+1)$-valued loss function. For example, in Section 4.11 we will work with excess losses $f(h,h',X,Y) = \ell(h(X),Y) - \gamma\ell(h'(X),Y)$, and there we define $\mathcal{Z} = \mathcal{H} \times \mathcal{X} \times \mathcal{Y}$, so that $h \in \mathcal{H}$ is the first argument of f and $Z = (h',X,Y) \in \mathcal{Z}$ is the second argument of f. For $j \in \{1,\ldots,K\}$ let $f_{|j}(\cdot,\cdot) = \mathbb{1}(f(\cdot,\cdot) \geq b_j)$. Let $\mathcal{D}_Z$ be an unknown distribution on $\mathcal{Z}$. For $h \in \mathcal{H}$ let $F(h) = \mathbb{E}_{\mathcal{D}_Z}[f(h,Z)]$ and $F_{|j}(h) = \mathbb{E}_{\mathcal{D}_Z}[f_{|j}(h,Z)]$. Let $S = \{Z_1,\ldots,Z_n\}$ be an i.i.d. sample according to $\mathcal{D}_Z$ and $\hat{F}_{|j}(h,S) = \frac{1}{n}\sum_{i=1}^n f_{|j}(h,Z_i)$.

Theorem 4.38 (PAC-Bayes-Split-kl Inequality (Wu et al., 2024))**.** *For any distribution π on $\mathcal{H}$ that is independent of S and any $\delta \in (0,1)$:*

$$\mathbb{P}\left(\exists \rho \in \mathcal{P} : \mathbb{E}_\rho[F(h)] \geq b_0 + \sum_{j=1}^K \alpha_j \,\mathrm{kl}^{-1,+}\left(\mathbb{E}_\rho[\hat{F}_{|j}(h,S)], \frac{\mathrm{KL}(\rho\|\pi) + \ln\frac{2K\sqrt{n}}{\delta}}{n}\right)\right) \leq \delta,$$

where $\mathcal{P}$ is the set of all possible probability distributions on $\mathcal{H}$ that can depend on S.

Proof. We have $f(\cdot,\cdot) = b_0 + \sum_{j=1}^K \alpha_j f_{|j}(\cdot,\cdot)$ and $F(h) = b_0 + \sum_{j=1}^K \alpha_j F_{|j}(h)$. Therefore,

$$\mathbb{P}\left(\exists \rho \in \mathcal{P} : \mathbb{E}_\rho[F(h)] \geq b_0 + \sum_{j=1}^K \alpha_j \operatorname{kl}^{-1,+}\left(\mathbb{E}_\rho[\hat{F}_{|j}(h,S)], \frac{\mathrm{KL}(\rho\|\pi) + \ln\frac{2K\sqrt{n}}{\delta}}{n}\right)\right)$$
$$\leq \mathbb{P}\left(\exists \rho \in \mathcal{P} \text{ and } \exists j : \mathbb{E}_\rho[F_{|j}(h)] \geq \operatorname{kl}^{-1,+}\left(\mathbb{E}_\rho[\hat{F}_{|j}(h,S)], \frac{\mathrm{KL}(\rho\|\pi) + \ln\frac{2K\sqrt{n}}{\delta}}{n}\right)\right) \leq \delta,$$

where the first inequality is by the decomposition of F and the second inequality is by the union bound and application of Theorem 4.26 to $F_{|j}$ (note that $f_{|j}$ is a zero-one loss function). □

4.11 Recursive PAC-Bayes

In this section we consider sequential processing of the data. Imagine that we have a data set S that is split into T non-overlapping subsets, $S = S_1 \cup \cdots \cup S_T$, and we process them one after the other. Sequential processing may be unavoidable if the data arrive sequentially, but as we will see, it may also be beneficial when all the data are available in advance.

In PAC-Bayesian analysis we start with a prior distribution over prediction rules and after observing the data we update it to a posterior distribution. The posterior can then be turned into a prior for processing more data. However, a challenge is that the denominator in PAC-Bayes bounds, e.g., in Theorem 4.26, is the amount of data used for construction of the posterior, whereas information on how much data were used for constructing the prior, which reflects confidence in the prior, is lost. This leads to a leaky processing pipeline, because confidence information on the prior is lost every time a new posterior replaces the old prior.

To fix the leak, Wu et al. (2024) have proposed *Recursive PAC-Bayes*, which provides a way to preserve confidence information on the prior from one processing round to the next. In order to present the idea, we let $\pi_0^*, \pi_1^*, \ldots, \pi_T^*$ be a sequence of distributions over $\mathcal{H}$. The first of them, π_0^*, is an initial prior distribution that does not depend on S. For $t \in \{1,\ldots,T\}$, π_t^* is a posterior distribution over $\mathcal{H}$ that is obtained by starting from a prior distribution π_{t-1}^* and processing data subset S_t. After S_t has been processed, π_t^* becomes a prior for processing the next chunk, S_{t+1}. The final goal is to obtain a good generalization bound on the expected loss of the last posterior, $\mathbb{E}_{\pi_T^*}[L(h)]$.

Wu et al. (2024) came up with the following recursive decomposition of the loss, which lays the foundation for Recursive PAC-Bayes:

$$\mathbb{E}_{\pi_t}[L(h)] = \mathbb{E}_{\pi_t}[L(h) - \gamma_t \mathbb{E}_{\pi_{t-1}}[L(h')]] + \gamma_t \mathbb{E}_{\pi_{t-1}}[L(h')], \tag{4.35}$$

where $\gamma_t \in [0,1]$.

For example, for $T = 3$ the decomposition becomes

$$\mathbb{E}_{\pi_3}[L(h)] = \mathbb{E}_{\pi_3}[L(h) - \gamma_3 \mathbb{E}_{\pi_2}[L(h')]] + \gamma_3 \mathbb{E}_{\pi_2}[L(h) - \gamma_2 \mathbb{E}_{\pi_1}[L(h')]] + \gamma_3\gamma_2 \mathbb{E}_{\pi_1}[L(h')],$$

see Figure 4.9 for a graphical illustration and discussion.

We make a few observations based on (4.35).

- $\mathbb{E}_{\pi_{t-1}}[L(h')]$ is "of the same kind" as $\mathbb{E}_{\pi_t}[L(h)]$, which is why we can apply the decomposition recursively.
- The decomposition in (4.35) applies to any loss function, including unbounded losses, assuming all the expectations are well-defined.
- $\mathbb{E}_{\pi_t}[L(h) - \gamma_t \mathbb{E}_{\pi_{t-1}}[L(h')]] = \mathbb{E}_{\pi_t}[\mathbb{E}_{\pi_{t-1},X,Y}[\ell(h(X),Y) - \gamma_t \ell(h'(X),Y)]]$ is an expected *excess loss*. If $\ell(h(X),Y)$ and $\mathbb{E}_{\pi_{t-1}}[\ell(h'(X),Y)]$ are correlated, then the excess loss has lower variance and, therefore, potentially tighter bound than the plain loss $\mathbb{E}_{\pi_t}[L(h)]$. Excess loss has to be bounded using PAC-Bayes bounds that are capable of exploiting small variance, for example, PAC-Bayes-split-kl. Basic PAC-Bayes bounds, such as PAC-Bayes-kl, are "blind" to the variance and destroy the advantage of working with excess losses.

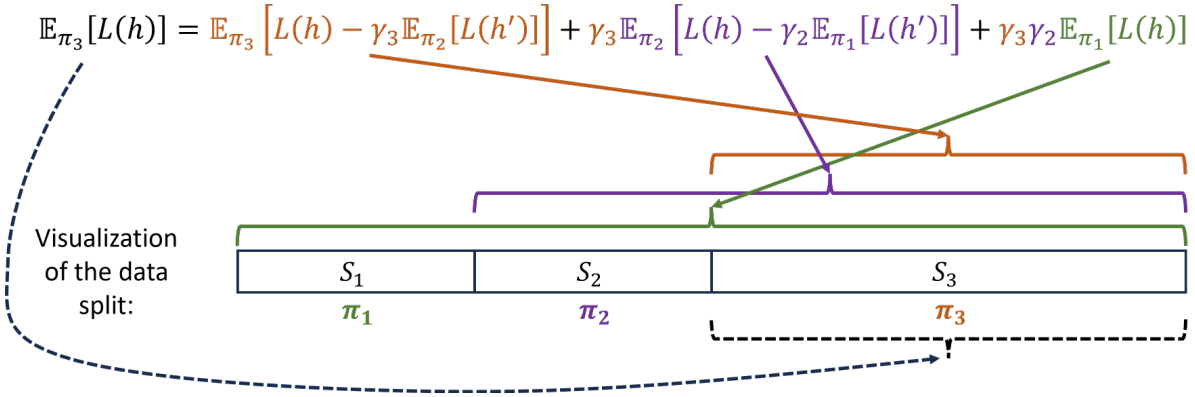


Figure 4.9: **Recursive Decomposition into Three Terms.** The figure illustrates recursive decomposition of $\mathbb{E}_{\pi_3}[L(h)]$ into three terms based on equation (4.35), and a geometric data split. The bottom line illustrates which data are used for construction of which distribution: S_1 for π_1; S_2 for π_2; and S_3 for π_3. The brackets above the data show which data are used for computing PAC-Bayes bounds for which term: $S_1 \cup S_2 \cup S_3$ for $\mathbb{E}_{\pi_1}[L(h)]$; $S_2 \cup S_3$ for $\mathbb{E}_{\pi_2}[L(h) - \gamma_2\mathbb{E}_{\pi_1}[L(h')]]$; and S_3 for $\mathbb{E}_{\pi_3}[L(h) - \gamma_3\mathbb{E}_{\pi_2}[L(h')]]$. Note that a direct computation of a PAC-Bayes bound on $\mathbb{E}_{\pi_3}[L(h)]$ would only use the data in S_3, as shown by the black dashed line. The figure illustrates that recursive decomposition provides more efficient use of the data. We also note that initially we start with poor priors, and so the $\mathrm{KL}(\pi_t \| \pi_{t-1})$ term for small t is expected to be large, but this is compensated by a small multiplicative factor $\prod_{i=t+1}^{T} \gamma_i$ and availability of a lot of data $\bigcup_{i=t}^{T} S_i$ for computing the PAC-Bayes bound. For example, $\mathbb{E}_{\pi_1}[L(h)]$ is multiplied by $\gamma_3\gamma_2$ and we can use all the data for computing a PAC-Bayes bound on this term. By the time we reach higher t, the priors π_{t-1} get better, and the $\mathrm{KL}(\pi_t \| \pi_{t-1})$ term in the bounds gets much smaller, and additionally the bounds benefit from the small variance of the excess loss. With geometric split of the data, we use little data to quickly move π_t to a good region, and then we still have enough data for a good estimation of the later terms, like $\mathbb{E}_{\pi_3}[L(h) - \gamma_3\mathbb{E}_{\pi_2}[L(h')]]$. (The figure is borrowed from Wu et al. (2024).)

- In order to bound the excess loss we are going to look at $f(h, (h', X, Y)) = \ell(h(X), Y) - \gamma_t \ell(h'(X), Y)$ and $F(h) = \mathbb{E}_{h', X, Y}[f(h, (h', X, Y)]$. We will construct samples of triplets (h', X, Y), where (X, Y) come from the sample S and h' is sampled according to π_{t-1}, and then proceed with PAC-Bayesian bounding of $\mathbb{E}_{\pi_t}[F(h)]$ using PAC-Bayes-split-kl bound.

- Recursive PAC-Bayes exploits the following important property of PAC-Bayes. Even though we only use S_t in the construction of π_t^*, we can use $S_t, S_{t+1}, \ldots, S_T$ for computing empirical estimates of $F(h)$ and bounding $\mathbb{E}_{\pi_t^*}[F(h)]$. This is because the prior π_{t-1}^* is independent of $S_t, S_{t+1}, \ldots, S_T$ and the posterior is allowed, but not required to depend on all the data. And note that even though we will use $S_{t+1}, \ldots, S_T$ for estimation of $\mathbb{E}_{\pi_t^*}[F(h)]$, π_t^* will stay independent of $S_{t+1}, \ldots, S_T$, as required from a prior for π_{t+1}^*. Thus, S_t is used for construction of π_t^* and estimation of $\mathbb{E}_{\pi_s^*}[F(h)]$ for all $s \leq t$, yielding an efficient use of the data.

We now present a generic Recursive Bound. The bound can be seen as a generic shell that needs to be filled in with concrete bounds $\mathrm{B}_1(\pi_1)$ on $\mathbb{E}_{\pi_1}[L(h)]$ and $\mathcal{E}_t(\pi_t, \gamma_t)$ on the excess losses $\mathbb{E}_{\pi_t}[L(h) - \gamma_t \mathbb{E}_{\pi_{t-1}^*}[L(h')]]$ for $t \geq 2$. The filling of the shell will happen one theorem later.

Theorem 4.39 (Recursive Bound (Wu et al., 2024)). *Let $S = S_1 \cup \cdots \cup S_T$ be an i.i.d. sample of size n split in an arbitrary way into T non-overlapping subsamples, and let $U_t^{\mathrm{train}} = \bigcup_{s=1}^{t} S_s$ and $U_t^{\mathrm{val}} = \bigcup_{s=t}^{T} S_s$. Let $\pi_0^*, \pi_1^*, \ldots, \pi_T^*$ be a sequence of distributions on $\mathcal{H}$, where π_t^* is allowed to depend on U_t^{train}, but not the rest of the data. Let $\gamma_2, \ldots, \gamma_T$ be a sequence of coefficients, where γ_t is allowed to depend on U_{t-1}^{train}, but not the rest of the data. For $t \in \{1, \ldots, T\}$ let $\mathcal{P}_t$ be a set of distributions on $\mathcal{H}$, which are allowed to depend on U_t^{train}. Let $\delta \in (0, 1)$. Assume there exists a function $\mathrm{B}_1(\pi_1)$ that satisfies*

$$\mathbb{P}(\exists \pi_1 \in \mathcal{P}_1 : \mathbb{E}_{\pi_1}[L(h)] \geq \mathrm{B}_1(\pi_1)) \leq \frac{\delta}{T}. \tag{4.36}$$

For $t \geq 2$ assume there exist functions $\mathcal{E}_t(\pi_t, \gamma_t)$ that satisfy

$$\mathbb{P}\Big(\exists \pi_t \in \mathcal{P}_t : \mathbb{E}_{\pi_t}[L(h) - \gamma_t \mathbb{E}_{\pi^*_{t-1}}[L(h')]] \geq \mathcal{E}_t(\pi_t, \gamma_t)\Big) \leq \frac{\delta}{T}. \tag{4.37}$$

Let

$$\mathrm{B}_t(\pi_t) = \mathcal{E}_t(\pi_t, \gamma_t) + \gamma_t \, \mathrm{B}_{t-1}(\pi^*_{t-1}). \tag{4.38}$$

Then

$$\mathbb{P}(\exists t \in \{1, \ldots, T\} \;\; \textit{and} \; \pi_t \in \mathcal{P}_t : \mathbb{E}_{\pi_t}[L(h)] \geq \mathrm{B}_t(\pi_t)) \leq \delta. \tag{4.39}$$

Proof. The proof follows directly by induction, because the definition of $\mathrm{B}_t(\pi_t)$ in equation (4.38) matches the recursive decomposition of the loss in equation (4.35), and so

$$\begin{aligned}
&\mathbb{P}(\exists t \in \{1, \ldots, T\} \text{ and } \pi_t \in \mathcal{P}_t : \mathbb{E}_{\pi_t}[L(h)] \geq \mathrm{B}_t(\pi_t)) \\
&\quad \leq \mathbb{P}\Big(\exists \pi_1 \in \mathcal{P}_1 : \mathbb{E}_{\pi_1}[L(h)] \geq \mathrm{B}_1(\pi_1) \quad \text{OR} \quad \exists \pi_t \in \mathcal{P}_t \text{ for } t \geq 2 : \mathbb{E}_{\pi_t}[L(h) - \gamma_t \mathbb{E}_{\pi^*_{t-1}}[L(h')]] \geq \mathcal{E}_t(\pi_t, \gamma_t)\Big) \\
&\quad \leq \delta,
\end{aligned}$$

where the last step is by the union bound. □

Theorem 4.39 only states that if we could control $\mathbb{E}_{\pi_1}[L(h)]$ and $\mathbb{E}_{\pi_t}[L(h) - \gamma_t \mathbb{E}_{\pi^*_{t-1}}[L(h')]]$ for all $t \geq 2$, then we would have $\mathbb{E}_{\pi_t}[L(h)]$ under control for all t. We can use any basic PAC-Bayes bound, for example, Theorem 4.26, to control the plain loss $\mathbb{E}_{\pi_1}[L(h)]$ and any bound for non-binary losses, for example, Theorem 4.38, to control the excess losses $\mathbb{E}_{\pi_t}[L(h) - \gamma_t \mathbb{E}_{\pi^*_{t-1}}[L(h')]]$.

Next we present one concrete way of defining the bounds using PAC-Bayes-split-kl inequality. In order to apply the inequality we define

$$F_{\gamma_t, \pi^*_{t-1}}(h) = L(h) - \gamma_t \mathbb{E}_{\pi^*_{t-1}}[L(h')] = \mathbb{E}_{\pi^*_{t-1} \times \mathcal{D}}[\ell(h(X), Y) - \gamma_t \ell(h'(X), Y)],$$

where $\pi^*_{t-1} \times \mathcal{D}$ is a product distribution on $\mathcal{H} \times \mathcal{X} \times \mathcal{Y}$ and $h' \in \mathcal{H}$ is sampled according to π^*_{t-1}. Then $\mathbb{E}_{\pi_t}[L(h) - \gamma_t \mathbb{E}_{\pi^*_{t-1}}[L(h')]] = \mathbb{E}_{\pi_t}[F_{\gamma_t, \pi^*_{t-1}}(h)]$. We further define

$$f_{\gamma_t}(h, (h', X, Y)) = \ell(h(X), Y) - \gamma_t \ell(h'(X), Y) \in \{-\gamma_t, 0, 1 - \gamma_t, 1\},$$

then $F_{\gamma_t, \pi^*_{t-1}}(h) = \mathbb{E}_{\pi^*_{t-1} \times \mathcal{D}}[f_{\gamma_t}(h, (h', X, Y))]$. In order to apply Theorem 4.38, we represent f_{γ_t} as a superposition of binary functions. For this purpose we let $\{b_{t|0}, b_{t|1}, b_{t|2}, b_{t|3}\} = \{-\gamma_t, 0, 1 - \gamma_t, 1\}$ and define $f_{\gamma_t|j}(h, (h', X, Y)) = \mathbb{1}\big(f_{\gamma_t}(h, (h', X, Y)) \geq b_{t|j}\big)$. We let $F_{\gamma_t, \pi^*_{t-1}|j}(h) = \mathbb{E}_{\pi^*_{t-1} \times \mathcal{D}}[f_{\gamma_t|j}(h, (h', X, Y))]$, then $F_{\gamma_t, \pi^*_{t-1}}(h) = -\gamma_t + \sum_{j=1}^{3} (b_{t|j} - b_{t|j-1}) F_{\gamma_t, \pi^*_{t-1}|j}(h)$.

Now we construct an empirical estimate of $F_{\gamma_t, \pi^*_{t-1}|j}(h)$. We already have a sample of (X, Y) pairs, which we need to complement with a sample of h' to obtain (h', X, Y) triplets. We first let $\hat{\pi}^*_{t-1} = \left\{h_1^{\pi^*_{t-1}}, h_2^{\pi^*_{t-1}}, \ldots\right\}$ be a sequence of prediction rules sampled independently according to π^*_{t-1}. We define $\hat{\pi}^*_{t-1} \circ U_t^{\mathrm{val}} = \left\{\left(h_i^{\pi^*_{t-1}}, X_i, Y_i\right) : (X_i, Y_i) \in U_t^{\mathrm{val}}\right\}$. In words, for every sample $(X_i, Y_i) \in U_t^{\mathrm{val}}$ we sample a prediction rule $h_i^{\pi^*_{t-1}}$ according to π^*_{t-1} and place the triplet $(h_i^{\pi^*_{t-1}}, X_i, Y_i)$ in $\hat{\pi}^*_{t-1} \circ U_t^{\mathrm{val}}$. The triplets $(h_i^{\pi^*_{t-1}}, X_i, Y_i)$ correspond to the random variables Z in Theorem 4.38. We note that $|U_t^{\mathrm{val}}| = |\hat{\pi}^*_{t-1} \circ U_t^{\mathrm{val}}|$, and we let $n_t^{\mathrm{val}} = |U_t^{\mathrm{val}}|$. We define an empirical estimate of $F_{\gamma_t, \pi^*_{t-1}|j}(h)$ as $\hat{F}_{\gamma_t|j}(h, \hat{\pi}^*_{t-1} \circ U_t^{\mathrm{val}}) = \frac{1}{n_t^{\mathrm{val}}} \sum_{(h', X, Y) \in \hat{\pi}^*_{t-1} \circ U_t^{\mathrm{val}}} f_{\gamma_t|j}(h, (h', X, Y))$. Note that $\mathbb{E}_{\pi^*_{t-1} \times \mathcal{D}}[\hat{F}_{\gamma_t|j}(h, \hat{\pi}^*_{t-1} \circ U_t^{\mathrm{val}})] = F_{\gamma_t, \pi^*_{t-1}|j}(h)$, therefore, we can use Theorem 4.38 to bound $\mathbb{E}_{\pi_t}[F_{\gamma_t, \pi^*_{t-1}}(h)]$ using its empirical estimates. We are now ready to state the bound.

Theorem 4.40 (Recursive PAC-Bayes-split-kl Inequality (Wu et al., 2024))**.** *Let*

$$\mathrm{B}_1(\pi_1) = \mathrm{kl}^{-1,+}\left(\mathbb{E}_{\pi_1}[\hat{L}(h, S)], \frac{\mathrm{KL}(\pi_1 \| \pi^*_0) + \ln \frac{2T\sqrt{n}}{\delta}}{n}\right)$$

and

$$\mathcal{E}_t(\pi_t, \gamma_t) = -\gamma_t + \sum_{j=1}^{3} (b_{t|j} - b_{t|j-1}) \operatorname{kl}^{-1,+}\left(\mathbb{E}_{\pi_t}\left[\hat{F}_{\gamma_t|j}(h, \hat{\pi}^*_{t-1} \circ U^{\mathrm{val}}_t)\right], \frac{\mathrm{KL}(\pi_t \| \pi^*_{t-1}) + \ln \frac{6T\sqrt{n^{\mathrm{val}}_t}}{\delta}}{n^{\mathrm{val}}_t} \right).$$

Then B_1 *satisfies* (4.36) *and* $\mathcal{E}_t$ *satisfies* (4.37)*, and the statement* (4.39) *of Theorem 4.39 holds.*

Proof. Statement (4.36) follows by Theorem 4.26 and statement (4.37) follows by Theorem 4.38. □

Next we dicuss practical aspects of how to construct π^*_t, how to select γ_t, and how to split the sample.

Construction of $\pi^*_1, \ldots, \pi^*_T$

An interesting point about Recursive PAC-Bayes is that π^*_t is constructed using S_t, but evaluated using $U^{\mathrm{val}}_t = S_t \bigcup \left(\bigcup_{s=t+1}^{T} S_s\right) = S_t \bigcup U^{\mathrm{val}}_{t+1}$. For the construction of π^*_t we take the triplets (h', X, Y) in $\hat{\pi}^*_{t-1} \circ S_t$. We could then consider a "construction bound"

$$\mathcal{E}^{\mathtt{const}}_t(\pi, \gamma_t, n_t) = -\gamma_t + \sum_{j=1}^{3} (b_{t|j} - b_{t|j-1}) \operatorname{kl}^{-1,+}\left(\mathbb{E}_{\pi}\left[\hat{F}_{\gamma_t|j}(h, \hat{\pi}^*_{t-1} \circ S_t)\right], \frac{\mathrm{KL}(\pi \| \pi^*_{t-1}) + \ln \frac{6T\sqrt{n_t}}{\delta}}{n_t} \right),$$

which would be a high-probability bound on $\mathbb{E}_{\pi}[L(h) - \gamma_t \mathbb{E}_{\pi^*_{t-1}}[L(h')]]$ based on S_t alone. It involves $\mathbb{E}_{\pi}\left[\hat{F}_{\gamma_t|j}(h, \hat{\pi}^*_{t-1} \circ S_t)\right]$ and $n_t = |S_t|$. We could then take $\pi^*_t = \arg\min_{\pi} \mathcal{E}^{\mathtt{const}}_t(\pi, \gamma_t, n_t)$, which would minimize this bound. However, we know that when we reach the evaluation stage, $\mathcal{E}_t(\pi_t, \gamma_t)$ will look at more data and involve $\mathbb{E}_{\pi_t}\left[\hat{F}_{\gamma_t|j}(h, \hat{\pi}^*_{t-1} \circ U^{\mathrm{val}}_t)\right]$ and n^{val}_t. So the denominator of the bound $n^{\mathrm{val}}_t \geq n_t$ will be larger. But it is also likely that $\mathbb{E}_{\pi_t}\left[\hat{F}_{\gamma_t|j}(h, \hat{\pi}^*_{t-1} \circ U^{\mathrm{val}}_t)\right]$ will be larger than $\mathbb{E}_{\pi}\left[\hat{F}_{\gamma_t|j}(h, \hat{\pi}^*_{t-1} \circ S_t)\right]$, because $\mathbb{E}_{\pi_t}\left[\hat{F}_{\gamma_t|j}(h, \hat{\pi}^*_{t-1} \circ U^{\mathrm{val}}_t)\right]$ is a weighted average of $\mathbb{E}_{\pi}\left[\hat{F}_{\gamma_t|j}(h, \hat{\pi}^*_{t-1} \circ S_t)\right]$ and $\mathbb{E}_{\pi}\left[\hat{F}_{\gamma_t|j}(h, \hat{\pi}^*_{t-1} \circ U^{\mathrm{val}}_{t+1})\right]$. The latter is an unbiased estimate of $\mathbb{E}_{\pi^*_t}[L(h) - \gamma_t \mathbb{E}_{\pi^*_{t-1}}[L(h')]]$, but the former is an underestimate of this quantity, because π^*_t is tailored to S_t. Therefore, we could exploit the knowledge that we are going to have more data at the evaluation phase and redefine the "construction bound" as

$$\mathcal{E}^{\mathtt{const}}_t(\pi, \gamma_t, n^{\mathrm{val}}_t) = -\gamma_t + \sum_{j=1}^{3} (b_{t|j} - b_{t|j-1}) \operatorname{kl}^{-1,+}\left(\mathbb{E}_{\pi}\left[\hat{F}_{\gamma_t|j}(h, \hat{\pi}^*_{t-1} \circ S_t)\right], \frac{\mathrm{KL}(\pi \| \pi^*_{t-1}) + \ln \frac{6T\sqrt{n^{\mathrm{val}}_t}}{\delta}}{n^{\mathrm{val}}_t} \right)$$

(i.e., increase the denominator from n_t to n^{val}_t while still considering only S_t in the loss estimates) and take $\pi^*_t = \arg\min_{\pi} \mathcal{E}^{\mathtt{const}}_t(\pi, \gamma_t, n^{\mathrm{val}}_t)$. This would allow more aggressive overfitting of S making $\mathbb{E}_{\pi}\left[\hat{F}_{\gamma_t|j}(h, \hat{\pi}^*_{t-1} \circ S_t)\right]$ smaller, but the effect on $\mathbb{E}_{\pi_t}\left[\hat{F}_{\gamma_t|j}(h, \hat{\pi}^*_{t-1} \circ U^{\mathrm{val}}_t)\right]$ depends on the data. We emphasize that $\mathcal{E}_t(\pi_t, \gamma_t)$ is a valid bound for any choice of π^*_t and both choices of the "construction bound" are admissible, but which of them would yield a tighter bound depends on the data. Since the bounds are conservative, empirically replacing n_t with n^{val}_t in the "construction bound" usually performs better.

Selection of $\gamma_2, \ldots, \gamma_T$

We naturally want to have an improvement in the bound as we proceed from one chunk of the data to the next, meaning that we want $\mathrm{B}_t(\pi^*_t) < \mathrm{B}_{t-1}(\pi^*_{t-1})$. Substituting this inequality into (4.38) yields $\mathrm{B}_{t-1}(\pi^*_{t-1}) > \mathrm{B}_t(\pi^*_t) = \mathcal{E}_t(\pi^*_t, \gamma_t) + \gamma_t \mathrm{B}_{t-1}(\pi^*_{t-1})$, which leads to $\gamma_t < 1 - \frac{\mathcal{E}_t(\pi^*_t, \gamma_t)}{\mathrm{B}_{t-1}(\pi^*_{t-1})}$. Therefore, γ_t should be strictly smaller than 1, and it should also be non-negative. Note that $\gamma_t = 0$ recovers $\mathbb{E}_{\pi_t}[L(h)] = \mathbb{E}_{\pi_t}[L(h)]$. Since γ_t is only used once we reach S_t it can be constructed sequentially based on U^{train}_{t-1}. It cannot depend on S_t, because it would bias the estimates of $F_{\gamma_t, \pi^*_{t-1}}(h)$, but we can select γ_t from a grid of values if we take a union bound over the selection. We note that Wu et al. (2024) have simply taken $\gamma_t = \frac{1}{2}$.

How to split the sample

When data arrive sequentially the data split is determined by the arrival process. But when all data are available offline in advance, there is a question of how to split the data S into subsets $S_1, \ldots, S_T$. One could split the data uniformly, so that $|S_t| = \frac{1}{T}|S|$, but it leaves relatively little data for estimation of the final term, $\mathcal{E}_T(\pi_T^*, \gamma_t)$. In order to address this issue, Wu et al. (2024) have proposed to work with a geometric split of the data, where $|S_T| = \frac{1}{2}|S|$, $|S_{T-1}| = \frac{1}{4}|S|$, ... This approach has two advantages. First, when we start, the prior is typically poor, and so we use little data to bring the prior to a reasonably good region. The large $\mathrm{KL}(\pi_t \| \pi_{t-1})$ term in the first steps is compensated by a small multiplicative factor $\prod_{s=t}^{T} \gamma_s$ and by the large size of U_t^{val}, which provides a large denominator. By the time we reach later processing steps, the prior is going to be good, so that the $\mathrm{KL}(\pi_t \| \pi_{t-1})$ will be small, and we will still have a lot of data to compute a good bound $\mathcal{E}(\pi_T^*, \gamma_T)$. See Figure 4.9 for an illustration.

