Notes for COMS 4773 Fall 2026

Daniel Hsu

November 24, 2026

Contents

1 Using expert advice **4**
- 1.1 Basic scenario and modeling . . . 4
  - 1.1.1 Expert advice scenario . . . 4
  - 1.1.2 Modeling . . . 4
- 1.2 Basic learning algorithms for using expert advice . . . 4
  - 1.2.1 Follow the Leader . . . 4
  - 1.2.2 Halving . . . 5
  - 1.2.3 Barrier for all learning algorithms . . . 5
- 1.3 Modeling and coping with imperfection . . . 6
  - 1.3.1 Experts with bounded imperfection . . . 6
  - 1.3.2 Halving with Resurrection . . . 6
  - 1.3.3 Barrier for all learning algorithms . . . 7
  - 1.3.4 Weighted Majority . . . 7

2 Online allocation **10**
- 2.1 Basic scenario and performance measures . . . 10
  - 2.1.1 Online allocation scenario . . . 10
  - 2.1.2 Performance measures . . . 10
- 2.2 Connection to the expert advice scenario . . . 11
  - 2.2.1 Randomized Weighted Majority . . . 11
  - 2.2.2 Non-adaptive adversaries and the connection to online allocation . . . 12
  - 2.2.3 Foreshadowing: breaking through the pre-factor of two . . . 12
- 2.3 Algorithm for online allocation . . . 13
  - 2.3.1 Hedge algorithm . . . 13
  - 2.3.2 Entropy and relative entropy . . . 15
  - 2.3.3 Proof of the Hedge theorem . . . 16
  - 2.3.4 Improved analysis of Hedge for bounded losses . . . 17
- 2.4 Application to zero-sum games . . . 18
  - 2.4.1 Zero-sum games . . . 18
  - 2.4.2 Proof of the easy part . . . 19
  - 2.4.3 Proof of the other part via no-regret online allocation . . . 20
- 2.5 Application to online convex optimization . . . 21
  - 2.5.1 Convex functions . . . 21
  - 2.5.2 Generalization of online allocation . . . 21
- 2.6 Optimality of Hedge . . . 22
  - 2.6.1 Lower bound for any learning algorithm . . . 22
  - 2.6.2 Proof strategy . . . 23
  - 2.6.3 Most of the proof of the lower bound . . . 23

3 Online concept learning **25**
3.1 Concept classes and the basic scenario . . . 25
3.1.1 Concept classes . . . 25
3.1.2 Online concept learning scenario . . . 25
3.2 Learning concepts over the Boolean domain . . . 25
3.2.1 Concepts over the Boolean n-cube . . . 25
3.2.2 Monotone conjunctions . . . 26
3.2.3 Monotone disjunctions . . . 27
3.2.4 Conjunctions . . . 28
3.2.5 Linear threshold functions . . . 28
3.3 Perceptron and margins . . . 30
3.3.1 Linear threshold functions over Euclidean space . . . 30
3.3.2 Margins . . . 31
3.3.3 Perceptron . . . 32
3.4 Optimal mistake bounds . . . 34
3.4.1 Mistake trees . . . 34
3.4.2 Littlestone dimension . . . 35

4 Statistical learning **38**
4.1 Supervised machine learning . . . 38
4.1.1 Machine learning as programming-by-example . . . 38
4.1.2 Statistical framework . . . 38
4.1.3 Probably Approximately Correct criterion . . . 40
4.1.4 Role of test data . . . 40
4.2 PAC learning from online concept learning . . . 41
4.2.1 Batch versus online learning . . . 41
4.2.2 From online to batch learning . . . 41
4.3 Detour: Tools to bound the probability of large deviations . . . 44
4.3.1 Chebyshev's inequality . . . 44
4.3.2 Logarithmic moment generating function and the Cramér-Chernoff method . . . 45
4.3.3 Rate function and Cramér's theorem . . . 47
4.3.4 Subgaussianity and Hoeffding's inequality . . . 49
4.3.5 Poisson heuristic and Bennett's inequality . . . 50
4.4 PAC learning by minimizing training error rate . . . 52
4.4.1 Union bound . . . 52
4.4.2 Learning by finding a consistent hypothesis . . . 52
4.4.3 Learning by finding a hypothesis with low training error rate . . . 54
4.4.4 Occam's razor . . . 56

5 Uniform convergence **58**
5.1 Uniform convergence of training error rates . . . 58
5.1.1 Finite hypothesis classes . . . 58
5.1.2 Impossibility for some infinite hypothesis classes . . . 58
5.2 Vapnik-Chervonenkis theory . . . 59
5.2.1 Empirical covers and empirical processes . . . 59
5.2.2 VC dimension . . . 59

5.2.3 Sauer's lemma . . . 61
5.2.4 Population and empirical averages . . . 64
5.2.5 Zero-one loss class . . . 69
5.2.6 Lower bound for uniform convergence . . . 69

6 Optimal sample complexity **71**
6.1 Comparing sample complexity upper bounds for PAC learning . . . 71
6.2 Sample complexity lower bound for PAC learning . . . 72
6.2.1 Statement of the lower bound . . . 72
6.2.2 Heuristic argument for constant ϵ . . . 73
6.2.3 Heuristic argument for all $\epsilon > 0$ . . . 73
6.2.4 Rigorous proof for all $\epsilon > 0$ . . . 74

7 Boosting **76**
7.1 Weak versus strong PAC learning . . . 76
7.2 In-sample boosting . . . 76
7.2.1 In-sample weak learning . . . 76
7.2.2 An alternative characterization of the weak learning assumption . . . 78
7.2.3 Approximately solving the game using Hedge . . . 79
7.3 Out-of-sample performance of boosting . . . 82
7.3.1 VC dimension of majority votes over hypotheses . . . 82
7.3.2 Generalization guarantee for boosting . . . 83
7.4 Margins theory . . . 84
7.4.1 An empirical puzzle . . . 84
7.4.2 Margin errors . . . 84
7.4.3 Uniform covers . . . 85
7.4.4 Boosting as an optimization algorithm . . . 88

8 Linear learning **91**
8.1 Linear learning paradigm . . . 91
8.1.1 Computational intractability . . . 91
8.1.2 Convex surrogate losses . . . 93
8.2 Stochastic gradient descent . . . 95
8.2.1 Stochastic convex optimization . . . 95
8.2.2 Stochastic gradients . . . 95
8.2.3 Stochastic (sub)gradient descent algorithm . . . 95
8.3 Approximation theory . . . 97
8.3.1 Linear learning with convex surrogates and SGD . . . 97
8.3.2 Approximation of Lipschitz functions by polynomials . . . 98
8.3.3 Approximation of Lipschitz ridge functions by polynomials . . . 100
8.4 Limitations of linear learning . . . 104
8.4.1 PAC learning by linear learning . . . 104
8.4.2 Some simple representations . . . 104
8.4.3 Lower bounds . . . 105
8.4.4 Parity functions . . . 106

Topic 1: Using expert advice

1.1 Basic scenario and modeling

1.1.1 Expert advice scenario

Imagine you are a weather reporter for the morning news. Every morning, you must predict the weather, which for simplicity is a binary outcome: either "rain" or "shine". At the end of each day, you become aware of the true outcome, and hence also whether you were correct or you made a mistake with your prediction. Because you are a television personality rather than a professional meteorologist, you consult some experts (e.g., Claude, Gemini, Kimi, Qwen) to get their predictions before you make your own. Can you learn which experts to trust, which are unreliable, etc.?

The scenario, which we call the *expert advice scenario*, is an example of *online learning*, where the "learning" process is on-going rather than something that only happens "offline".

1.1.2 Modeling

A typical modeling approach from statistics would posit that the outcome is governed by some simple mechanism, such as a small Markov chain. The experts may use different types of statistical methods to generate their predictions.

Our modeling approach, for now, is rather different. We make no assumption about the experts or outcomes: they could be controlled by an adversary seeking to make the weather reporter look foolish. To counterbalance the adversary (for now), we assume that (at least) one expert makes no mistakes. This is called the *realizability assumption*.

1.2 Basic learning algorithms for using expert advice

A *learning algorithm* for the expert advice scenario is a protocol for the weather reporter—a.k.a. the *learner*—to follow everyday to make predictions. For now, we only consider deterministic protocols.

1.2.1 Follow the Leader

One of the simplest learning algorithms is the *Follow the Leader (FTL)* algorithm:

> Sort the experts (e.g., alphabetically by name); designate the first expert as the leader. Each day, copy the prediction of the leader; upon a mistake, designate as leader the first expert who has not yet made a mistake.

Theorem 1.1. *In the expert advice scenario with N experts and the realizability assumption, FTL makes at most $N - 1$ mistakes.*

Proof. Say an expert is "alive" at a point in time if they have not yet made a mistake, and "dead" otherwise (yikes!). If FTL makes a mistake, then the expert it followed was alive at the start of the day but dies by the end of the day. With the realizability assumption, at least one expert never makes a mistake, so at most $N - 1$ experts can "die". So FTL makes at most $N - 1$ mistakes. □

Theorem 1.1 cannot generally be improved: assuming FTL picks an (alive) expert to follow without considering the expert predictions, it can be forced to make $N-1$ mistakes. To see this, imagine that the weather and the experts are controlled by an adversary who knows the details of how FTL is implemented. The adversary coordinates the weather and the expert predictions so that only the expert chosen by FTL makes a mistake (and hence so does FTL). The adversary can do this on $N-1$ days until only one expert remains. So the analysis of FTL in Theorem 1.1 is said to be "tight".

The *mistake bound* of a learning algorithm is an upper bound on the worst-case number of mistakes that it ever makes. So FTL has a mistake bound of $N-1$ with the realizability assumption.

1.2.2 Halving

One way to improve on the FTL algorithm is to improve on its mistake bound. The *Halving* (or *Majority*) algorithm of Barzdin and Freivald (1972), described below, does just that:

> Each day, copy the most popular prediction made by the "alive" experts.

(Here, we adopt the "alive"/"dead" terminology from the proof of Theorem 1.1.)

Theorem 1.2. *In the expert advice scenario with N experts and the realizability assumption, Halving makes at most $\lfloor \log_2(N) \rfloor$ mistakes.*

Proof. When Halving makes a mistake in a certain day, then at least half of the experts who are alive at the start of the day—specifically, those who also make the more popular prediction—also make a mistake and hence die at the end of the day. So the number of experts who are alive is at most half of what it was at the start of the day. All N experts are initially alive, and this number can be reduced this way at most $\lfloor \log_2(N) \rfloor$ times before it is 1. With the realizability assumption, one expert must always remain alive. Therefore Halving can make at most $\lfloor \log_2(N) \rfloor$ mistakes. □

Exercise 1.1. Show that the bound in Theorem 1.2 is tight. ◊

1.2.3 Barrier for all learning algorithms

Can any learning algorithm improve on the mistake bound of Halving? In general, the answer is no: every learner can be forced to make $\lfloor \log_2(N) \rfloor$ mistakes for a worst-case sequence of the N expert predictions and outcomes.

Theorem 1.3. *In the expert advice scenario with N experts and the realizability assumption, for each learning algorithm, there is some sequence of expert predictions and outcomes on which the learner makes at least $\lfloor \log_2(N) \rfloor$ mistakes.*

Proof. Consider $N' := 2^{\lfloor \log_2(N) \rfloor}$ of the N experts and any learning algorithm. We show that among these N' experts, for there is a sequence of the N' expert predictions and outcomes such that the learner makes $\log_2(N') = \lfloor \log_2(N) \rfloor$ mistakes, and yet one of the N' experts makes no mistakes. To see this, consider the rooted perfect binary tree of depth $\log_2(N')$, with leaves corresponding to the N' experts, and each edge annotated with either "rain" or "shine" so that each binary sequence in $\{\text{rain}, \text{shine}\}^{\log_2(N')}$ appears along the edges of some root-to-leaf path. The binary sequence leading to a leaf corresponding to an expert gives the predictions of that expert for the first $\log_2(N')$ days. Simulate the learning algorithm for $\log_2(N')$ days on a sequence of outcomes that always differs from the learner's predictions. The learner thus makes $\log_2(N')$ mistakes on this sequence of outcomes, but the expert corresponding to the leaf at the end of this root-to-leaf path makes no mistakes. □

1.3 Modeling and coping with imperfection

1.3.1 Experts with bounded imperfection

Another way to improve on the learning algorithms described above is to relax the assumptions under which it can reliably operate.