For empirical evaluation of Recursive PAC-Bayes see the work of Wu et al. (2024).

4.12 Exercises

Exercise 4.1 (*Experiment design*)**.**

1. You are working at a hospital and you have collected an i.i.d. sample of 2000 patients and annotated it for presence or absence of some disease (binary annotation). You organize a competition to find a classifier for the disease. You have 20 teams that have signed up for the competition and your boss requires you to provide a confidence interval of 0.05 on the prediction accuracy of the best classifier that will hold with probability at least 95%. In other words, with probability at least 95% the estimate of the expected error should not underestimate the true expected zero-one error of the selected classifier by more than 0.05 (one-sided error). How many samples do you have to keep aside in order to satisfy this requirement, assuming that you accept 1 solution from each team? Provide a complete calculation, numerical answers without any derivations or explanations will not be accepted.

2. You have conducted the competition above, but were not satisfied with the prediction accuracy of the winner. You decided to make another competition and were very lucky to convince your boss to support annotation of another 1000 patients. You decided to release the old 2000 patients data for training and keep the new 1000 samples for evaluating the outcome of the new competition. Your boss requires from you the same confidence interval of 0.05 with probability at least 95%. How many teams can you accept to take part in the competition assuming that you accept only 1 solution from each team? Provide a complete calculation, numerical answers without any derivations or explanations will not be accepted.

Exercise 4.2 (*Combining datasets*)**.** You are approached by a big and a small company, and each proposes you a classifier for a problem of interest, $h_{\texttt{big}}$ and $h_{\texttt{small}}$. They also provide you the data they have used for training their classifiers. The big company has collected a dataset $S_{\texttt{big}}$ of 10000 samples and the small company has collected a dataset $S_{\texttt{small}}$ of only 1000 samples. We assume that all the data are i.i.d. and both samples come from the same distribution. The companies provide no details on how they produced the classifiers. You have no own data to test the solutions, so instead you test $h_{\texttt{big}}$ on $S_{\texttt{small}}$ and $h_{\texttt{small}}$ on $S_{\texttt{big}}$. You obtain $\hat{L}(h_{\texttt{big}}, S_{\texttt{small}}) = 0.03$ and $\hat{L}(h_{\texttt{small}}, S_{\texttt{big}}) = 0.06$.

You need to pick a classifier and provide a generalization bound that will hold with probability at least 95%. Explain which of the two classifiers you will pick and provide a generalization bound for it.

Exercise 4.3 (*How to split data into training and test sets*)**.** In this question you will analyze one possible approach to the question of how to split a dataset S into training and test sets, $S^{\texttt{train}}$ and $S^{\texttt{test}}$. As we have already discussed, overly small test sets lead to unreliable loss estimates, whereas overly large test sets leave too little data for training, thus producing poor prediction models. The optimal trade-off depends on the data and the prediction model. So can we let the data speak for itself? We will give it a try.

1. We want to find a good balance between the sizes of $S^{\texttt{train}}$ and $S^{\texttt{test}}$. We consider m possible splits $\{(S_1^{\texttt{train}}, S_1^{\texttt{test}}), \ldots, (S_m^{\texttt{train}}, S_m^{\texttt{test}})\}$, where the sizes of the test sets are $n_1, \ldots, n_m$, correspondingly.

For example, it could be $(10\%, 90\%), (20\%, 80\%), \ldots, (90\%, 10\%)$ splits or anything else with a reasonable coverage of the possible options. We train m prediction models $\hat{h}_1^*, \ldots, \hat{h}_m^*$, where $\hat{h}_i^*$ is trained on $S_i^{\texttt{train}}$. We calculate the test loss of the i-th model on the i-th test set $\hat{L}(\hat{h}_i^*, S_i^{\texttt{test}})$. Derive a bound on $L(\hat{h}_i^*)$ in terms of $\hat{L}(\hat{h}_i^*, S_i^{\texttt{test}})$ and n_i that holds for all $\hat{h}_i^*$ simultaneously with probability at least $1 - \delta$.

Comment: No theorem from the book applies directly to this setting, because they all have a fixed sample size n, whereas here the sample sizes $n_1, \ldots, n_m$ vary. You have to provide a complete derivation.

2. We expect that most readers will treat all the splits in the previous point equally. Note, however, that models trained on more data are a-priori expected to perform better. Propose a way to give them an advantage by using a non-uniform treatment [a "prior"] that will give preference to classifiers trained on more samples and repeat the analysis. You have to propose one explicit prior and do the analysis with that prior.

Exercise 4.4 (*Efficient use of data*)**.** Most of the theoretical results in the book use part of the data for training prediction rules and another part for validating them. This way some data are only used for training and some data are only used for validation. But pay attention that if we would have trained a prediction rule on the validation set and validated it on the training set, we would have also gotten an unbiased estimate of the loss. (Remember: "it's not about how you call it, it's about how you use it"!). So could we use the data more efficiently?

The approach of using part of the data for training and part for validation, and then reverting the roles, somewhat resembles cross-validation. But a warning would be in place here. Even though the standard cross-validation technique is widely used, it is a heuristic, and if it is used to validate too many prediction rules, it is prone to overfitting, in exactly the same way as the standard validation technique is prone to overfitting, unless generalization bounds are used to control the overfitting.

What you will do next is inspired by cross-validation, but it is different from the standard cross-validation approach.

So, we have a data set S of size n (assume that n is even). We split the data set into two equal halves, $S = S_0 \cup S_1$. We train M models $\{h_{0,1}, \ldots, h_{0,M}\}$ on the first half of the data and validate them on the remaining half. Let $\hat{L}(h_{0,i}, S_1)$ for $i \in \{1, \ldots, M\}$ be the corresponding validation losses. Then we train another M models $\{h_{1,1}, \ldots, h_{1,M}\}$ on the second half of the data and validate them on the first half. Let $\hat{L}(h_{1,i}, S_0)$ for $i \in \{1, \ldots, M\}$ be the corresponding validation losses. Finally, we select the model $h_{j^*,i^*} = \arg \min_{j \in \{0,1\}, i \in \{1,\ldots,M\}} \hat{L}(h_{j,i}, S_{1-j})$ with the smallest validation loss.

Derive a high-probability generalization bound for the expected loss of h_{j^*,i^*}. (I.e., a bound on $L(h_{j^*,i^*})$ that holds with probability at least $1 - \delta$.)

Comment: no theorem from the book directly applies to the question, because they all assume that $\hat{L}(h, S)$ is computed on the same S for all h. You have to make a custom derivation, but it will not be very different from derivations you can find in the book.

Exercise 4.5 (*Learning by discretization*)**.** We want to learn an arbitrary binary function on a unit square by discretizing the square into a uniform grid with d^2 cells. The hypothesis space is the space of all possible uniform grids with d^2 cells for $d \in \{1, 2, 3, \ldots\}$, where each cell gets a binary label.

We have a sample S of size n to learn the function. Let $\mathcal{H}_d$ be the hypothesis set of uniform grids with d^2 cells. Let $\mathcal{H} = \bigcup_{d=1}^{\infty} \mathcal{H}_d$ be the hypothesis set of all possible uniform grids. Let $f(h)$ denote the number of cells in the hypothesis h. Let $d(h) = \sqrt{f(h)}$, then $d(h) \in \{1, 2, 3, \ldots\}$ and $h \in \mathcal{H}_{d(h)}$.

1. Derive a generalization bound for learning with $\mathcal{H}_d$. (I.e., a bound on $L(h)$ that holds for all $h \in \mathcal{H}_d$ with probability at least $1 - \delta$: $\mathbb{P}(\forall h \in \mathcal{H}_d : L(h) \leq \ldots) \geq 1 - \delta$, your task is to fill in the dots.)

2. Derive a generalization bound for learning with $\mathcal{H}$. (I.e., a bound on $L(h)$ that holds for all $h \in \mathcal{H}$ with probability at least $1 - \delta$.)

3. Write down a selection rule for selecting a prediction rule $h \in \mathcal{H}$ that is optimal according to the bound in the previous point. Ideally, you answer should be in a form $h^* = \ldots$, where $\ldots$ is a mathematical expression using the bound.

4. What is the maximal number of cells as a function of n, for which your bound is non-vacuous? (It is sufficient to derive an order of magnitude, you do not need to make a precise calculation.)

5. Explain how the density of the grid $d(h)$ affects the bound. Which terms in the bound (if any) increase as the density of the grid increases and which terms in the bound (if any) decrease as the density of the grid increases?

Exercise 4.6 (*Early stopping*)**.** Early stopping is a widely used technique to avoid overfitting in models trained by iterative methods, such as gradient descent. In particular, it is used to avoid overfitting in training neural networks. In this question we analyze several ways of implementing early stopping. The technique sets aside a validation set $S^{\texttt{val}}$, which is used to monitor the improvement of the training process. Let $h_1, h_2, h_3, \ldots$ be a sequence of models obtained after $1, 2, 3, \ldots$ epochs of training a neural network or any other prediction model (you do not need to know any details about neural networks or their training procedure to answer the question). Let $\hat{L}(h_1, S^{\texttt{val}})$, $\hat{L}(h_2, S^{\texttt{val}})$, $\hat{L}(h_3, S^{\texttt{val}}), \ldots$ be the corresponding sequence of validation errors on the validation set $S^{\texttt{val}}$.

1. Let h_{t^*} be the neural network returned after training with early stopping. In which of the following cases $\hat{L}(h_{t^*}, S^{\texttt{val}})$ is an unbiased estimate of $L(h_{t^*})$ and in which cases it is not? Please, explain your answer.

   (a) Predefined stopping: the training procedure always stops after 100 epochs and always returns the last model $h_{t^*} = h_{100}$.

   (b) Non-adaptive stopping: the training procedure is executed for a fixed number of epochs T, and returns the model h_{t^*} with the lowest validation error observed during the training process, i.e., $t^* = \arg\min_{t \in \{1,\ldots,T\}} \hat{L}(h_t, S^{\texttt{val}})$.

   (c) Adaptive stopping: the training procedure stops when no improvement in $\hat{L}(h_t, S^{\texttt{val}})$ is observed for a significant number of epochs. It then returns the best model observed ever during training. (This procedure is proposed in Goodfellow et al. (2016, Algorithm 7.1) or `https://www.quora.com/How-does-one-employ-early-stopping-in-TensorFlow`, but again, you do not need to know the details of the training procedure.)

2. Derive a high-probability bound (a bound that holds with probability at least $1 - \delta$) on $L(h_{t^*})$ in terms of $\hat{L}(h_{t^*}, S^{\texttt{val}})$, δ, and the size n of the validation set $S^{\texttt{val}}$ for the three cases above. In the second case the bound may additionally depend on the total number of epochs T, while in the third case the bound may additionally depend on the index t^* of the epoch providing the optimal model. Please, solve the last case using the series $\sum_{i=1}^{\infty} \frac{1}{i(i+1)} = 1$.[2]

3. The adaptive approach suggests stopping when "no improvement in $\hat{L}(h_t, S^{\texttt{val}})$ is observed for a significant number of epochs". A natural way of redefining the stopping criterion once we have the generalization bound is to stop when "no improvement in the generalization bound is observed for a significant number of epochs". The adaptive approach does not limit the number of epochs in advance, but what is the maximal number of epochs $T_{\max}$, after which it makes no sense to continue training according to the bound you derived in Point 2? Express $T_{\max}$ in terms of the number of validation samples n. It is sufficient to provide an order of magnitude of $T_{\max}$ in terms of n, you do not have to calculate the explicit constants.

4. How would your answer to the previous point change if you were to use the series $\sum_{i=1}^{\infty} \frac{1}{2^i} = 1$ for deriving the bound? (You should get that with this series you can run significantly less epochs in the adaptive approach compared to the series used in Point 2. Thus, unlike in the case of decision trees in Section 4.4.1, here the choice of the series has a significant impact.)

5. In this question we compare the adaptive procedure with non-adaptive. Assume that the two procedures use the same initialization, so that the corresponding models at epoch t are identical, and assume that the adaptive procedure has considered all the models h_t for $t \in \{1, \ldots, T_{\max}\}$. Let t^* be the index of the model h_{t^*} minimizing the adaptive bound and let T^* be the index of the

[2] We have $\sum_{i=1}^{\infty} \frac{1}{i(i+1)} = \sum_{i=1}^{\infty} \left(\frac{1}{i} - \frac{1}{i+1}\right) = 1$.

model h_{T^*} minimizing the non-adaptive bound. Show that the generalization bound for adaptive stopping in Point 2 is never much worse than the generalization bound for non-adaptive stopping, but in some cases the adaptive bound can be significantly lower.

Guidance: To simplify the analysis, throughout the question we assume that the confidence parameter $\delta \leq \frac{1}{2}$. For $T \geq 1$ it gives $\delta \leq \frac{1}{2} \leq \frac{T}{T+1}$.

(a) First, assume that $T \leq T_{\max}$. Let t^* be the index of the epoch selected by the adaptive procedure and T^* be the index of the epoch selected by the non-adaptive procedure. Since the adaptive procedure has selected t^* we know that the adaptive bound for epoch t^* is lower than the adaptive bound for epoch T^*. We also know that $T^* \leq T$, where T is the number of epochs in the non-adaptive approach. Use this information and do some bounding to show that for any confidence parameter $\delta \leq \frac{1}{2}$, the adaptive bound can be at most a multiplicative factor of $\sqrt{2}$ larger than the non-adaptive bound.

(b) [Optional] Now consider the case $T > T_{\max}$. Show that in this case the non-adaptive bound is at least $\frac{1}{\sqrt{2}}$. Since the losses are upper bounded by 1, any bound can be truncated at 1 and still be a valid bound. In other words, for any "bound" we can define a "truncated bound" = max(1, "bound") and it will still be a valid bound. So in this case the truncated adaptive bound also cannot exceed the non-adaptive bound by more than a multiplicative factor of $\sqrt{2}$.

(c) You have shown that under the assumption that $\delta \leq \frac{1}{2}$ the adaptive bound never exceeds the non-adaptive bound by more than a multiplicative factor of $\sqrt{2}$. Now provide *two* examples of sequences of empirical losses $\hat{L}(h_1, S^{\texttt{val}}), \hat{L}(h_2, S^{\texttt{val}}), \ldots$, for which the adaptive bound can be significantly smaller than the non-adaptive bound. In both cases you should have $T < T_{\max}$ and $\delta \leq \frac{1}{2}$.

(d) [Optional] Show that irrespective of the choice of T, there always exists a sequence of losses $\hat{L}(h_1, S^{\texttt{val}}), \hat{L}(h_2, S^{\texttt{val}}), \ldots$, for which $\frac{\text{adaptive bound}}{\text{non adaptive bound}} \leq \left(\frac{2 \ln \frac{2}{\delta}}{n}\right)^{\frac{1}{4}}$.

Conclusion: depending on the data, the generalization bound for adaptive stopping can be significantly smaller than the generalization bound for non-adaptive stopping, and at the same time it is guaranteed that it is never worse by more than a multiplicative factor of $\sqrt{2}$.

Exercise 4.7 (*Occam's razor with* kl *inequality*)**.** In this exercise we derive a version of Occam's razor bound based on the kl inequality.

1. Prove the following theorem.

   Theorem 4.41 (Occam's kl-razor inequality)**.** *Let S be an i.i.d. sample of n points, let ℓ be a loss function bounded in the $[0,1]$ interval, let $\mathcal{H}$ be countable, and let $\pi(h)$ be such that it is independent of the sample S and satisfies $\pi(h) \geq 0$ for all h and $\sum_{h \in \mathcal{H}} \pi(h) \leq 1$. Let $\delta \in (0,1)$. Then*

   $$\mathbb{P}\left(\exists h \in \mathcal{H} : \mathrm{kl}(\hat{L}(h,S) \| L(h)) \geq \frac{\ln \frac{1}{\pi(h)\delta}}{n}\right) \leq \delta.$$

   You should prove the theorem directly, and not through relaxation of the PAC-Bayes-kl bound. Briefly emphasize where in your proof are you using the assumption that $\pi(h)$ is independent of S, and why is it necessary.

2. The bound in Theorem 4.41 is somewhat implicit. Prove the following corollary, which makes it more explicit, and clearly shows the "fast convergence rate" that it provides.

   Corollary 4.42. *Under the assumptions of Theorem 4.41*

   $$\mathbb{P}\left(\exists h \in \mathcal{H} : L(h) \geq \hat{L}(h,S) + \sqrt{\frac{2\hat{L}(h,S) \ln \frac{1}{\pi(h)\delta}}{n}} + \frac{2 \ln \frac{1}{\pi(h)\delta}}{n}\right) \leq \delta.$$

3. Briefly compare Theorem 4.42 with Occam's razor bound in Theorem 4.3. What are the advantages and when are they most prominent?

Exercise 4.8 (*The Airline Question*)**.**

1. An airline knows that any person making a reservation on a flight will not show up with probability of 0.05 (5 percent). They introduce a policy to sell 100 tickets for a flight that can hold only 99 passengers. Bound the probability that the number of people that show up for a flight will be larger than the number of seats (assuming they show up independently).

2. An airline has collected an i.i.d. sample of 10000 flight reservations and figured out that in this sample 5 percent of passengers who made a reservation did not show up for the flight. They introduce a policy to sell 100 tickets for a flight that can hold only 99 passengers. Bound the probability of observing such sample and getting a flight overbooked.

   There are multiple ways to approach this question. We will guide you through two options. You are asked to solve the question in both ways.

   (a) Let p be the true probability of showing up for a flight (remember that p is unknown). In the first approach we consider two events: the first is that in the sample of 10000 passengers, where each passenger shows up with probability p, we observe 95% of show-ups. The second event is that in the sample of 100 passengers, where each passenger shows up with probability p, everybody shows up. Note that these two events are independent. Bound the probability that they happen simultaneously assuming that p is known. And then find the worst case p (you can do it numerically). With a simple approach you can get a bound of around 0.61. If you are careful and use the right bounds you can get down to around 0.0068.

   It is advised to visualize the problem (the $[0, 1]$ interval with 0.95 point for the 95% show-ups and 1 for the 100% show-ups and p somewhere in $[0, 1]$). This should help you understand the problem, understand where the worst case p should be, and understand what direction of inequalities you need.

   Attention: This is a frequentist rather than a Bayesian question. In case you are familiar with the Bayesian approach, it cannot be applied here, because we do not provide a prior on p. In case you are unfamiliar with the Bayesian approach, you can safely ignore this comment.

   (b) The second approach considers an alternative way of generating the two samples, using the same idea as in the proof of the VC-bound. Consider the following process of generating the two samples:

      i. We sample 10100 passenger show up events independently at random according to an unknown distribution p.
      ii. We then split them into 10000 passengers in the collected sample and 100 passengers booked for the 99-seats flight.

   Bound the probability of observing a sample of 10000 with 95% show ups and a 99-seats flight with all 100 passengers showing up by following the above sampling protocol. If you do things right, you can get a bound of about 0.0062 (there may be some variations depending on how exactly you do the calculation).

Exercise 4.9 (*The Growth Function*)**.**

1. Let $\mathcal{H}$ be a finite hypothesis set with $|\mathcal{H}| = M$ hypotheses. Prove that $m_{\mathcal{H}}(n) \leq \min\{M, 2^n\}$.

2. Let $\mathcal{H}$ be a hypothesis space with 2 hypotheses (i.e., $|\mathcal{H}| = 2$). Prove that $m_{\mathcal{H}}(n) = 2$. (Pay attention that you are asked to prove an equality, $m_{\mathcal{H}}(n) = 2$, not an inequality.)

3. Prove that $m_{\mathcal{H}}(2n) \leq m_{\mathcal{H}}(n)^2$.

Exercise 4.10 (*The VC-dimension*)**.**

1. Let $\mathcal{H}$ be a finite hypothesis set with $|\mathcal{H}| = M$ hypotheses. Bound the VC-dimension of $\mathcal{H}$.

2. Let $\mathcal{H}$ be a hypothesis space with 2 hypotheses (i.e., $|\mathcal{H}| = 2$). Prove that $d_{\text{VC}}(\mathcal{H}) = 1$. (Pay attention that you are asked to prove an equality, not an inequality.)

3. Let $\mathcal{H}_+$ be the class of "positive" circles in $\mathbb{R}^2$ (each $h \in \mathcal{H}_+$ is defined by the center of the circle $c \in \mathbb{R}^2$ and its radius $r \in \mathbb{R}$; all points inside the circle are labeled positively and outside negatively). Prove that $d_{VC}(\mathcal{H}_+) \geq 3$.

4. Let $\mathcal{H} = \mathcal{H}_+ \cup \mathcal{H}_-$ be the class of "positive" and "negative" circles in $\mathbb{R}^2$ (the "negative" circles are negative inside and positive outside). Prove that $d_{VC}(\mathcal{H}) \geq 4$.

5. ***Optional question (0 points)*** Prove the matching upper bounds $d_{VC}(\mathcal{H}_+) \leq 3$ and $d_{VC}(\mathcal{H}) \leq 4$. [Doing this formally is not easy, but will earn you extra honor.]

6. What is the VC-dimension of the hypothesis space $\mathcal{H}_d$ of binary decision trees of depth d?

7. What is the VC-dimension of the hypothesis space $\mathcal{H}$ of binary decision trees of unlimited depth?

Exercise 4.11 (*Steps in the Proof of the VC Bound*)**.**

1. Prove Theorem 4.15. (Hint: use induction.)

2. Verify that Theorem 4.8, Theorem 4.13, and Theorem 4.15 together yield Theorem 4.16.

Exercise 4.12 (*The VC bound*)**.**

1. What should be the relation between $d_{\text{VC}}(\mathcal{H})$ and n in the VC generalization bound in Theorem 4.16 in order for the bound to be non-trivial? [A bound on the loss that is greater than or equal to 1 is trivial, because we know that the loss is always bounded by 1. You do not have to make an exact calculation, giving an order of magnitude is sufficient.]

2. In the case of a finite hypothesis space, $|\mathcal{H}| = M$, compare the generalization bound that you can obtain with Theorem 4.16 with the generalization bound in Theorem 4.2. In what situations which of the two bounds is tighter?

3. How many samples do you need in order to ensure that the empirical loss of a linear classifier selected out of a set of linear classifiers in $\mathbb{R}^{10}$ does not underestimate the expected loss by more than 0.01 with 99% confidence?

   Clarifications: (1) you are allowed to use the fact that the VC-dimension of general separating hyperplanes in $\mathbb{R}^d$ (not necessarily passing through the origin) is $d + 1$, see Abu-Mostafa et al. (2012, Exercise 2.4); (2) solving the question analytically is a bit tricky, you are allowed to provide a numerical solution. In either case (numerical or analytical solution), please, explain clearly in your report what you did.

4. You have a sample of 100,000 points and you have managed to find a linear separator that achieves $\hat{L}_{\texttt{FAT}}(h, S) = 0.01$ with a margin of 0.1. Provide a bound on its expected loss that holds with probability of 99%. The input space is assumed to be within the unit ball and the hypothesis space is the space of linear separators.

5. **The fine details of the lower bound.** We have shown that if a hypothesis space $\mathcal{H}$ has an infinite VC-dimension, it is possible to construct a worst-case data distribution that will lead to overfitting, i.e., with probability at least $\frac{1}{8}$ it will be possible to find a hypothesis for which $L(h) \geq \hat{L}(h, S) + \frac{1}{8}$. But does it mean that hypothesis spaces with infinite VC-dimension are always deemed to overfit? Well, the answer is that it depends on the data distribution. If the data distribution is not the worst-case for $\mathcal{H}$, there may still be hope.

   Construct a data distribution $p(X, Y)$ and a hypothesis space $\mathcal{H}$ with infinite VC-dimension, such that for any sample S of more than 100 points with probability at least 0.95 we will have $L(h) \leq \hat{L}(h, S) + 0.01$ for all h in $\mathcal{H}$.

   Hint: this can be achieved with an extremely simple example.

Exercise 4.13 (*Occam's* kl*-razor vs. PAC-Bayes-*kl)**.** In this question we compare Occam's kl-razor inequality with the PAC-Bayes-kl inequality.

1. Prove the following theorem, which extends Theorem 4.41 from Exercise 4.7 to soft selection.

Theorem 4.43 (Occam's kl-razor inequality for soft selection)**.** *Under the conditions of Theorem 4.41*

$$\mathbb{P}\left(\exists \rho : \mathrm{kl}\left(\mathbb{E}_\rho\left[\hat{L}(h,S)\right] \middle\| \mathbb{E}_\rho\left[L(h)\right]\right) \geq \frac{\mathbb{E}_\rho\left[\ln \frac{1}{\pi(h)}\right] + \ln \frac{1}{\delta}}{n}\right) \leq \delta.$$

2. Compare Theorem 4.43 to PAC-Bayes-kl inequality in Theorem 4.26. What are the advantages of PAC-Bayes-kl and what are the disadvantages?

Exercise 4.14 (*PAC-Bayesian Aggregation*)**.** In this question you are asked to reproduce an experiment from Thiemann et al. (2017, Section 6, Figure 2) (the paper is an outcome of a master project). Figure 2 corresponds to "the second experiment" in Section 6 of the paper "Experimental Results". You are only asked to reproduce the experiment for the first dataset, Ionosphere, which you can download from the UCI repository[3] (Asuncion and Newman, 2007). You are allowed to use any SVM solver you choose. Please, document carefully what you do and clearly annotate your graphs, including legend and axis labels.

Comments:

1. Assuming you have read Sections 4.8 and 4.9, it should be sufficient to read only the "Experimental Results" section of the paper in order to reproduce the experiment, but you are of course welcome to read the full article.

2. Theorem 6 in the paper corresponds to our Theorem 4.30.

3. Ideally, you should repeat the experiment several times, say 10, and report the average + some form of deviation, e.g. standard deviation or quantiles, over the repetitions. We have committed a sin by not doing it in the paper. We encourage you to make a proper experiment, but in order to save time you are allowed to repeat the sin (we will not take points for that). Please, do not do it in real papers.

4. Pay attention that you are required to find ρ through alternating minimization of the bound in Theorem 6 of the paper (Theorem 4.30 in this text), but then you report the loss of predictions by the ρ-weighted majority vote (defined in (4.31)) rather than the loss of the randomized classifier defined by ρ. A simple bound on the loss of the weighted majority vote is twice the bound on the loss of the randomized classifier (Theorem 4.31), although in practice weighted majority vote typically performs better than the randomized classifier. The paper reports the bound on the loss of the randomized classifier, which is, strictly speaking, incorrect, because the bound on the loss of the weighted majority vote is twice as large, but you are allowed to do the same.

Hint: Direct computation of the update rule for ρ,

$$\rho(h) = \frac{\pi(h) e^{-\lambda(n-r)\hat{L}^{\mathrm{val}}(h,S)}}{\sum_{h'} \pi(h') e^{-\lambda(n-r)\hat{L}^{\mathrm{val}}(h',S)}},$$

is numerically unstable, since for large $n-r$ it leads to division of zero by zero. A way to fix the problem is to normalize by $e^{-\lambda(n-r)\hat{L}^{\mathrm{val}}_{\min}}$, where $\hat{L}^{\mathrm{val}}_{\min} = \min_h \hat{L}^{\mathrm{val}}(h,S)$. This leads to

$$\rho(h) = \frac{\pi(h) e^{-\lambda(n-r)\hat{L}^{\mathrm{val}}(h,S)}}{\sum_{h'} \pi(h') e^{-\lambda(n-r)\hat{L}^{\mathrm{val}}(h',S)}} = \frac{\pi(h) e^{-\lambda(n-r)\left(\hat{L}^{\mathrm{val}}(h,S) - \hat{L}^{\mathrm{val}}_{\min}\right)}}{\sum_{h'} \pi(h') e^{-\lambda(n-r)\left(\hat{L}^{\mathrm{val}}(h',S) - \hat{L}^{\mathrm{val}}_{\min}\right)}}.$$

Calculation of the latter expression for $\rho(h)$ does not lead to numerical instability problems.

[3]The dataset can be downloaded from: `https://archive.ics.uci.edu/ml/datasets/Ionosphere`

Optional Add-on Repeat the experiment with the tandem bound on the weighted majority vote from Theorem 4.36, which corresponds to Masegosa et al. (2020, Theorem 9). You can find the details of optimization procedure for the bound in Masegosa et al. (2020, Appendix G). Tandem losses should be evaluated on overlaps of validation sets and n in the bounds should be replaced with the minimal overlap size for all pairs of hypotheses. Compare the results to the first order bound.

Exercise 4.15 (*Majority Vote*)**.** In this question we illustrate a few properties of the majority vote. Let MV denote a uniformly weighted majority vote.

1. Design an example of $\mathcal{H}$ and decision space $\mathbf{X}$, where $L(\mathrm{MV}) = 0$ and $L(h) \geq \frac{1}{3}$ for all h. (Hint: three hypotheses and $|\mathbf{X}| = 3$ is sufficient.)
2. Design an example of $\mathcal{H}$ and $\mathbf{X}$, where $L(\mathrm{MV}) > L(h)$ for all h.
   (a) Optional: design an example, where $L(\mathrm{MV}) \xrightarrow[|\mathcal{H}|\to\infty]{} 2\max_h L(h)$.
3. Let $\mathcal{H}$ be a hypothesis space, such that $|\mathcal{H}| = M$ and all $h \in \mathcal{H}$ have the same expected error, $L(h) = \frac{1}{2} - \varepsilon$ for $\varepsilon > 0$, and that the hypotheses in $\mathcal{H}$ make independent errors. Prove that $L(\mathrm{MV}) \xrightarrow[|\mathcal{H}|\to\infty]{} 0$. (In words: derive a bound for $L(\mathrm{MV})$ and show that as M grows the bound converges to zero, even though $L(h)$ can be almost as bad as 1/2.)