So far we have considered learning algorithms that crucially rely on the realizability assumption, so that assumption is a prime target for relaxation. Note that without any assumption about the experts, it could be that every learner makes a mistake in every round[1]. A less drastic relaxation is to assume that at least one expert makes at most K mistakes, for some non-negative integer K. Call this the *K-mistake assumption*. The strength of the K-mistake assumption is controlled by the modeling parameter K.

1.3.2 Halving with Resurrection

Neither FTL and Halving is well-defined when all experts are dead, and that is a possibility even with the 1-mistake assumption. A simple fix is to "resurrect" all experts whenever all of them are dead. Let *Halving with Resurrection* be the variant of Halving with this change:

> Proceed as in Halving, except: whenever all experts are dead at the end of a round, immediately "resurrect" them (i.e., declare all of them to be alive).

Theorem 1.4. *In the expert advice scenario with N experts and the K-mistake assumption, Halving with Resurrection makes at most $(\lfloor \log_2(N) \rfloor + 1) \times K + \lfloor \log_2(N) \rfloor$ mistakes.*

Proof. Partition all rounds into contiguous blocks of rounds between resurrections. If a resurrection occurs, then every expert has made at least one mistake since the last resurrection (or since before the first round). So the number of resurrections is at most K.

Within a block of rounds before a resurrection, the Halving with Resurrection can make at most $\lfloor \log_2(N) \rfloor + 1$ mistakes. This is because every mistake corresponds to at least half of the experts who are alive having made a mistake; this is the same argument as in the proof of Theorem 1.2, with the extra +1 coming from the one additional mistake that results in all experts being dead.

There can be one additional block of rounds after the last resurrection, and in this block, Halving with Resurrection makes at most $\lfloor \log_2(N) \rfloor$ mistakes, by the same argument as in the proof of Theorem 1.2.

So the total number of mistakes is bounded by

$$\begin{array}{lll} (\#\text{ mistakes in block before resurrection}) & \times (\#\text{ resurrections}) & + (\#\text{ mistakes after last resurrection}) \\ \leq (\lfloor \log_2(N) \rfloor + 1) & \times K & + \lfloor \log_2(N) \rfloor \end{array}$$

as claimed. □

Exercise 1.2. Show that the bound in Theorem 1.4 is tight. ◊

Exercise 1.3. Analyze the following two learning algorithms in the expert advice scenario with N experts and the K-mistake assumption:

[1] It is more common to use the term "round" or "trial" instead of "day", so we will henceforth adopt this terminology.

- Majority-of-Leaders: Let P_m be the set of experts who have made exactly m mistakes. Each day, copy the most popular prediction made by the experts in $P_{m^\star}$, where $m^\star = \min\{m \mid P_m \neq \emptyset\}$.
- Majority-of-Survivors: Each day, copy the most popular prediction made by experts who have made at most K mistakes. ◊

1.3.3 Barrier for all learning algorithms

The K-mistake assumption can be thought of as a promise about the "best expert", i.e., the expert who ultimately makes the fewest mistakes. The mistake bound in Theorem 1.4 shows a bound on the number of mistakes Halving with Resurrection can make in terms of the number of mistakes made by the best expert: roughly $\log(N) \times K$.

Is there a learning algorithm that improves on this mistake bound? The following theorem establishes a barrier on how much improvement is possible: roughly speaking, the $\log(N)$ factor cannot be reduced below two.

Theorem 1.5 (Cover, 1965)**.** *In the expert advice scenario with $N \geq 2$ experts and the K-mistake assumption, for each learning algorithm, there is some sequence of expert predictions and outcomes on which the learner makes at least $2K + 1$ mistakes.*

Proof. Consider two experts, one that always predicts "rain", and another that always predicts "shine". Now consider any learning algorithm, and simulate it for $2K + 1$ rounds on a sequence of outcomes that always differs from the learner's predictions. The learner thus makes $2K + 1$ mistakes on this sequence of outcomes, but one of the two experts makes at most K mistakes. □

Exercise 1.4. Improve Theorem 1.5 to $2K + \lfloor \log_2(N) \rfloor$ mistakes. ◊

There is still a large gap between $2K + \lfloor \log_2(N) \rfloor$ and $(\lfloor \log_2(N) \rfloor + 1) \times K + \lfloor \log_2(N) \rfloor$.

1.3.4 Weighted Majority

We now describe a gentler version of Halving that has a mistake bound with the K-mistake assumption that is much closer to the lower bound from Theorem 1.5. The algorithm is called *Weighted Majority (WM)*, due to Littlestone and Warmuth (1994). The description of the algorithm uses the following notation for the expert advice scenario with N experts.

- The N experts are denoted by $[N] := \{1, 2, \ldots, N\}$.
- The possible outcomes are denoted by $\{0, 1\}$.
- The rounds are indexed by $t \in \mathbb{N} := \{1, 2, 3, \ldots\}$.
- In round t, expert i's prediction is denoted by $b_{t,i} \in \{0, 1\}$, the learner prediction is denoted by $a_t \in \{0, 1\}$, and the actual outcome is denoted by $y_t \in \{0, 1\}$.

The algorithm begins by initializing "weights" associated with the experts. In a given round, the current weights are used along with the experts' predictions to form the learner's own prediction; the weights are updated at the end of the round after observing the outcome.

Algorithm 1.1: Weighted Majority for expert advice scenario with N experts.
Hyperparameter: $\beta \in (0,1)$.

- Initially, set $w_{1,i} := 1$ for all $i \in [N]$. This is the "weight" of expert i.
- In round t:
  - Observe experts' predictions $b_{t,1}, \ldots, b_{t,N} \in \{0,1\}$.
  - For each possible outcome $y \in \{0,1\}$, let

$$Z_{t,y} := \sum_{i=1}^{N} \mathbb{1}\{b_{t,i} = y\} w_{t,i}$$

    be the total weight of experts that predict y.
  - Predict

$$a_t := \begin{cases} 1 & \text{if } Z_{t,1} > Z_{t,0}, \\ 0 & \text{otherwise.} \end{cases}$$

  - Observe outcome $y_t \in \{0,1\}$.
  - Update experts' weights: for each $i \in [N]$,

$$w_{t+1,i} := \begin{cases} w_{t,i} & \text{if } y_t = b_{t,i}, \\ \beta w_{t,i} & \text{if } y_t \neq b_{t,i}. \end{cases}$$

We can think of the weight of an expert as a measure of its reliability. We consider all experts to be equally reliable at the start. And experts who make a mistake have their reliability score reduced by multiplying it by the hyperparameter $\beta \in (0,1)$.

For simplicity, we first analyze Weighted Majority with the concrete hyperparameter $\beta = 1/2$.

Theorem 1.6 (Littlestone and Warmuth, 1994)**.** *In the expert advice scenario with N experts and the K-mistake assumption, Weighted Majority with hyperparameter $\beta = 1/2$ makes at most $\frac{1}{\log_2(4/3)}(K + \log_2(N)) \approx 2.41(K + \log_2(N))$ mistakes.*

The idea behind the proof is similar to the analyses of FTL and Halving. Each mistake of the learner is charged to the weighted fraction of experts that also make a mistake. That weighted fraction is at least half the total weight. The proof of Theorem 1.6 will track the total weight of experts as it evolves over time. (We regard the total weight of experts as a "potential function".)

Proof of Theorem 1.6. For any round t, define

$$Z_t := \sum_{i=1}^{N} w_{t,i} = Z_{t,0} + Z_{t,1}$$

to be the total weight of experts at the start of round t. At the start of round 1, we have $Z_1 = N$.

Suppose Weighted Majority makes a mistake in round t. Let

$$\alpha_t := \frac{Z_{t,1-y_t}}{Z_t}$$

be the fraction (or proportion) of the total weight of experts that were mistaken. Since $\beta = 1/2$, the weight of these experts are halved in the update; the weight of the remaining experts are unchanged. We know $\alpha_t \geq 1/2$ by the way Weighted Majority chooses its (mistaken) prediction. Therefore

$$\begin{aligned} Z_{t+1} &= Z_{t,y_t} + \frac{1}{2} Z_{t,1-y_t} \\ &= (1-\alpha_t) Z_t + \frac{1}{2}\alpha_t Z_t \\ &= Z_t - \frac{1}{2}\alpha_t Z_t \\ &\leq Z_t - \frac{1}{4} Z_t \quad (\text{since } \alpha_t \geq 1/2) \\ &= \frac{3}{4} Z_t. \end{aligned}$$

Therefore, by induction, if Weighted Majority makes M mistakes after t rounds, then

$$Z_{t+1} \leq \left(\frac{3}{4}\right)^M Z_1 = \left(\frac{3}{4}\right)^M N.$$

The weight $w_{t+1,i}$ of expert i at the start of round $t+1$ is determined by the number of mistakes expert i has made after t rounds: if expert i has made M_i mistakes, then $w_{t+1,i} = 2^{-M_i}$. Therefore, if there is an expert who has made at most K mistakes after t rounds, then

$$Z_{t+1} \geq 2^{-K}.$$

We now have upper- and lower-bounds on Z_{t+1}, so we can combine the bounds

$$2^{-K} \leq Z_{t+1} \leq \left(\frac{3}{4}\right)^M N$$

and rearrange to get $M \leq (K + \log_2 N)/\log_2(4/3)$. □

For general $\beta \in (0,1)$, the same analysis from the proof of Theorem 1.6 gives the following.

Theorem 1.7 (Littlestone and Warmuth, 1994)**.** *In the expert advice scenario with N experts and the K-mistake assumption, Weighted Majority with hyperparameter $\beta \in (0,1)$ makes at most*

$$\frac{\log(1/\beta)K + \log N}{\log(2/(1+\beta))}. \tag{1.1}$$

mistakes.

The base of the logarithm clearly does not matter in (1.1), but to make it concrete, let us always assume log uses base e. Here is what the mistake bound in (1.1) looks like for different values of β.

β	Equation (1.1)
1/2	$\approx 2.41K + 3.48\ln(N)$
2/3	$\approx 2.22K + 5.48\ln(N)$
3/4	$\approx 2.15K + 7.49\ln(N)$
4/5	$\approx 2.12K + 9.49\ln(N)$

The limit of the pre-factor on K is 2 as $\beta \to 1$ (as can be checked by L'Hôpital's rule), but the pre-factor on $\ln(N)$ diverges to $+\infty$ as $\beta \to 1$.

Topic 2: Online allocation

2.1 Basic scenario and performance measures

2.1.1 Online allocation scenario

Online allocation is an abstract online learning scenario proposed by Freund and Schapire (1997). Define the following sets of N-vectors:

$$\mathbb{R}_+^N := \left\{x = (x_1, \ldots, x_N) \in \mathbb{R}^N \;\middle|\; x_i \geq 0 \text{ for all } i \in [N]\right\},$$
$$\Delta^{N-1} := \left\{x = (x_1, \ldots, x_N) \in \mathbb{R}_+^N \;\middle|\; \sum_{i=1}^N x_i = 1\right\}.$$

The set $\mathbb{R}_+^N$ is the non-negative orthant of $\mathbb{R}^N$, and the set Δ^{N-1} is the probability simplex in $\mathbb{R}^N$. Also define the standard inner product $\langle \cdot, \cdot \rangle : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ by

$$\langle x, y \rangle := \sum_{i=1}^N x_i y_i \quad \text{for } x = (x_1, \ldots, x_N) \in \mathbb{R}^N \text{ and } y = (y_1, \ldots, y_N) \in \mathbb{R}^N.$$

Similar to the expert advice scenario, the online allocation scenario involves a sequence of rounds in which the learner makes choices and observes/incurs consequences. In round t:

- The learner chooses an allocation vector $p_t \in \Delta^{N-1}$.
- A loss vector $\ell_t \in \mathbb{R}_+^N$ is revealed to the learner.
- The learner incurs loss $\langle p_t, \ell_t \rangle$. (Freund and Schapire call this the *mixture loss*.)

Online allocation is an interesting learning scenario in its own right, but there are at least three additional motivations. First, it provides a connection to the expert advice scenario and shows how to seemingly break one of the lower bounds. Second, algorithms for online allocation can be generalized to work in many other learning scenarios (such as multi-armed bandits and online convex optimization). Third, algorithms for online allocation can be used to give simple proofs of important theorems that are used in learning theory.

2.1.2 Performance measures

The cumulative (a.k.a. total, net) loss of the learner after T rounds is

$$L_T := \sum_{t=1}^T \langle p_t, \ell_t \rangle.$$

The number T is called the *horizon*. The cumulative loss of the allocation vector $q \in \Delta^{N-1}$ is

$$L_{T,q} := \sum_{t=1}^T \langle q, \ell_t \rangle.$$

If q is an elementary basis vector[1], then we write $L_{T,i}$ to mean L_{T,e_i}. The *regret of the learner to q after T rounds* is

$$R_{T,q} := L_T - L_{T,q}$$

(with $R_{T,i} = R_{T,e_i}$), and

$$R_T := \max_{q \in \Delta^{N-1}} R_{T,q} = L_T - \min_{q \in \Delta^{N-1}} L_{T,q}$$

is the *regret of the learner (to the best allocation) after T rounds*.

Regret is named as such because it measures the extra loss incurred by the learner compared to having used the "best-in-hindsight" allocation in all T rounds. By linearity, we have $\sum_{t=1}^T \langle q, \ell_t \rangle = \langle q, \sum_{t=1}^T \ell_t \rangle$, so the "best-in-hindsight" cumulative loss achievable by a single allocation vector is determined by minimizing a linear function (given by $\sum_{t=1}^T \ell_t$) over the probability simplex. This minimum value will always be achieved at a vertex of the simplex, i.e., one of the N elementary basis vectors. So R_T can also be written as

$$R_T = \max_{i \in [N]} L_T - L_{T,i} = L_T - \min_{i \in [N]} L_{T,i}.$$

2.2 Connection to the expert advice scenario

2.2.1 Randomized Weighted Majority

Randomized Weighted Majority (RWM) from Littlestone and Warmuth (1994) is the following.

Algorithm 2.1: Randomized Weighted Majority for expert advice scenario with N experts. Hyperparameter: $\beta \in (0, 1)$.

- Initially, set $w_{1,i} := 1$ for all $i \in [N]$.
- In round t:
  - Let $p_t := (p_{t,1}, \ldots, p_{t,N}) \in \Delta^{N-1}$ be defined by

    $$p_{t,i} := \frac{w_{t,i}}{Z_t} \quad \text{for all } i \in [N],$$

    where $Z_t := \sum_{i=1}^N w_{t,i}$ is the total weight (as in the proof of Theorem 1.7).
  - Randomly draw $i_t \in [N]$ so that, for each $i \in [N]$, $i_t = i$ with probability $p_{t,i}$.
  - Observe experts' predictions $b_t := (b_{t,1}, \ldots, b_{t,N}) \in \{0, 1\}^N$.
  - Predict $a_t := b_{t,i_t} \in \{0, 1\}$.
  - Observe outcome $y_t \in \{0, 1\}$.
  - Update experts' weights: for each $i \in [N]$,

    $$w_{t+1,i} := \begin{cases} w_{t,i} & \text{if } y_t = b_{t,i}, \\ \beta w_{t,i} & \text{if } y_t \neq b_{t,i}. \end{cases}$$

[1]The N *elementary (or coordinate) basis vectors* (in $\mathbb{R}^N$) are $e_1, \ldots, e_N$, where e_i has a 1 in the i-th entry and 0 elsewhere.

RWM explicitly uses randomness to make its predictions via the random indices i_t. Specifically, in each round t, the learner's prediction a_t is determined as a function of both i_t and b_t.

Note that use of randomness alone does not allow RWM to subvert the lower bound from Theorem 1.5 (for the expert advice scenario with N experts and the K-mistake assumption). The proof given for Theorem 1.5 was stated for deterministic learning algorithms. However, the argument still holds by allowing the adversary to adaptively pick y_t as a function of a_t in each round t: simply set $y_t := 1 - a_t$ on-the-fly. So the learner makes a mistake in each of the first $2K + 1$ rounds, but one of the "constant" experts makes at most K mistakes.

2.2.2 Non-adaptive adversaries and the connection to online allocation

To subvert the lower bound—and to connect RWM to online allocation—it will suffice to assume that the adversary is *non-adaptive*: in each round t, the adversary chooses the b_t and y_t in a way that does not depend on the random choices of the learner in round t. The assumption ensures that (b_t, y_t) and i_t are conditionally independent given p_t. In this case,

$$\begin{aligned}
\Pr(a_t \neq y_t \mid p_t, b_t, y_t) &= \sum_{i=1}^{N} \Pr(i_t = i \mid p_t, b_t, y_t) \mathbb{1}\{b_{t,i} \neq y_t\} \\
&= \sum_{i=1}^{N} \Pr(i_t = i \mid p_t) \mathbb{1}\{b_{t,i} \neq y_t\} \\
&= \sum_{i=1}^{N} p_{t,i} \mathbb{1}\{b_{t,i} \neq y_t\} \\
&= \langle p_t, \ell_t \rangle
\end{aligned}$$

where

$$\ell_{t,i} := \mathbb{1}\{b_{t,i} \neq y_t\} \in \{0,1\} \quad \text{and} \quad \ell_t := (\ell_{t,1}, \ldots, \ell_{t,N}) \in \{0,1\}^N.$$

RWM chooses p_t based on $\ell_1, \ldots, \ell_{t-1}$ from previous rounds (and the initial weights $w_{1,1}, \ldots, w_{1,N}$), and hence it can be viewed as a learning algorithm for online allocation where ℓ_t is the loss vector in round t. The total loss of RWM after T rounds is

$$\sum_{t=1}^{T} \langle p_t, \ell_t \rangle = \sum_{t=1}^{T} \Pr(a_t \neq y_t \mid p_t, b_t, y_t).$$

The expected value of the above quantity is the expected number of mistakes committed by RWM after T rounds.

2.2.3 Foreshadowing: breaking through the pre-factor of two

For the expert advice scenario with N experts, the K-mistake assumption, and the non-adaptive adversary assumption, the expected number of mistakes of RWM can be made to be at most

$$(1 + \epsilon)K + \frac{2 \ln(N)}{\epsilon}$$

for any $\epsilon \in (0, 1)$ (with a suitable choice of hyperparameter β depending on ϵ, K, and N). The key point is that the pre-factor on K can be made arbitrarily close to 1. See Corollary 2.2.

2.3 Algorithm for online allocation

2.3.1 Hedge algorithm

We now describe an algorithm called Hedge due to Freund and Schapire (1997) for the online allocation scenario. It is a generalization of RWM (when viewed as an algorithm for a special case of online allocation).

Algorithm 2.2: Hedge for online allocation in N-space.

Hyperparameter: $\eta > 0$.

- Initially, set $w_{1,i} := 1$ for all $i \in [N]$.
- In round t:
  - Choose allocation $p_t := (p_{t,1}, \ldots, p_{t,N}) \in \Delta^{N-1}$, defined by

$$p_{t,i} := \frac{w_{t,i}}{Z_t} \quad \text{for all } i \in [N],$$

    where $Z_t := \sum_{i=1}^N w_{t,i}$ is the total weight.
  - Observe loss vector $\ell_t := (\ell_{t,1}, \ldots, \ell_{t,N}) \in \mathbb{R}_+^N$.
  - Update weights: for each $i \in [N]$,

$$w_{t+1,i} := w_{t,i} \exp(-\eta \ell_{t,i}).$$

We have changed the hyperparameter from the fraction $\beta \in (0,1)$ to the positive number $\eta > 0$. But really nothing has changed, since in the special case where $\ell_{t,i} \in \{0,1\}$, the weight updates for Hedge and RWM are exactly the same when we take $\beta = \exp(-\eta)$. It turns out to be convenient to (hyper-)parameterize the weight updates with η to make certain standard Taylor approximations more recognizable in the analysis.

Note that Hedge is a completely deterministic. Therefore, if the loss vectors in online allocation are chosen by an adversary, the entire sequence of loss vectors may as well be fixed in advance. It is only in Hedge/RWM's interpretation as a randomized learning algorithm for the expert advice scenario where the distinction between adaptive and non-adaptive adversaries makes a difference.

Theorem 2.1 (Freund and Schapire, 1997)**.** *In the online allocation scenario in N-space, for any T and any sequence of loss vectors $\ell_1, \ldots, \ell_T \in \mathbb{R}_+^N$, Hedge with hyperparameter $\eta > 0$ guarantees that*

$$R_T \le \frac{\ln(N)}{\eta} + \frac{\eta}{2} \sum_{t=1}^T \langle p_t, \ell_t^2 \rangle$$

where $\ell_t^2 := (\ell_{t,1}^2, \ldots, \ell_{t,N}^2) \in \mathbb{R}_+^N$ for all t.

Corollary 2.1. *In the setup of Theorem 2.1, if $\ell_t \in [0,1]^N$ for all t, then*

$$R_T \le \frac{\ln(N)}{\eta} + \frac{\eta T}{2} \quad \textit{and} \quad (1 - \eta/2) L_T \le L^\star + \frac{\ln(N)}{\eta}$$

where

$$L^\star := \min_{i \in [N]} L_{T,i}.$$

In particular, there is a choice of $\eta > 0$ (depending on N and T) such that

$$R_T \leq \sqrt{2T \ln(N)},$$

and there is a choice of $\eta \in (0, 1)$ (depending on N, T, and $L^\star$, assuming $L^\star > 0$) such that

$$R_T \leq \sqrt{2L^\star \ln(N) + \ln^2(N)} + \ln(N).$$

Exercise 2.1. Determine the values of η that give the final regret bounds in Corollary 2.1. ◊

We say a learner has *no-regret* if its time-averaged regret to the best allocation after T rounds is non-positive in the limit as $T \to \infty$:

$$\limsup_{T \to \infty} \frac{R_T}{T} \leq 0.$$

In other words, the regret is sublinear in T. This is indeed the case for Hedge with an appropriate choice of hyperparameter, as seen in Corollary 2.1. No-regret is an important property of a learning algorithm for online allocation and is taken as the standard for what it means to successfully learn.

The second guarantee in Corollary 2.1 is called an *$L^\star$-bound* or *first-order bound* in that it replaces the dependence on the horizon T with a first-order quantity, namely $L^\star$. (A second-order quantity would be something involving the variability of per-round losses.) In terms of the time-averaged regret, the first-order bound interpolates between two different "rates" (up to constant factors):

$$L^\star = T \implies \frac{R_T}{T} \leq O\left(\sqrt{\frac{\log(N)}{T}}\right);$$
$$L^\star = 0 \implies \frac{R_T}{T} \leq O\left(\frac{\log(N)}{T}\right).$$

The $T^{-1/2}$ rate is referred to as a *slow rate*, while the T^{-1} rate is referred to as a *fast rate*.

Corollary 2.2. *In the expert advice scenario with N experts, the K-mistake assumption, and the non-adaptive adversary assumption, Randomized Weighted Majority with hyperparameter $\beta = \exp(-\eta)$ for $\eta \in (0, 2)$ makes at most*

$$\frac{1}{1 - \eta/2}\left(K + \frac{\ln(N)}{\eta}\right)$$

mistakes in expectation.

The statement of Theorem 2.1 is slightly different from how it is stated by Freund and Schapire (1997), but not in any essential way for our purposes. The proof below is based on that from Freund and Schapire (1999), which uses concepts from information theory, discussed next.

2.3.2 Entropy and relative entropy

The *(Shannon) entropy* $\mathsf{H}(q)$ of a (discrete) probability distribution $q = (q_1, \ldots, q_N) \in \Delta^{N-1}$ over $[N]$ is defined by

$$\mathsf{H}(q) := \sum_{i=1}^{N} q_i \ln\left(\frac{1}{q_i}\right).$$

(Here, we regard $0\ln(0)$ as 0 rather than $-\infty$.) Entropy is a central concept in information theory and many other fields, with several interpretations. One way to think about entropy is as a measure of average "surprise" one experiences when observing the value of a random variable. The idea is that you experience more surprise when you see rare events than when you see common events. Letting $\ln(1/q_i)$ be the quantitative measure of surprise for the outcome i, we then see $\mathsf{H}(q)$ to be exactly the expected surprise. (There are many other interpretations of entropy, such as the average number of bits needed to communicate a random message $X \sim q$, but we will not discuss them here.) The maximum value of $\mathsf{H}(q)$ is $\ln(N)$, achieved by the uniform distribution $(1/N, \ldots, 1/N)$.