Bottom line: *If* the errors are independent *and* $L(h) < \frac{1}{2}$ for all $h \in \mathcal{H}$, the majority vote improves over individual classifiers. However, if $L(h) > \frac{1}{2}$ for some h the errors may get amplified, and if there is correlation it may play in either direction, depending on whether $L(h)$ is above or below $\frac{1}{2}$ and whether it is correlation or anti-correlation.

Exercise 4.16 (*PAC-Bayes-Unexpected-Bernstein*)**.**

Background The kl and PAC-Bayes-kl inequalities that we have studied in the course work well for binary random variables (the zero-one loss), but, even though they apply to any random variables bounded in the $[0, 1]$ interval, they are not necessarily a good choice if a random variable is non-binary and has a high probability mass inside the interval, because the kl inequality does not exploit small variance. For example, if you have a sample of Bernoulli random variables taking values $\{0, 1\}$ with probability half-half, and you have another sample of non-Bernoulli random variables from a distribution, which is concentrated on $\frac{1}{2}$ (i.e., the random variables always take the value $\frac{1}{2}$), the kl bound on the expectation will be the same in both cases, because it is only based on the empirical average $\hat{p}_n$, even though in the second case the random variables are much more concentrated than in the first.

Non-binary random variables occur, for example, if the loss of false positives and false negatives is asymmetric; in learning with abstention, where an algorithm is occasionally allowed to abstain from prediction and pay an abstention cost $c \in (0, \frac{1}{2})$; in working with continuous loss functions, such as the square or the absolute loss (although the Unexpected Bernstein inequality you will derive in this question still requires that the loss is one-side bounded); and many other problems (Wu and Seldin, 2022).

In this question you will derive a concentration of measure inequality belonging to the family of Bernstein's inequalities, which exploit small variance to provide tighter concentration guarantees.

Guidance The question is built step-by-step, and if you fail in one of the steps you can still proceed to the next, because the outcomes of the intermediate steps are given. While it is possible to find alternative derivations of the inequality in the literature, you are asked to follow the steps.

1. Let $Z \leq 1$ be a random variable. Show that for any $\lambda \in [0, \frac{1}{2}]$:

$$\mathbb{E}\left[e^{-\lambda Z - \lambda^2 Z^2}\right] \leq e^{-\lambda \mathbb{E}[Z]}.$$

   Point out where you are using the assumption that $\lambda \in [0, \frac{1}{2}]$ and where you are using the assumption that $Z \leq 1$.

   Hint: the following two inequalities are helpful for the proof. For any $z \geq -\frac{1}{2}$ we have $z - z^2 \leq \ln(1 + z)$ (Cesa-Bianchi et al., 2007, Lemma 1). And for any z, we have $1 + z \leq e^z$.

2. Prove that for $Z \leq 1$ and $\lambda \in [0, \frac{1}{2}]$,

$$\mathbb{E}\left[e^{\lambda(\mathbb{E}[Z]-Z)-\lambda^2 Z^2}\right] \leq 1.$$

3. Let $Z_1, \ldots, Z_n$ be independent random variables upper bounded by 1. Show that for any $\lambda \in [0, \frac{1}{2}]$

$$\mathbb{E}\left[e^{\lambda \sum_{i=1}^n (\mathbb{E}[Z_i]-Z_i)-\lambda^2 \sum_{i=1}^n Z_i^2}\right] \leq 1.$$

4. Let $Z_1, \ldots, Z_n$ be independent random variables upper bounded by 1. Show that for any $\lambda \in (0, \frac{1}{2}]$

$$\mathbb{P}\left(\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n Z_i\right] \geq \frac{1}{n}\sum_{i=1}^n Z_i + \frac{\lambda}{n}\sum_{i=1}^n Z_i^2 + \frac{\ln\frac{1}{\delta}}{\lambda n}\right) \leq \delta.$$

5. [Unexpected Bernstein inequality] Explanation: the right hand side of the inequality inside the probability above is minimized by $\lambda^*(Z_1, \ldots, Z_n) = \min\left\{\frac{1}{2}, \sqrt{\frac{\ln\frac{1}{\delta}}{\sum_{i=1}^n Z_i^2}}\right\}$, but we cannot plug $\lambda^*(Z_1, \ldots, Z_n)$ into the bound, because it depends on the sample $Z_1, \ldots, Z_n$, and if you trace the proof back to Point 1, it assumes that λ is independent of the sample. And, while the bound in Point 4 holds for any λ, it does not hold for all λ simultaneously. What you will do instead is take a grid of λ values and a union bound over the grid, and select λ from the grid, which minimizes the bound.

   Your task: Let $\Lambda = \{\lambda_1, \ldots, \lambda_k\}$ be a grid of k values of λ, such that $\lambda_i \in (0, \frac{1}{2}]$ for all i. Prove that:

$$\mathbb{P}\left(\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n Z_i\right] \geq \frac{1}{n}\sum_{i=1}^n Z_i + \min_{\lambda\in\Lambda}\left(\frac{\lambda}{n}\sum_{i=1}^n Z_i^2 + \frac{\ln\frac{k}{\delta}}{\lambda n}\right)\right) \leq \delta.$$

   We will call the above inequality an Unexpected Bernstein inequality.

6. [Empirical comparison of the kl and Unexpected Bernstein inequalities.] We compare the Unexpected Bernstein inequality with the kl inequality. Take a ternary random variable (a random variable taking three values) $Z \in \{0, \frac{1}{2}, 1\}$. Let $p_0 = \mathbb{P}(Z=0)$, $p_{\frac{1}{2}} = \mathbb{P}(Z=\frac{1}{2})$, and $p_1 = \mathbb{P}(Z=1)$. Set $p_0 = p_1 = (1-p_{\frac{1}{2}})/2$, i.e., the probability of getting $Z=0$ or $Z=1$ is equal, and we have one parameter $p_{\frac{1}{2}}$, which controls the probability mass of the central value. We will compare the bounds as a function of $p_{\frac{1}{2}} \in [0,1]$. Let $p = \mathbb{E}[Z]$ (in the constructed example, for any value of $p_{\frac{1}{2}}$ we have $p = \frac{1}{2}$, because $p_0 = p_1$). For each value of $p_{\frac{1}{2}}$ in a grid covering the $[0,1]$ interval draw a random sample $Z_1, \ldots, Z_n$ from the distribution we have constructed and let $\hat{p}_n = \frac{1}{n}\sum_{i=1}^n Z_i$ and $\hat{v}_n = \frac{1}{n}\sum_{i=1}^n Z_i^2$. Generate a figure, where you plot the Unexpected Bernstein bound on $p - \hat{p}_n$ and the kl bound on $p - \hat{p}_n$ as a function of $p_{\frac{1}{2}}$ for $p_{\frac{1}{2}} \in [0,1]$. The Unexpected Bernstein bound on $p - \hat{p}_n$ is $\min_{\lambda\in\Lambda}\left(\lambda\hat{v}_n + \frac{\ln\frac{k}{\delta}}{\lambda n}\right)$, and the kl bound on $p - \hat{p}_n$ is $\mathrm{kl}^{-1^+}\left(\hat{p}_n, \frac{\ln\frac{1}{\delta}}{n}\right) - \hat{p}_n$; pay attention that in contrast to Exercise 3.8 we subtract the value of $\hat{p}_n$ after inversion of kl to get a bound on the difference $p - \hat{p}_n$ rather than on p. Take the following values for the comparison: $n = 100$, $\delta = 0.05$, $|\Lambda| = k = \lceil \log_2(\sqrt{n/\ln(1/\delta)}/2)\rceil$, and $\Lambda = \{\frac{1}{2}, \frac{1}{2^2}, \ldots, \frac{1}{2^k}\}$. Briefly comment on the result of empirical evaluation.

   Explanation: The kl inequality depends only on the empirical first moment of the sample, $\hat{p}_n = \frac{1}{n}\sum_{i=1}^n Z_i$ and, therefore, it is "blind" to the variance and cannot exploit it. The Unexpected Bernstein inequality depends on the empirical first and second moments of the sample, $\hat{p}_n$ and $\hat{v}_n = \frac{1}{n}\sum_{i=1}^n Z_i^2$. The second moment is directly linked to the variance, $\mathbb{V}[Z] = \mathbb{E}[Z^2] - \mathbb{E}[Z]^2$ and, therefore, the Unexpected Bernstein inequality is able to exploit small variance.

7. Let S be an i.i.d. sample, h a prediction rule, and $\ell(y', y)$ a loss function upper bounded by 1. Define $\hat{V}(h,S) = \frac{1}{n}\sum_{i=1}^n \ell(h(X_i), Y_i)^2$ and $L(h)$ and $\hat{L}(h,S)$ as usual. Show that for any $\lambda \in [0, \frac{1}{2}]$ we have

$$\mathbb{E}\left[e^{n\left(\lambda\left(L(h)-\hat{L}(h,S)\right)-\lambda^2\hat{V}(h,S)\right)}\right] \leq 1.$$

8. Let S and ℓ be as before. Let $\mathcal{H}$ be a set of prediction rules, let π be a distribution on $\mathcal{H}$ that is independent of S. Show that for any $\lambda \in (0, \frac{1}{2}]$

$$\mathbb{P}\left(\exists \rho : \mathbb{E}_\rho\left[L(h)\right] \geq \mathbb{E}_\rho\left[\hat{L}(h,S)\right] + \lambda \mathbb{E}_\rho\left[\hat{V}(h,S)\right] + \frac{\mathrm{KL}(\rho\|\pi) + \ln\frac{1}{\delta}}{n\lambda}\right) \leq \delta,$$

where ρ denotes a distribution on $\mathcal{H}$.

Hint: Take $f(h,S) = n\left(\lambda\left(L(h) - \hat{L}(h,S)\right) - \lambda^2 \hat{V}(h,S)\right)$ and use PAC-Bayes bounding procedure and the result from the previous point.

Side remark: Note that the "optimal" value of λ (the one that minimizes the right hand side of the inequality inside the probability) depends on the data, and that the bound does not hold for all λ simultaneously. In the next step we resolve this issue by taking a grid of λ values and using the best value in the grid.

9. [PAC-Bayes-Unexpected-Bernstein Inequality.] Let $\Lambda = \{\lambda_1, \ldots, \lambda_k\}$ be a grid of k values of λ, such that $\lambda_i \in (0, \frac{1}{2}]$ for all i. Let S, ℓ, and π be as before. Prove that:

$$\mathbb{P}\left(\exists \rho : \mathbb{E}_\rho\left[L(h)\right] \geq \mathbb{E}_\rho\left[\hat{L}(h,S)\right] + \min_{\lambda\in\Lambda}\left(\lambda \mathbb{E}_\rho\left[\hat{V}(h,S)\right] + \frac{\mathrm{KL}(\rho\|\pi) + \ln\frac{k}{\delta}}{n\lambda}\right)\right) \leq \delta.$$

Chapter 5

Supervised Learning - Regression

In this chapter we consider the regression problem, which is another special case of supervised learning with $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \mathbb{R}$.

5.1 Linear Least Squares

Linear regression with square loss $\ell(Y', Y) = (Y' - Y)^2$ is also known as linear least squares. Let $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ be our sample. We are looking for a prediction rule of a form $h(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$, where $\mathbf{w}^T\mathbf{x}$ is the dot-product (also known as the inner product) between a vector $\mathbf{w} \in \mathbb{R}^d$ and a data point $\mathbf{x} \in \mathbb{R}^d$. We will use $\mathbf{w}$ to denote the above prediction rule. Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ be a matrix holding $\mathbf{x}_1^T, \ldots, \mathbf{x}_n^T$ as its rows

$$\mathbf{X} = \begin{pmatrix} — \ \mathbf{x}_1^T \ — \\ \vdots \\ — \ \mathbf{x}_n^T \ — \end{pmatrix}$$

and let $\mathbf{y} = (y_1, \ldots, y_n)^T$ be the vector of labels. We are looking for $\mathbf{w}$ that minimizes the empirical loss $\hat{L}(\mathbf{w}, S) = \sum_{i=1}^n \ell(\mathbf{w}^T\mathbf{x}_i, y_i) = \sum_{i=1}^n \left(\mathbf{w}^T\mathbf{x}_i - y_i\right)^2 = \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2$.

When the number of constraints n (the number of points in S) is larger than the number of unknowns d (the number of entries in $\mathbf{w}$), most often the linear system $\mathbf{X}\mathbf{w} = \mathbf{y}$ has no solutions (unless $\mathbf{y}$ by chance falls in the linear span of the columns of $\mathbf{X}$). Therefore, we are looking for the best approximation of $\mathbf{y}$ by a linear combination of the columns of $\mathbf{X}$, which means that we are looking for a *projection* of $\mathbf{y}$ onto the column space of $\mathbf{X}$. There are two ways to define projections, analytical and algebraic, which lead to two ways of solving the problem. In the analytical formulation the projection is a point of a form $\mathbf{X}\mathbf{w}$ that has minimal distance to $\mathbf{y}$. In the algebraic formulation the projection is a vector $\mathbf{X}\mathbf{w}$ that is perpendicular to the remainder $\mathbf{y} - \mathbf{X}\mathbf{w}$. We present both ways in detail below.

5.1.1 Analytical Approach

We are looking for

$$\min_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 = \min_{\mathbf{w}} (\mathbf{X}\mathbf{w} - \mathbf{y})^T(\mathbf{X}\mathbf{w} - \mathbf{y}) = \min_{\mathbf{w}} \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} - 2\mathbf{y}^T\mathbf{X}\mathbf{w} + \mathbf{y}^T\mathbf{y}.$$

By taking a derivative of the above and equating it to zero we have[1]

$$\frac{d(\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} - 2\mathbf{y}^T\mathbf{X}\mathbf{w} + \mathbf{y}^T\mathbf{y})}{d\mathbf{w}} = 2\mathbf{X}^T\mathbf{X}\mathbf{w} - 2\mathbf{X}^T\mathbf{y} = 0.$$

Which gives

$$\mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{X}^T\mathbf{y}.$$

If we assume that the *columns* of $\mathbf{X}$ are linearly independent ($dim(\mathbf{X}) = d$) then $\mathbf{X}^T\mathbf{X} \in \mathbb{R}^{d \times d}$ is invertible (see Chapter C) and we obtain

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

[1] See Appendix D for details on calculation of derivatives [gradients] of multidimensional functions.

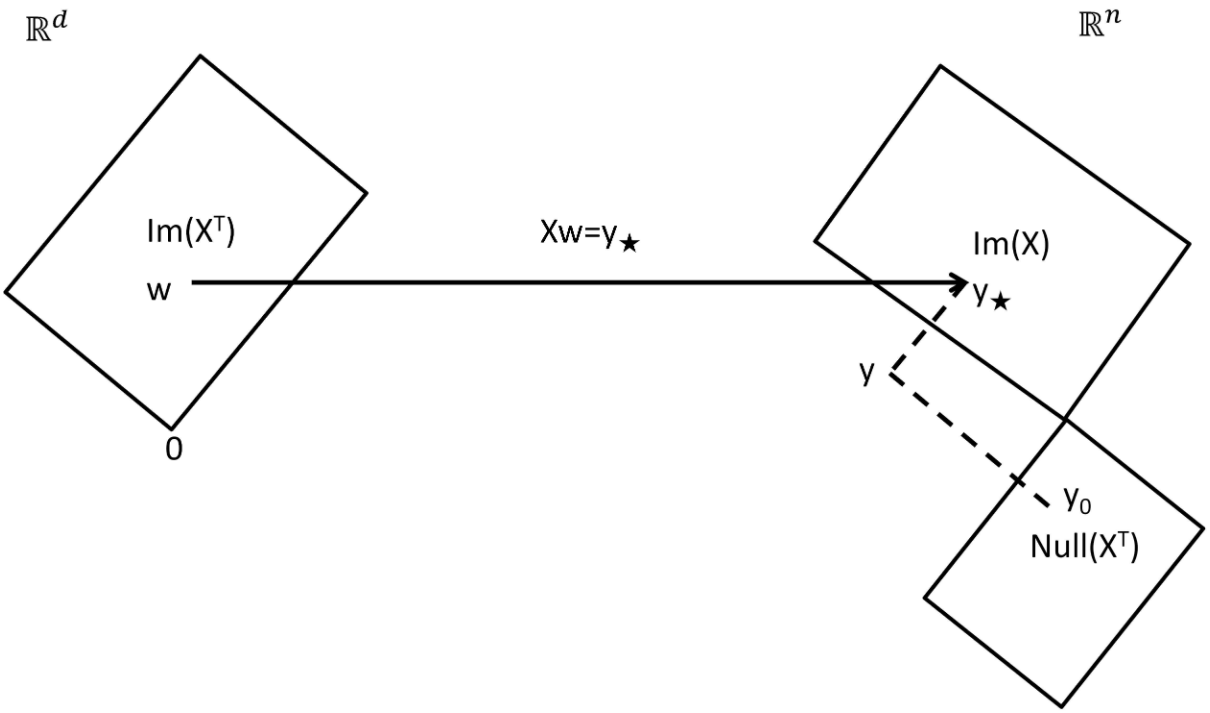


Figure 5.1: **Illustration of algebraic solution of linear least squares.**

5.1.2 Algebraic Approach - Fast Track

The projection $\mathbf{Xw}$ is a vector that is orthogonal to the remainder $\mathbf{y} - \mathbf{Xw}$ (so that $\mathbf{y}$ is a sum of the projection and the remainder, $\mathbf{y} = \mathbf{Xw} + (\mathbf{y} - \mathbf{Xw})$, and there is a right angle between the two). Two vectors are orthogonal if and only if their inner product is zero. Thus, we are looking for $\mathbf{w}$ that satisfies

$$(\mathbf{Xw})^T (\mathbf{y} - \mathbf{Xw}) = 0,$$

which is equivalent to $\mathbf{w}^T\mathbf{X}^T (\mathbf{y} - \mathbf{Xw}) = 0$. It is sufficient to find $\mathbf{w}$ that satisfies $\mathbf{X}^T (\mathbf{y} - \mathbf{Xw}) = 0$ to solve this equation, which is equivalent to $\mathbf{X}^T\mathbf{Xw} = \mathbf{X}^T\mathbf{y}$. By multiplying both sides by $\left(\mathbf{X}^T\mathbf{X}\right)^{-1}$ (which is defined, since the columns are linearly independent) we obtain a solution $\mathbf{w} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$.

This solution is, actually, unique due to independence of the columns of $\mathbf{X}$. Assume there is another solution $\mathbf{w}'$, such that $\mathbf{Xw}' = \mathbf{y}$. Then $\mathbf{Xw} - \mathbf{Xw}' = \mathbf{X}(\mathbf{w} - \mathbf{w}') = 0$, but since the columns of $\mathbf{X}$ are linearly independent the only their linear combination that yields zero is the zero vector, meaning that $\mathbf{w} - \mathbf{w}' = 0$ and $\mathbf{w} = \mathbf{w}'$.

5.1.3 Algebraic Approach - Complete Picture

Linear Least Squares is a great opportunity to revisit a number of basic concepts from linear algebra. Once the complete picture is understood, the algebraic solution of the problem is just one line. We refer the reader to Chapter C for a quick review of basic concepts from linear algebra. We are looking for a solution of $\mathbf{Xw} = \mathbf{y}$, where $\mathbf{y}$ (most likely) lies outside of the column space of $\mathbf{X}$ and the equation has no solution. Therefore, the best we can do is to solve $\mathbf{Xw} = \mathbf{y}_\star$, where $\mathbf{y}_\star$ is a projection of $\mathbf{y}$ onto the column space of $\mathbf{X}$ (see Figure 5.1). We assume that $dim(\mathbf{X}) = d$ and thus the matrix $\mathbf{X}^T\mathbf{X}$ is invertible. The projection $\mathbf{y}_\star$ is then given by $\mathbf{y}_\star = \mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$, which means that the best we can do is to solve $\mathbf{Xw} = \mathbf{y}_\star = \mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$ and the solution is $\mathbf{w} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$.

5.1.4 Using Linear Least Squares for Learning Coefficients of Non-linear Models

Linear Least Squares can be used for learning coefficients of non-linear models. For example, assume that we want to fit our data $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ (where both x_i-s and y_i-s are real numbers) with a polynomial of degree d. I.e., we want to have a model of a form $y = a_d x^d + a_{d-1} x^{d-1} + \cdots + a_1 x + a_0$. All we have to do is to map our features x_i-s into feature vectors $x_i \to (x_i^d, x_i^{d-1}, \ldots, x_i, 1)$ and apply linear least squares to the following system:

$$\begin{pmatrix} x_1^d & x_1^{d-1} & \ldots & x_1 & 1 \\ x_2^d & x_2^{d-1} & \ldots & x_2 & 1 \\ & & \vdots & & \\ x_n^d & x_n^{d-1} & \ldots & x_n & 1 \end{pmatrix} \begin{pmatrix} a_d \\ a_{d-1} \\ \vdots \\ a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

to get the parameters vector $(a_d, a_{d-1}, \ldots, a_1, a_0)^T$.

Chapter 6

Limitations and Pitfalls of the Classical Batch Learning

This chapter is dedicated to discussion of limitations of the classical batch learning model that has been the focus of Chapters 2, 4, and 5. Some of the limitations are addressed by the online learning model discussed in Chapter 7, but others are general limitations of learning based on selection.

6.1 The i.i.d. Assumption

The assumption that the data are i.i.d. and that new data come from the same distribution as the data used for training is behind everything discussed so far. (We depart from this assumption in Chapter 7, where we introduce online learning.) As a consequence, violation of the i.i.d. assumption leads to break down of the guarantees derived in Chapters 3 and 4. There are two ways in which the i.i.d. assumption can be violated. First, if the training data are not independent, the empirical error $\hat{L}(h, S)$ might not be converging to the expected error $L(h)$ at the same rate as in the theorems, or might not be converging at all, as in Exercise 3.4. Second, if the new data are not coming from the same distribution as the training data, then the empirical estimates clearly do not represent the quantity we are interested in.

Note that the assumption that future data come from the same distribution as the training data implies that generalization guarantees derived in Chapter 4 are not applicable in situations, where deployment of a learning algorithm leads to feedback loops, namely, when the algorithm changes its application environment. For example, past data can be used to predict congestion level on a network link, but if the predictions are used to reroute traffic they will get invalidated.

6.2 Overfitting

The central theme of Chapter 4 was to derive tools to control the deviation of empirical estimates of prediction accuracy from the expected accuracy. The control is based on balancing two competing forces - *concentration* and *selection*. For example, for selection from a finite set of prediction rules $\mathcal{H}$ with $|\mathcal{H}| = M$ based on a sample of size n we have

$$\mathbb{P}\Big(\exists h \in \mathcal{H} : L(h) \geq \hat{L}(h, S) + \varepsilon\Big) \leq \underbrace{M}_{\text{selection}} \times \underbrace{e^{-2n\varepsilon^2}}_{\text{concentration}} .$$

The power of concentration of each individual estimate depends on the number of points used to validate the models, whereas the power of selection depends on the richness of the class of prediction rules from which the selection is done. If the richness of selection grows (super-)exponentially with the number of points, the control over generalization may be lost, leading to overfitting. Two points are important to emphasize in this regard.

"Big data" is not a universal cure to overfitting The abundance of data on its own does not prevent overfitting. If the amount of selection (measured in the relevant way) is (super-)exponential

in the number of data points used to validate the prediction rules, overfitting may occur. In modern prediction models involving billions of parameters, the effective amount of selection may easily surpass the exponent of the number of points used for validation.

Internal and External selection The second point is that selection may be *internal* and *external*. By internal selection we mean selection done within a particular algorithm, for example, selection of a separating hyperplane in linear separation. Since an algorithm is in direct control of the set of prediction rules $\mathcal{H}$ it is selecting from, it can directly control overfitting. By external selection we mean selection happening outside of algorithms. For example, multiple research groups may apply various algorithms to a publicly available dataset, compare the results, and publish the best ones. In this case the same limited data (the publicly available dataset) are used to select from many more prediction rules than those used within any individual prediction algorithm. And this may easily lead to overfitting. In fact, many of the popular publicly available datasets are heavily overfitted. The only way to control overfitting in external selection is to turn it into internal selection. In other words, there should be a careful bookkeeping of the union of all the hypothesis classes that have been applied to a dataset and generalization bounds should be computed based on this union. Moreover, the selection of the hypothesis classes should be independent of the outcomes of preceding hypothesis classes on the validation set or, otherwise, fresh data are required for valid generalization guarantees.

6.3 Human Perception of Uncertainty

Humans are not very good at quantitative judgment of uncertainty (Kahneman, 2011). Moreover, when presented with various studies they tend to ignore confidence information altogether (if it happened to be there in the first place) and focus on the average number. For example, a newspaper article might say that "a study has shown that X is $p\%$ better than Y", but it might often omit the confidence interval (that would depend on the variation of outcomes and the number of subjects used in the study). Even if the confidence information is reported, it might be often ignored by the readers, who would keep in mind just the key number. Therefore, when reporting results it is important to report uncertainty information (confidence intervals), and preferably in a way that would make it hard to ignore the confidence information. For example, one may opt to plot a cloud of individual points or trajectories rather than the mean and standard deviation. Avoiding plotting the mean will make it harder to replace the outcomes with just one number, and would force the reader to give at least some consideration to uncertainty of the outcomes.

6.4 Correlation $\neq$ Causation

Machine learning provides a wealth of tools for correlation mining, and humans have a predisposition for making causal interpretations of correlation data (Kahneman, 2011). Therefore, it is important to emphasize that correlation does not imply causation. For example, boxers typically do not wear glasses (a correlation), but this does not imply that taking boxing classes is likely to improve anyone's vision (a causal relation). Machine learning can reliably identify the correlation between boxing and reasonable vision based on data. Moreover, if someone practices boxing, it is a reliable predictor that they have reasonable vision, so correlations can be used for reliable predictions. But methods discussed so far assume that the data are i.i.d. and are not built to gauge the effect of interventions (taking boxing classes), which would violate the assumption that new data come from the same distribution as the training data.

While the topic of causality is completely outside of the scope of this book, we note that online learning, discussed in Chapter 7, is based on exploiting causal dependencies between actions and outcomes (rather than mere correlations). In this respect a difference between online learning and causality is that causality aims to understand the causal structure of the world, whereas online learning aims to identify actions (interventions) that lead to a desired outcomes.

Chapter 7

Online Learning

So far in the book has considered *batch* learning. Batch learning starts with some training data, analyzes it, and then "ships the result of the analysis into the world" (see Figure 7.1). "The result" can be a fixed classifier h, a distribution over classifiers ρ, or anything else, the important point is that it does not change from the moment the selection procedure is over. It takes no new information into account. This is also the reason why we had to assume that new samples come from the same distribution as the samples in the training set, because the classifier was not designed to adapt.

Online learning is a learning framework, where data collection, analysis, and application of inferred knowledge are in a perpetual loop, see Figure 7.1. Examples of problems, which fit into this framework include:

- Investment in the stock market.
- Online advertizing and personalization.
- Online routing.
- Games.
- Robotics.
- And so on ...

The recurrent nature of online learning problems makes them closely related to repeated games. They also borrow some of the terminology from the game theory, including calling the problems *games* and every "Act - Observe - Analyze" cycle a *game round*. In general, we may need online learning in the following scenarios:

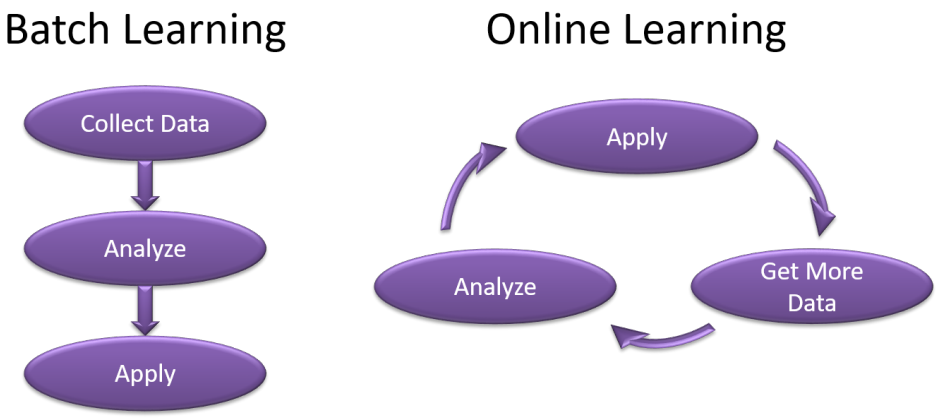


Figure 7.1: **Online learning vs. batch.**

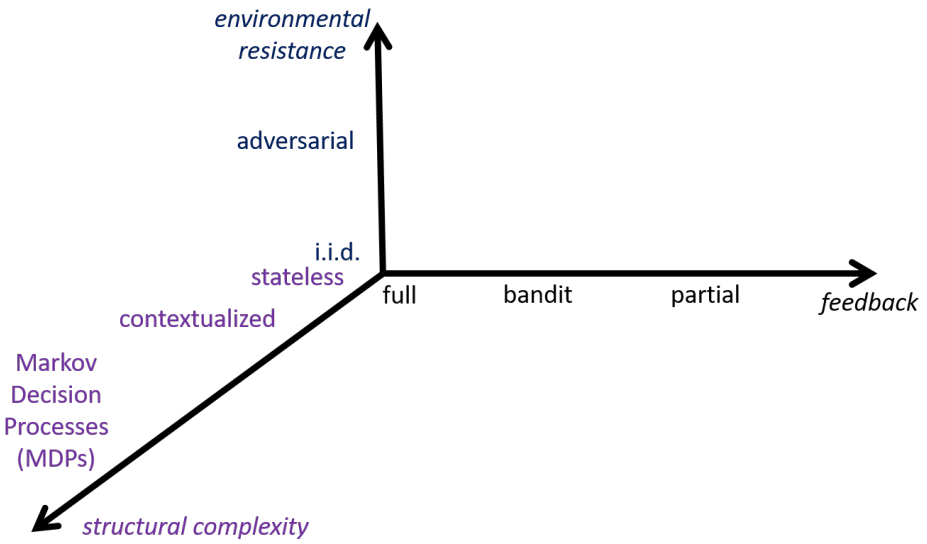


Figure 7.2: **The Space of Online Learning Problems.**

- Interactive learning: situations, where an algorithm continuously gets new information and taking it into account may improve the quality of future actions. For example, interaction with Internet users falls under this category – it makes sense to adapt to user behavior as the algorithm proceeds from one user query to the next.