Now consider two probability distributions, $p, q \in \Delta^{N-1}$. The *cross entropy* of q from p is

$$\mathsf{CE}(q, p) := \sum_{i=1}^{N} q_i \ln\left(\frac{1}{p_i}\right).$$

Cross entropy measures the average surprise when outcomes are distributed according to q, but the measure of surprise is (perhaps mistakenly) based on p. The *relative entropy*[2] of q from p is

$$\mathsf{RE}(q, p) := \sum_{i=1}^{N} q_i \ln\left(\frac{q_i}{p_i}\right) = \mathsf{CE}(q, p) - \mathsf{H}(q)$$

is the excess average surprise in this scenario. Note that $\mathsf{RE}(q, p)$ is well-defined only if q is dominated by p (written $q \ll p$, which means that for any i, we have $p_i = 0$ only if $q_i = 0$ as well). If $q \not\ll p$, then we define $\mathsf{RE}(q, p) := +\infty$. Also note that $\mathsf{RE}(q, p)$ is potentially (very) different from $\mathsf{RE}(p, q)$.

Proposition 2.1 (Gibbs' inequality)**.** $\mathsf{RE}(q, p) \geq 0$ *with equality if and only if* $q = p$.

Proof. A Taylor expansion of ln gives

$$\ln(1 + x) \leq x \quad \text{for all } x > -1. \tag{2.1}$$

Without loss of generality, assume $q \ll p$. Then

$$\begin{aligned}
\mathsf{RE}(q, p) &= -\sum_{i=1}^{N} q_i \ln\left(1 + \frac{p_i}{q_i} - 1\right) \\
&\geq -\sum_{i=1}^{N} q_i \left(\frac{p_i}{q_i} - 1\right) && \text{(by (2.1))} \\
&= -\sum_{i=1}^{N} p_i - q_i = 0.
\end{aligned}$$

Equality holds iff $p_i/q_i - 1 = 0$ for all i, which is equivalent to $p_i = q_i$ for all i. □

[2] Relative entropy is also known as *Kullback-Leibler divergence* and *I-divergence*.

2.3.3 Proof of the Hedge theorem

To prove Theorem 2.1, it suffices to show that, for any $q \in \Delta^{N-1}$, we have

$$\sum_{t=1}^{T} \langle p_t - q, \ell_t \rangle \le \frac{\ln(N)}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \langle p_t, \ell_t^2 \rangle.$$

The following is the key lemma in the proof.

Lemma 2.1. *Fix any $p \in \Delta^{N-1}$ and any $\ell \in \mathbb{R}_+^N$, and define $p' \in \Delta^{N-1}$ by*

$$p'_i := \frac{p_i \exp(-\eta \ell_i)}{\sum_{j=1}^{N} p_j \exp(-\eta \ell_j)} \quad \text{for all } i \in [N].$$

Then for any $q \in \Delta^{N-1}$ such that $q \ll p$ and any $\eta > 0$,

$$\mathsf{RE}(q, p') - \mathsf{RE}(q, p) \le \eta \langle q - p, \ell \rangle + \frac{\eta^2}{2} \langle p, \ell^2 \rangle.$$

where $\ell^2 := (\ell_1^2, \ldots, \ell_N^2)$.

Proof. Taylor expansion of exp gives

$$\exp(x) \le 1 + x + \frac{x^2}{2} \quad \text{for all } x \le 0. \tag{2.2}$$

Therefore

$$\begin{aligned}
\mathsf{RE}(q, p') - \mathsf{RE}(q, p) = \sum_{i=1}^{N} q_i \ln\left(\frac{p_i}{p'_i}\right) &= \sum_{i=1}^{N} q_i \ln\left(\frac{\sum_{j=1}^{N} p_j \exp(-\eta \ell_j)}{\exp(-\eta \ell_i)}\right) \\
&= \eta \langle q, \ell \rangle + \ln\left(\sum_{j=1}^{N} p_j \exp(-\eta \ell_j)\right) \\
&\le \eta \langle q, \ell \rangle + \ln\left(\sum_{j=1}^{N} p_j \left(1 - \eta \ell_j + \frac{\eta^2}{2} \ell_j^2\right)\right) \qquad \text{(by (2.2))} \\
&= \eta \langle q, \ell \rangle + \ln\left(1 - \eta \langle p, \ell \rangle + \frac{\eta^2}{2} \langle p, \ell^2 \rangle\right) \\
&\le \eta \langle q, \ell \rangle - \eta \langle p, \ell \rangle + \frac{\eta^2}{2} \langle p, \ell^2 \rangle \qquad \text{(by (2.1)).} \quad \square
\end{aligned}$$

Using Lemma 2.1, we have for any t and any $q \in \Delta^{N-1}$,

$$\eta \langle p_t - q, \ell_t \rangle \le \mathsf{RE}(q, p_t) - \mathsf{RE}(q, p_{t+1}) + \frac{\eta^2}{2} \langle p_t, \ell_t^2 \rangle.$$

Sum both sides over $t = 1, \dots, T$ to obtain

$$\begin{aligned}\eta \sum_{t=1}^{T} \langle p_t - q, \ell_t \rangle &\le \sum_{t=1}^{T} \left(\mathsf{RE}(q, p_t) - \mathsf{RE}(q, p_{t+1}) + \frac{\eta^2}{2} \langle p_t, \ell_t^2 \rangle \right) \\ &= \underbrace{\mathsf{RE}(q, p_1)}_{= \ln(N) - \mathsf{H}(q)} - \underbrace{\mathsf{RE}(q, p_{T+1})}_{\ge 0} + \frac{\eta^2}{2} \sum_{t=1}^{T} \langle p_t, \ell_t^2 \rangle \\ &\le \ln(N) + \frac{\eta^2}{2} \sum_{t=1}^{T} \langle p_t, \ell_t^2 \rangle\end{aligned}$$

where the last inequality uses Proposition 2.1 and the definition of p_1. □

Exercise 2.2. Suppose we consider a generalization of online allocation in which loss vectors may have components that are negative. Explain how the analysis of (and performance guarantees for) Hedge in Theorem 2.1 and Corollary 2.1 should change. For Corollary 2.1, you may assume that $\max_{i \in [N]} \ell_{t,i} - \min_{i \in [N]} \ell_{t,i} \le 1$ for all t. ♢

2.3.4 Improved analysis of Hedge for bounded losses

The analysis of Hedge in Theorem 2.1 allows for arbitrary non-negative losses, while Corollary 2.1 derives consequences of this analysis for bounded losses. In Theorem 2.2, we directly use boundedness in the analysis to obtain somewhat sharper results.

Theorem 2.2. *In the online allocation scenario in N-space, for any T and any sequence of loss vectors $\ell_1, \dots, \ell_T \in [0,1]^N$, Hedge with hyperparameter $\eta > 0$ guarantees that, for all $q \in \Delta^{N-1}$,*

$$\sum_{t=1}^{T} \langle p_t - q, \ell_t \rangle \le \frac{\mathsf{RE}(q, p_1)}{\eta} + \frac{\eta T}{8}$$

and

$$\sum_{t=1}^{T} \langle p_t - q, \ell_t \rangle \le \frac{\mathsf{RE}(q, p_1)}{\eta} + \left(\frac{\exp(\eta) - \eta - 1}{\eta} \right) \sum_{t=1}^{T} \big(\langle p_t, \ell_t^2 \rangle - \langle p_t, \ell_t \rangle^2 \big)$$

where $\ell_t^2 := (\ell_{t,1}^2, \dots, \ell_{t,N}^2) \in [0,1]^N$ for all t.

The proof of Theorem 2.2 is based on the following improved version of Lemma 2.1, which uses results about logarithmic moment generating functions (log mgfs) of bounded random variables, discussed in Section 4.3.

Lemma 2.2. *Fix any $p \in \Delta^{N-1}$ and any $\ell \in [0,1]^N$, and define $p' \in \Delta^{N-1}$ by*

$$p'_i := \frac{p_i \exp(-\eta \ell_i)}{\sum_{j=1}^{N} p_j \exp(-\eta \ell_j)} \quad \text{for all } i \in [N].$$

Then for any $q \in \Delta^{N-1}$ such that $q \ll p$ and any $\eta > 0$,

$$\mathsf{RE}(q, p') - \mathsf{RE}(q, p) \le \eta \langle q - p, \ell \rangle + \min \left\{ \frac{\eta^2}{8}, \ (\exp(\eta) - \eta - 1) \big(\langle p, \ell^2 \rangle - \langle p, \ell \rangle^2 \big) \right\}$$

where $\ell^2 := (\ell_1^2, \dots, \ell_N^2)$.

Proof. In the proof of Lemma 2.1, the quantity $\ln(\sum_{j=1}^N p_j \exp(-\eta \ell_j))$ can be interpreted as the log mgf of a random variable $\mathbf{z}$ with $\Pr(\mathbf{z} = -\ell_j) = p_j$ for each $j \in [N]$, evaluated at η:

$$\ln\left(\sum_{j=1}^N p_j \exp(-\eta \ell_j)\right) = \mathsf{K}_{\mathbf{z}}(\eta) = \eta \operatorname{E}[\mathbf{z}] + \mathsf{K}_{\mathbf{z}-\operatorname{E}[\mathbf{z}]}(\eta).$$

Observe that $\mathbf{z} - \operatorname{E}[\mathbf{z}]$ takes values in $[-1 - \operatorname{E}[\mathbf{z}], -\operatorname{E}[\mathbf{z}]]$, so by Lemma 4.1,

$$\mathsf{K}_{\mathbf{z}-\operatorname{E}[\mathbf{z}]}(\eta) \le \frac{\eta^2}{8},$$

and by Lemma 4.3,

$$\mathsf{K}_{\mathbf{z}-\operatorname{E}[\mathbf{z}]}(\eta) \le \operatorname{var}(\mathbf{z})(\exp(\eta) - \eta - 1).$$

Moreover,

$$\operatorname{E}[\mathbf{z}] = -\langle p, \ell \rangle, \quad \operatorname{var}(\mathbf{z}) = \langle p, \ell^2 \rangle - \langle p, \ell \rangle^2.$$

The rest of the proof is the same as in Lemma 2.1. □

2.4 Application to zero-sum games

2.4.1 Zero-sum games

A *zero-sum game* is a two-player game in which the loss of one player is the gain of the other player. The loss/gain is specified by an $N \times d$ payoff matrix A; for simplicity, assume the entries of A are from $[0, 1]$. The "row" player has N possible actions, and the "column" player has d possible actions. For example, suppose the row player is the goalkeeper, who can dive left, stay put, or dive right; and the column player is the penalty kicker, who can kick left, kick straight, or kick right. The payoff matrix (where the actions are ordered as in the previous sentence) is

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

So, for example, if the goalkeeper stays put, then the penalty kicker gains 1 by kicking left or kicking right (either will do). On the other hand, if the penalty kicker kicks right, then the goalkeeper loses 0 by diving left. Notice that given the action of one player, there is a (possibly non-unique) best action that the other player can choose; this is called the *best response* for that other player.

The players are permitted to use *mixed strategies*, which are probability distributions over their respective action sets. If the row player chooses $p \in \Delta^{N-1}$ and the column player chooses $q \in \Delta^{d-1}$, then the loss of the row player (equivalently, the gain of the column player) is $p^\mathsf{T} A q$. For example, the penalty kicker may choose to kick left and kick right with probability $1/2$ each ($q = (1/2, 0, 1/2)$). This is a best response mixed strategy given that the goalkeeper chooses to stay put ($p = (0, 1, 0)$).

We have been ambiguous about exactly when the choices of the players are made. If the row player chooses p first, and then the column player chooses q in response to seeing p, then the best possible loss of the row player (assuming the column player chooses a best response) is

$$\min_{p \in \Delta^{N-1}} \max_{q \in \Delta^{d-1}} p^\mathsf{T} A q. \tag{2.3}$$

However, if the column player chooses q first, and then the row player chooses p in response to seeing q, then the best possible gain of the column player (assuming the row player chooses a best response) is

$$\max_{q \in \Delta^{d-1}} \min_{p \in \Delta^{N-1}} p^\mathsf{T} A q. \tag{2.4}$$

Von Neumann's "minimax theorem" states that (2.3) and (2.4) are equal; there is no disadvantage to "going first" in a zero-sum game. This common value is called the *value of the game*.

Theorem 2.3 (Minimax theorem of von Neumann, 1928)**.** *For any $A \in [0,1]^{N \times d}$,*

$$\max_{q \in \Delta^{d-1}} \min_{p \in \Delta^{N-1}} p^\mathsf{T} A q = \min_{p \in \Delta^{N-1}} \max_{q \in \Delta^{d-1}} p^\mathsf{T} A q.$$

Exercise 2.3. Find the optimal mixed strategies for both players with the payoff matrix given at the start of Section 2.4.1. What is the value of the game? ◊

To prove Theorem 2.3, it suffices to show that the max min is at most the min max, and also that min max is at most the max min. The first part is rather intuitive and easy to prove. The other part is less obvious; Freund and Schapire (1999) showed that this part follows from the existence of no-regret algorithms for online allocation such as Hedge.

2.4.2 Proof of the easy part

Proposition 2.2. *For any $A \in [0,1]^{N \times d}$,*

$$\max_{q \in \Delta^{d-1}} \min_{p \in \Delta^{N-1}} p^\mathsf{T} A q \le \min_{p \in \Delta^{N-1}} \max_{q \in \Delta^{d-1}} p^\mathsf{T} A q.$$

Proof. Fix any $q' \in \Delta^{d-1}$. Then, for every $p \in \Delta^{N-1}$,

$$p^\mathsf{T} A q' \le \max_{q \in \Delta^{d-1}} p^\mathsf{T} A q.$$

Let p' be the choice of $p \in \Delta^{N-1}$ that minimizes the right-hand side, so

$$\begin{aligned}(p')^\mathsf{T} A q' &\le \max_{q \in \Delta^{d-1}} (p')^\mathsf{T} A q \\ &= \min_{p \in \Delta^{N-1}} \max_{q \in \Delta^{d-1}} p^\mathsf{T} A q.\end{aligned}$$

We clearly also have

$$(p')^\mathsf{T} A q' \ge \min_{p \in \Delta^{N-1}} p^\mathsf{T} A q',$$

so combining the previous two displayed inequalities gives

$$\min_{p \in \Delta^{N-1}} p^\mathsf{T} A q' \le \min_{p \in \Delta^{N-1}} \max_{q \in \Delta^{d-1}} p^\mathsf{T} A q$$

Since this inequality holds for every $q' \in \Delta^{d-1}$, it also holds for the choice of q' that maximizes the left-hand side:

$$\max_{q \in \Delta^{d-1}} \min_{p \in \Delta^{N-1}} p^\mathsf{T} A q \le \min_{p \in \Delta^{N-1}} \max_{q \in \Delta^{d-1}} p^\mathsf{T} A q.$$ □

2.4.3 Proof of the other part via no-regret online allocation

The second part is the "hard" direction, which involves the execution of Hedge in an instance of the online allocation scenario that is constructed on-the-fly.

Proposition 2.3. *For any $A \in [0,1]^{N \times d}$,*

$$\min_{p \in \Delta^{N-1}} \max_{q \in \Delta^{d-1}} p^\mathsf{T} A q \le \max_{q \in \Delta^{d-1}} \min_{p \in \Delta^{N-1}} p^\mathsf{T} A q.$$

Proposition 2.3 is a direct consequence of the following proposition.

Proposition 2.4. *For any $A \in [0,1]^{N \times d}$, consider the execution of Hedge with $\eta := 2\sqrt{2\ln(N)/T}$ for selecting $p_1, \dots, p_T \in \Delta^{N-1}$ with loss vectors $\ell_1, \dots, \ell_T \in [0,1]^N$ defined by $\ell_t := A q_t$ for all t, where $q_t \in \{e_1, \dots, e_d\}$ is chosen so that*

$$p_t^\mathsf{T} A q_t = \max_{q \in \Delta^{d-1}} p_t^\mathsf{T} A q.$$

Define

$$\bar{p} := \frac{1}{T} \sum_{t=1}^{T} p_t \quad \text{and} \quad \bar{q} := \frac{1}{T} \sum_{t=1}^{T} q_t.$$

Then

$$\min_{p \in \Delta^{N-1}} \max_{q \in \Delta^{d-1}} p^\mathsf{T} A q \le \max_{q \in \Delta^{d-1}} \bar{p}^\mathsf{T} A q \le \max_{q \in \Delta^{d-1}} \min_{p \in \Delta^{N-1}} p^\mathsf{T} A q + \sqrt{\frac{\ln(N)}{2T}},$$

$$\text{and} \quad \min_{p \in \Delta^{N-1}} p^\mathsf{T} A \bar{q} \ge \min_{p \in \Delta^{N-1}} \max_{q \in \Delta^{d-1}} p^\mathsf{T} A q - \sqrt{\frac{\ln(N)}{2T}}.$$

Proposition 2.3 follows from Proposition 2.4 by letting $T \to \infty$. Proposition 2.4 is useful because it additionally shows how to construct $\bar{p}$ and $\bar{q}$ that approximately achieve the value of the game. In fact, the constructed $\bar{q}$ is sparse, which is a useful property in some applications.

Proof of Proposition 2.4. By Theorem 2.2 (with the given setting of η), we have

$$\sum_{t=1}^{T} \langle p_t, \ell_t \rangle \le \min_{p \in \Delta^{N-1}} \sum_{t=1}^{T} \langle p, \ell_t \rangle + \sqrt{\frac{T \ln(N)}{2}}.$$

This implies

$$\frac{1}{T} \sum_{t=1}^{T} p_t^\mathsf{T} A q_t \le \min_{p \in \Delta^{N-1}} \frac{1}{T} \sum_{t=1}^{T} p^\mathsf{T} A q_t + \sqrt{\frac{\ln(N)}{2T}}. \tag{2.5}$$

Then

$$\begin{aligned}
\min_{p\in\Delta^{N-1}} \max_{q\in\Delta^{n-1}} p^\mathsf{T} A q &\le \max_{q\in\Delta^{d-1}} \bar{p}^\mathsf{T} A q \quad (\text{since } \bar{p}\in\Delta^{N-1}) \\
&= \max_{q\in\Delta^{d-1}} \frac{1}{T}\sum_{t=1}^{T} p_t^\mathsf{T} A q \\
&\le \frac{1}{T}\sum_{t=1}^{T} \max_{q\in\Delta^{n-1}} p_t^\mathsf{T} A q \\
&= \frac{1}{T}\sum_{t=1}^{T} p_t^\mathsf{T} A q_t \quad (\text{by definition of } q_t) \\
&\le \min_{p\in\Delta^{N-1}} \frac{1}{T}\sum_{t=1}^{T} p^\mathsf{T} A q_t + \sqrt{\frac{\ln(N)}{2T}} \quad (\text{by (2.5)}) \\
&= \min_{p\in\Delta^{N-1}} p^\mathsf{T} A \bar{q} + \sqrt{\frac{\ln(N)}{2T}} \\
&\le \max_{q\in\Delta^{d-1}} \min_{p\in\Delta^{N-1}} p^\mathsf{T} A q + \sqrt{\frac{\ln(N)}{2T}}.
\end{aligned}$$

□

2.5 Application to online convex optimization

2.5.1 Convex functions

A function $f\colon \mathbb{R}^N \to \mathbb{R}$ is *convex*[3] if, for every $x_\mathrm{o} \in \mathbb{R}^N$, there exists $\lambda \in \mathbb{R}^N$ such that

$$f(x) \ge f(x_\mathrm{o}) + \langle x - x_\mathrm{o}, \lambda\rangle, \quad \forall x \in \mathbb{R}^N. \tag{2.6}$$

Linear and affine functions are convex functions where the inequality can be made to be an equality. That is, if $f(x) = \langle x, m\rangle + b$ for some $m \in \mathbb{R}^N$ and $b \in \mathbb{R}$, then (2.6) holds with equality with $\lambda = m$. Sums of convex functions are also convex functions. The set of vectors $\lambda \in \mathbb{R}^N$ satisfying (2.6), denoted $\partial f(x_\mathrm{o})$, is the set of *subgradients* of f at x_o. If f is also differentiable at x_o, then $\partial f(x_\mathrm{o})$ contains only the gradient of f evaluated at x_o.

2.5.2 Generalization of online allocation

Consider the following generalization of the online allocation scenario, called *online convex optimization* (over the probability simplex Δ^{N-1}). In round t:

- The learner chooses an allocation vector $p_t \in \Delta^{N-1}$.
- A convex loss function $f_t \in \mathbb{R}^N \to \mathbb{R}$ is revealed to the learner. (It is not obvious how such a function is "revealed" to the learner, but we shall make this more precise later.)

[3] There are other equivalent definitions of convex functions, but the one based on (2.6) is the most directly relevant for our purposes. For twice-differentiable functions, convexity is equivalent to positive semi-definiteness of the matrix of second derivatives at all points in $\mathbb{R}^N$.

- The learner incurs loss $f_t(p_t)$.

The notions of cumulative loss and regret naturally extend to this scenario, with the loss $\langle q, \ell_t \rangle$ replaced by $f_t(q)$ for all $q \in \Delta^{N-1}$.

In particular, suppose in round t, the learner acquires a vector $\lambda_t \in \partial f_t(p_t)$ after having selected p_t. This is the sense in which the learner needs to "observe" f_t. Then for any $q \in \Delta^{N-1}$, the inequality (2.6) implies

$$f_t(q) \geq f_t(p_t) + \langle q - p_t, \lambda_t \rangle,$$

which rearranges to

$$f_t(p_t) - f_t(q) \leq \langle p_t - q, \lambda_t \rangle.$$

Therefore, a learner that treats λ_t as the loss vector ℓ_t in round t in the original online allocation scenario and guarantees a regret bound will also guarantee a regret bound in online convex optimization.

Corollary 2.3. *In the online convex optimization scenario over the probability simplex Δ^{N-1}, for any T and any sequence of convex loss functions $f_1, \ldots, f_T \colon \mathbb{R}^N \to \mathbb{R}$, Hedge with hyperparameter $\eta > 0$—treating $\lambda_t \in \partial f_t(p_t)$ as the loss vector ℓ_t in all rounds t—guarantees that*

$$R_T \leq \frac{\ln(N)}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \langle p_t, \ell_t^2 \rangle$$

where $\ell_t^2 := ((\lambda_{t,1} - m_t)^2, \ldots, (\lambda_{t,N} - m_t)^2) \in \mathbb{R}_+^N$ and $m_t := \min_{i \in [N]} \lambda_{t,i}$ for all t.[4]

Exercise 2.4. Suppose in the online convex optimization scenario over the probability simplex Δ^{N-1}, for some positive scalars $\gamma, \theta > 0$, the loss function $f_t \colon \mathbb{R}^N \to \mathbb{R}$ in each round t has the form

$$f_t(x) = \max\{0, \gamma - y_t(\langle x, z_t \rangle - \theta)\}$$

for some $z_t \in \{0,1\}^N$ and $y_t \in \{-1, 1\}$.

(a) Confirm that each f_t is a convex function.

(b) Show that Hedge can be used (with some setting of hyperparameter η) to choose $p_1, \ldots, p_T \in \Delta^{N-1}$ satisfying

$$\sum_{t=1}^{T} f_t(p_t) \leq \min_{q \in \Delta^{N-1}} \sum_{t=1}^{T} f_t(q) + \sqrt{2T \ln(N)}.$$

$\Diamond$

2.6 Optimality of Hedge

2.6.1 Lower bound for any learning algorithm

While Hedge is a no-regret learning algorithm for online allocation, it is natural to ask if its regret bound can be improved. It turns out that no learning algorithm can guarantee a better regret bound (up to constants) in the worst case that holds for all T and N. This is the content of Theorem 2.4. In this sense, Hedge is an optimal learning algorithm for online allocation.