- Adversarial or game-theoretic settings, where we cannot assume that "the future behaves similarly to the past". For example, in spam filtering we cannot assume that new spam messages are generated from the same distribution as the old ones. Or, in playing chess we cannot assume that the moves of the opponent are sampled i.i.d..

- Intelligent data collection. To cite Thompson (1933), "*there can be no objection to the use of data, however meagre, as a guide to action required before more can be collected*". Thompson was one of the pioneers of online learning and the theory of adaptive medical trials. In the context of adaptive medical trials the message is that it makes sense to look into the outcome of completed treatments before deciding on further treatments, as opposed to the more classical approach of A/B testing, where the size of treatment and control groups are decided upon before experimentation begins.[1]

As with many other problems in computer science, having loops (as in Figure 7.1) makes things much more challenging, but also much richer and more fun to work on. For example, online learning allows to treat adversarial environments, which is impossible to do in the batch setting.

7.1 The Space of Online Learning Problems

Online learning problems are characterized by three major parameters:

1. The amount of *feedback* that the algorithm receives in every round of interaction with the environment.

2. The *environmental resistance* to the algorithm.

[1] My other favourite quote on the topic is by Robbins (1952): "Until recently, statistical theory has been restricted to the design and analysis of sampling experiments in which the size and composition of the samples are completely determined before the experimentation begins. The reasons for this are partly historical, dating back to the time when the statistician was consulted, if at all, only after the experiment was over, and partly intrinsic in the mathematical difficulty of working with anything but a fixed number of independent random variables. A major advance now appears to be in the making with the creation of a theory of the *sequential design* of experiments, in which the size and composition of the samples are not fixed in advance but are functions of the observations themselves."

3. The *structural complexity* of a problem.

Jointly they define *the space of online learning problems*, see Figure 7.2. It is not really a space, but a convenient way to organize the material and get initial orientation in the zoo of online learning settings. We discuss the three axes of the space with some examples below.

Feedback

Feedback refers to the amount of information that the algorithm receives in every round of interaction with the environment. The most basic forms of feedback are *full information* and *limited* (better known as *bandit*[2]) feedback.

A classical example of a full information game is investment in the stock market. In every round of this game an algorithm distributes wealth over a set of stocks and the next day it observes the rates of all the stocks, which constitutes full information. With full information the algorithm can evaluate the quality of its own investment strategy, as well as any alternative investment strategy.

A classical example of a bandit feedback game are medical treatments. An algorithm has a set of *actions* (in this case treatments), but it can only apply one treatment to a given patient. The algorithm observes the outcome of the applied treatment, but not of the other treatments, resulting in limited feedback. With limited feedback the algorithm only observes the quality of the selected strategy, but it gets no direct access to the quality of alternative strategies that could have been selected. Limited feedback leads to the *exploration-exploitation trade-off*, which is the trade-mark signature of online learning. The essence of the exploration-exploitation trade-off is that in order to estimate the quality of actions the algorithm has to try them out (to explore). If it explores too little, it risks missing some good actions and end up performing suboptimally. However, exploration has a cost, because trying out suboptimal actions for too long is also undesirable. The goal is to balance exploration (trying new actions) with exploitation, which is taking actions, which are currently believed to be the optimal ones. The "Act-Observe-Analyze" cycle comes into play here, because unlike in batch learning the training set is not given, but is built by the algorithm for itself: if it does not try an action, it gets no data from it.

There are many other problems that fall within the bandit feedback framework, with another popular example being online advertising. A simplistic way of modeling online advertising is by assuming that there is a pool of advertisements, but on every round of the game it is only allowed to show one advertisement to a user. Since the advertiser only observes feedback for the advertisement that has been presented, the problem can be formulated as an online learning problem with bandit feedback.

There are other feedback models, which we will only touch briefly. In the bandit feedback model the algorithm observes a noisy estimate of the quality of selected action, for example, whether an advertisement was clicked or not. In *partial* feedback model studied under *partial monitoring* the feedback has some relation to the action, but not necessarily its quality. For example, in dynamic pricing the seller only observes whether a proposed price was above or below the value of a product for a buyer, but not the value itself (the maximal price the buyer would be ready to pay for the product). Bandit feedback is a special case of partial feedback, where the observation is the value. Another example is *dueling bandit* feedback, where the feedback is a relative preference over a pair of items rather than the absolute value of the items. For example, an answer to the question "Do you prefer fish or chicken?" is an example of dueling bandit feedback. Dueling bandit feedback model is used in information retrieval systems, since humans are much better in providing relative preferences rather than absolute utility values.

Environmental Resistance

Environmental resistance is concerned with how much the environment resists to the algorithm. Two classical examples are i.i.d. (a.k.a. *stochastic*) and *adversarial* environments. An example of an i.i.d. environment is the weather. It has a high degree of uncertainty, but it does not play against the algorithm. Another example of an i.i.d. environment are outcomes of medical treatments. Here also there is uncertainty in the outcomes, but the patients are not playing against the algorithm. An example of an adversarial environment is spam filtering. Here the spammers are deliberately changing distribution of the spam messages in order to outplay the spam filtering algorithm. Another classical example of an

[2] "The name derives from an imagined slot machine .... (Ordinary slot machines with one arm are one-armed bandits, since in the long run they are as effective as human bandits in separating the victim from his money.)" (Lai and Robbins, 1985)

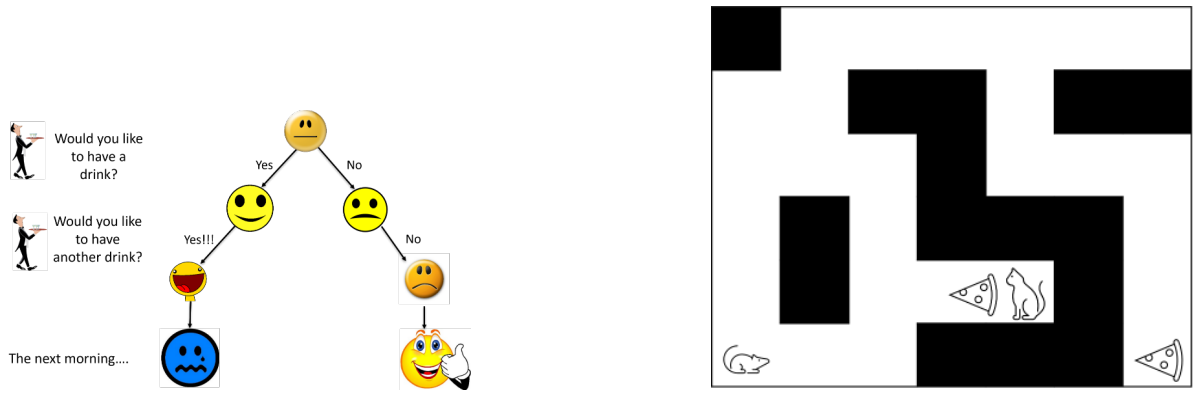


Figure 7.3: **Planning.** Even when the immediate outcomes are known, long-term goals require planning.

adversarial environment is the stock market. Even though the stock market does not play directly against an individual investor (assuming the investments are small), it is not stationary, because if there would be regularity in the market it would be exploited by other investors and would be gone.

The environment may also be collaborative, for example, when several agents are jointly solving a common task. Yet another example are slowly changing environments, where the parameters of a distribution are slowly changing with time.

Structural Complexity

In structural complexity we distinguish between *stateless* problems, *contextualized* problems (or problems with state), and *Markov decision processes*. In stateless problems actions are taken without taking any additional information except the history of the outcomes into account. In contextualized problems in every round of the game the algorithm observes a context (or state) and takes an action within the observed context. An example of a context is a medical record of a patient or, in the advertising example, it could be the parameters of the user that opened a web page.

Markov decision processes (MDPs) are concerned with processes with evolving state. The difference between contextualized problems and Markov decision processes is that in the former the actions of the algorithm do not influence the next state, whereas in the latter they do. For example, subsequent treatments of the same patient are changing his or her state and, therefore, depend on each other. In contrast, in subsequent treatments of different patients treatment of one patient does not influence the state of the next patient and, thus, can be modeled as a contextualized problem.

Markov decision processes are studied within the field of *reinforcement learning* (RL). There is no clear cut distinction between online learning and reinforcement learning, and one could be seen as a subfield of the other, or the other way around. But as a rule of thumb, problems involving evolution of states, such as Markov decision processes, are part of reinforcement learning, and problems that do not involve evolution of states are part of online learning.

When actions impact the state of the environment and the agent, they may have long-term consequences and, therefore, require *planning*. For example, getting to the "safe" slice of pizza in Figure 7.3 requires the mouse to plan a sequence of actions. Even if the immediate outcomes of every action in every state are known (there is no noise in execution of motor commands), there is still computational work to be done to infer the optimal action in each state. This is in contrast to online learning, where if the outcome of every action is known (e.g., the outcome of every treatment), inferring the best action is trivial. In many situations planning is combined with uncertainty estimation, for example, if the floor is slippery, the mouse might need to infer the relation between its actions and transitions between states. To summarize, online learning is primarily focused on uncertainty estimation, whereas reinforcement learning is focused on uncertainty estimation and planning, and the latter may be interesting and non-trivial even in absence of the former.

Online Learning	Reinforcement Learning
Uncertainty Estimation	Uncertainty Estimation
—	Planning

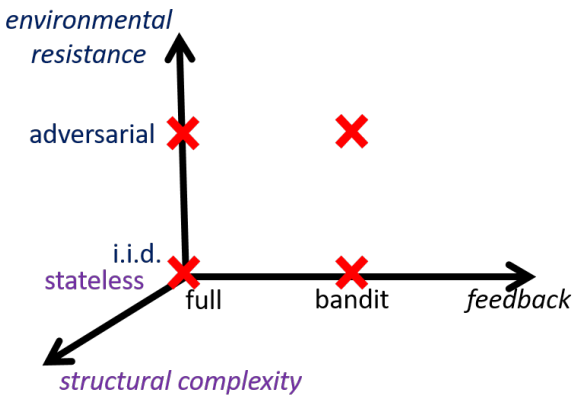


Figure 7.4: **The four basic online learning problems.**

There are many other online learning problems, which do not fit directly into Figure 7.2, but can still be discussed in terms of feedback, environmental resistance, and structural complexity. For example, in *combinatorial bandits* the goal is to select a set of actions, potentially with some constraints, and the quality of the set is evaluated jointly. An instance of a combinatorial bandit problem is selection of a path in a graph, such as communication or transport network. In this case an action can be decomposed into sub-actions corresponding to selection of edges in the graph. The goal is to minimize the length of a path, which may correspond to the delay between the source and the target nodes. Various forms of feedback can be considered, including bandit feedback, where the total length of the path is observed; semi-bandit feedback, where the length of each of the selected edges is observed; cascading bandit feedback, where the lengths of the edges are observed in a sequence until a terminating node (e.g., a server that is down) or the target is reached; or a full information feedback, where the length of all edges is observed.

Summary

To summarize, online and reinforcement learning introduce three new elements that we have not seen in batch learning: (1) incomplete feedback and *exploration* to deal with it, (2) the ability to handle *adversarial environments*, and (3) *planning*. And there is an infinite world of novel problem formulations that can be modeled in online and reinforcement learning.

In the following sections we consider in detail a number of the most basic online learning problems, and key tools for dealing with uncertainty and addressing the exploration-exploitation trade-off, and for handling i.i.d. and adversarial environments. (The topic of planning is left outside of the scope of the book.)

7.2 A General Basic Setup

We start with four most basic games in online learning, depicted by the red crosses in Figure 7.4: stateless i.i.d. full information, stateless i.i.d. adversarial, stateless i.i.d. bandit, and stateless adversarial bandit. The first setting is very easy and is studied in Exercise 7.2. The stateless i.i.d. adversarial setting is known as *prediction with expert advice*, and the two bandit settings are known as *stochastic multiarmed bandits* and *adversarial multiarmed bandits*, respectively. We first provide a general setup that encompasses all the four problems, and then specialize it to each of them.

We are given a $K \times \infty$ matrix of losses $\ell_{t,a}$, where $t \in \{1, 2, \dots\}$ and $a \in \{1, \dots, K\}$ and $\ell_{t,a} \in [0, 1]$.

$$
\begin{array}{rcccc}
 & \ell_{1,1}, & \ell_{2,1}, & \cdots & \ell_{t,1}, & \cdots \\
 & \vdots & \vdots & \cdots & \vdots & \cdots \\
\textit{Losses} & \ell_{1,a}, & \ell_{2,a}, & \cdots & \ell_{t,a}, & \cdots \\
 & \vdots & \vdots & \cdots & \vdots & \cdots \\
 & \ell_{1,K}, & \ell_{2,K}, & \cdots & \ell_{t,K}, & \cdots
\end{array}
$$

$$
\xrightarrow[time]{}
$$

The matrix is fixed before the game starts according to a protocol defined below, but not revealed to the algorithm. The game proceeds in the following way.

Game Protocol

For $t = 1, 2, \dots$:

1. Pick a row A_t
2. Suffer ℓ_{t,A_t}
3. Observe ... [the observations are defined below]

Definition of the four games There are two basic ways to generate the matrix of losses and two basic ways to define the observations, which jointly make up the four games, as summarized in the table below.

The first way to generate the matrix is to sample $\ell_{t,a}$-s independently, so that the mean of the losses in each row is fixed, $\mathbb{E}[\ell_{t,a}] = \mu(a)$. The second is to generate $\ell_{t,a}$-s arbitrarily. The second model of generation of losses is known as an *oblivious adversary*, since the generation happens before the game starts and does not take actions of the algorithm into account.[3]

The first way to define the observations is to reveal the full column $\ell_{t,1}, \dots, \ell_{t,K}$ after the algorithm has played the row A_t. The second is to reveal only the selected entry ℓ_{t,A_t}.

Jointly the two ways of generating the matrix of losses and the two ways of defining the observations define the four variants of the game.

Matrix generation \ Observations	Observe $\ell_{t,1}, \dots, \ell_{t,K}$	Observe ℓ_{t,A_t}
$\ell_{t,a}$-s are sampled i.i.d. with $\mathbb{E}[\ell_{t,a}] = \mu(a)$	I.I.D. Prediction with expert advice	Stochastic multiarmed bandits
$\ell_{t,a}$ are selected arbitrarily (by an adversary)	Prediction with expert advice (adversarial)	Adversarial multiarmed bandits

Performance Measure The goal of the algorithm is to play so that the loss it suffers will not be significantly larger than the loss of the best row in hindsight. There are several ways to formalize this goal. The basic performance measure is the *regret* defined by

$$R_T = \sum_{t=1}^{T} \ell_{t,A_t} - \min_a \sum_{t=1}^{T} \ell_{t,a}.$$

In adversarial problems we analyze the *expected regret*[4] defined by

$$\mathbb{E}[R_T] = \mathbb{E}\left[\sum_{t=1}^{T} \ell_{t,A_t}\right] - \mathbb{E}\left[\min_a \sum_{t=1}^{T} \ell_{t,a}\right].$$

[3] It is also possible to consider an *adaptive adversary*, which generates losses as the game proceeds and takes past actions of the algorithm into account. We do not discuss this model in the book.

[4] It is also possible to analyze the regret, but we do not do it here.

If the sequence of losses is deterministic, we can remove the second expectation and obtain

$$\mathbb{E}\left[R_T\right] = \mathbb{E}\left[\sum_{t=1}^{T} \ell_{t,A_t}\right] - \min_a \sum_{t=1}^{T} \ell_{t,a}.$$

In stochastic problems we analyze the *pseudo regret* defined by

$$\bar{R}_T = \mathbb{E}\left[\sum_{t=1}^{T} \ell_{t,A_t}\right] - \min_a \mathbb{E}\left[\sum_{t=1}^{T} \ell_{t,a}\right] = \mathbb{E}\left[\sum_{t=1}^{T} \ell_{t,A_t}\right] - T \min_a \mu(a).$$

Note that since for random variables X and Y we have $\mathbb{E}\left[\min\left\{X, Y\right\}\right] \leq \min\left\{\mathbb{E}\left[X\right], \mathbb{E}\left[Y\right]\right\}$ [it is recommended to verify this identity], we have $\bar{R}_T \leq \mathbb{E}\left[R_T\right]$.

The different notions of regret Let us briefly discuss the relations and differences between regret, expected regret, and pseudo-regret. First, we note that in the oblivious adversarial setting the losses are considered to be selected deterministically and, therefore, the expectation in the second term can be dropped, resulting in $\mathbb{E}\left[\min_a \sum_{t=1}^{T} \ell_{t,a}\right] = \min_a \mathbb{E}\left[\sum_{t=1}^{T} \ell_{t,a}\right] = \min_a \sum_{t=1}^{T} \ell_{t,a}$. Thus, in the oblivious adversarial setting the notions of expected regret and pseudo-regret coincide. (This is not true for the adaptive setting, but we do not delve into it here.) The difference between regret and expected/pseudo-regret in the adversarial setting is thus only in the first term – whether we want to obtain guarantees on the expected performance of an algorithm, $\mathbb{E}\left[\sum_{t=1}^{T} \ell_{t,A_t}\right]$, or individual roll-outs of its play, $\sum_{t=1}^{T} \ell_{t,A_t}$. Both are valid targets, we focus on the expected performance because it is a tiny bit easier.

In the stochastic setting $\mathbb{E}\left[\min_a \sum_{t=1}^{T} \ell_{t,a}\right] \leq \min_a \mathbb{E}\left[\sum_{t=1}^{T} \ell_{t,a}\right] = T \min_a \mu(a)$, and so the expected regret and the pseudo-regret are not the same. In pseudo-regret the performance baseline is the expected performance of a best action, $T \min_a \mu(a)$, defined by $\mu(a)$, whereas in expected regret it is the expected best roll-out of any action. Imagine that there are K actions and the loss of every action at every round is a Bernoulli random variable with bias $\frac{1}{2}$. Then $\mu(a) = \frac{1}{2}$ for all a and the pseudo-regret baseline is $\frac{1}{2}T$. And for any algorithm $\mathbb{E}\left[\sum_{t=1}^{T} \ell_{t,A_t}\right] = \frac{1}{2}T$, thus $\bar{R}_T = 0$. However, $\mathbb{E}\left[\min_a \sum_{t=1}^{T} \ell_{t,a}\right] \leq \frac{1}{2}T$, because the baseline (the competitor of the algorithm) is allowed to select the best out of K roll-outs of T Bernoulli random variables with bias $\frac{1}{2}$. In fact, $\mathbb{E}\left[\min_a \sum_{t=1}^{T} \ell_{t,a}\right] \approx \frac{1}{2}T - \sqrt{\frac{1}{2}T \ln K}$ (see Section 7.4.1), leading to $\mathbb{E}\left[R_T\right] \approx \sqrt{\frac{1}{2}T \ln K}$. Even though the loss of any algorithm cannot be smaller than $\frac{1}{2}T$ in expectation, and the loss of any fixed row is also $\frac{1}{2}T$ in expectation, the expected regret grows as $\sqrt{\frac{1}{2}T \ln K}$, because the competitor has the advantage of being allowed to make K tries of sampling T Bernoulli losses and selecting the best, whereas the algorithm gets just one try. Thus, comparing the performance of an algorithm to the best row in expectation rather than the expected best roll-out is considered more reasonable. We will be able to derive bounds on pseudo-regret in the stochastic setting that grow at the rate of $\ln T$, whereas the fluctuations of $\sum_{t=1}^{T} \ell_{t,a}$ and $\sum_{t=1}^{T} \ell_{t,A_t}$ can be as large as $\sqrt{T}$, and so the best possible bounds on the regret and the expected regret in the stochastic setting are of order $\sqrt{T}$.

Explanation of the Names In the complete definition of prediction with expert advice game, in every round of the game the player gets an advice from K experts and then takes an action, which may be a function of the advice. The player suffers a loss depending on the action taken, and the experts suffer losses depending on their advice. Hence the name, prediction with expert advice. If we restrict the actions of the player to following the advice of a single expert, then from the perspective of the playing strategy the actual advice does not matter and it is only the loss that defines the strategy. We consider the restricted setting, because it allows to highlight the relation to multiarmed bandits.

The name multiarmed bandits comes from the analogy with slot machines, which are one-armed bandits. In this game the actions are the "arms" of a slot machine.

Losses vs. Rewards In some games it is more natural to consider rewards (also called gains), rather than losses. In fact, in the literature on stochastic problems it is more popular to work with rewards, whereas in the literature on adversarial problems it is more popular to work with losses. There is a simple transformation $r = 1 - \ell$, which brings a losses game into a gains game and the other way around. Interestingly, in the adversarial setting working with losses leads to tighter and simpler results (see Exercise 7.13). In the stochastic setting the choice does not matter.

7.3 I.I.D. (stochastic) Multiarmed Bandits

In this section we consider a multiarmed bandit game, where the outcomes (the sequence of losses) are generated i.i.d. with fixed, but unknown means. In this game there is no difference between working with losses or rewards, and since most of the literature is based on games with rewards we are going to use rewards in order to be consistent. The treatment of losses is identical - see Seldin (2015).

Notations We are given a $K \times \infty$ matrix of rewards (or gains) $r_{t,a}$, where $t \in \{1, 2, \dots\}$ and $a \in \{1, \dots, K\}$.

$$
\begin{array}{c}
\textit{Action rewards} \\
\\
\\
\\
\\
\\
\end{array}
\begin{array}{cccccc}
r_{1,1}, & r_{2,1}, & \cdots & r_{t,1}, & \cdots \\
\vdots & \vdots & \cdots & \vdots & \cdots \\
r_{1,a}, & r_{2,a}, & \cdots & r_{t,a}, & \cdots \\
\vdots & \vdots & \cdots & \vdots & \cdots \\
r_{1,K}, & r_{2,K}, & \cdots & r_{t,K}, & \cdots \\
\multicolumn{5}{c}{\longrightarrow} \\
\multicolumn{5}{c}{\textit{time}}
\end{array}
$$

We assume that $r_{t,a}$-s are in $[0, 1]$ and that they are generated independently, so that $\mathbb{E}[r_{t,a}] = \mu(a)$. We use $\mu^* = \max_a \mu(a)$ to denote the expected reward of an optimal action and $\Delta(a) = \mu^* - \mu(a)$ to denote the *suboptimality gap* (or simply the *gap*) of action a. The suboptimality gap $\Delta(a)$ measures by how much, in expectation, playing action a is worse than playing the optimal action. We use $a^* \in \arg\max_a \mu(a)$ to denote *a best action* (note that there may be more than one best action, in such case we let a^* be any of them).

Game Definition

For $t = 1, 2, \dots$:

1. Pick a row A_t

2. Observe & accumulate r_{t,A_t}

Performance Measure Let $N_t(a)$ denote the number of times action a was played up to round t. We measure the performance using the pseudo regret and we rewrite it in the following way (note that since we are working with rewards, we subtract the expected reward of the algorithm from the expected

reward of a best arm, whereas for losses it was the other way around)

$$\begin{aligned}
\bar{R}_T &= \max_a \mathbb{E}\left[\sum_{t=1}^{T} r_{t,a}\right] - \mathbb{E}\left[\sum_{t=1}^{T} r_{t,A_t}\right] \\
&= T\mu^* - \mathbb{E}\left[\sum_{t=1}^{T} r_{t,A_t}\right] \\
&= \sum_{t=1}^{T} \mathbb{E}\left[\mu^* - r_{t,A_t}\right] \\
&= \sum_{t=1}^{T} \mathbb{E}\left[\mathbb{E}\left[\mu^* - r_{t,A_t} | A_t\right]\right] \qquad (7.1) \\
&= \sum_{t=1}^{T} \mathbb{E}\left[\mu^* - \mu(A_t)\right] \\
&= \sum_{t=1}^{T} \mathbb{E}\left[\Delta(A_t)\right] \\
&= \sum_{a} \Delta(a)\mathbb{E}\left[N_T(a)\right].
\end{aligned}$$

In step (7.1) we note that $\mathbb{E}\left[r_{t,A_t}\right]$ is an expectation over two random variables, the selection of A_t, which is based on the history of the game, and the draw of r_{t,A_t}, for which $\mathbb{E}\left[r_{t,A_t}|A_t\right] = \mu(A_t)$. We have $\mathbb{E}\left[r_{t,A_t}\right] = \mathbb{E}\left[\mathbb{E}\left[r_{t,A_t}|A_t\right]\right]$, where the inner expectation is with respect to the draw of r_{t,A_t} and the outer expectation is with respect to the draw of A_t.

Note that in the i.i.d. setting the reward of an algorithm is compared to the expected reward of the best action in expectation, $\max_a \mathbb{E}\left[\sum_{t=1}^{T} r_{t,a}\right] = T \max_a \mu(a)$, whereas in the adversarial setting the reward of an algorithm is compared to the reward of the best action in hindsight, $\max_a \sum_{t=1}^{T} r_{t,a}$.

Exploration-exploitation trade-off: A simple approach

I.I.D. multiarmed bandits is the simplest problem, where we face the exploration-exploitation trade-off. In general, the goal is to play a best arm in all the rounds, but since the identity of the best arm is unknown, it has to be identified first. In order to identify a best arm we need to explore all the arms. However, rounds used for exploration of suboptimal arms increase the regret, because every time we play a suboptmal arm a, we pay $\Delta(a)$ in the regret. The total regret is $\bar{R} = \sum_a \Delta(a)\mathbb{E}\left[N_T(a)\right]$, where $\mathbb{E}\left[N_T(a)\right]$ is the expected number of times a suboptimal action a was played. At the same time, saving too much on exploration may lead to confusion between a best and a suboptimal arm, which may eventually lead to even higher regret if we start exploiting a wrong arm.

So let us make a first attempt at quantifying this trade-off. Assume that we have just two actions, and we know the suboptimality gap Δ, so that $\mu(a) = \mu(a^*) - \Delta$, but we do not know which of the two actions is the better one, so we need to figure it out. Assume that we know the time horizon T we are going to play the game. A possible approach is to start with εT exploration rounds, where we play each of the two arms $\frac{1}{2}\varepsilon T$ times, followed by $(1-\varepsilon)T$ exploitation rounds, where we play the arm that yielded the highest reward by the end of the exploration phase. What should be the length of the exploration phase εT and what will be the pseudo-regret of this playing strategy?

Let $\delta(\varepsilon)$ denote the probability that we misidentify the best arm at the end of the exploration phase, namely, due to statistical fluctuations the suboptimal arm a happens to yield a higher reward than the optimal arm a^*. The pseudo regret can be bounded by:

$$\bar{R}_T \leq \underbrace{\frac{1}{2}\Delta\varepsilon T}_{\text{exploration}} + \underbrace{\delta(\varepsilon)\Delta(1-\varepsilon)T}_{\text{exploitation}} \leq \frac{1}{2}\Delta\varepsilon T + \delta(\varepsilon)\Delta T = \left(\frac{1}{2}\varepsilon + \delta(\varepsilon)\right)\Delta T,$$

where the first term is a bound on the pseudo regret during the exploration phase and the second term is a bound on the pseudo regret during the exploitation phase in case we select a wrong arm at the end of the exploration phase. Now what is $\delta(\varepsilon)$? Let $\hat{\mu}_t(a)$ denote the empirical mean of observed rewards of arm a up to round t. For the exploitation phase it is natural to select the arm that maximizes $\hat{\mu}_{\varepsilon T}(a)$ at the end of the exploration phase. Therefore:

$$\begin{aligned}\delta(\varepsilon) &= \mathbb{P}(\hat{\mu}_{\varepsilon T}(a) \geq \hat{\mu}_{\varepsilon T}(a^*)) \\ &\leq \mathbb{P}\left(\hat{\mu}_{\varepsilon T}(a) \geq \mu(a) + \frac{1}{2}\Delta\right) + \mathbb{P}\left(\hat{\mu}_{\varepsilon T}(a^*) \leq \mu^* - \frac{1}{2}\Delta\right) \\ &\leq 2e^{-2\frac{\varepsilon T}{2}\left(\frac{1}{2}\Delta\right)^2} = 2e^{-\varepsilon T\Delta^2/4},\end{aligned}$$

where the last line is by Hoeffding's inequality. By substituting this back into the regret bound we obtain:

$$\bar{R}_T \leq \left(\frac{1}{2}\varepsilon + 2e^{-\varepsilon T\Delta^2/4}\right)\Delta T.$$

In order to minimize $\frac{1}{2}\varepsilon + 2e^{-\varepsilon T\Delta^2/4}$ we take a derivative and equate it to zero, which leads to $\varepsilon = \frac{\ln(T\Delta^2)}{T\Delta^2/4}$. It is easy to check that the second derivative is positive, confirming that this is the minimum. Note that ε must be non-negative, so strictly speaking we have $\varepsilon = \max\left\{0, \frac{\ln(T\Delta^2)}{T\Delta^2/4}\right\}$. If we substitute this back into the regret bound we obtain:

$$\bar{R}_T \leq \min\left\{\Delta T, \left(\frac{2\ln(T\Delta^2)}{T\Delta^2} + 2e^{-\ln(T\Delta^2)}\right)\Delta T\right\} = \min\left\{\Delta T, \frac{2\ln(T\Delta^2)}{\Delta} + \frac{2}{\Delta}\right\}.$$

Note that the number of exploration rounds is $\varepsilon T = \max\left\{0, \frac{\ln(T\Delta^2)}{\Delta^2/4}\right\}$.

Pay attention that the regret bound has an inverse dependence on Δ, meaning that it gets *larger* as Δ gets *smaller*. Although intuitively when Δ is small we do not care that much about playing a suboptimal action as opposed to the case when Δ is large, problems with small Δ are actually harder and lead to larger regret. The reason is that the number of rounds that it takes to identify the best action (the number of exploration steps εT) grows with $1/\Delta^2$. Even though in each exploration round we only suffer a regret of Δ, the fact that the number of exploration rounds grows with $1/\Delta^2$ makes problems with small Δ harder and makes the regret grow at the rate of $1/\Delta$. However, note that if the time horizon T is very small in relation to $1/\Delta^2$, then there is not enough time to identify the best action, and the ΔT term dominates the minimum in the regret bound, see Exercise 7.4 for further details.

The above approach has three drawbacks: (1) it assumes knowledge of the time horizon T, (2) it assumes knowledge of the gap Δ, and (3) if we would generalize it to more than one arm, the length of the exploration phase would depend on the smallest gap, even if there are many arms with larger gap that are much easier to eliminate. The following approach resolves all the three problems.

The Upper Confidence Bound (UCB) algorithm

We present the UCB1 algorithm of Auer et al. (2002a).[5]

Algorithm 3 UCB1 (Auer et al., 2002a)

Initialization: Play each action once.
for $t = K+1, K+2, \ldots$ **do**

$$\text{Play } A_t = \arg\max_a \hat{\mu}_{t-1}(a) + \sqrt{\frac{3\ln t}{2N_{t-1}(a)}}.$$

end for

The expression $U_t(a) = \hat{\mu}_{t-1}(a) + \sqrt{\frac{3\ln t}{2N_{t-1}(a)}}$ is called an *upper confidence bound*. Why? Because $U_t(a)$ upper bounds $\mu(a)$ with high probability. UCB approach follows the *optimism in the face of uncertainty principle*. That is, we take an optimistic estimate of the reward of every arm by taking the upper limit of the confidence bound. UCB1 algorithm has the following regret guarantee.

[5]See Exercise 7.5 for an improved parametrization and analysis.

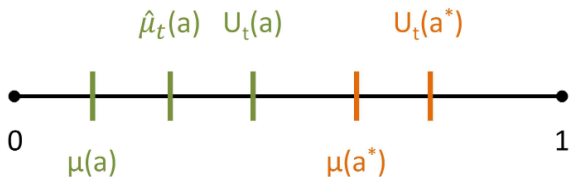


Figure 7.5: Illustration for UCB analysis.

Theorem 7.1. *For any time T the regret of UCB1 satisfies:*

$$\bar{R}_T \leq 6 \sum_{a:\Delta(a)>0} \frac{\ln T}{\Delta(a)} + \left(1 + \frac{\pi^2}{3}\right) \sum_a \Delta(a).$$

Proof. For the analysis it is convenient to have the following picture in mind – see Figure 7.5. A suboptimal arm may be played if $U_t(a) \geq U_t(a^*)$. Our goal is to show that this does not happen too often. The analysis is based on the following three points, which bound the corresponding distances in Figure 7.5.

1. We show that $U_t(a^*) > \mu(a^*)$ for almost all rounds. A bit more precisely, let $F(a^*)$ be the number of rounds when $U_t(a^*) \leq \mu(a^*)$, then $\mathbb{E}[F(a^*)] \leq \frac{\pi^2}{6}$.

2. In a similar way, we show that $\hat{\mu}_t(a) < \mu(a) + \sqrt{\frac{3 \ln t}{2N_t(a)}}$ for almost all rounds. A bit more precisely, let $F(a)$ be the number of rounds when $\hat{\mu}_t(a) \geq \mu(a) + \sqrt{\frac{3 \ln t}{2N_t(a)}}$, then $\mathbb{E}[F(a)] \leq \frac{\pi^2}{6}$. (Note that $L_t(a) = \hat{\mu}_t(a) - \sqrt{\frac{3 \ln t}{2N_t(a)}}$ is a lower confidence bound for $\mu(a)$, meaning that with high probability $L_t(a) < \mu(a)$.)

3. When Point 2 holds we have that $U_t(a) = \hat{\mu}_{t-1}(a) + \sqrt{\frac{3 \ln t}{2N_{t-1}(a)}} \leq \mu(a) + 2\sqrt{\frac{3 \ln t}{2N_{t-1}(a)}} = \mu(a^*) - \Delta(a) + 2\sqrt{\frac{3 \ln t}{2N_{t-1}(a)}}$.

Let us fix time horizon T and analyze what happens by time T (note that the algorithm does not depend on T). We have that for most rounds $t \leq T$:

$$U_t(a) < \mu(a^*) - \Delta(a) + \sqrt{\frac{6 \ln t}{N_{t-1}(a)}} \leq \mu(a^*) - \Delta(a) + \sqrt{\frac{6 \ln T}{N_{t-1}(a)}},$$
$$U_t(a^*) > \mu(a^*).$$

Thus, we can play a suboptimal action a only in the following cases:

- Either $\sqrt{\frac{6 \ln T}{N_{t-1}(a)}} \geq \Delta(a)$, which means that $N_{t-1}(a) \leq \frac{6 \ln T}{\Delta(a)^2}$. (As long as a suboptimal arm has not been played $\frac{6 \ln T}{\Delta(a)^2}$ times, its confidence interval is not tight enough to reliably disambiguate it from the best arm.)
- Or one of the confidence intervals in Points 1 or 2 has failed.

In other words, after a suboptimal action a has been played for $\left\lceil \frac{6 \ln T}{\Delta(a)^2} \right\rceil$ rounds, it can only be played again only if one of the confidence intervals fails. Therefore,

$$\mathbb{E}[N_T(a)] \leq \left\lceil \frac{6 \ln T}{\Delta(a)^2} \right\rceil + \mathbb{E}[F(a^*)] + \mathbb{E}[F(a)] \leq \frac{6 \ln T}{\Delta(a)^2} + 1 + \frac{\pi^2}{3}$$

and since $\bar{R}_T(a) = \sum_a \Delta(a) \mathbb{E}[N_T(a)]$ the result follows.

To complete the proof it is left to prove Points 1 and 2. We prove Point 1, the proof of Point 2 is identical. We start by looking at

$$\mathbb{P}(U_t(a^*) \le \mu(a^*)) = \mathbb{P}\left(\hat{\mu}_{t-1}(a^*) + \sqrt{\frac{3\ln t}{2N_{t-1}(a^*)}} \le \mu(a^*)\right) = \mathbb{P}\left(\mu(a^*) - \hat{\mu}_{t-1}(a^*) \ge \sqrt{\frac{3\ln t}{2N_{t-1}(a^*)}}\right).$$

The delicate point is that $N_{t-1}(a^*)$ is a random variable dependent on $\hat{\mu}_{t-1}(a^*)$, and thus we cannot apply Hoeffding's inequality directly. Instead, we introduce a series of random variables $X_1, X_2, \dots$, such that X_i-s have the same distribution as r_{t,a^*}-s. Let $\bar{\mu}_s = \frac{1}{s}\sum_{i=1}^{s} X_i$ be the average of the first s elements of the sequence. Then we have:

$$\begin{aligned}
\mathbb{P}\left(\mu(a^*) - \hat{\mu}_{t-1}(a^*) \ge \sqrt{\frac{3\ln t}{2N_{t-1}(a^*)}}\right) &\le \mathbb{P}\left(\exists s \in \{1,\dots,t\} : \mu(a^*) - \bar{\mu}_s \ge \sqrt{\frac{3\ln t}{2s}}\right) \\
&\le \sum_{s=1}^{t} \mathbb{P}\left(\mu(a^*) - \bar{\mu}_s \ge \sqrt{\frac{3\ln t}{2s}}\right) \\
&\le \sum_{s=1}^{t} \frac{1}{t^3} = \frac{1}{t^2},
\end{aligned}$$

where in the first line we decouple $\hat{\mu}_t(a^*)$-s from $N_t(a^*)$-s via the use of $\bar{\mu}_s$-s, and in the last line we apply Hoeffding's inequality (Theorem 3.5). Note that $3\ln t = \ln t^3$ corresponds to $\ln\frac{1}{\delta}$ in Hoeffding's inequality, and thus $\delta = \frac{1}{t^3}$. Finally, we have:

$$\mathbb{E}\left[F(a^*)\right] = \sum_{t=1}^{\infty} \mathbb{P}\left(\mu(a^*) - \hat{\mu}_{t-1}(a^*) \ge \sqrt{\frac{3\ln t}{2N_{t-1}(a^*)}}\right) \le \sum_{t=1}^{\infty} \frac{1}{t^2} = \frac{\pi^2}{6}.$$

□

We refer the reader to Exercise 7.5 for an improved parametrization and a tighter regret bound for the UCB1 algorithm.

7.4 Prediction with Expert Advice

Now we turn our attention to stateless adversarial game with full information feedback.

Notations We are given a $K \times \infty$ matrix of expert losses $\ell_{t,a}$, where $t \in \{1, 2, \dots\}$ and $a \in \{1, \dots, K\}$.

$$\begin{array}{c}
\textit{Expert Losses} \\
\\
\end{array}
\begin{array}{ccccc}
\ell_{1,1}, & \ell_{2,1}, & \cdots & \ell_{t,1}, & \cdots \\
\vdots & \vdots & \cdots & \vdots & \cdots \\
\ell_{1,a}, & \ell_{2,a}, & \cdots & \ell_{t,a}, & \cdots \\
\vdots & \vdots & \cdots & \vdots & \cdots \\
\ell_{1,K}, & \ell_{2,K}, & \cdots & \ell_{t,K}, & \cdots
\end{array}$$

$$\xrightarrow[time]{}$$

Game Definition

For $t = 1, 2, \dots$:

1. Pick a row A_t
2. Observe the column $\ell_{t,1}, \dots, \ell_{t,K}$ & suffer ℓ_{t,A_t}

Performance Measure The performance is measured by *regret*

$$R_T = \sum_{t=1}^{T} \ell_{t,A_t} - \min_a \left(\sum_{t=1}^{T} \ell_{t,a} \right).$$

In the notes we analyze the *expected regret* $\mathbb{E}[R_T]$.

The Hedge Algorithm We consider the Hedge algorithm (a.k.a. exponential weights and weighted majority) for playing this game.

Algorithm 4 Hedge (a.k.a. Exponential Weights), (Vovk, 1990, Littlestone and Warmuth, 1994)

Input: Learning rates $\eta_1 \geq \eta_2 \geq \cdots > 0$
$\forall a : L_0(a) = 0$
for $t = 1, 2, ...$ **do**
  $\forall a : \; p_t(a) = \frac{e^{-\eta_t L_{t-1}(a)}}{\sum_{a'} e^{-\eta_t L_{t-1}(a')}}$
  Sample A_t according to p_t and play it
  Observe $\ell_{t,1}, \ldots, \ell_{t,K}$ and suffer ℓ_{t,A_t}
  $\forall a : \; L_t(a) = L_{t-1}(a) + \ell_{t,a}$
end for

Analysis We analyze the Hedge algorithm in a slightly simplified setting, where the time horizon T is known. Unknown time horizon can be handled by using the doubling trick (see Exercise 7.10) or, much more elegantly, by a more careful analysis (see, e.g., Bubeck and Cesa-Bianchi (2012)).

The analysis is based on the following lemma.

Lemma 7.2. *Let $\{X_{1,a}, X_{2,a}, \ldots\}_{a \in \{1,\ldots,K\}}$ be K sequences of non-negative numbers ($X_{t,a} \geq 0$ for all a and t). Let $L_t(a) = \sum_{s=1}^{t} X_{s,a}$, let $L_0(a)$ be zero for all a, and let $\eta > 0$. Finally, let $p_t(a) = \frac{e^{-\eta L_{t-1}(a)}}{\sum_{a'} e^{-\eta L_{t-1}(a')}}$. Then:*

$$\sum_{t=1}^{T} \sum_{a=1}^{K} p_t(a) X_{t,a} - \min_a L_T(a) \leq \frac{\ln K}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \sum_{a=1}^{K} p_t(a) \left(X_{t,a}\right)^2.$$

Proof. We define $W_t = \sum_a e^{-\eta L_t(a)}$ and study how this quantity evolves. We start with an upper bound.

$$\begin{aligned}
\frac{W_t}{W_{t-1}} &= \frac{\sum_a e^{-\eta L_t(a)}}{\sum_a e^{-\eta L_{t-1}(a)}} \\
&= \frac{\sum_a e^{-\eta X_{t,a}} e^{-\eta L_{t-1}(a)}}{\sum_a e^{-\eta L_{t-1}(a)}} && (7.2) \\
&= \sum_a e^{-\eta X_{t,a}} \frac{e^{-\eta L_{t-1}(a)}}{\sum_{a'} e^{-\eta L_{t-1}(a')}} \\
&= \sum_a e^{-\eta X_{t,a}} p_t(a) && (7.3) \\
&\leq \sum_a \left(1 - \eta X_{t,a} + \frac{1}{2}\eta^2 \left(X_{t,a}\right)^2\right) p_t(a) && (7.4) \\
&= 1 - \eta \sum_a X_{t,a} p_t(a) + \frac{\eta^2}{2} \sum_a \left(X_{t,a}\right)^2 p_t(a) \\
&\leq e^{-\eta \sum_a X_{t,a} p_t(a) + \frac{\eta^2}{2} \sum_a (X_{t,a})^2 p_t(a)}, && (7.5)
\end{aligned}$$

where in (7.2) we used the fact that $L_t(a) = X_{t,a} + L_{t-1}(a)$, in (7.3) we used the definition of $p_t(a)$, in (7.4) we used the inequality $e^x \leq 1 + x + \frac{1}{2}x^2$, which holds for $x \leq 0$ (this is a delicate point, because the

inequality does not hold for $x > 0$ and, therefore, we must check that the condition $x \leq 0$ is satisfied; it is satisfied under the assumptions of the lemma), and inequality (7.5) is based on inequality $1 + x \leq e^x$, which holds for all x.

Now we consider the ratio $\frac{W_T}{W_0}$. On the one hand:

$$\frac{W_T}{W_0} = \frac{W_1}{W_0} \times \frac{W_2}{W_1} \times \cdots \times \frac{W_T}{W_{T-1}} \leq e^{-\eta \sum_{t=1}^{T} \sum_a X_{t,a} p_t(a) + \frac{\eta^2}{2} \sum_{t=1}^{T} \sum_a (X_{t,a})^2 p_t(a)}.$$

On the other hand:

$$\frac{W_T}{W_0} = \frac{\sum_a e^{-\eta L_T(a)}}{K} \geq \frac{\max_a e^{-\eta L_T(a)}}{K} = \frac{e^{-\eta \min_a L_T(a)}}{K},$$

where we lower-bounded the sum by its maximal element. By taking the two inequalities together and applying logarithm we obtain:

$$-\eta \min_a L_T(a) - \ln K \leq -\eta \sum_{t=1}^{T} \sum_a X_{t,a} p_t(a) + \frac{\eta^2}{2} \sum_{t=1}^{T} \sum_a (X_{t,a})^2 p_t(a).$$

Finally, by changing sides and dividing by η we get:

$$\sum_{t=1}^{T} \sum_a X_{t,a} p_t(a) - \min_a L_T(a) \leq \frac{\ln K}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \sum_a (X_{t,a})^2 p_t(a)$$

□

Now we are ready to present an analysis of the Hedge algorithm.

Theorem 7.3. *The expected regret of the Hedge algorithm with a fixed learning rate η satisfies:*

$$\mathbb{E}[R_T] \leq \frac{\ln K}{\eta} + \frac{\eta}{2} T.$$

The expected regret is minimized by $\eta = \sqrt{\frac{2 \ln K}{T}}$, which leads to

$$\mathbb{E}[R_T] \leq \sqrt{2T \ln K}.$$

Proof. We note that $\ell_{t,a}$-s are positive and apply Lemma 7.2 to obtain:

$$\sum_{t=1}^{T} \sum_{a=1}^{K} p_t(a) \ell_{t,a} - \min_a L_T(a) \leq \frac{\ln K}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \sum_{a=1}^{K} p_t(a) (\ell_{t,a})^2.$$

Note that $\sum_a p_t(a) \ell_{t,a}$ is the expected loss of Hedge at round t and $\sum_{t=1}^{T} \sum_{a=1}^{K} p_t(a) \ell_{t,a}$ is the expected cumulative loss of Hedge after T rounds. Thus, the left hand side of the inequality is the expected regret of Hedge. Also note that $\ell_{t,a} \leq 1$, and thus $(\ell_{t,a})^2 \leq 1$ and $\sum_a p_t(a) (\ell_{t,a})^2 \leq 1$. Thus, $\sum_{t=1}^{T} \sum_{a=1}^{K} p_t(a) (\ell_{t,a})^2 \leq T$. Altogether, we get that

$$\mathbb{E}[R_T] \leq \frac{\ln K}{\eta} + \frac{\eta}{2} T.$$

By taking the derivative of the right hand side and equating it to zero we obtain that $-\frac{\ln K}{\eta^2} + \frac{T}{2} = 0$, and thus $\eta = \sqrt{\frac{2 \ln K}{T}}$ is an extreme point. The second derivative is $\frac{2 \ln K}{\eta^3}$, and since $\eta > 0$ it is positive. Thus, the extreme point is the minimum. □

See Exercise 7.8 for a tighter analysis of the Hedge algorithm that matches the lower bound presented in the next section.

7.4.1 Lower Bound

A lower bound for the expected regret in prediction with expert advice is based on the following construction. We draw a $K \times \infty$ matrix of losses with each loss drawn according to a Bernoulli distribution with bias $1/2$. In this game the expected loss of any algorithm after T rounds is $T/2$, irrespective of what the algorithm is doing. However, the loss of the best action in hindsight is lower, because we are selecting the "best" out of K rows. For each individual row the expected loss is $T/2$, but the expectation of the minimum of the losses is lower. The reduction is quantified in the following theorem, see Cesa-Bianchi and Lugosi (2006) for a proof.

Theorem 7.4. *Let $\ell_{t,a}$ be i.i.d. Bernoulli random variables with bias $1/2$, then*

$$\lim_{T\to\infty}\lim_{K\to\infty}\frac{T/2 - \mathbb{E}\left[\min_a \sum_{t=1}^{T} \ell_{t,a}\right]}{\sqrt{\frac{1}{2}T\ln K}} = 1.$$

Note that the numerator in the above expression, $T/2 - \mathbb{E}\left[\min_a \sum_{t=1}^{T} \ell_{t,a}\right]$, is the expectation of the expected regret with respect to generation of the matrix of losses. Thus, if the adversary generates the matrix of losses according to the construction described above, then in expectation with respect to generation of the matrix and in the limit of K and T going to infinity the expected regret cannot be smaller than $\sqrt{\frac{1}{2}T\ln K}$.

The lower bound in Theorem 7.4 matches up to constants the refined upper bound on the expected regret of Hedge provided in Exercise 7.8, which is an extremely rare case. It shows that Hedge is a minimax optimal algorithm for this game.

7.5 Adversarial Multiarmed Bandits

We proceed to stateless adversarial game with bandit feedback.

Game definition We are working with the same matrix of losses as in prediction with expert advice, but now at each round of the game we are allowed to observe only the loss of the row that we have played:
For $t = 1, 2, \ldots$:

1. Pick a row A_t
2. Observe & suffer ℓ_{t,A_t}. ($\ell_{t,a}$-s for $a \neq A_t$ remain unobserved)

Importance-Weighted loss estimates: Emulating full information under bandit feedback In the bandit setting an algorithm only observes the loss ℓ_{t,A_t} of the action A_t played at round t. An elegant way to convert bandit feedback into an imaginary full information feedback is to use importance-weighted loss estimates defined by

$$\tilde{\ell}_{t,a} = \frac{\ell_{t,a}\mathbb{1}(A_t = a)}{p_t(a)} = \begin{cases} \frac{\ell_{t,a}}{p_t(a)}, & \text{if } A_t = a \\ 0, & \text{otherwise} \end{cases},$$

where

$$\mathbb{1}(A_t = a) = \begin{cases} \frac{\ell_{t,a}}{p_t(a)}, & \text{If } A_t = a \\ 0, & \text{otherwise} \end{cases}$$

is the indicator function. Note that $\tilde{\ell}_{t,a}$ is well-defined for all a, even though we do not observe $\ell_{t,a}$ for $a \neq A_t$. As we will show in a moment, it is also an unbiased estimate of $\ell_{t,a}$, namely, $\mathbb{E}\left[\tilde{\ell}_{t,a}\right] = \ell_{t,a}$, although it has a higher variance.

The EXP3 Algorithm The importance-weighted loss estimates can be used as a substitute for the true losses in running the Hedge algorithm. This leads to the EXP3 algorithm proposed by Auer et al. (2002b) and summarized in Algorithm 5 display.[6] Its name, EXP3, stands for EXPonential EXPloration-EXPloitation.

Algorithm 5 EXP3 (Auer et al., 2002b)

Input: Learning rates $\eta_1 \geq \eta_2 \geq \cdots > 0$
$\forall a : \tilde{L}_0(a) = 0$
for $t = 1, 2, \ldots$ **do**
  $\forall a : \; p_t(a) = \frac{e^{-\eta_t \tilde{L}_{t-1}(a)}}{\sum_{a'} e^{-\eta_t \tilde{L}_{t-1}(a')}}$
  Sample A_t according to p_t and play it
  Observe and suffer ℓ_{t,A_t}
  Set $\tilde{\ell}_{t,a} = \frac{\ell_{t,a}\mathbb{1}(A_t=a)}{p_t(a)} = \begin{cases} \frac{\ell_{t,a}}{p_t(a)}, & \text{If } A_t = a \\ 0, & \text{otherwise} \end{cases}$
  $\forall a : \; \tilde{L}_t(a) = \tilde{L}_{t-1}(a) + \tilde{\ell}_{t,a}$
end for

Properties of importance-weighted loss estimates Before analyzing the EXP3 algorithm we discuss a number of important properties of importance-weighted loss estimates.

1. The samples $\tilde{\ell}_{t,a}$ are not independent in two ways. First, for a fixed t, the set $\left\{\tilde{\ell}_{t,1}, \ldots, \tilde{\ell}_{t,K}\right\}$ is dependent (if we know that one of $\tilde{\ell}_{t,a}$-s is non-zero, we automatically know that all the rest are zero). And second, $\tilde{\ell}_{t,a}$ depends on all $\tilde{\ell}_{s,a'}$ for $s < t$ and all a' since $p_t(a)$ depends on $\left\{\tilde{\ell}_{s,a}\right\}_{1 \leq s < t, a \in \{1,\ldots,K\}}$, which is the history of the game up to round t. In other words, $p_t(a)$ itself is a random variable.

2. Even though $\tilde{\ell}_{t,a}$-s are not independent, they are unbiased estimates of the true losses. Specifically,

$$\begin{aligned}
\mathbb{E}\left[\tilde{\ell}_{t,a}\right] &= \mathbb{E}\left[\frac{\ell_{t,a}\mathbb{1}(A_t = a)}{p_t(a)}\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\frac{\ell_{t,a}\mathbb{1}(A_t = a)}{p_t(a)}\middle| A_1, \ldots, A_{t-1}\right]\right] \\
&= \mathbb{E}\left[\frac{\ell_{t,a}}{p_t(a)}\mathbb{E}\left[\mathbb{1}(A_t = a)|A_1, \ldots, A_{t-1}\right]\right] \\
&= \mathbb{E}\left[\frac{\ell_{t,a}}{p_t(a)}p_t(a)\right] \\
&= \ell_{t,a}.
\end{aligned}$$

   The first expectation above is with respect to $A_1, \ldots, A_t$. In the nested expectations, the external expectation is with respect to $A_1, \ldots, A_{t-1}$ and the internal is with respect to A_t. Note that $p_t(a)$ is a random variable depending on $A_1, \ldots, A_{t-1}$, thus after the conditioning on $A_1, \ldots, A_{t-1}$ it is deterministic.

3. Since $\ell_{t,a} \in [0, 1]$, we have $\tilde{\ell}_{t,a} \in \left[0, \frac{1}{p_t(a)}\right]$.

4. What is important is that the second moment of $\tilde{\ell}_{t,a}$-s is by an order of magnitude smaller than the second moment of a general random variable in the corresponding range. This is because the

[6] We note that the original algorithm of Auer et al. (2002b) was formulated for the gains game. Here we present an improved and simplified version of the algorithm for the losses game (Stoltz, 2005, Bubeck, 2010). We refer to Exercise 7.13 for the difference between the two.

expectation of $\tilde{\ell}_{t,a}$-s is in the $[0,1]$ interval. Specifically:

$$\begin{aligned}
\mathbb{E}\left[\left(\tilde{\ell}_{t,a}\right)^2\right] &= \mathbb{E}\left[\left(\frac{\ell_{t,a}\mathbb{1}(A_t=a)}{p_t(a)}\right)^2\right] \\
&= \mathbb{E}\left[\frac{\left(\ell_{t,a}\right)^2\left(\mathbb{1}(A_t=a)\right)^2}{p_t(a)^2}\right] \\
&= \mathbb{E}\left[\frac{\left(\ell_{t,a}\right)^2\mathbb{1}(A_t=a)}{p_t(a)^2}\right] \\
&\leq \mathbb{E}\left[\frac{\mathbb{1}(A_t=a)}{p_t(a)^2}\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\frac{\mathbb{1}(A_t=a)}{p_t(a)^2}\middle| A_1,\ldots,A_{t-1}\right]\right] \\
&= \mathbb{E}\left[\frac{1}{p_t(a)^2}\mathbb{E}\left[\mathbb{1}(A_t=a)|A_1,\ldots,A_{t-1}\right]\right] \\
&= \mathbb{E}\left[\frac{1}{p_t(a)}\right],
\end{aligned}$$

where we have used $\left(\mathbb{1}(A_t=a)\right)^2 = \mathbb{1}(A_t=a)$ and $\left(\ell_{t,a}\right)^2 \leq 1$ (since $\ell_{t,a}\in[0,1]$).

Analysis Now we are ready to present an analysis of the algorithm.

Theorem 7.5. *The expected regret of the EXP3 algorithm with a fixed learning rate η satisfies:*

$$\mathbb{E}\left[R_T\right] \leq \frac{\ln K}{\eta} + \frac{\eta}{2}KT.$$

The expected regret is minimized by $\eta = \sqrt{\frac{2\ln K}{KT}}$, which leads to

$$\mathbb{E}\left[R_T\right] \leq \sqrt{2KT\ln K}.$$

Note that the extra payment for being able to observe just one entry rather than the full column is the multiplicative $\sqrt{K}$ factor in the regret bound.

Proof. The proof of the theorem is based on Lemma 7.2. We note that $\tilde{\ell}_{t,a}$-s are all non-negative and, thus, by Lemma 7.2 we have:

$$\sum_{t=1}^{T}\sum_a p_t(a)\tilde{\ell}_{t,a} - \min_a \tilde{L}_T(a) \leq \frac{\ln K}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\sum_a p_t(a)\left(\tilde{\ell}_{t,a}\right)^2.$$

By taking expectation of the two sides of the inequality we obtain:

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_a p_t(a)\tilde{\ell}_{t,a}\right] - \mathbb{E}\left[\min_a \tilde{L}_T(a)\right] \leq \frac{\ln K}{\eta} + \frac{\eta}{2}\mathbb{E}\left[\sum_{t=1}^{T}\sum_a p_t(a)\left(\tilde{\ell}_{t,a}\right)^2\right].$$

We note that $\mathbb{E}\left[\min\left[\cdot\right]\right] \leq \min\left[\mathbb{E}\left[\cdot\right]\right]$ and thus:

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_a p_t(a)\tilde{\ell}_{t,a}\right] - \min_a \mathbb{E}\left[\tilde{L}_T(a)\right] \leq \frac{\ln K}{\eta} + \frac{\eta}{2}\mathbb{E}\left[\sum_{t=1}^{T}\sum_a p_t(a)\left(\tilde{\ell}_{t,a}\right)^2\right].$$

And now we consider the three expectation terms in this inequality.

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_a p_t(a)\tilde{\ell}_{t,a}\right] = \mathbb{E}\left[\sum_{t=1}^{T}\sum_a \mathbb{E}\left[p_t(a)\tilde{\ell}_{t,a}\middle| A_1,\ldots,A_{t-1}\right]\right] = \mathbb{E}\left[\sum_{t=1}^{T}\sum_a p_t(a)\ell_{t,a}\right],$$

which is the expected loss of EXP3.

$$\mathbb{E}\left[\tilde{L}_T(a)\right] = \mathbb{E}\left[\sum_{t=1}^{T} \tilde{\ell}_{t,a}\right] = \sum_{t=1}^{T} \ell_{t,a},$$

which is the cumulative loss of row a up to time T. And, finally,

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{a} p_t(a)\left(\tilde{\ell}_{t,a}\right)^2\right] = \mathbb{E}\left[\sum_{t=1}^{T}\sum_{a}\mathbb{E}\left[p_t(a)\left(\tilde{\ell}_{t,a}\right)^2 \middle| A_1,\ldots,A_{t-1}\right]\right] \le \mathbb{E}\left[\sum_{t=1}^{T}\sum_{a} p_t(a)\frac{1}{p_t(a)}\right] = KT.$$

This step is known as "the bandit magic". We have shown earlier that the second moment of importance-weighted loss estimates satisfies $\mathbb{E}\left[\left(\tilde{\ell}_{t,a}\right)^2\right] \le \mathbb{E}\left[\frac{1}{p_t(a)}\right]$, which can still be a large number if $p_t(a)$ is small (and it is expected to be small for suboptimal a). However, when the actions are sampled according to $p_t(a)$, the weighted second moment satisfies $\mathbb{E}\left[\sum_a p_t(a)\left(\tilde{\ell}_{t,a}\right)^2\right] \le K$, so it is perfectly under control. Recall that in the full information setting we had $\sum_a p_t(a)\left(\ell_{t,a}\right)^2 \le 1$, and so K is the estimator's variance price that we pay for having limited rather than full information feedback.

Plugging the bounds on the three expectation terms back into the inequality we obtain the first statement of the theorem. And, as before, we find η that minimizes the bound. □

7.5.1 Lower Bound

The lower bound is based on construction of $K+1$ games. In the 0-th game all the losses are sampled from Bernoulli distribution with bias $1/2$. In the i-th game for $i \in \{1,\ldots,K\}$ all the losses are Bernoulli with bias $1/2$ except the losses of the i-th arm, which are Bernoulli with bias $1/2-\varepsilon$ for $\varepsilon = \sqrt{cK/T}$, where c is a properly selected constant. With T/K pulls it is impossible to distinguish between Bernoulli distribution with bias $1/2$ and Bernoulli distribution with bias $1/2-\sqrt{K/T}$, because they induce indistinguishable distributions over sequences of length T/K. As a result, within T pulls the player cannot distinguish between the 0-th game and the i-th games. Therefore, if the adversary picks an i-th game at random, the player's regret will on average (with respect to the adversary's and the players choices) be at least $\Omega(\varepsilon T) = \Omega\left(\sqrt{KT}\right)$. For the details of the proof see Cesa-Bianchi and Lugosi (2006) or Bubeck and Cesa-Bianchi (2012).