[4] Exercise 2.2 explores how Hedge handles loss vectors that may have negative components.

Theorem 2.4 (Freund and Schapire, 1999; Arora, Hazan, and Kale, 2012)**.** *There exists an absolute constant $C > 0$ such that the following holds. Let A denote a learning algorithm for the online allocation scenario in N-space where loss vectors come from $[0,1]^N$, and let $R_T(A; \ell_1, \ldots, \ell_T)$ denote the regret of A after T rounds with loss vectors $\ell_1, \ldots, \ell_T \in [0,1]^N$. For all sufficiently large N and T, for all learning algorithms A, there exist $\ell_1, \ldots, \ell_T \in \{0,1\}^N$ such that*

$$R_T(A; \ell_1, \ldots, \ell_T) \geq C\sqrt{T \ln(N)}.$$

2.6.2 Proof strategy

The key limitation of learning algorithms is what "information" they have access to when choosing the allocations in each round. This limitation is sufficient to leverage the probabilistic method to establish the lower bound in Theorem 2.4. The main idea is the following:

1. Define a probability distribution for the (random) loss vectors $\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_T$, and then show that

$$\Pr\Big(R_T(A; \boldsymbol{\ell}_1, \ldots, \ell_T) \geq C\sqrt{T \ln(N)}\Big) > 0. \tag{2.7}$$

2. Therefore, there must be some element in the sample space, which ultimately determines some $\ell_1, \ldots, \ell_T$ and $R_T(A; \ell_1, \ldots, \ell_T)$, such that

$$R_T(A; \ell_1, \ldots, \ell_T) \geq C\sqrt{T \ln(N)}.$$

So it suffices to show that (2.7) holds for some suitable probability distribution for $\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_T$. One distribution that works treats $\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_T$ as independent random vectors, with each ℓ_t uniformly distributed over $\{0,1\}^N$.

2.6.3 Most of the proof of the lower bound

Let $\mathbf{p}_t$ denote the allocation vector chosen by A in round t, and recall that $\mathbf{p}_t$ depends only on $\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_{t-1}$. By definition of regret and the tower property of expectations,

$$\begin{aligned}
\mathrm{E}[R_T(A; \boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_T)] &= \mathrm{E}\left[\sum_{t=1}^T \langle \mathbf{p}_t, \boldsymbol{\ell}_t \rangle - \min_{i \in [N]} \sum_{t=1}^T \boldsymbol{\ell}_{t,i}\right] \\
&= \sum_{t=1}^T \mathrm{E}\Big[\mathrm{E}[\langle \mathbf{p}_t, \boldsymbol{\ell}_t \rangle \mid \boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_{t-1}]\Big] - \mathrm{E}\left[\min_{i \in [N]} \sum_{t=1}^T \boldsymbol{\ell}_{t,i}\right] \\
&= \frac{T}{2} - \mathrm{E}\left[\min_{i \in [N]} \sum_{t=1}^T \boldsymbol{\ell}_{t,i}\right].
\end{aligned}$$

It will be notationally cleaner to introduce the random vectors $\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_T$, where $\boldsymbol{\sigma}_t = (\boldsymbol{\sigma}_{t,1}, \ldots, \boldsymbol{\sigma}_{t,N})$ is defined by $\boldsymbol{\sigma}_{t,i} := 1 - 2\boldsymbol{\ell}_{t,i}$ for each $i \in [N]$. So the $\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_T$ are independent random vectors, and each $\boldsymbol{\sigma}_t$ is uniform distributed over $\{-1,1\}^N$. Then

$$\min_{i \in [N]} \sum_{t=1}^T \boldsymbol{\ell}_{t,i} = \min_{i \in [N]} \sum_{t=1}^T \frac{1 - \boldsymbol{\sigma}_{t,i}}{2} = \min_{i \in [N]} \frac{T}{2} - \frac{1}{2} \sum_{t=1}^T \boldsymbol{\sigma}_{t,i} = \frac{T}{2} - \frac{1}{2} \max_{i \in [N]} \sum_{t=1}^T \boldsymbol{\sigma}_{t,i}.$$

Therefore, we have

$$\mathrm{E}[R_T(A;\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T)] = \frac{T}{2} - \mathrm{E}\left[\frac{T}{2} - \frac{1}{2}\max_{i\in[N]}\sum_{t=1}^{T}\boldsymbol{\sigma}_{t,i}\right] = \frac{1}{2}\,\mathrm{E}\left[\max_{i\in[N]}\sum_{t=1}^{T}\boldsymbol{\sigma}_{t,i}\right].$$

For each $i \in [N]$, the random variable $\mathbf{s}_{T,i} := \sum_{t=1}^{T}\boldsymbol{\sigma}_{t,i}$ is the sum of T independent random variables, each uniform over $\{-1,1\}$. It turns out that, for large enough T and N,

$$\mathrm{E}\left[\max_{i\in[N]}\mathbf{s}_{T,i}\right] \geq C\sqrt{T\ln(N)}$$

for some absolute constant $C > 0$. We will not prove this, but just give some the intuition for why it should be believable. By the Central Limit Theorem, as $T \to \infty$,

$$\frac{1}{\sqrt{T}}(\mathbf{s}_{T,1},\ldots,\mathbf{s}_{T,N}) \xrightarrow{\text{dist.}} (\mathbf{z}_1,\ldots,\mathbf{z}_N) \sim \mathrm{Normal}(0,1)^N.$$

So let us just heuristically pretend that

$$\mathrm{E}\left[\max_{i\in[N]}\mathbf{s}_{T,i}\right] \overset{\text{☺}}{=} \sqrt{T}\,\mathrm{E}\left[\max_{i\in[N]}\mathbf{z}_i\right].$$

(The ☺ is meant to caution you from taking this equality literally.) The distribution of $\max_{i\in[N]}\mathbf{z}_i$ is also well understood in the limit $N \to \infty$; after a certain centering and scaling, its distribution tends to a Gumbel distribution, which is one of the central distributions that features in the study of extreme value statistics. The results in this literature lead us to

$$\mathrm{E}\left[\max_{i\in[N]}\mathbf{z}_i\right] \overset{\text{☺}}{=} \sqrt{2\ln(N)}(1-o(1)),$$

where the $o(1)$ term goes to 0 as $N \to \infty$. So, we find

$$\mathrm{E}[R_T(A;\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T)] \overset{\text{☺}}{=} \sqrt{\frac{T\ln(N)}{2}}(1-o(1)).$$

Theorem 2.4 is proved by using the appropriate non-asymptotic versions of these heuristic arguments.

Topic 3: Online concept learning

3.1 Concept classes and the basic scenario

3.1.1 Concept classes

Another departure from the expert advice scenario puts structure on the experts' predictions. Consider a scenario in which experts correspond to $\{0,1\}$-valued functions from some class of functions $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$ defined on a domain $\mathcal{X}$. We refer to the function class $\mathcal{C}$ as a *concept class*, and the individual functions as *concepts*. Each domain element $x \in \mathcal{X}$—a.k.a. *input* or *input point*—represents the available information about a present circumstance, and a concept $c \in \mathcal{C}$ represents a way of classifying the circumstance as positive ($c(x) = 1$) or negative ($c(x) = 0$) as a function of the available information. The output $c(x)$ is also called the *label* of x provided by c.

3.1.2 Online concept learning scenario

We will be interested in learning algorithms that take advantage of the structure of a concept class. Just as in the expert advice scenario, a learner makes predictions in a sequence of rounds. But unlike the expert advice scenario, and at the start of each round, the learner is provided an input $x \in \mathcal{X}$, which determines the predictions of all concepts $c \in \mathcal{C}$. We call this the *online concept learning scenario*. The *realizability assumption* from the expert advice scenario can be adapted to online concept learning. The *$\mathcal{C}$-realizability assumption* is the assumption that, at any time, all actual outcomes that have been revealed (i.e., the "true" labels) are consistent with the labels provided by some "true" concept $c^\star \in \mathcal{C}$. We call the pair (x, y) a *labeled example*.[1] The goal of the learner is to never make more than a finite number of mistakes, i.e., ensure a finite mistake bound.[2]

The online concept learning scenario can capture the expert advice scenario: consider $\mathcal{X} = \{0,1\}^N$ and $\mathcal{C} = \mathsf{EXPERTS}_N := \{c_i \mid i \in [N]\}$ where $c_i(x_1, \ldots, x_N) = x_i$ for all $i \in [N]$.[3] Conversely, if $|\mathcal{C}| < \infty$, then every algorithm we have considered for the expert advice scenario can be used in the online concept learning scenario, too, simply by treating each concept as an expert and computing the experts' predictions in each round. However, we will be particularly interested in efficient algorithms when $|\mathcal{C}|$ is very large compared to the size of the input x.

3.2 Learning concepts over the Boolean domain

3.2.1 Concepts over the Boolean n-cube

In this section, we let $\mathcal{X} = \{0,1\}^n$ for some positive integer n, i.e., the *Boolean n-cube*. So each input $x = (x_1, \ldots, x_n) \in \{0,1\}^n$ is described by n Boolean attributes (with 1 corresponding to true and 0 corresponding to false). A concept $c\colon \{0,1\}^n \to \{0,1\}$ maps each of the 2^n possible settings of the n attribute values to either true (1) or false (0). A concept can be described by its "truth table"

[1] A labeled example (x, y) is called a positive example if $y = 1$, and a negative example if $y = 0$.

[2] The online concept learning scenario with the $\mathcal{C}$-realizability assumption is also called the *online mistake bound learning model* with concept class $\mathcal{C}$.

[3] In social choice theory, the class $\mathsf{EXPERTS}_N$ is also sometimes called a class of *dictators*.

which gives this mapping explicitly; there are exactly 2^{2^n} possible truth tables. Thus every concept class $\mathcal{C}$ on the domain $\{0,1\}^n$ has cardinality $|\mathcal{C}| \leq 2^{2^n}$, so the Halving algorithm ensures a mistake bound of $\log_2(|\mathcal{C}|) \leq 2^n$ with the realizability assumption. This mistake bound is also achieved by the naïve algorithm that simply memorizes every input/label-pair that has been encountered and hence can avoid mistakes on repeated inputs.

3.2.2 Monotone conjunctions

The first class we consider is that of *monotone conjunctions*. For each subset $S \subseteq [n] := \{1, \ldots, n\}$, define the concept $c_S \colon \{0,1\}^n \to \{0,1\}$ by

$$c_S(x_1, \ldots, x_n) := \bigwedge_{i \in S} x_i.$$

The value of $c_S(x_1, \ldots, x_n)$ is true precisely when all x_i for $i \in S$ are true. The concept class $\mathsf{MCONJ}_n := \{c_S \mid S \subseteq [n]\}$ contains all 2^n such concepts c_S, and each concept is succinctly described by an attribute index set $S \subseteq [n]$.

We may adapt the Halving algorithm from Section 1.2 to guarantee a mistake bound of $\log_2(|\mathsf{MCONJ}_n|) = \log_2(2^n) = n$ with the realizability assumption. However, a naïve adaptation of Halving would explicitly track which of the 2^n concepts are "alive" at any point in time, which is computationally expensive. An alternative is the following algorithm, called *Attribute Elimination*, essentially due to Valiant (1984) and Angluin (1988).

Algorithm 3.1: Attribute Elimination for online learning monotone conjunctions over $\{0,1\}^n$.

- Initially, set $S_1 := [n]$.
- In round t:
  - Observe $x_t = (x_{t,1}, \ldots, x_{t,n}) \in \{0,1\}^n$.
  - Predict $a_t := c_{S_t}(x_t)$.
  - Observe label $y_t \in \{0,1\}$.
  - Update hypothesis:

$$S_{t+1} := \begin{cases} S_t & \text{if } a_t = y_t, \\ S_t \setminus \{i \in [n] \mid x_{t,i} = 0\} & \text{if } a_t \neq y_t. \end{cases}$$

Attribute Elimination maintains a "hypothesis" set of attribute indices that is guaranteed to always contain the index set $S^\star$ for a "true" concept $c_{S^\star}$ (under the realizability assumption). After a mistake, the Attribute Elimination updates the hypothesis set by removing indices of attributes that can be deduced to not belong to $S^\star$. Both the "size" of the hypothesis and the number of operations needed for each update is linear in the number of attributes n.