Note that there is a gap of $\sqrt{\ln K}$ factor between the lower bound and the upper bound for the EXP3 algorithm. There exists a different algorithm, Tsallis-INF, which closes this gap and achieves $O(\sqrt{KT})$ regret upper bound. The algorithm was proposed by Audibert and Bubeck (2009, 2010) and refined by Zimmert and Seldin (2021).

7.6 Adversarial Multiarmed Bandits with Expert Advice

In this section we move outside the stateless plane and introduce a contextual setting.

Game setting We are, again, working with the same matrix of losses as in prediction with expert advice. But now in every round of the game we get advice of N experts indexed by h in a form of a distribution over the K arms. More formally:
For $t = 1, 2, \ldots$:

1. Observe $q_{t,1},\ldots,q_{t,N}$, where $q_{t,h}$ is a probability distribution over $\{1,\ldots,K\}$.
2. Pick a row A_t.
3. Observe & suffer ℓ_{t,A_t}. ($\ell_{t,a}$-s for $a \neq A_t$ remain unobserved)

Performance measure We compare the expected loss of the algorithm to the expected loss of the best expert in hindsight, where the expectation of the loss of expert h is taken with respect to its advice vector q_h. Specifically:

$$\mathbb{E}\left[R_T\right] = \sum_{t=1}^{T}\sum_{a} p_t(a)\ell_{t,a} - \min_h \sum_{t=1}^{T}\sum_{a} q_{t,h}(a)\ell_{t,a}.$$

The EXP4 Algorithm One approach to playing this game, proposed by Auer et al. (2002b), is quite similar to the EXP3 algorithm.[7] Its name, EXP4, stands for EXPonential EXPloration-EXPloitation with EXPert advice. Note that now $\tilde{L}_t(h)$ tracks cumulative importance-weighted estimates of expert losses, rather than of individual arm losses.

Algorithm 6 EXP4 (Auer et al., 2002b)

Input: Learning rates $\eta_1 \geq \eta_2 \geq \cdots > 0$
$\forall h : \tilde{L}_0(h) = 0$
for $t = 1, 2, \ldots$ **do**
  $\forall h$: $w_t(h) = \frac{e^{-\eta_t \tilde{L}_{t-1}(h)}}{\sum_{h'} e^{-\eta_t \tilde{L}_{t-1}(h')}}$
  Observe $q_{t,1}, \ldots, q_{t,N}$
  $\forall a$: $p_t(a) = \sum_h w_t(h) q_{t,h}(a)$
  Sample A_t according to p_t and play it
  Observe and suffer ℓ_{t,A_t}
  Set $\tilde{\ell}_{t,a} = \frac{\ell_{t,a}\mathbb{1}(A_t=a)}{p_t(a)} = \begin{cases} \frac{\ell_{t,a}}{p_t(a)}, & \text{if } A_t = a \\ 0, & \text{otherwise} \end{cases}$
  Set $\tilde{\ell}_{t,h} = \sum_a q_t(h)\tilde{\ell}_{t,a}$
  $\forall h$: $\tilde{L}_t(h) = \tilde{L}_{t-1}(h) + \tilde{\ell}_{t,h}$
end for

Analysis The EXP4 algorithm satisfies the following regret guarantee.

Theorem 7.6. *The expected regret of the EXP4 algorithm with a fixed learning rate η satisfies:*

$$\mathbb{E}\left[R_T\right] \leq \frac{\ln N}{\eta} + \frac{\eta}{2}KT.$$

The expected regret is minimized by $\eta = \sqrt{\frac{2\ln N}{KT}}$, which leads to

$$\mathbb{E}\left[R_T\right] \leq \sqrt{2KT\ln N}.$$

Note that the $\ln N$ term plays the role of complexity of the class of experts in a very similar way to the complexity terms we saw earlier in supervised learning (specifically, in Theorem 4.2).

Proof. The analysis is quite similar to the analysis of the EXP3 algorithm. We note that $\tilde{\ell}_{t,h}$-s are all non-negative and that w_t is a distribution over $\{1, \ldots, N\}$ defined in the same way as p_t in Lemma 7.2. Thus, by Lemma 7.2 we have:

$$\sum_{t=1}^{T}\sum_{h} w_t(h)\tilde{\ell}_{t,h} - \min_h \tilde{L}_T(h) \leq \frac{\ln N}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\sum_{h} w_t(h)\left(\tilde{\ell}_{t,h}\right)^2.$$

By taking expectations of the two sides of this expression we obtain:

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{h} w_t(h)\tilde{\ell}_{t,h}\right] - \mathbb{E}\left[\min_h \tilde{L}_T(h)\right] \leq \frac{\ln N}{\eta} + \frac{\eta}{2}\mathbb{E}\left[\sum_{t=1}^{T}\sum_{h} w_t(h)\left(\tilde{\ell}_{t,h}\right)^2\right].$$

[7] As with the EXP3 algorithm, we present a slightly improved version of the algorithm for the game with losses. The original algorithm was designed for the game with rewards.

As before, $\mathbb{E}[\min[\cdot]] \leq \min[\mathbb{E}[\cdot]]$ and thus:

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{h} w_t(h)\tilde{\ell}_{t,h}\right] - \min_h \mathbb{E}\left[\tilde{L}_T(h)\right] \leq \frac{\ln N}{\eta} + \frac{\eta}{2}\mathbb{E}\left[\sum_{t=1}^{T}\sum_{h} w_t(h)\left(\tilde{\ell}_{t,h}\right)^2\right].$$

And now we consider the three expectation terms in this inequality.

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T}\sum_{h} w_t(h)\tilde{\ell}_{t,h}\right] &= \mathbb{E}\left[\sum_{t=1}^{T}\sum_{h} w_t(h)\sum_{a} q_{t,h}(a)\tilde{\ell}_{t,a}\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T}\sum_{a}\left(\sum_{h} w_t(h)q_{t,h}(a)\right)\tilde{\ell}_{t,a}\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T}\sum_{a} p_t(a)\tilde{\ell}_{t,a}\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T}\sum_{a} p_t(a)\ell_{t,a}\right],
\end{aligned}$$

where the first equality is by the definition of $\tilde{\ell}_{t,h}$ and the last equality is due to unbiasedness of $\tilde{\ell}_{t,a}$. Thus, the first expectation is the expected loss of EXP4.

$$\mathbb{E}\left[\tilde{L}_T(h)\right] = \mathbb{E}\left[\sum_{t=1}^{T}\tilde{\ell}_{t,h}\right] = \mathbb{E}\left[\sum_{t=1}^{T}\sum_{a} q_t(a)\tilde{\ell}_{t,a}\right] = \mathbb{E}\left[\sum_{t=1}^{T}\sum_{a} q_t(a)\ell_{t,a}\right],$$

where we can remove tilde due to unbiasedness of $\tilde{\ell}_{t,a}$, and we obtain the expected cumulative loss of expert h over T rounds. And, finally,

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T}\sum_{h} w_t(h)\left(\tilde{\ell}_{t,h}\right)^2\right] &= \mathbb{E}\left[\sum_{t=1}^{T}\sum_{h} w_t(h)\left(\sum_{a} q_{t,h}(a)\tilde{\ell}_{t,a}\right)^2\right] \\
&\leq \mathbb{E}\left[\sum_{t=1}^{T}\sum_{h} w_t(h)\sum_{a} q_{t,h}(a)\left(\tilde{\ell}_{t,a}\right)^2\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T}\sum_{a}\left(\sum_{h} w_t(h)q_{t,h}(a)\right)\left(\tilde{\ell}_{t,a}\right)^2\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T}\sum_{a} p_t(a)\left(\tilde{\ell}_{t,a}\right)^2\right] \\
&\leq KT,
\end{aligned}$$

where the first inequality is by Jensen's inequality (Theorem B.30) and convexity of x^2, and the last inequality is along the same lines as the analogous inequality in the analysis of EXP3. By substituting the three expectations back into the inequality we obtain the first statement of the theorem. And, as before, we find η that minimizes the bound. □

7.6.1 Lower Bound

It is relatively easy to show that the regret of adversarial multiarmed bandits with expert advice must be at least $\Omega\left(\sqrt{KT\frac{\ln N}{\ln K}}\right)$. The lower bound is based on construction of $\frac{\ln N}{\ln K}$ independent bandit problems, each according to the construction of the lower bound for multiarmed bandits in Section 7.5.1, and construction of expert advice, so that for every possible selection of best arms for the subproblems there is an expert that recommends that selection. For details of the proof see Agarwal et al. (2012), Seldin and Lugosi (2016).

With a bit more work it is possible to derive a tight lower bound of $\Omega\left(\sqrt{KT \ln \frac{N}{K}}\right)$ (Chase et al., 2025). And by replacing EXP3-style approach with Tsallis-INF-style techniques it is possible to derive an algorithm with matching $O\left(\sqrt{KT \ln \frac{N}{K}}\right)$ regret upper bound (Kale, 2014).

7.7 Exercises

Exercise 7.1 (*Find an online learning problem from real life*)**.** Find two examples of real life problems that fit into the online learning framework (online, not reinforcement!). For each of the two examples explain what is the set of actions an algorithm can take, what are the losses (or rewards) and what is the range of the losses/rewards, whether the problem is stateless or contextual, and whether the problem is i.i.d. or adversarial, and with full information or bandit feedback.

Exercise 7.2 (*Follow The Leader (FTL) algorithm for i.i.d. full information games*)**.** Follow the leader (FTL) is a playing strategy that at round t plays the action that was most successful up to round t ("the leader"). Derive a bound for the pseudo regret of FTL in i.i.d. full information games with K possible actions and outcomes bounded in the $[0, 1]$ interval (you can work with rewards or losses, as you like). You can use the following guidelines (which assume a game with rewards):

1. You are allowed to solve the problem for $K = 2$. (The guidelines are not limited to $K = 2$.)
2. It may be helpful to write the algorithm down explicitly. For the analysis it does not matter how you decide to break ties.
3. Let $\mu(a)$ be expected reward of action a and let $\hat{\mu}_t(a)$ be empirical estimate of the reward of action a at round t (the average of rewards observed so far). Let a^* be an optimal action (there may be more than one optimal action, but then things only get better [convince yourself that this is true], so we can assume that there is a single a^*). Let $\Delta(a) = \mu(a^*) - \mu(a)$. FTL may play $a \neq a^*$ at rounds t for which $\hat{\mu}_{t-1}(a) \geq \max_{a'} \hat{\mu}_{t-1}(a')$ (in the case of two arms it means $\hat{\mu}_{t-1}(a) \geq \hat{\mu}_{t-1}(a^*)$). So you should analyze how often this may happen.
4. Note that the number of times an action a was played can be written as $N_T(a) = \sum_{t=1}^{T} \mathbb{1}(A_t = a)$, and that $\mathbb{E}[\mathbb{1}(A_t = a)] \leq \mathbb{P}(\hat{\mu}_{t-1}(a) \geq \max_{a'} \hat{\mu}_{t-1}(a')) \leq \mathbb{P}(\hat{\mu}_{t-1}(a) \geq \hat{\mu}_{t-1}(a^*))$, where $\mathbb{1}$ is the indicator function.
5. Bound $\mathbb{P}(\hat{\mu}_{t-1}(a) \geq \hat{\mu}_{t-1}(a^*))$.
6. At some point in the proof you will need to sum up a geometric series. A geometric series is a series of a form $\sum_{t=0}^{\infty} r^t$, and for $r < 1$ we have $\sum_{t=0}^{\infty} r^t = \frac{1}{1-r}$. In your case r will be an exponent $r = e^{\alpha}$ for some constant α.
7. At the end you should get a bound of a form $\bar{R}_T \leq \sum_{a:\Delta(a)>0} \frac{c}{1-\exp(-\Delta(a)^2/2)} \Delta(a)$, where c is a constant.

Important observations to make:

1. *Note that in the full information i.i.d. setting the regret does not grow with time!!! (Since the bound is independent of T.)*
2. *Note that even though you have used $\Delta(a)$ in the analysis of the algorithm, you do not need to know it in order to define the algorithm! I.e., you can run the algorithm even if you do not know $\Delta(a)$.*

Exercise 7.3 (*Decoupling exploration and exploitation in i.i.d. multiarmed bandits*)**.** Assume an i.i.d. multiarmed bandit game, where the observations are not coupled to the actions. Specifically, we assume that at each round of the game the player is allowed to observe the reward of a single arm, but it does not have to be the same arm that was played at that round (and if it is not the same arm, the player does not observe its own reward, but instead observes the reward of the alternative option).

Derive a playing strategy and a regret bound for this game. (You should solve the problem for a general K and you should get that the regret does not grow with time.)

Remark: note that in this setting the exploration is "for free", because we do not have to play suboptimal actions in order to test their quality. And if we contrast this with the standard multiarmed bandit setting we observe that the regret stops growing with time instead of growing logarithmically with time. Actually, the result that you should get is much closer to the regret bound for FTL with full information than to the regret bound for multiarmed bandits. Thus, it is not the fact that we have just a single observation that makes i.i.d. multiarmed bandits harder than full information games, but the fact that this single observation is linked to the action. In adversarial multiarmed bandits the effect of decoupling is more involved (Avner et al., 2012, Seldin et al., 2014, Rouyer and Seldin, 2020).

Exercise 7.4 (*The worst case gap for a fixed T*)**.** "Exploration-exploitation trade-off: A simple approach" in Section 7.3 shows that problems with small gap Δ are harder (have higher pdeudo-regret) than problems with large gap Δ. This is true if the time horizon is unlimited, but for a limited time horizon there is actually a limit on how large the pseudo-regret can be.

1. Argue why in the two-arms case the regret never exceeds ΔT.
2. Show that the worst-case gap (the gap that leads to the highest pseudo-regret) is of order $\Delta = \theta\left(\sqrt{\frac{\ln T}{T}}\right)$.
3. Show that the regret satisfies $\bar{R}_T \leq \theta\left(\sqrt{T \ln T}\right)$

Conclusion: Indeed, as Δ decreases the regret increases, but if the time horizon is fixed T, it only increases up to a limit of order $\sqrt{T \ln T}$, and thereafter starts decreasing. Bounds depending on Δ (as in Section 7.3) are known as instance-dependent bounds *(they depend on the instance of a problem with gap Δ), whereas bounds that are independent of Δ and only depend on T (as the one you derived above) are known as* worst-case bounds.

Exercise 7.5 (*Improved Parametrization of UCB1*)**.** In this question we refine the upper confidence bound for the UCB1 algorithm from Section 7.3 and derive a tighter regret bound.

1. Show that if we replace the upper confidence bounds in UCB1 with

$$U_t(a) = \hat{\mu}_{t-1}(a) + \sqrt{\frac{\ln t}{N_{t-1}(a)}}$$

   then its pseudo-regret satisfies

$$\bar{R}_T \leq 4 \sum_{a:\Delta(a)>0} \frac{\ln T}{\Delta(a)} + (2\ln(T) + 3) \sum_a \Delta(a).$$

   Hint: The T-th harmonic number, $\sum_{t=1}^{T} \frac{1}{t}$, satisfies $\sum_{t=1}^{T} \frac{1}{t} \leq \ln(T) + 1$.

2. Write a simulation to compare numerically the performance of UCB1 from Section 7.3 with performance of UCB1 with modified confidence bounds proposed above. Instructions for the simulation:
   - Generate Bernoulli rewards for two actions, a^* and a, so that $\mathbb{E}[r_{t,a^*}] = \frac{1}{2} + \frac{1}{2}\Delta$ and $\mathbb{E}[r_{t,a}] = \frac{1}{2} - \frac{1}{2}\Delta$. (The rewards may be generated dynamically as you run the algorithms and, actually, you only need them for the actions that are played by the algorithms.)
   - Run the experiment with $\Delta = \frac{1}{4}$, $\Delta = \frac{1}{8}$, and $\Delta = \frac{1}{16}$. (Three different experiments.)
   - Take $T = 100000$. (In general, the time horizon should be large in relation to $\frac{1}{\Delta^2}$.)
   - Plot the empirical pseudo regret defined by $\hat{R}_t = \sum_{s=1}^{t} \Delta(A_s)$ for the two algorithms as a function of time for $1 \leq t \leq T$. (To remind you: A_s is the action taken by the algorithm in round s and $\Delta(a) = \max_{a'} \mathbb{E}[r_{s,a'}] - \mathbb{E}[r_{s,a}] = \mu(a^*) - \mu(a)$.) To make the plot you should make 20 runs of each algorithm and plot the average pseudo regret over the 20 runs and the average pseudo regret + one standard deviation over the 20 runs. Do not forget to add a legend to your plot.

- Answer the following questions:
  - Which values of Δ lead to higher regret?
  - What can you say about the relative performance of the two parametrizations?

Comment: The UCB1 algorithm in Section 7.3 takes confidence intervals $\sqrt{\frac{3 \ln t}{2N_{t-1}(a)}} = \sqrt{\frac{\ln t^3}{2N_{t-1}(a)}}$*, corresponding to confidence parameter* $\delta = \frac{1}{t^3}$*. The modified UCB1 algorithm in this question takes confidence intervals* $\sqrt{\frac{2 \ln t}{2N_{t-1}(a)}} = \sqrt{\frac{\ln t^2}{2N_{t-1}(a)}}$*, corresponding to confidence parameter* $\delta = \frac{1}{t^2}$*. The original algorithm and analysis by Auer et al. (2002a) uses confidence intervals* $\sqrt{\frac{4 \ln t}{2N_{t-1}(a)}} = \sqrt{\frac{\ln t^4}{2N_{t-1}(a)}}$*, corresponding to confidence parameter* $\delta = \frac{1}{t^4}$*, due to one unnecessary union bound. The reason we can move from* $\delta = \frac{1}{t^3}$ *to* $\delta = \frac{1}{\delta^2}$ *is not due to elimination of additional union bounds (we still need two of them), but due to a compromise on the last term in the regret bound.*

Exercise 7.6 (*Introduction of New Products*)**.** Imagine that we have an established product on the market, which sells with probability 0.5. We have received a new product, which sells with an unknown probability μ. Assume that at every sales round we can offer only one product, so we have to choose between offering the established or the new product. Propose a strategy for maximizing the number of sales and analyze its pseudo-regret. Write your answer in terms of the gap $\Delta = 0.5 - \mu$. (The regret bound for $\Delta > 0$ is different from the regret bound for $\Delta < 0$, i.e., you should obtain two separate answers for positive and negative gap.)

Hint: The solution is *not* an application of an existing algorithm. You should design a *new* algorithm tailored for the problem. The new algorithm will not be very different from something we have studied, but you have to make a small adaptation; this is the whole point of the question.

Pay attention that if $\mu > 0.5$ (the new product is better than the old one) then $\Delta < 0$, and if $\mu < 0.5$ (the new product is worse than the old one) then $\Delta > 0$. If you do things correctly, for $\Delta < 0$ the regret of your algorithm should be bounded by a constant that is independent of the number of game rounds T, and for $\Delta > 0$ it should grow logarithmically with T. The algorithm should not know whether Δ is positive or negative, but the analysis of the two cases can be done separately.

Exercise 7.7 (*Empirical evaluation of algorithms for adversarial environments*)**.** Is it possible to evaluate experimentally the quality of algorithms for adversarial environments? If yes, how would you design such an experiment? If no, explain why it is not possible.

Hint: Think what kind of experiments can certify that an algorithm for an adversarial environment is good and what kind of experiments can certify that the algorithm is bad? How easy or hard is it to construct the corresponding experiments?

Exercise 7.8 (*A tighter analysis of the Hedge algorithm*)**.**

1. Apply Hoeffding's lemma (Theorem 3.6) in order to derive a better parametrization and a tighter bound for the expected regret of the Hedge algorithm (Algorithm 4 in Section 7.4). [Do not confuse Hoeffding's lemma (Theorem 3.6) with Hoeffding's inequality (Theorem 3.3)!] Guidance:

   (a) Traverse the analysis of the Hedge algorithm that we did in class. There will be a place where you will have to bound expectation of an exponent of a function ($\sum_a p_t(a) e^{-\eta X_{t,a}}$). Instead of going the way we did, apply Hoeffding's lemma.

   (b) Find the value of η that minimizes the new bound. (You should get $\eta = \sqrt{\frac{8 \ln K}{T}}$ - please, prove this formally.)

   (c) At the end you should obtain $\mathbb{E}[R_T] \leq \sqrt{\frac{1}{2} T \ln K}$. (I.e., you will get an improvement by a factor of 2 compared to what we did in class.)

   Remark: Note that the regret upper bound matches the lower bound up to the constants. This is an extremely rare case.

2. Explain why the same approach cannot be used to tighten the regret bound of the EXP3 algorithm.

Exercise 7.9 (*Empirical comparison of FTL and Hedge*)**.** Assume that you have to predict a binary sequence $X_1, X_2, \ldots$ and that you know that X_i-s are i.i.d. Bernoulli random variables with an unknown bias μ. (You know that X_i-s are i.i.d., but you do not know the value of μ.) At every round you can predict "0" or "1" (i.e., you have two actions - "predict 0" or "predict 1") and your loss is the zero-one loss depending on whether your prediction matches the outcome. The regret is computed with respect to the performance of the best out of the two possible actions.

1. Write a simulation that compares numerically the performance of Follow The Leader (FTL) algorithm with performance of the Hedge algorithm (Algorithm 4 in Section 7.4) with $\eta = \sqrt{\frac{2 \ln K}{T}}$, and performance of the reparametrized Hedge algorithm from Exercise 7.8, with $\eta = \sqrt{\frac{8 \ln K}{T}}$. The Hedge algorithm should operate with the aforementioned two actions. To make things more interesting we will add an "anytime" version of Hedge to the comparison. "Anytime" algorithm is an algorithm that does not depend on the time horizon. Let t be a running time index ($t = 1, \ldots, T$). Anytime Hedge corresponding to the simple analysis uses $\eta_t = \sqrt{\frac{\ln K}{t}}$ and anytime Hedge corresponding to the tighter analysis in Exercise 7.8 uses $\eta_t = 2\sqrt{\frac{\ln K}{t}}$ (the learning rate η_t of anytime Hedge changes with time and does not depend on the time horizon). Some instructions for the simulation:
   - Take time horizon $T = 2000$. (In general, the time horizon should be large in comparison to $\frac{1}{(\mu - \frac{1}{2})^2}$.)
   - Test several values of μ. We suggest $\mu = \frac{1}{2} - \frac{1}{4}$, $\mu = \frac{1}{2} - \frac{1}{8}$, $\mu = \frac{1}{2} - \frac{1}{16}$.
   - Plot the empirical pseudo regret defined by $\hat{R}_t = \sum_{s=1}^{t} \Delta(A_s)$ of the five algorithms with respect to the best out of "0" and "1" actions as a function of t for $1 \leq t \leq T$ and for the different values of μ (make a separate plot for each μ). Make 10 runs of each algorithm and report the average empirical pseudo regret over the 10 runs and the average empirical pseudo regret + one standard deviation over the 10 runs. (Generate a new sequence $X_1, X_2, \ldots$ for each run of the algorithm.) Do not forget to add a legend to your plot.
2. Which values of μ lead to higher regret? How the relative performance of the algorithms evolves with time and does it depend on μ?
3. Design an adversarial (non-i.i.d.) sequence, which makes the FTL algorithm perform poorly. Ideally, your solution should not depend on the tie breaking approach of FTL. (If you need to make assumptions about tie breaking, please, state them clearly. We may take a few points, because it would make the problem easier.) Explain the design of your adversarial sequence and report a plot with a simulation, where you compare the performance of FTL with the different versions of Hedge. As before, make 10 repetitions of the experiment and report the average regret (in this case you should use regret and not pseudo regret) and the average + one standard deviation. Comment on your observations.

Exercise 7.10 (*The doubling trick*)**.** Consider the following forecasting strategy (the "doubling trick"): time is divided into periods $(2^m, \ldots, 2^{m+1} - 1)$, where $m = 0, 1, 2, \ldots$. (In other words, the periods are $(1), (2, 3), (4, \ldots, 7), (8, \ldots, 15), \ldots$.) In period $(2^m, \ldots, 2^{m+1} - 1)$ the strategy uses the optimized Hedge forecaster (from the previous question) initialized at time 2^m with parameter $\eta_m = \sqrt{\frac{8 \ln K}{2^m}}$. Thus, the Hedge forecaster is reset at each time instance that is an integer power of 2 and restarted with a new value of η. By the analysis of optimized Hedge we know that with $\eta_m = \sqrt{\frac{8 \ln K}{2^m}}$ its expected regret within the period $(2^m, \ldots, 2^{m+1} - 1)$ is bounded by $\sqrt{\frac{1}{2} 2^m \ln K}$.

1. Prove that for any $T = 2^m - 1$ the overall expected regret (considering the time period $(1, \ldots, T)$) of this forecasting strategy satisfies
$$\mathbb{E}[R_T] \leq \frac{1}{\sqrt{2} - 1} \sqrt{\frac{1}{2} T \ln K}.$$
(Hint: at some point in the proof you will have to sum up a geometric series.)

2. Prove that for any arbitrary time T the expected regret of this forecasting strategy satisfies

$$\mathbb{E}[R_T] \leq \frac{\sqrt{2}}{\sqrt{2}-1}\sqrt{\frac{1}{2}T\ln K}.$$

Remark: The expected regret of "anytime" Hedge with $\eta_t = 2\sqrt{\frac{\ln K}{t}}$ satisfies $\mathbb{E}[R_T] \leq \sqrt{T\ln K}$ for any T. For comparison, $\frac{1}{\sqrt{2}(\sqrt{2}-1)} \approx 1.7$ and $\frac{1}{\sqrt{2}-1} \approx 2.4$. Thus, anytime Hedge is both more elegant and more efficient than Hedge with the doubling trick.

Exercise 7.11 (*Regularization by relative entropy and the Gibbs distribution*)**.** In this question we will show that regularization by relative entropy leads to solutions in a form of the Gibbs distribution. Let's assume that we have a finite hypothesis class $\mathcal{H}$ of size m and we want to minimize

$$\mathcal{F}(\rho) = \alpha\mathbb{E}_\rho\left[\hat{L}(h,S)\right] + \mathrm{KL}(\rho\|\pi) = \alpha\sum_{h=1}^{m}\rho(h)\hat{L}(h,S) + \sum_{h=1}^{m}\rho(h)\ln\frac{\rho(h)}{\pi(h)}$$

with respect to the distribution ρ. This objective is closely related to the objective of PAC-Bayes-λ inequality when λ is fixed and this sort of minimization problem appears in many other places in machine learning. Let's slightly simplify and formalize the problem. Let $\rho = (\rho_1,\ldots,\rho_m)$ be the posterior distribution, $\pi = (\pi_1,\ldots,\pi_m)$ the prior distribution, and $L = (L_1,\ldots,L_m)$ the vector of losses. You should solve

$$\begin{aligned} \min_{\rho_1,\ldots,\rho_m} \quad & \alpha\sum_{h=1}^{m}\rho_h L_h + \sum_{h=1}^{m}\rho_h\ln\frac{\rho_h}{\pi_h} \\ s.t. \quad & \sum_{h=1}^{m}\rho_h = 1 \\ & \forall h: \rho_h \geq 0 \end{aligned} \tag{7.6}$$

and show that the solution is $\rho_h = \frac{\pi_h e^{-\alpha L_h}}{\sum_{h'=1}^{m}\pi_{h'}e^{-\alpha L_{h'}}}$. Distribution of this form is known as the Gibbs distribution.

Guidelines:

1. Take a shortcut. Instead of solving minimization problem (7.6), solve the following minimization problem

$$\begin{aligned} \min_{\rho_1,\ldots,\rho_m} \quad & \alpha\sum_{h=1}^{m}\rho_h L_h + \sum_{h=1}^{m}\rho_h\ln\frac{\rho_h}{\pi_h} \\ s.t. \quad & \sum_{h=1}^{m}\rho_h = 1, \end{aligned} \tag{7.7}$$

   i.e., drop the last constraint in (7.6).

2. Use the method of Lagrange multipliers to show that the solution of the above problem has a form of $\rho_h = \pi_h e^{-\alpha L_h + \texttt{something}}$, where `something` is something involving the Lagrange multiplier.

3. Show that $\rho_h \geq 0$ for all h. (This is trivial. But it gives us that the solutions of (7.6) and (7.7) are identical.)

4. Finally, $e^{\texttt{something}}$ should be such that the constraint $\sum_{h=1}^{m}\rho_h = 1$ is satisfied. So you can easily get the solution. You even do not have to compute the Lagrange multiplier explicitly.

Exercise 7.12 (*Empirical comparison of UCB1 and EXP3 algorithms*)**.** Implement and compare the performance of UCB1 with improved parametrization derived in Exercise 7.5 and EXP3 in the i.i.d. multiarmed bandit setting. For EXP3 take time-varying $\eta_t = \sqrt{\frac{\ln K}{tK}}$.

1. Use the following settings:

   - Time horizon $T = 10000$.
   - Take a single best arm with Bernoulli distribution with bias $\mu^* = 0.5$.
   - Take $K - 1$ suboptimal arms for $K = 2, 4, 8, 16$.
   - For suboptimal arms take Bernoulli distributions with bias $\mu = \mu^* - \frac{1}{4}$, $\mu = \mu^* - \frac{1}{8}$, $\mu = \mu^* - \frac{1}{16}$ (in total 12 different experiments corresponding to the four values of K and three values of μ, with all suboptimal arms in each experiment sharing the same μ).

   Make 20 repetitions of each experiment and for each experiment plot the average empirical pseudo regret (over the 20 repetitions) as a function of time and the average empirical pseudo regret + one standard deviation (over the 20 repetitions). The empirical pseudo regret is defined as $\hat{R}_T = \sum_{t=1}^{T} \Delta(A_t)$.

   Important: do not forget to add a legend and labels to the axes.