Theorem 3.1. *In the online concept learning scenario with the MCONJ_n-realizability assumption, Attribute Elimination makes at most n mistakes.*

Proof. Let $S^\star \subseteq [n]$ be attribute set for a "true" concept $c_{S^\star}$ that never makes a mistake. Also, define M_t to be the number of mistakes of Attribute Elimination through round t, with $M_0 := 0$.

We show by induction that, for all rounds $t \in \mathbb{N}$, we have

$$S^\star \subseteq S_t \quad \text{and} \quad M_{t-1} \leq |S_1| - |S_t|. \tag{3.1}$$

The claim is true for $t = 1$ since $S_1 = [n]$ and $M_0 = 0$ by definition. So consider any round $t \geq 1$, and assume that (3.1) holds as the inductive hypothesis. We now show that $S^\star \subseteq S_{t+1}$ and $M_t \leq |S_1| - |S_{t+1}|$.

If $a_t = y_t$, then $S^\star \subseteq S_{t+1} = S_t$ and $M_t = M_{t-1} \leq |S_1| - |S_t| = |S_1| - |S_{t+1}|$. So assume instead that $a_t \neq y_t$. Since $S^\star \subseteq S_t$ by (3.1), we have

$$a_t = \underbrace{\left(\bigwedge_{i \in S^\star} x_{t,i} \right)}_{y_t} \wedge \left(\bigwedge_{i \in S_t \setminus S^\star} x_{t,i} \right).$$

If $y_t = 0$, then $a_t = 0$ as well, which contradicts the assumption that the learner makes a mistake. So it must be that $y_t = 1$ and $a_t = 0$, and

$$\left(\bigwedge_{i \in S^\star} x_{t,i} \right) = 1 \quad \text{and} \quad \left(\bigwedge_{i \in S_t \setminus S^\star} x_{t,i} \right) = 0.$$

This implies the following facts:

1. $x_{t,i} = 1$ for all $i \in S^\star$;
2. there exists some $i \in S_t \setminus S^\star$ such that $x_i = 0$.

Combining the first fact with (3.1) gives $S^\star \subseteq S_{t+1}$. The second fact implies $|S_{t+1}| \leq |S_t| - 1$; combining this with (3.1) (and the assumption $a_t \neq y_t$) gives

$$M_t = M_{t-1} + 1 \leq |S_1| - (|S_t| - 1) \leq |S_1| - |S_{t+1}|.$$

This completes the inductive step. The theorem follow since $|S_1| = n$. □

3.2.3 Monotone disjunctions

The second class we consider is that of *monotone disjunctions*. For each subset $S \subseteq [n]$, define the concept $d_S \colon \{0,1\}^n \to \{0,1\}$ by

$$d_S(x_1, \ldots, x_n) := \bigvee_{i \in S} x_i.$$

The value of $d_S(x_1, \ldots, x_n)$ is true precisely when any x_i for $i \in S$ is true. The concept class $\mathsf{MDISJ}_n := \{d_S \mid S \subseteq [n]\}$ contains all 2^n such concepts d_S.

By De Morgan's law,

$$\neg \bigvee_{i \in S} x_i = \bigwedge_{i \in S} \neg x_i$$

(where $\neg b = 1 - b$ for $b \in \{0,1\}$). In other words, the negation of a monotone disjunction is the same as a monotone conjunction over the negated attributes. Therefore, learning monotone disjunctions reduces to learning monotone conjunctions (and vice versa) by a simple transformation of the inputs $(x_{t,1}, \ldots, x_{t,n}) \mapsto (\neg x_{t,1}, \ldots, \neg x_{t,n})$ and labels $y_t \mapsto \neg y_t$.

Theorem 3.2. *In the online concept learning scenario with the* MDISJ_n*-realizability assumption, Attribute Elimination with the appropriate input and label transformations makes at most* n *mistakes.*

Exercise 3.1. Design a learning algorithm for the online concept learning scenario with the $\mathsf{MCONJ}_n \cup \mathsf{MDISJ}_n$-realizability assumption with mistake bound $O(n)$ and time complexity $O(n)$ per round. ◊

3.2.4 Conjunctions

The third class we consider is that of *conjunctions*. For each pair of disjoint subsets $S, T \subseteq [n]$, define the concept $c_{S,T} \colon \{0,1\}^n \to \{0,1\}$ by

$$c_{S,T}(x_1, \ldots, x_n) := \left(\bigwedge_{i \in S} x_i \right) \wedge \left(\bigwedge_{i \in T} \neg x_i \right).$$

The value of $c_{S,T}(x_1, \ldots, x_n)$ is true precisely when all x_i for $i \in S$ are true, and all x_i for $i \in T$ are false. The concept class $\mathsf{CONJ}_n := \{c_{S,T} \mid S, T \subseteq [n], S \cap T = \emptyset\}$ contains all 3^n such concepts $c_{S,T}$.

Suppose we extend the n Boolean attributes $(x_1, \ldots, x_n)$ to $2n$ Boolean attributes $(x_1, \ldots, x_{2n})$, where $x_{n+i} := \neg x_i$ for all $i \in [n]$. Then

$$c_{S,T}(x_1, \ldots, x_n) = \bigwedge_{i \in S \cup (n+T)} x_i.$$

In other words, $c_{S,T}$ has a representation as a monotone conjunction over the extended attributes $(x_1, \ldots, x_{2n})$. Therefore, learning conjunctions reduces to learning monotone conjunctions (over a larger domain with $2n$ attributes) by a simple transformation of the inputs $(x_{t,1}, \ldots, x_{t,n}) \mapsto (x_{t,1}, \ldots, x_{t,n}, \neg x_{t,1}, \ldots, \neg x_{t,n})$.

Theorem 3.3. *In the online concept learning scenario with the* CONJ_n*-realizability assumption, Attribute Elimination with the appropriate input transformation makes at most* $2n$ *mistakes.*

Exercise 3.2. Let DISJ_n be the concept class of *disjunctions*, defined as you would expect. What are the appropriate input and label transformations needed for Attribute Elimination for the online concept learning scenario with the DISJ_n-realizability assumption? ◊

Exercise 3.3. Improve the bound from Theorem 3.3 to $n + 1$ mistakes. ◊

3.2.5 Linear threshold functions

The class of *linear threshold functions (LTFs)* is a very well-studied concept class in part for its connections to neural networks (Minsky and Papert, 1969). A function $f \colon \{0,1\}^n \to \{0,1\}$ is a *linear threshold function (LTF)* on $\{0,1\}^n$ if there exists $w \in \mathbb{R}^n$ and $\theta \in \mathbb{R}$ such that

$$f(x) = \mathbb{1}\{\langle w, x \rangle > \theta\} = \begin{cases} 1 & \text{if } \sum_{i=1}^n w_i x_i > \theta, \\ 0 & \text{otherwise} \end{cases}$$

That is, $f(x)$ is true precisely when the standard inner product $\langle w, x \rangle$ of w and x exceeds θ. In this case, we say that the LTF f is represented by (w, θ), where w is the *weight vector* w, and θ is

the *threshold*. Let LTF_n denote the concept class of all LTFs on $\{0,1\}^n$. Although every LTF has infinitely-many representations, the number of LTFs is finite; in fact, Muroga (1971) showed that the number of LTFs satisfies

$$\log_2(|\mathsf{LTF}_n|) = \Theta(n^2).$$

(An even sharper result of Zuev (1989) implies that $\lim_{n\to\infty} \log_2(|\mathsf{LTF}_n|)/n^2 = 1$.) The upper bound $\log_2(|\mathsf{LTF}_n|) = O(n^2)$ is a consequence of Sauer's lemma, which will be discussed later in Section 5.2.3. The lower bound $\log_2(|\mathsf{LTF}_n|) = \Omega(n^2)$ is the following proposition.

Proposition 3.1. $|\mathsf{LTF}_n| \geq 2^{2+\frac{n(n-1)}{2}}$.

Proof. We first claim that every $f \in \mathsf{LTF}_n$ has a representation $(w, \theta) \in \mathbb{R}^n \times \mathbb{R}$ such that

$$|\{\langle w, x\rangle \mid x \in \{0,1\}^n\}| = 2^n. \tag{3.2}$$

There are many ways to see this; here is one. Pick any representation $(a, t) \in \mathbb{R}^n \times \mathbb{R}$ for f. Let

$$\gamma := \min\{\langle a, x\rangle - t \mid x \in \{0,1\}^n, \langle a, x\rangle > t\} > 0.$$

Let $v := (2^0, 2^1, \ldots, 2^{n-1})$; set $w := a + \epsilon v$ and $\theta := t + \gamma/2$. We show that there is a choice of $\epsilon > 0$ such that (w, θ) is a representation for f and (3.2) holds.

- If $\epsilon \leq \gamma/2^{n+1}$, then for all $x \in \{0,1\}^n$, $0 \leq \epsilon\langle v, x\rangle \leq \epsilon 2^n \leq \gamma/2$. Therefore:

  $$\begin{aligned} \langle a, x\rangle \leq t &\implies \langle w, x\rangle = \langle a, x\rangle + \epsilon\langle v, x\rangle \leq t + \gamma/2 \leq \theta; \\ \langle a, x\rangle > t &\implies \langle w, x\rangle = \langle a, x\rangle + \epsilon\langle v, x\rangle \geq t + \gamma > \theta. \end{aligned}$$

  So (w, θ) is a representation for f.

- Consider any distinct $x, x' \in \{0,1\}^n$. Then $\langle v, x\rangle \neq \langle v, x'\rangle$. So if $\langle w, x\rangle = \langle w, x'\rangle$ for a particular choice of ϵ, then it must be that

  $$\epsilon = -\frac{\langle a, x\rangle - \langle a, x'\rangle}{\langle v, x\rangle - \langle v, x'\rangle}.$$

  Consequently, there are finitely-many choices of $\epsilon > 0$ that lead to a weight vector w with $\langle w, x\rangle = \langle w, x'\rangle$ for some distinct $x, x' \in \{0,1\}^n$. If ϵ is strictly smaller than all such choices, then (3.2) holds.

We now claim that, for all $n \in \mathbb{N}$,

$$|\mathsf{LTF}_{n+1}| \geq 2^n |\mathsf{LTF}_n|.$$

Fix any $f \in \mathsf{LTF}_n$, and take a representation (w, θ) for f such that (3.2) holds. From this f, we construct 2^n distinct LTFs over $\{0,1\}^{n+1}$ that behave the same as f on the sub-cube $\{(x_1, \ldots, x_n, x_{n+1}) \in \{0,1\}^{n+1} \mid x_{n+1} = 0\}$, but behave differently from each other on the other sub-cube $\{(x_1, \ldots, x_n, x_{n+1}) \in \{0,1\}^{n+1} \mid x_{n+1} = 1\}$. Set $S := \{\langle w, x\rangle \mid x \in \{0,1\}^n\}$. For each $s \in S$, define the LTF $F_s \colon \{0,1\}^{n+1} \to \{0,1\}$ by

$$F_s(x, x_{n+1}) := \mathbb{1}\{\langle w, x\rangle + (\theta - s)x_{n+1} > \theta\}.$$

Observe that $F_s(x,0) = f(x)$, and $F_s(x,1) = \mathbb{1}\{\langle w, x\rangle > s\}$. Since the values in $S = \{\langle w, x\rangle \mid x \in \{0,1\}^n\}$ are distinct, the 2^n choices of $s \in S$ lead to 2^n distinct LTFs over $\{0,1\}^{n+1}$. So we have the desired construction of 2^n distinct LTFs over $\{0,1\}^{n+1}$ corresponding to f. This gives the claimed bound $|\mathsf{LTF}_{n+1}| \geq 2^n |\mathsf{LTF}_n|$.

For $n = 1$, there are $|\mathsf{LTF}_1| = 4$ distinct LTFs. Therefore

$$|\mathsf{LTF}_n| \geq 4 \prod_{k=1}^{n-1} 2^k = 2^{2+\frac{n(n-1)}{2}}$$

as claimed. □

So there are many more LTFs than there are, say, conjunctions or disjunctions. In fact, conjunctions and disjunctions can be represented as LTFs. So algorithms for learning LTFs can be used to learn conjunctions or disjunctions (as well as many other well-known concept classes).