2. ***Break UCB1*** Now design an adversarial sequence for which you expect UCB1 to suffer linear regret. The sequence should be oblivious (i.e., it should be independent of the actions taken by UCB1). Ideally, your solution should not depend on the tie breaking approach of UCB1. Explain how you design the sequence, and execute UCB1 and EXP3 on it and report the observations (in a form of a plot). You should make several repetitions of the experiment (say, 20), because there is internal randomness in the algorithms.

   Hint: there will always be a tie in the very first round (i.e., in the initial "play each action once" phase, if UCB1 starts in random order), so you need to find a way how to start. After that ties are unlikely, and you can avoid them altogether by using rewards in $[0, 1]$ *rather than* $\{0, 1\}$*. You need to explain how.*

You are welcome to try other settings (not for submission). For example, check what happens when μ^* is close to 1 or 0. Or what happens when the best arm is not static, but switches between rounds. Even though you can break UCB1, it is actually pretty robust, unless you design adversarial sequences that exploit the knowledge of the algorithm.

Exercise 7.13 (*Rewards vs. losses*)**.** The original EXP3 algorithm for multiarmed bandits was designed for the game with rewards rather than losses (see Auer et al. (2002b, Page 6)). In the game with rewards we have an infinite matrix of rewards $\{r_t^a\}_{a \in \{1,\dots,K\}, t \geq 1}$, where $r_t^a \in [0, 1]$. On each round of the game the algorithm plays an action A_t and accumulates and observed reward $r_t^{A_t}$. The remaining rewards r_t^a for $a \neq A_t$ remain unobserved.

On the one hand, we can easily convert rewards into losses by taking $\ell_t^a = 1 - r_t^a$ and apply the EXP3 algorithm we saw in class. On the other hand, a bit surprisingly, working directly with rewards (as Auer et al. did) turns to be more cumbersome and less efficient. The high-level reason is that the games with rewards and losses have a different dynamics. In the rewards game when an action is played its relative quality (expressed by the cumulative reward) increases. Therefore, we need explicit exploration to make sure that we do not get locked on a suboptimal action. In the losses game when an action is played its relative quality (expressed by the cumulative loss) decreases. Therefore, we never get locked on any particular action and exploration happens automatically without the need to add it explicitly (sometimes this is called implicit exploration). The low-level reason when it comes down to the analysis of the algorithm is that it is easier to upper bound the exponent of x for negative x as opposed to positive x.

The original EXP3 algorithm for the rewards game looks as follows, where the most important difference with the algorithm for the losses game is highlighted in red and two additional minor differences are highlighted in blue (we explicitly emphasize that the sign in the exponent changes from "-" to "+"). $\tilde{R}_t$ is used to denote cumulative importance-weighted rewards.

1. Explain why the analysis of the EXP3 algorithm for the rewards game without the addition of explicit exploration $\frac{\eta}{K}$ in Line 3 of the algorithm would not work. More specifically - if you would try to follow the lines of the analysis of EXP3 with losses, at which specific point you would get stuck and why?

Algorithm 7 The EXP3 Algorithm for the game with rewards and fixed time horizon

1: $\forall a : \tilde{R}_0(a) = 0$
2: **for** $t = 1, \ldots, T$ **do**
3: $\quad \forall a : \; p_t(a) = (1-\eta)\frac{e^{+\eta\tilde{R}_{t-1}(a)}}{\sum_{a'} e^{+\eta\tilde{R}_{t-1}(a')}} + \frac{\eta}{K}$
4: $\quad$ Sample A_t according to p_t and play it
5: $\quad$ Observe r_{t,A_t}
6: $\quad \forall a : \; \tilde{r}_{t,a} = \frac{r_{t,a}\mathbb{1}(A_t=a)}{p_t(a)} = \begin{cases} \frac{r_{t,a}}{p_t(a)}, & \text{if } A_t = a \\ 0, & \text{otherwise} \end{cases}$
7: $\quad \forall a : \; \tilde{R}_t(a) = \tilde{R}_{t-1}(a) + \tilde{r}_{t,a}$
8: **end for**

2. How the addition of explicit exploration term $\frac{\eta}{K}$ in Line 3 of the algorithm allows the analysis to go through? (You can check the analysis of the algorithm in Auer et al. (2002b, Page 7).)

3. By how much the expected regret guarantee for EXP3 with rewards is weaker than the expected regret guarantee for EXP3 with losses? (Check Auer et al. (2002b, Corollary 3.2) and assume that g takes its worst-case value, which is T.)

4. You are mostly welcome to experiment and see whether theoretical analysis reflects the performance in practice. I.i.d. experiments will be the easiest to construct, but you are also welcome to try adversarial settings.

Exercise 7.14 (*Offline Evaluation of Bandit Algorithms*)**.**

1. Evaluation of algorithms for online learning with limited feedback in real life (as opposed to simulations) is a challenging topic. The straightforward way is to implement an algorithm, execute it "live", and measure the performance, but most often this is undesirable. Give two reasons why.

2. There are two alternative offline evaluation methods: importance-sampling and rejection sampling. In both cases we have to know the distribution that was used for collecting the data. Note that we only observe the reward (or loss) for an action taken by the algorithm when the action matches the action of the logging policy. In the importance-sampling approach we reweigh the reward by inverse probability of the action being taken by the logging policy when there is a match and assign zero reward otherwise. The rejection sampling approach requires that the logging policy samples all actions uniformly at random. At the evaluation phase rejection sampling scrolls through the log of events until the first case where the action of the logging policy matches the action of the evaluated policy. The corresponding reward is assigned and all events that were scrolled over are discarded. You can read more about the rejection sampling approach in Li et al. (2011). The importance-sampling approach is more versatile, because it does not require a uniform logging policy. With importance-sampling it is possible to take data collected by an existing policy and evaluate new policies, as long as the logging distribution is known and strictly positive for all actions.

   The Theoretical Part of the Task Our theoretical aim is to modify the UCB1 and EXP3 algorithms to be able to apply them to logged data using the importance-sampling approach. For simplicity, we assume that the logging policy used uniform sampling. Pay attention that importance weighting in offline evaluation based on uniform sampling (the one you are asked to analyze) changes the range of the rewards from $[0, 1]$ to $[0, K]$, where K is the number of actions. Recall that the original versions of UCB1 and EXP3 assumed that the rewards are bounded in the $[0, 1]$ interval. Your task is to modify the two algorithms accordingly. Pay attention that the variance of the importance weighted estimates is "small" (of order K rather than of order K^2) and if you do the analysis carefully, you should be able to exploit it in the modified EXP3, but not in UCB1.

   (a) Modify the UCB1 algorithm with improved parametrization from Exercise 7.5 to work with importance-weighted losses generated by a logging policy based on uniform sampling. Provide

a pseudo-code of the modified algorithm (at the same level as the UCB1 pseudo-code in Algorithm 3) and all the necessary calculations supporting your modification, including a pseudo-regret bound. You do not need to redo the full derivation, it is sufficient to highlight the key points where you make changes and how they affect the regret bound, assuming you do it clearly.

(b) Briefly explain why you are unable to exploit the small variance in the modified UCB1.

(c) Modify the EXP3 algorithm with a fixed learning rate η and known time horizon T to work with importance-weighted losses generated by a logging policy based on uniform sampling. Provide a pseudo-code of the modified algorithm (at the same level as the EXP3 pseudo-code in Algorithm 4) and all the necessary calculations supporting your modification, including an expected regret bound. As already mentioned, with a careful analysis you should be able to exploit the small variance of importance-weighted losses.

(d) Anytime modification: In order to transform the fixed-horizon EXP3 from the previous task to an anytime EXP3 (an EXP3 that assumes no knowledge of the time horizon) you should replace the time horizon T in the learning rate by the running time t and reduce the learning rate by a factor of $\sqrt{2}$. The regret bound of anytime EXP3 is larger by a factor of $\sqrt{2}$ compared to the regret bound of EXP3 tuned for a specific T. In return, the bound holds for all t and not just for one specific time T the algorithm was tuned for. All you need to do for this point is to write the new learning rate and the new regret bound, you do not need to prove anything. You can find more details on the derivation in (Bubeck and Cesa-Bianchi, 2012), if you want. In the experiments you should use the anytime version of the algorithm and the anytime expected regret bound.

The Practical Part of the Task Now you should evaluate the modified UCB1 algorithm and the modified anytime EXP3 algorithm on the data.

In this question you will work with a preprocessed subset of R6B Yahoo! Front Page Today Module User Click Log Dataset[8]. The data is given in `data_preprocessed_features` file as space-separated numbers. There are 701682 rows in the file. Each row has the following format:

(a) First comes the ID of the action that was taken (the ID of an article displayed to a user). The subset has 16 possible actions, indexed from 0 to 15, corresponding to 16 articles.

(b) Then comes the click indicator (0 = no click = no reward; 1 = click = reward). You may notice that the clicks are very sparse.

(c) And then you have 10 binary features for the user, which you can ignore. (Optionally, you can try to use the features to improve the selection strategy, but this is not for submission.)

You are given that the actions were selected uniformly at random and you should work with importance-weighted approach in this question.

In the following we refer to the quality of arms by their cumulative reward at the final time step $T = 701682$. Provide plots as described in the next two points for the following subsets of arms:

i. All arms.

ii. Extract rounds with the best and the worst arm (according to the reward at T) and repeat the experiment with just these two arms. Pay attention that after the extraction you can assume that you make offline evaluation with just two arms that were sampled uniformly at random [out of two arms]. The time horizon will get smaller.

iii. The same with the best and two worst arms.

iv. The best and three worst arms.

v. The best, the median, and the worst arm. (Since the number of arms is even, there are two median arms, the "upper" and the "lower" median; you can pick any of the two.)

[8] `https://webscope.sandbox.yahoo.com/catalog.php?datatype=r`

(d) Provide one plot per each setup described above, where you report the estimate of the *regret* of EXP3 and UCB1 based on offline importance-weighted samples as a function of time t. For each of the algorithms you should make 10 repetitions and report the mean and the mean $\pm$ one standard deviation over the repetitions. Pay attention that the regret at running time t should be computed against the action that is the best at time t, not the one that is the best at the final time T!

(e) The same plot, where you add the expected regret bound for EXP3 and the regret of a random strategy, which picks actions uniformly at random. (We leave you to think why we are not asking to provide a bound for UCB1.)

(f) Discussion of the results.

Optional, not for submission (for those who have taken "Machine Learning B" course): since the mean rewards are close to zero and have small variance, algorithms based on the kl-inequality, such as kl-UCB (Cappé et al., 2013), or algorithms that are able to exploit small variance, for example, by using the Unexpected Bernstein inequality, are expected to be able to exploit the small variance of importance-weighted losses and perform much better than Hoeffding-based UCB1.

Appendix A

Set Theory Basics

In this chapter we provide a number of basic definitions and notations from the set theory that are used in the notes.

Countable and Uncountable sets A set is called *countable* if its elements can be counted or, in other words, if every element in a set can be associated with a natural number. For example, the set of integer numbers is countable and the set of rational numbers (ratios of two integers) is also countable. Finite sets are countable as well. A set is called *uncountable* if its elements cannot be enumerated. For example, the set of real numbers $\mathbb{R}$ is uncountable and the set of numbers in a $[0, 1]$ interval is also uncountable.

Relations between sets For two sets A and B we use $A \subseteq B$ to denote that A is a subset of B.

Operations on sets For two sets A and B we use $A \cup B$ to denote the union of A and B; $A \cap B$ the intersection of A and B; and $A \setminus B$ the difference of A and B (the set of elements that are in A, but not in B).

The empty set We use $\varnothing$ or $\{\}$ to denote the empty set.

Disjoint sets Two sets A and B are called *disjoint* if $A \cap B = \varnothing$.

Appendix B

Probability Theory Basics

This chapter provides a number of basic definitions and results from the probability theory. It is partially based on Mitzenmacher and Upfal (2005).

B.1 Axioms of Probability

We start with a definition of a probability space.

Definition B.1 (Probability space)**.** *A probability space is a tuple* $(\Omega, \mathcal{F}, \mathbb{P})$*, where*

- Ω *is a* sample space*, which is the set of all possible outcomes of the random process modeled by the probability space.*
- $\mathcal{F}$ *is a family of sets representing the allowable events, where each set in* $\mathcal{F}$ *is a subset of the sample space* Ω*.*
- $\mathbb{P}$ *is a probability function* $\mathbb{P} : \mathcal{F} \to [0, 1]$ *satisfying Definition B.4.*

Elements of Ω are called *simple* or *elementary* events.

Example B.2. For coin flips the sample space is $\Omega = \{H, T\}$, where H stands for "heads" and T for "tails".

In dice rolling the sample space is $\Omega = \{1, 2, 3, 4, 5, 6\}$, where 1,...,6 label the sides of a dice (you should consider them as labels rather than numerical values, we get back to this later in Example B.15).

If we simultaneously flip a coin and roll a dice the sample space is
$\Omega = \{(H, 1), (T, 1), (H, 2), (T, 2), \ldots, (H, 6), (T, 6)\}$.

If Ω is countable (including finite), the probability space is *discrete*. In discrete probability spaces the family $\mathcal{F}$ consists of all subsets of Ω. In particular, $\mathcal{F}$ always includes the empty set $\varnothing$ and the complete sample space Ω. If Ω is uncountably infinite (for example, the real line or the $[0, 1]$ interval) a proper definition of $\mathcal{F}$ requires concepts from the measure theory, which go beyond the scope of these notes.

Example B.3. In the coin flipping experiment $\mathcal{F} = \{\varnothing, \{H\}, \{T\}, \{H, T\}\}$.

Definition B.4 (Probability Axioms)**.** *A probability function is any function* $\mathbb{P} : \mathcal{F} \to \mathbb{R}$ *that satisfies the following conditions*

1. *For any event* $E \in \mathcal{F}$*,* $0 \leq \mathbb{P}(E) \leq 1$*.*
2. $\mathbb{P}(\Omega) = 1$*.*
3. *For any finite or countably infinite sequence of mutually disjoint events* $E_1, E_2, \ldots$

$$\mathbb{P}\left(\bigcup_{i \geq 1} E_i\right) = \sum_{i \geq 1} \mathbb{P}(E_i).$$

We now consider a number of basic properties of probabilities.

Lemma B.5 (Monotonicity)**.** *Let A and B be two events, such that $A \subseteq B$. Then*

$$\mathbb{P}(A) \leq \mathbb{P}(B).$$

Proof. We have that $B = A \cup (B \setminus A)$ and the events A and $B \setminus A$ are disjoint. Thus,

$$\mathbb{P}(B) = \mathbb{P}(A) + \mathbb{P}(B \setminus A) \geq \mathbb{P}(A),$$

where the equality is by the third axiom of probabilities and the inequality is by the first axiom of probabilities, since $\mathbb{P}(B \setminus A) \geq 0$. □

The next simple, but very important result is known as the *union bound*.

Lemma B.6 (The union bound)**.** *For any finite or countably infinite sequence of events $E_1, E_2, \ldots$,*

$$\mathbb{P}\left(\bigcup_{i\geq 1} E_i\right) \leq \sum_{i\geq 1} \mathbb{P}(E_i).$$

Proof. We have

$$\bigcup_{i\geq 1} E_i = E_1 \cup (E_2 \setminus E_1) \cup (E_3 \setminus (E_1 \cup E_2)) \cup \cdots = \bigcup_{i\geq 1} F_i,$$

where the events $F_i = E_i \setminus \bigcup_{j=1}^{i-1} E_j$ are disjoint, $F_i \subseteq E_i$, and $\bigcup_{i\geq 1} F_i = \bigcup_{i\geq 1} E_i$. Therefore,

$$\mathbb{P}\left(\bigcup_{i\geq 1} E_i\right) = \mathbb{P}\left(\bigcup_{i\geq 1} F_i\right) = \mathbb{P}\left(\bigcup_{i\geq 1} F_i\right) = \sum_{i\geq 1} \mathbb{P}(F_i) \leq \sum_{i\geq 1} \mathbb{P}(E_i),$$

where the second equality is by the third axiom of probabilities and the inequality is by monotonicity of the probability (Lemma B.5). □

Example B.7. Let $E_1 = \{1, 3, 5\}$ be the event that the outcome of a dice roll is odd and $E_2 = \{1, 2, 3\}$ be the event that the outcome is at most 3. Then $\mathbb{P}(E_1 \cup E_2) = \mathbb{P}(1, 2, 3, 5) \leq \mathbb{P}(E_1) + \mathbb{P}(E_2)$. Note that this is true irrespective of the choice of the probability measure $\mathbb{P}$. In particular, this is true irrespective of whether the dice is fair or not.

Definition B.8 (Independence)**.** *Two events A and B are called* independent *if and only if*

$$\mathbb{P}(A \cap B) = \mathbb{P}(A) \cdot \mathbb{P}(B).$$

Definition B.9 (Pairwise independence)**.** *Events $E_1, \ldots, E_n$ are called* pairwise independent *if and only if for any pair i, j*

$$\mathbb{P}(E_i \cap E_j) = \mathbb{P}(E_i)\mathbb{P}(E_j).$$

Definition B.10 (Mutual independence)**.** *Events $E_1, \ldots, E_n$ are called* mutually independent *if and only if for any subset of indices $I \subseteq \{1, \ldots, n\}$*

$$\mathbb{P}\left(\bigcap_{i\in I} E_i\right) = \prod_{i\in I} \mathbb{P}(E_i).$$

Note that pairwise independence does not imply mutual independence. Take the following example: assume we roll a fair tetrahedron (a three-dimensional object with four faces) with faces colored in red, blue, green, and the fourth face colored in all three colors, red, blue, and green. Let E_1 be the event that we observe red color, E_2 be the even that we observe blue color, and E_3 be the event that we observe green color. Then for all i we have $\mathbb{P}(E_i) = \frac{1}{2}$ and for any pair $i \neq j$ we have $\mathbb{P}(E_i \cap E_j) = \frac{1}{4} = \mathbb{P}(E_i)\mathbb{P}(E_j)$. However, $\mathbb{P}(E_1 \cap E_2 \cap E_3) = \frac{1}{4} \neq \mathbb{P}(E_1)\mathbb{P}(E_2)\mathbb{P}(E_3)$ and, thus, the events are pairwise independent, but not mutually independent. If we say that events $E_1, \ldots, E_n$ are independent without further specifications we imply mutual independence.

Definition B.11 (Conditional probability)**.** *The* conditional probability *that event A occurs given that event B occurs is*

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}.$$

The conditional probability is well-defined only if $\mathbb{P}(B) > 0$.

By the definition we have that $\mathbb{P}(A \cap B) = \mathbb{P}(B)\mathbb{P}(A|B) = \mathbb{P}(A)\mathbb{P}(B|A)$.

Example B.12. For a fair dice let $A = \{1, 6\}$ and $B = \{1, 2, 3, 4\}$. Then

$$\begin{aligned}
\mathbb{P}(A) &= \frac{1}{3}, \\
\mathbb{P}(B) &= \frac{2}{3}, \\
A \cap B &= \{1\}, \\
\mathbb{P}(A \cap B) &= \frac{1}{6}, \\
\mathbb{P}(A|B) &= \frac{\frac{1}{6}}{\frac{2}{3}} = \frac{1}{4}.
\end{aligned}$$

Lemma B.13 (The law of total probability)**.** *Let $E_1, E_2, \ldots, E_n$ be mutually disjoint events, such that $\bigcup_{i=1}^{n} E_n = \Omega$. Then*

$$\mathbb{P}(A) = \sum_{i=1}^{n} \mathbb{P}(A \cap E_i) = \sum_{i=1}^{n} \mathbb{P}(A|E_i)\mathbb{P}(E_i).$$

Proof. Since the E_i-s are disjoint and cover the entire space it follows that $A = \bigcup_{i=1}^{n}(A \cap E_i)$ and the events $A \cap E_i$ are mutually disjoint. Therefore,

$$\mathbb{P}(A) = \mathbb{P}\left(\bigcup_{i=1}^{n}(A \cap E_i)\right) = \sum_{i=1}^{n} \mathbb{P}(A \cap E_i) = \sum_{i=1}^{n} \mathbb{P}(A|E_i)\mathbb{P}(E_i).$$

□

B.2 Discrete Random Variables

We now define another basic concept in probability theory, a *random variable*.

Definition B.14. *A* random variable *X on a sample space Ω is a real-valued function on Ω, that is $X : \Omega \to \mathbb{R}$. A* discrete *random variable is a random variable that takes on only a finite or countably infinite number of values.*

Example B.15. For a coin we can define a random variable X, such that $X(H) = 1$ and $X(T) = 0$. We can also define another random variable Y, such that $Y(H) = 1$ and $Y(T) = -1$.

For a dice we can define a random variable X, such that $X(1) = 1, X(2) = 2, X(3) = 3, X(4) = 4, X(5) = 5, X(6) = 6$. We can also define a random variable Y, such that $Y(1) = 3, Y(2) = 2.4, Y(3) = -6, Y(4) = 8, Y(5) = 8, Y(6) = 0$. This example emphasizes the difference between labeling of events and assignment of numerical values to events. Note that the random variable Y does not distinguish between faces 4 and 5 of the dice, even though they are separate events in the probability space.

Functions of random variables are also random variables. In the last example, a random variable $Z = X^2$ takes values $Z(1) = 1, Z(2) = 4, Z(3) = 9, \ldots, Z(6) = 36$.

Definition B.16 (Independence of random variables)**.** *Two random variables X and Y are* independent *if and only if*

$$\mathbb{P}((X = x) \cap (Y = y)) = \mathbb{P}(X = x)\mathbb{P}(Y = y).$$

for all values x and y.

Definition B.17 (Pairwise independence)**.** *Random variables $X_1, \ldots, X_n$ are* pairwise independent *if and only if for any pair i, j and any values x_i, x_j*

$$\mathbb{P}((X_i = x_i) \cap (X_j = x_j)) = \mathbb{P}(X_i = x_i)\mathbb{P}(X_j = x_j).$$

Definition B.18 (Mutual independence)**.** *Random variables $X_1, \ldots, X_n$ are* mutually independent *if for any subset of indices $I \subseteq \{1, \ldots, n\}$ and any values x_i, $i \in I$*

$$\mathbb{P}\left(\bigcap_{i \in I}(X_i = x_i)\right) = \prod_{i \in I}\mathbb{P}(X_i = x_i).$$

Similar to the example given earlier, pairwise independence of random variables does not imply their mutual independence. If we say that random variables are independent without further specifications we imply mutual independence.

B.3 Expectation

Expectation is the most basic characteristic of a random variable.

Definition B.19 (Expectation)**.** *Let X be a discrete random variable and let $\mathcal{X}$ be the set of all possible values that it can take. The* expectation *of X, denoted by $\mathbb{E}[X]$, is given by*

$$\mathbb{E}[X] = \sum_{x \in \mathcal{X}} x\mathbb{P}(X = x).$$

The expectation is finite if $\sum_{x \in \mathcal{X}} |x|\mathbb{P}(X = x)$ converges; otherwise the expectation is unbounded.

Example B.20. For a fair dice with faces numbered 1 to 6 let $X(i) = i$ (the i-th face gets value i). Then

$$\mathbb{E}[X] = \sum_{i=1}^{6} i\frac{1}{6} = \frac{7}{2}.$$

Take another random variable $Z = X^2$ then

$$\mathbb{E}[Z] = \mathbb{E}\left[X^2\right] = \sum_{i=1}^{6} i^2\frac{1}{6} = \frac{91}{6}.$$

Expectation satisfies a number of important properties (these properties also hold for continuous random variables). We leave a proof of these properties as an exercise.

Lemma B.21 (Multiplication by a constant)**.** *For any constant c*

$$\mathbb{E}[cX] = c\mathbb{E}[X].$$

Theorem B.22 (Linearity)**.** *For any pair of random variables X and Y, not necessarily independent,*

$$\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y].$$

Theorem B.23. *If X and Y are independent random variables, then*

$$\mathbb{E}[XY] = \mathbb{E}[X]\,\mathbb{E}[Y].$$

We emphasize that in contrast with Theorem B.22, this property does not hold in the general case (if X and Y are not independent).

B.4 Variance

Variance is the second most basic characteristic of a random variable.

Definition B.24 (Variance)**.** *The* variance *of a random variable X (discrete or continuous), denoted by $\mathbb{V}[X]$, is defined by*

$$\mathbb{V}[X] = \mathbb{E}\left[(X - \mathbb{E}[X])^2\right] = \mathbb{E}\left[X^2\right] - (\mathbb{E}[X])^2.$$

We invite the reader to prove that $\mathbb{E}\left[(X - \mathbb{E}[X])^2\right] = \mathbb{E}\left[X^2\right] - (\mathbb{E}[X])^2$.

Example B.25. For a fair dice with faces numbered 1 to 6 let $X(i) = i$ (the i-th face gets value i). Then

$$\mathbb{V}[X] = \mathbb{E}\left[X^2\right] - (\mathbb{E}[X])^2 = \frac{91}{6} - \frac{49}{4} = \frac{35}{12}.$$

Theorem B.26. *If $X_1, \ldots, X_n$ are independent random variables then*

$$\mathbb{V}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathbb{V}[X_i].$$

The proof is based on Theorem B.23 and the result does not necessarily hold when X_i-s are not independent. We leave the proof as an exercise.

B.5 The Bernoulli and Binomial Random Variables

Two most basic discrete random variables are Bernoulli and binomial.

Definition B.27 (Bernoulli random variable)**.** *A random variable X taking values $\{0, 1\}$ is called a* Bernoulli random variable*. The parameter $p = \mathbb{P}(X = 1)$ is called the* bias *of X.*

Bernoulli random variable has the following property (which does not hold in general):

$$\mathbb{E}[X] = 0 \cdot (1 - p) + 1 \cdot p = p = \mathbb{P}(X = 1).$$

Definition B.28 (Binomial random variable)**.** *A* binomial random variable *Y with parameters n and p, denoted by $B(n, p)$, is defined by the following probability distribution on $k \in \{0, 1, \ldots, n\}$:*

$$\mathbb{P}(Y = k) = \binom{n}{k} p^k (1 - p)^{n-k}.$$

Binomial random variable can be represented as a sum of independent identically distributed Bernoulli random variables.

Lemma B.29. *Let $X_1, \ldots, X_n$ be independent Bernoulli random variables with bias p. Then $Y = \sum_{i=1}^{n} X_i$ is a binomial random variable with parameters n and p.*

A proof of this lemma is left as an exercise to the reader.

B.6 Jensen's Inequality

Jensen's inequality is one of the most basic in probability theory.

Theorem B.30 (Jensen's inequality)**.** *If f is a convex function and X is a random variable, then*

$$\mathbb{E}[f(X)] \geq f(\mathbb{E}[X]).$$

For a proof see, for example, Mitzenmacher and Upfal (2005) or Cover and Thomas (2006).

Appendix C

Linear Algebra

We revisit a number of basic concepts from linear algebra. This is only a brief revision of the main concepts that we are using in the book. For more details, please, refer to Strang (2009) or some other textbook on linear algebra.

Vectors

- We use $\mathbf{u} = \begin{pmatrix} u_1 \\ \vdots \\ u_d \end{pmatrix}$ to denote a *vector* in $\mathbb{R}^d$. By default, vectors are column vectors.
- For two vectors $\mathbf{u} = \begin{pmatrix} u_1 \\ \vdots \\ u_d \end{pmatrix}$ and $\mathbf{v} = \begin{pmatrix} v_1 \\ \vdots \\ v_d \end{pmatrix}$ in $\mathbb{R}^d$, the *inner product* is defined by $\mathbf{u}^T\mathbf{v} = \sum_{i=1}^d u_i v_i$. The same quantity is also known as the *scalar product*, and the *dot product*. An alternative notation is $\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^T\mathbf{v}$. Note that $\mathbf{u}^T\mathbf{v} = \mathbf{v}^T\mathbf{u}$ and that the result is a scalar (a number).
- Two vectors $\mathbf{u}$ and $\mathbf{v}$ are perpendicular, $\mathbf{u} \perp \mathbf{v}$, if and only if their inner product $\mathbf{u}^T\mathbf{v} = 0$.
- The *outer product* is defined as $\mathbf{u}\mathbf{v}^T = \begin{pmatrix} u_1v_1 & \dots & u_1v_d \\ \vdots & \ddots & \vdots \\ u_dv_1 & \dots & u_dv_d \end{pmatrix}$. Note that the outer product is a matrix in $\mathbb{R}^{d\times d}$, and that $\mathbf{u}^T\mathbf{v} \neq \mathbf{u}\mathbf{v}^T$, unless $\mathbf{u}^T\mathbf{v}$ is a symmetric matrix. Also note that $\mathbf{u}^T\mathbf{v}$ is only defined when $\mathbf{u}$ and $\mathbf{v}$ have the same dimension, whereas $\mathbf{u}\mathbf{v}^T$ is defined also when the dimensions are not the same.
- If you consider $\mathbf{u}$ and $\mathbf{v}$ as matrices in $\mathbb{R}^{d\times 1}$, then it is easy to see that the definition of the inner and the outer product follow directly from the rules of matrix multiplication.
- The *Euclidean norm* of a vector $\mathbf{u}$, which corresponds to the length of $\mathbf{u}$, is denoted by $\|\mathbf{u}\| = \sqrt{\sum_{i=1}^d u_i^2}$. Note that the square norm satisfies $\|\mathbf{u}\|^2 = \mathbf{u}^T\mathbf{u}$.

Matrices A matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ takes vectors in $\mathbb{R}^d$ and maps them into $\mathbb{R}^n$. There are two fundamental subspaces associated with a matrix $\mathbf{X}$. The *image* of $\mathbf{X}$, denoted $Im(\mathbf{X}) \subseteq \mathbb{R}^n$, is the space of all vectors $\mathbf{v} \in \mathbb{R}^n$ that can be obtained through multiplication of $\mathbf{X}$ with a vector $\mathbf{w}$. The image $Im(\mathbf{X})$ is a linear subspace of $\mathbb{R}^n$ and it is also called a *column space* of $\mathbf{X}$. The second subspace is the *nullspace* of $\mathbf{X}$, denoted $Null(\mathbf{X}) \subseteq \mathbb{R}^d$, which is the space of all vectors $\mathbf{w}$ for which $\mathbf{X}\mathbf{w} = 0$. The nullspace is a linear subspace of $\mathbb{R}^d$. The subspaces are illustrated in Figure C.1.

Matrix transpose Matrix transpose $\mathbf{X}^T$ takes vectors in $\mathbb{R}^n$ and maps them into $\mathbb{R}^d$. The corresponding subspaces are $Im(\mathbf{X}^T)$, the *row space* of $\mathbf{X}$, and $Null(\mathbf{X}^T)$.

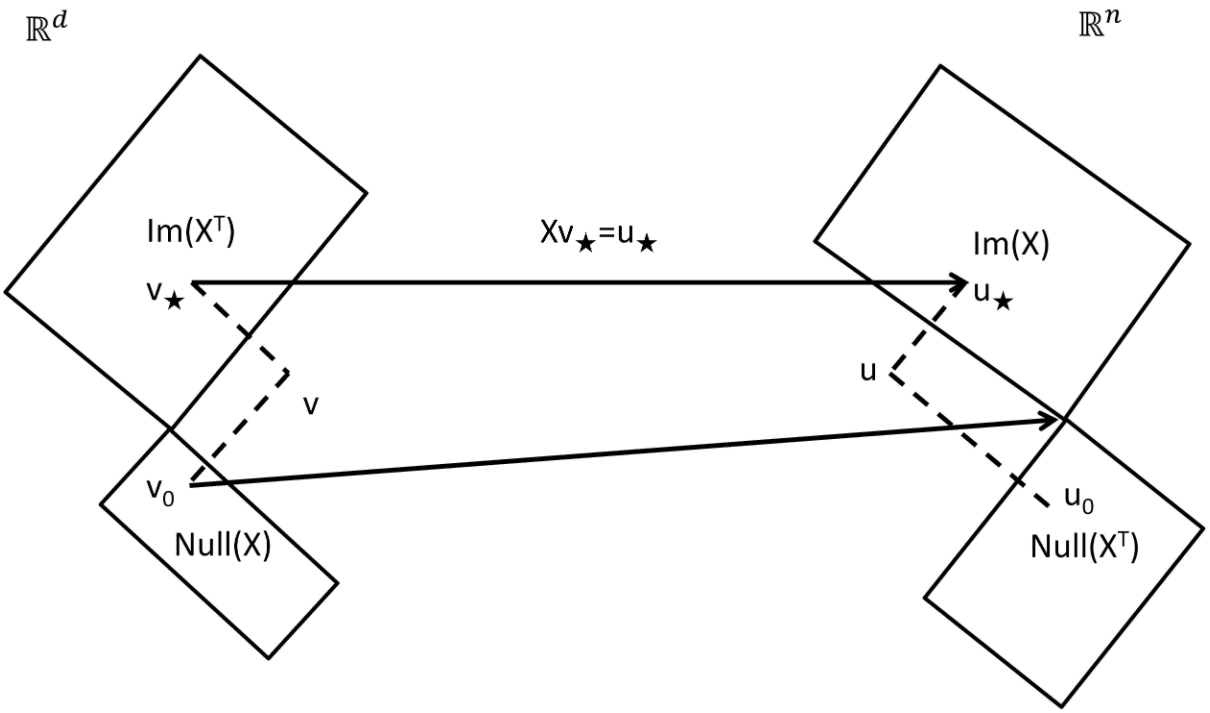


Figure C.1: **The four fundamental subspaces of a matrix X.** There is a right angle between $Im(\mathbf{X})$ and $Null(\mathbf{X}^T)$, as well as between $Im(\mathbf{X}^T)$ and $Null(\mathbf{X})$.

Orthogonality of the fundamental subspaces $Im(\mathbf{X}) \perp Null(\mathbf{X}^T)$ and $Im(\mathbf{X}^T) \perp Null(\mathbf{X})$ There is an important and extremely beautiful relation between the four fundamental subspaces associated with a matrix $\mathbf{X}$ and its transpose. Namely, the image of $\mathbf{X}$ is orthogonal to the nullspace of $\mathbf{X}^T$ and the image of $\mathbf{X}^T$ is orthogonal to the nullspace of $\mathbf{X}$. It means that if we take any two vectors $\mathbf{u} \in Im(\mathbf{X})$ and $\mathbf{v} \in Null(\mathbf{X}^T)$ then $\mathbf{u}^T\mathbf{v} = 0$ (and the same for the second pair of subspaces). The proof of this fact is short and elegant. Any vector in $\mathbf{u} \in Im(\mathbf{X})$ can be represented as a linear combination of the rows of $\mathbf{X}$, meaning that $\mathbf{u} = \mathbf{X}\mathbf{z}$. At the same time, by definition of a nullspace, if $\mathbf{v} \in Null(\mathbf{X}^T)$ then $\mathbf{X}^T\mathbf{v} = 0$. By putting these two facts together we obtain:

$$\mathbf{u}^T\mathbf{v} = (\mathbf{X}\mathbf{z})^T\mathbf{v} = \mathbf{z}^T\mathbf{X}^T\mathbf{v} = \mathbf{z}^T(\mathbf{X}^T\mathbf{v}) = 0.$$

Complete relation between $Im(\mathbf{X})$, $Im(\mathbf{X}^T)$, $Null(\mathbf{X})$, and $Null(\mathbf{X}^T)$ Not only the pairs $Im(\mathbf{X})$ with $Null(\mathbf{X}^T)$ and $Im(\mathbf{X}^T)$ with $Null(\mathbf{X})$ are orthogonal, they also complement each other. Let $dim(\mathbf{A})$ denote dimension of a matrix $\mathbf{A}$. The dimension is equal to the number of independent columns, which is equal to the number of independent rows (this fact can be shown by bringing $\mathbf{A}$ to a diagonal form). Then we have the following relations:

1. $dim(Im(\mathbf{X})) = dim(Im(\mathbf{X}^T)) = dim(\mathbf{X})$.
2. $dim(Null(\mathbf{X})) = d - dim(Im(\mathbf{X}^T))$ and $dim(Null(\mathbf{X}^T)) = n - dim(Im(\mathbf{X}))$.
3. $Im(\mathbf{X}) \perp Null(\mathbf{X}^T)$ and $Im(\mathbf{X}^T) \perp Null(X)$.

Together these properties mean that a combination of bases for $Im(\mathbf{X}^T)$ and $Null(\mathbf{X})$ makes a basis for $\mathbb{R}^d$ and a combination of bases for $Im(\mathbf{X})$ and $Null(\mathbf{X}^T)$ make a basis for $\mathbb{R}^n$. It means that any vector $\mathbf{v} \in \mathbb{R}^d$ can be represented as $\mathbf{v} = \mathbf{v}_\star + \mathbf{v}_0$, where $\mathbf{v}_\star \in Im(\mathbf{X}^T)$ belongs to the row space of $\mathbf{X}$ and $v_0 \in Null(\mathbf{X})$ belongs to the nullspace of $\mathbf{X}$.

The mapping between $Im(\mathbf{X}^T)$ and $Im(\mathbf{X})$ is one-to-one and, thus, invertible Every vector $\mathbf{u}$ in the column space comes from one and only one vector in the row space $\mathbf{v}$. The proof of this fact is also simple. Assume that $\mathbf{u} = \mathbf{X}\mathbf{v} = \mathbf{X}\mathbf{v}'$ for two vectors $\mathbf{v}, \mathbf{v}' \in Im(\mathbf{X}^T)$. Then $\mathbf{X}(\mathbf{v} - \mathbf{v}') = 0$ and

the vector $\mathbf{v} - \mathbf{v}' \in Null(\mathbf{X})$. But $Null(\mathbf{X})$ is perpendicular to $Im(\mathbf{X}^T)$, which means that $\mathbf{v} - \mathbf{v}'$ is orthogonal to itself and, therefore, must be the zero vector.

$\mathbf{X}^T\mathbf{X}$ is invertible if and only if X has linearly independent columns $(\mathbf{X}^T\mathbf{X})^{-1}$ is a very important matrix. We show that $\mathbf{X}^T\mathbf{X}$ is invertible if and only if $\mathbf{X}$ has linearly independent columns, meaning that $dim(X) = d$. We show this by proving that $\mathbf{X}$ and $\mathbf{X}^T\mathbf{X}$ have the same nullspace. Let $\mathbf{v} \in Null(\mathbf{X})$, then $\mathbf{X}\mathbf{v} = 0$ and, therefore, $\mathbf{X}^T\mathbf{X}\mathbf{v} = 0$ and $\mathbf{v} \in Null(\mathbf{X}^T\mathbf{X})$. In the other direction, let $\mathbf{v} \in Null(\mathbf{X}^T\mathbf{X})$. Then $\mathbf{X}^T\mathbf{X}\mathbf{v} = 0$ and we have:

$$\|\mathbf{X}\mathbf{v}\|^2 = (\mathbf{X}\mathbf{v})^T(\mathbf{X}\mathbf{v}) = \mathbf{v}^T\mathbf{X}^T\mathbf{X}\mathbf{v} = \mathbf{v}^T(\mathbf{X}^T\mathbf{X}\mathbf{v}) = 0.$$

Since $\|\mathbf{X}\mathbf{v}\|^2 = 0$ if and only if $\mathbf{X}\mathbf{v} = 0$, we have $\mathbf{v} \in Null(\mathbf{X})$.

$\mathbf{X}^T\mathbf{X}$ is a $d \times d$ square matrix, therefore $dim(\mathbf{X}^T\mathbf{X}) = d - dim(Null(\mathbf{X}^T\mathbf{X})) = d - dim(Null(\mathbf{X}))$ and matrix $\mathbf{X}^T\mathbf{X}$ is invertible if and only if the dimension of the nullspace of $\mathbf{X}$ is zero, meaning that $\mathbf{X}$ has linearly independent columns. (Note that unless $n = d$, $\mathbf{X}$ itself is a rectangular matrix and that inverses are not defined for rectangular matrices.)

Projection onto a line A line in direction $\mathbf{u}$ is described by $\alpha\mathbf{u}$ for $\alpha \in \mathbb{R}$. Projection of vector $\mathbf{v}$ onto vector $\mathbf{u}$ means that we are looking for a projection vector $\mathbf{p} = \alpha\mathbf{u}$, such that the remainder $\mathbf{v} - \mathbf{p}$ is orthogonal to the projection. So we have:

$$\begin{aligned}(\mathbf{v} - \alpha\mathbf{u})^T\alpha\mathbf{u} &= 0,\\ \alpha\mathbf{v}^T\mathbf{u} &= \alpha^2\mathbf{u}^T\mathbf{u},\\ \alpha &= \frac{\mathbf{v}^T\mathbf{u}}{\mathbf{u}^T\mathbf{u}} = \frac{\mathbf{u}^T\mathbf{v}}{\mathbf{u}^T\mathbf{u}}.\end{aligned}$$

Thus, the projection $\mathbf{p} = \alpha\mathbf{u} = \frac{\mathbf{u}^T\mathbf{v}}{\mathbf{u}^T\mathbf{u}}\mathbf{u}$. Note that $\frac{\mathbf{u}^T\mathbf{v}}{\mathbf{u}^T\mathbf{u}}$ is a scalar, thus

$$\mathbf{p} = \frac{\mathbf{u}^T\mathbf{v}}{\mathbf{u}^T\mathbf{u}}\mathbf{u} = \mathbf{u}\frac{\mathbf{u}^T\mathbf{v}}{\mathbf{u}^T\mathbf{u}} = \frac{\mathbf{u}\mathbf{u}^T}{\mathbf{u}^T\mathbf{u}}\mathbf{v}.$$

The matrix $\mathbf{P} = \frac{\mathbf{u}\mathbf{u}^T}{\mathbf{u}^T\mathbf{u}}$ is a *projection matrix*. For any vector $\mathbf{v}$ the matrix $\mathbf{P}$ projects $\mathbf{v}$ onto u.

Projection onto a subspace A subspace can be described by a set of linear combinations $\mathbf{A}\mathbf{z}$, where the columns of matrix $\mathbf{A}$ span the subspace. Projection of a vector $\mathbf{v}$ onto a subspace described by $\mathbf{A}$ means that we are looking for a projection $\mathbf{p} = \mathbf{A}\mathbf{z}$, such that the remainder $\mathbf{v} - \mathbf{p}$ is perpendicular to the projection. The projection $\mathbf{p} = \mathbf{A}\mathbf{z}$ belongs to the image of $\mathbf{A}$, $Im(\mathbf{A})$. Thus, the remainder must be in the nullspace of $\mathbf{A}^T$, meaning that $\mathbf{A}^T(\mathbf{v} - \mathbf{p}) = 0$. Assuming that the columns of A are independent, we have:

$$\begin{aligned}\mathbf{A}^T(\mathbf{v} - \mathbf{A}\mathbf{z}) &= 0,\\ \mathbf{A}^T\mathbf{v} &= \mathbf{A}^T\mathbf{A}\mathbf{z},\\ \mathbf{z} &= (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{v},\end{aligned}$$

where we used independence of the columns of $\mathbf{A}$ in the last step to invert $\mathbf{A}^T\mathbf{A}$. The projection is $\mathbf{p} = \mathbf{A}\mathbf{z} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{v}$ and the projection matrix is $\mathbf{P} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$. The projection matrix $\mathbf{P}$ maps any vector $\mathbf{v}$ onto the space spanned by the columns of $\mathbf{A}$, $Im(\mathbf{A})$. Note how $(\mathbf{A}^T\mathbf{A})^{-1}$ plays the role of $\frac{1}{\mathbf{u}^T\mathbf{u}}$ in projection onto a line.

Projection matrices Projection matrices satisfy a number of interesting properties:

1. If $\mathbf{P}$ is a projection matrix then $\mathbf{P}^2 = \mathbf{P}$ (the second projection does not change the vector).
2. If $\mathbf{P}$ is a projection matrix projecting onto a subspace described by $\mathbf{A}$ then $\mathbf{I} - \mathbf{P}$ is also a projection matrix. It projects onto a subspace that is perpendicular to the subspace described by $\mathbf{A}$.

Appendix D

Calculus

We revisit some basic concepts from calculus.

D.1 Gradients

Gradients are vectors of partial derivatives. For a vector $\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix}$ and a function $f(\mathbf{x})$ the gradient of f is defined as

$$\nabla f(\mathbf{x}) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_d} \end{pmatrix}.$$

Gradient of a multivariate quadratic function $f(\mathbf{x}) = \mathbf{x}^T A\mathbf{x}$

Let A be a matrix with entries a_{ij}. Then

$$f(\mathbf{x}) = \mathbf{x}^T A\mathbf{x} = (x_1, \cdots, x_d) \begin{pmatrix} a_{11} & \cdots & a_{1d} \\ \vdots & & \vdots \\ a_{d1} & \cdots & a_{dd} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix} = (x_1, \cdots, x_d) \begin{pmatrix} \sum_{j=1}^d a_{1j}x_j \\ \vdots \\ \sum_{j=1}^d a_{dj}x_j \end{pmatrix} = \sum_{i=1}^d \sum_{j=1}^d a_{ij}x_i x_j.$$

The partial derivative $\frac{\partial f}{\partial x_k}$ then becomes:

$$\frac{\partial f}{\partial x_k} = \frac{\partial \left(\sum_{i=1}^d \sum_{j=1}^d a_{ij}x_i x_j \right)}{\partial x_k} = \sum_{j=1}^d a_{kj}x_j + \sum_{i=1}^d a_{ik}x_i,$$

where the first sum corresponds to the first element in the product $x_i x_j$ being x_k and the second sum corresponds to the second element in the product $x_i x_j$ being x_k. Putting all the derivatives together we obtain:

$$\begin{aligned} \nabla f(\mathbf{x}) &= \begin{pmatrix} \sum_{j=1}^d a_{1j}x_j + \sum_{i=1}^d a_{i1}x_i \\ \vdots \\ \sum_{j=1}^d a_{d1}x_j + \sum_{i=1}^d a_{id}x_i \end{pmatrix} \\ &= \begin{pmatrix} a_{11} & \cdots & a_{1d} \\ \vdots & & \vdots \\ a_{d1} & \cdots & a_{dd} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix} + \left((x_1, \cdots, x_d) \begin{pmatrix} a_{11} & \cdots & a_{1d} \\ \vdots & & \vdots \\ a_{d1} & \cdots & a_{dd} \end{pmatrix} \right)^T \\ &= A\mathbf{x} + A^T\mathbf{x} \\ &= (A + A^T)\mathbf{x}. \end{aligned}$$

A matrix A is called *symmetric* if $A^T = A$. For a symmetric matrix we have $\nabla f(\mathbf{x}) = 2A\mathbf{x}$ and for a general matrix we have $\nabla f(\mathbf{x}) = (A + A^T)\mathbf{x}$. Note the similarity and dissimilarity with the derivative of a univariate quadratic function $f(x) = ax^2$, which is $f'(x) = 2ax$.

Gradient of a linear function $f(\mathbf{x}) = \mathbf{b}^T\mathbf{x}$

Let $\mathbf{b} = \begin{pmatrix} b_1 \\ \vdots \\ b_d \end{pmatrix}$ be a vector and let $f(\mathbf{x}) = \mathbf{b}^T\mathbf{x} = \sum_{i=1}^{d} b_i x_i$. We leave it as an exercise to prove that the gradient $\nabla f(\mathbf{x}) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_d} \end{pmatrix} = \mathbf{b}$.

Appendix E

Vectorized Implementation of the K Nearest Neighbors Algorithm

Python, as well as other interpreter programming languages, provides built-in precompiled functions for vector and matrix operations, making it more efficient to work with vector operations rather than explicitly loop through vectors in the code. The appendix explains how to implement the K-NN algorithm using vector and matrix operations. Please, check the "Vectors" paragraph in Chapter C on Linear Algebra for a definition and basic facts about vectors (you do not need the rest of the appendix for this section).

For two points $\mathbf{x}$ and $\mathbf{z}$ in $\mathbb{R}^d$, the *Euclidean distance* between $\mathbf{x}$ and $\mathbf{z}$ is given by the norm of the vector from $\mathbf{x}$ to $\mathbf{z}$, $\text{dist}(\mathbf{x}, \mathbf{z}) = \|\mathbf{z} - \mathbf{x}\|$. The square distance can then be written as

$$\text{dist}(\mathbf{x}, \mathbf{z})^2 = \|\mathbf{x} - \mathbf{z}\|^2 = (\mathbf{x} - \mathbf{z})^T(\mathbf{x} - \mathbf{z}) = \mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T\mathbf{z} + \mathbf{z}^T\mathbf{z}. \tag{E.1}$$

Next we show how to exploit the above vectorized representation of a distance between two points for vectorized computation of pairwise distances across sets of multiple points.

Efficient Computation of Distances In order to label a target point $\mathbf{x}$, the K-NN algorithm requires sorting training points by their distance to the target point. Note that sorting the points by distance is equivalent to sorting them by the square distance, which allows to save the computation of the square root in the definition of the Euclidean norm. We show how to use the square distance representation in Equation (E.1) for efficient vectorized computation of distances from all training points to all target points without using any for-loops. Let $\mathbf{X}^{\texttt{train}} = \left(\begin{pmatrix} | \\ \mathbf{x}_1^{\texttt{train}} \\ | \end{pmatrix}, \cdots, \begin{pmatrix} | \\ \mathbf{x}_m^{\texttt{train}} \\ | \end{pmatrix} \right) \in \mathbb{R}^{d \times m}$ be a set of m training points and $\mathbf{X}^{\texttt{targ}} = \left(\begin{pmatrix} | \\ \mathbf{x}_1^{\texttt{targ}} \\ | \end{pmatrix}, \cdots, \begin{pmatrix} | \\ \mathbf{x}_n^{\texttt{targ}} \\ | \end{pmatrix} \right) \in \mathbb{R}^{d \times n}$ be a set of n target points, written as column vectors in the corresponding matrices. Let $\mathbf{x}_i^{\texttt{train}}$ be the i-th training point and $\mathbf{x}_j^{\texttt{targ}}$ be the j-th target point, then

$$\text{dist}(\mathbf{x}_i^{\texttt{train}}, \mathbf{x}_j^{\texttt{targ}}) = (\mathbf{x}_i^{\texttt{train}})^T\mathbf{x}_i^{\texttt{train}} - 2(\mathbf{x}_i^{\texttt{train}})^T\mathbf{x}_j^{\texttt{targ}} + (\mathbf{x}_j^{\texttt{targ}})^T\mathbf{x}_j^{\texttt{targ}}.$$

Our aim is to compute the matrix

$$D = \begin{pmatrix} \text{dist}(\mathbf{x}_1^{\texttt{train}}, \mathbf{x}_1^{\texttt{targ}}) & \dots & \text{dist}(\mathbf{x}_1^{\texttt{train}}\mathbf{x}_n^{\texttt{targ}}) \\ \vdots & \ddots & \vdots \\ \text{dist}(\mathbf{x}_m^{\texttt{train}}, \mathbf{x}_1^{\texttt{targ}}) & \dots & \text{dist}(\mathbf{x}_m^{\texttt{train}}\mathbf{x}_n^{\texttt{targ}}) \end{pmatrix} \in \mathbb{R}^{m \times n}$$

of all pairwise distances.

- For a square matrix $M = \begin{pmatrix} m_{1,1} & \dots & m_{1,d} \\ \vdots & \ddots & \vdots \\ m_{d,1} & \dots & m_{d,d} \end{pmatrix} \in \mathbb{R}^{d \times d}$ let $\text{diag}(M) = \begin{pmatrix} m_{1,1} \\ \vdots \\ m_{d,d} \end{pmatrix}$ be a column vector of the diagonal elements of M. (You can find a built-in Python function for extracting the diagonal.)
- Let $\mathbf{1}^d = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^d$ be a vector of d ones.
- Verify (convince yourself) that

$$D = \text{diag}\left((\mathbf{X}^{\texttt{train}})^T \mathbf{X}^{\texttt{train}}\right)(\mathbf{1}^n)^T - 2(\mathbf{X}^{\texttt{train}})^T \mathbf{X}^{\texttt{targ}} + \mathbf{1}^m \left(\text{diag}\left((\mathbf{X}^{\texttt{targ}})^T \mathbf{X}^{\texttt{targ}}\right)\right)^T.$$

  It should be a good idea to visualize the relevant vectors and matrices and their products to convince yourself. Appreciate the power of linear algebra — the above expression provides distances from all the training points to all the target points in just one line without any for-loops!

Additional Guidance

- Note that for a single data point you can compute the output of K-NN for all K in one shot using vector operations. No need in for-loops! And with a bit extra effort you should be able to do it without for-loops for the whole dataset.
- You may find the following functions useful:
  - Built-in sorting functions for sorting the distances.
  - Built-in functions for computing a cumulative sum of elements of a vector `v` (for computing the predictions of K-NN for all K at once).
- It may be a good idea to debug your code with a small subset of the data.

Bibliography

Yaser S. Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin. *Learning from data.* AMLbook, 2012.

Yaser S. Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin. *Learning from data. Dynamic E-Chapters.* AMLbook, 2015.

Alekh Agarwal, Miroslav Dudík, Satyen Kale, John Langford, and Robert E. Schapire. Contextual bandit learning with predictable rewards. In *Proceedings on the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2012.

A. Asuncion and D.J. Newman. UCI machine learning repository, 2007. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

Jean-Yves Audibert and Sébastien Bubeck. Minimax policies for adversarial and stochastic bandits. In *Proceedings of the Conference on Learning Theory (COLT)*, 2009.

Jean-Yves Audibert and Sébastien Bubeck. Regret bounds and minimax policies under partial monitoring. *Journal of Machine Learning Research*, 11, 2010.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47, 2002a.

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal of Computing*, 32(1), 2002b.

Orly Avner, Shie Mannor, and Ohad Shamir. Decoupling exploration and exploitation in multi-armed bandits. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2012.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities A Nonasymptotic Theory of Independence.* Oxford University Press, 2013.

Sébastien Bubeck. *Bandits Games and Clustering Foundations.* PhD thesis, Université Lille, 2010.

Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5, 2012.

Olivier Cappé, Aurélien Garivier, Odalric-Ambrym Maillard, Rémi Munos, and Gilles Stoltz. Kullback–Leibler upper confidence bounds for optimal sequential allocation. *The Annals of Statistics*, 41 (3), 2013.

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games.* Cambridge University Press, 2006.

Nicolò Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66, 2007.

Zachary Chase, Shinji Ito, and Idan Mehalel. A tight lower bound for non-stochastic multi-armed bandits with expert advice. Technical report, `https://arxiv.org/abs/2511.00257`, 2025.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* Wiley Series in Telecommunications and Signal Processing, 2nd edition, 2006.

Xiequan Fan, Ion Grama, and Quansheng Liu. Exponential inequalities for martingales with applications. *Electronic Journal of Probability*, 20, 2015.

Andrew Foong, Wessel Bruinsma, David Burt, and Richard Turner. How tight can pac-bayes be in the small data regime? In *Advances in Neural Information Processing Systems (NIPS)*, 2021.

Andrew Y. K. Foong, Wessel P. Bruinsma, and David R. Burt. A note on the chernoff bound for random variables in the unit interval. *arXiv preprint arXiv.2205.07880*, 2022.

Wei Gao and Zhi-Hua Zhou. On the doubt about margin explanation of boosting. *Artificial Intelligence*, 2013.

Pascal Germain, Alexandre Lacasse, François Laviolette, and Mario Marchand. PAC-Bayesian learning of linear classifiers. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2009.

Pascal Germain, Alexandre Lacasse, François Laviolette, Mario Marchand, and Jean-Francis Roy. Risk bounds for the majority vote: From a PAC-Bayesian analysis to a learning algorithm. *Journal of Machine Learning Research*, 16, 2015.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Daniel Kahneman. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, 2011.

Satyen Kale. Multiarmed bandits with limited expert advice. In *Proceedings of the Conference on Learning Theory (COLT)*, 2014.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6, 1985.

John Langford. Tutorial on practical prediction theory for classification. *Journal of Machine Learning Research*, 6, 2005.

Y. LeCun, C. Cortes, and C. J. C. Burges. The MNIST database of handwritten digits, 1994. MNIST was created in 1994 and released in 1998.

Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. 2011.

Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108, 1994.

Katalin Marton. A measure concentration inequality for contracting Markov chains. *Geometric and Functional Analysis*, 6(3), 1996.

Katalin Marton. A measure concentration inequality for contracting Markov chains Erratum. *Geometric and Functional Analysis*, 7(3), 1997.

Andrés R. Masegosa, Stephan S. Lorenzen, Christian Igel, and Yevgeny Seldin. Second order PAC-Bayesian bounds for the weighted majority vote. Technical report, `https://arxiv.org/abs/2007.13532`, 2020.

Andreas Maurer. A note on the PAC-Bayesian theorem. www.arxiv.org, 2004.

Andreas Maurer and Massimiliano Pontil. Empirical Bernstein bounds and sample variance penalization. In *Proceedings of the Conference on Learning Theory (COLT)*, 2009.

David McAllester. PAC-Bayesian stochastic model selection. *Machine Learning*, 51, 2003.

Zakaria Mhammedi, Peter Grünwald, and Benjamin Guedj. PAC-Bayes un-expected Bernstein inequality. In *Advances in Neural Information Processing Systems (NIPS)*, 2019.

Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.

Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 1952.

Chloé Rouyer and Yevgeny Seldin. Tsallis-INF for decoupled exploration and exploitation in multi-armed bandits. In *Proceedings of the Conference on Learning Theory (COLT)*, 2020.

Paul-Marie Samson. Concentration of measure inequalities for markov chains and ϕ-mixing processes. *The Annals of Probability*, 28(1), 2000.

Matthias Seeger. PAC-Bayesian generalization error bounds for Gaussian process classification. *Journal of Machine Learning Research*, 3, 2002.

Yevgeny Seldin. The space of online learning problems. ECML-PKDD Tutorial. `https://sites.google.com/site/spaceofonlinelearningproblems/`, 2015.

Yevgeny Seldin and Gábor Lugosi. A lower bound for multi-armed bandits with expert advice. In *Proceedings of the European Workshop on Reinforcement Learning (EWRL)*, 2016.

Yevgeny Seldin, François Laviolette, Nicolò Cesa-Bianchi, John Shawe-Taylor, and Peter Auer. PAC-Bayesian inequalities for martingales. *IEEE Transactions on Information Theory*, 58, 2012.

Yevgeny Seldin, Peter L. Bartlett, Koby Crammer, and Yasin Abbasi-Yadkori. Prediction with limited advice and multiarmed bandits with paid observations. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2014.

Gilles Stoltz. *Incomplete Information and Internal Regret in Prediction of Individual Sequences*. PhD thesis, Université Paris-Sud, 2005.

Gilbert Strang. *Introduction to linear algebra*. Wellesley-Cambridge Press, 4^{th} edition, 2009.

Niklas Thiemann, Christian Igel, Olivier Wintenberger, and Yevgeny Seldin. A strongly quasiconvex PAC-Bayesian bound. In *Proceedings of the International Conference on Algorithmic Learning Theory (ALT)*, 2017.

William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25, 1933.

Ilya Tolstikhin and Yevgeny Seldin. PAC-Bayes-Empirical-Bernstein inequality. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

Vladimir Vovk. Aggregating strategies. In *Proceedings of the Conference on Learning Theory (COLT)*, 1990.

John M. Wozengraft and Barney Reiffen. *Sequential Decoding*. The MIT Press, 1961.

Yi-Shan Wu and Yevgeny Seldin. Split-kl and PAC-Bayes-split-kl inequalities for ternary random variables. 2022.

Yi-Shan Wu, Andrés R. Masegosa, Stephan S. Lorenzen, Christian Igel, and Yevgeny Seldin. Chebyshev-cantelli pac-bayes-bennett inequality for the weighted majority vote. 2021.

Yi-Shan Wu, Yijie Zhang, Badr-Eddine Chérief-Abdellatif, and Yevgeny Seldin. Recursive PAC-Bayes: A frequentist approach to sequential prior updates with no information loss. 2024.

Julian Zimmert and Yevgeny Seldin. Tsallis-INF: An optimal algorithm for stochastic and adversarial bandits. *Journal of Machine Learning Research*, 2021.