We may adapt the Halving algorithm from Section 1.2 to guarantee a mistake bound of $\log_2(|\mathsf{LTF}_n|) = \log_2(2^{O(n^2)}) = O(n^2)$ for the online concept learning scenario with the LTF_n-realizability assumption; but as usual, a naïve implementation is computationally expensive. There is a polynomial-time algorithm that achieves mistake bound $O(n^2 \log(n))$ (Maass and Turán, 1994), but it is an open question whether there is a polynomial-time algorithm that achieves an asymptotically better mistake bound.

3.3 Perceptron and margins

3.3.1 Linear threshold functions over Euclidean space

Perhaps the most well-known algorithm for learning LTFs is (the online variant of) the Perceptron algorithm (Rosenblatt, 1958). Perceptron is applicable to learning LTFs over Euclidean n-space, $\mathbb{R}^n$ (or really, any inner product space):

$$\mathsf{LTF}_{\mathbb{R}^n} := \Big\{ f \colon \mathbb{R}^n \to \{0,1\} \;\Big|\; \exists (w,\theta) \in \mathbb{R}^n \times \mathbb{R} \text{ s.t. } f(x) = \mathbb{1}\{\langle w, x\rangle > \theta\} \Big\}.$$

To use Perceptron to learn LTFs over the Boolean n-cube, we can "embed" $\{0,1\}^n$ in $\mathbb{R}^n$.

The number of LTFs over $\mathbb{R}^n$ is not finite, so the Halving algorithm is not directly applicable. In fact, no algorithm can guarantee a finite mistake bound, even with the realizability assumption.

Theorem 3.4. *In the online concept learning scenario, for each learning algorithm, there is an infinite sequence of input/label-pairs such that: (i) every finite prefix is consistent with some $c^\star \in \mathsf{LTF}_{\mathbb{R}^1}$, (ii) the learner makes a mistake in every round.*

Proof. To construct the infinite sequence, simulate the learning algorithm in the online concept learning scenario and set the true labels $(y_t)_{t\in\mathbb{N}}$ to be the opposite of the learner's predictions $(a_t)_{t\in\mathbb{N}}$. We need to determine inputs $(x_t)_{t\in\mathbb{N}}$ so that for every $t \in \mathbb{N}$, the choice of x_t depends only on $y_1, \ldots, y_{t-1}$, and there is some $c^\star \in \mathsf{LTF}_{\mathbb{R}^1}$ consistent with $(x_1, y_1), \ldots, (x_t, y_t)$. This must be done for every possible sequence $(y_t)_{t\in\mathbb{N}}$.

For $t \in \mathbb{N}$, define

$$x_t := \sum_{i=1}^{t-1} (1 - y_i) 2^{-i} + 2^{-t}.$$

So, initially $x_1 = 1/2$; on round $t > 1$,

$$x_t = x_{t-1} + \left(\frac{1}{2} - y_{t-1}\right)2^{-(t-1)} = \begin{cases} x_{t-1} + 2^{-t} & \text{if } y_{t-1} = 0, \\ x_{t-1} - 2^{-t} & \text{if } y_{t-1} = 1. \end{cases}$$

We claim that for every $t \in \mathbb{N}$, there exists some $\theta_t \in \mathbb{R}$ such that

$$y_\tau = \mathbb{1}\{x_\tau > \theta_t\} \quad \text{for all } \tau \in \{1, \ldots, t\},$$

i.e., the concept $c^\star \in \mathsf{LTF}_{\mathbb{R}^1}$ given by $c^\star(x) = \mathbb{1}\{x > \theta_t\}$ is consistent with $(x_1, y_1), \ldots, (x_t, y_t)$. A choice for θ_t that works is

$$\theta_t := \sum_{i=1}^{t} (1 - y_i)2^{-i}.$$

Indeed, for any $\tau \in \{1, \ldots, t\}$,

$$x_\tau - \theta_t = \sum_{i=1}^{\tau-1} (1 - y_i)2^{-i} + 2^{-\tau} - \sum_{i=1}^{t} (1 - y_i)2^{-i} = y_\tau 2^{-\tau} - \sum_{i=\tau+1}^{t} (1 - y_i)2^{-i}$$

which is positive if and only if $y_\tau = 1$. □

The proof of Theorem 3.4 applies to the subset of $\mathsf{LTF}_{\mathbb{R}^1}$ in which the weight vector is required to be 1 and the threshold is required to lie in $[0, 1)$. The constructed sequence of input/label-pairs ensures that

$$\begin{aligned} x_t + 2^{-t} &= \min\{x_\tau \mid \tau \in \{1, \ldots, t-1\}, y_\tau = 1\} \cup \{1\}, \\ x_t - 2^{-t} &= \max\{x_\tau \mid \tau \in \{1, \ldots, t-1\}, y_\tau = 0\} \cup \{0\}. \end{aligned}$$

In other words, $x_t + 2^{-t}$ is the leftmost positive example seen so far (or 1), and $x_t - 2^{-t}$ is the rightmost negative example seen so far (or 0). Let $V_t := \{\theta \in (0, 1) \mid \mathbb{1}\{x_\tau > \theta\} = y_\tau \ \forall \tau \in \{1, \ldots, t-1\}\}$ be the possible values of the threshold that are consistent with the input/label-pairs prior to round t. The interval $(x_t - 2^{-t}, x_t + 2^{-t})$ is the "uncertainty region" for the threshold: for every $x \in (x_t - 2^{-t}, x_t + 2^{-t})$, there are thresholds $\theta, \theta' \in V_t$ for which $\mathbb{1}\{x > \theta\} = 0$ and $\mathbb{1}\{x > \theta'\} = 1$. The input in round t is placed in the middle of this interval, and no matter what the learner predicts, there is a threshold $\theta \in V_t$ for which $\mathbb{1}\{x_t > \theta\}$ differs from the learner's prediction. The width of the interval also shrinks exponentially with the number of rounds t.

It is pathologies like the above that defeat every learning algorithm for LTFs over $\mathbb{R}^n$. Fortunately, this is the only kind of pathology that can defeat Perceptron.

3.3.2 Margins

To rule out the pathology from Section 3.3.1, we formalize a strengthened version of the realizability assumption for LTFs. We assume that the sequence $((x_t, y_t))_{t \in \mathbb{N}}$ satisfies the following for some parameter $\gamma > 0$ called the *margin*:

- $\|x_t\| \leq 1$ for all $t \in \mathbb{N}$ (where $\|\cdot\|$ denotes the Euclidean norm); and

- there exists $(w^\star, \theta^\star) \in \mathbb{R}^n \times \mathbb{R}$ with $\|w^\star\| \leq 1$ and $|\theta^\star| \leq 1$ such that, for all $t \in \mathbb{N}$,

$$\begin{aligned} y_t = 1 &\implies \langle w^\star, x_t \rangle \geq \theta^\star + \gamma, \\ y_t = 0 &\implies \langle w^\star, x_t \rangle \leq \theta^\star - \gamma. \end{aligned} \tag{3.3}$$

This assumption (both parts together) is called the *γ-margin assumption*. The requirement that $\|x_t\| \leq 1$ and $\|w^\star\| \leq 1$ fixes a scale to ensures, via Cauchy-Schwarz, that $\langle w^\star, x_t \rangle$ lies in $[-1, 1]$. (The restriction $\theta^\star \in [-1, 1]$ is then without loss of generality.) The margin γ is on the same scale as $\langle w^\star, x_t \rangle$ and $\theta^\star$. The larger the value of γ, the stronger the assumption.

The γ-margin assumption rules out the pathology from Theorem 3.4: as soon as the width of the "uncertainty region" drops to 2γ, the "true" threshold $\theta^\star$ is determined.

3.3.3 Perceptron

The Perceptron algorithm of Rosenblatt (1958) is as follows.

Algorithm 3.2: Perceptron for online learning linear threshold functions over $\mathbb{R}^n$.

- Initially, set $w_1 := 0 \in \mathbb{R}^n$ and $\theta_1 := 0 \in \mathbb{R}$.
- In round t:
  - Observe $x_t \in \mathbb{R}^n$.
  - Predict $a_t := \mathbb{1}\{\langle w_t, x_t \rangle > \theta_t\}$.
  - Observe label $y_t \in \{0, 1\}$.
  - Update hypothesis: If $a_t \neq y_t$, then

$$w_{t+1} := \begin{cases} w_t + x_t & \text{if } y_t = 1, \\ w_t - x_t & \text{if } y_t = 0; \end{cases}$$

$$\theta_{t+1} := \begin{cases} \theta_t - 1 & \text{if } y_t = 1, \\ \theta_t + 1 & \text{if } y_t = 0. \end{cases}$$

  If $a_t = y_t$, then $w_{t+1} := w_t$ and $\theta_{t+1} := \theta_t$.

Intuitively, the update made by Perceptron after a mistake changes the value of the inner product between the weight vector and the input and the threshold to "move in the right direction". For example, if $y_t = 1$ but $a_t = 0$, then $\langle w_t, x_t \rangle$ was too small or θ_t was too large; so the update causes $\langle w_{t+1}, x_t \rangle$ to be larger than $\langle w_t, x_t \rangle$ and θ_{t+1} to be smaller than θ_t.

It is somewhat simpler to view Perceptron as a learning algorithm for *homogeneous linear threshold functions* over $\mathbb{R}^{n+1}$, denoted by $\mathsf{HLTF}_{\mathbb{R}^{n+1}}$, which are the LTFs over $\mathbb{R}^{n+1}$ with threshold 0. By mapping each input $x \in \mathbb{R}^n$ to $\tilde{x} := (x, -1) \in \mathbb{R}^{n+1}$ and each $(w, \theta) \in \mathbb{R}^n \times \mathbb{R}$ to a weight vector $\tilde{w} := (w, \theta) \in \mathbb{R}^{n+1}$, we see that

$$\langle w, x \rangle - \theta = \langle \tilde{w}, \tilde{x} \rangle.$$

So every LTF over $\mathbb{R}^n$ can be viewed as a homogeneous LTF over $\mathbb{R}^{n+1}$.

It is also somewhat simpler to use $\{-1, 1\}$-valued labels instead of $\{0, 1\}$-valued labels, so that the Perceptron update step (when $a_t \neq y_t$) can be written as

$$\tilde{w}_{t+1} := \tilde{w}_t + \tilde{y}_t \tilde{x}_t$$

where $\tilde{y}_t := 1$ if $y_t = 1$ and $\tilde{y}_t := -1$ if $y_t = 0$.

Under the γ-margin assumption (for $\mathsf{LTF}_{\mathbb{R}^n}$), and using $\{-1, 1\}$-valued labels, we have

- $\|\tilde{x}_t\| \leq \sqrt{2}$ for all $t \in \mathbb{N}$; and
- there exists $\tilde{w}^\star \in \mathbb{R}^{n+1}$ with $\|\tilde{w}^\star\| \leq \sqrt{2}$ such that, for all $t \in \mathbb{N}$,

$$\tilde{y}_t \langle \tilde{w}^\star, \tilde{x}_t \rangle \geq \gamma.$$

We call the above two bullets together the *homogeneous γ-margin assumption* (in $\mathbb{R}^{n+1}$, although the dimension of the space does not explicitly play a role in the following theorem).

Theorem 3.5 (Novikoff, 1962)**.** *In the online concept learning scenario with the homogeneous γ-margin assumption, Perceptron makes at most $4/\gamma^2$ mistakes.*

Note that if the norm bounds in the homogeneous γ-margin assumption are changed to $\|\tilde{x}_t\| \leq X$ and $\|\tilde{w}^\star\| \leq W$, then the mistake bound in Theorem 3.5 changes to $X^2 W^2 / \gamma^2$. Also, Theorem 3.5 does not only apply in cases where the $\tilde{x}_t$ and $\tilde{w}^\star$ are derived from the transformations described above; rather, the $\tilde{x}_t$ and $\tilde{w}^\star$ can be any vectors that satisfy the conditions in the two bullets.

Proof of Theorem 3.5. At the start of round $t = 1$, we have $\langle \tilde{w}^\star, \tilde{w}_1 \rangle = 0$ and $\|\tilde{w}_1\|^2 = 0$. Suppose Perceptron makes a mistake in round t. Using the sign-valued labels, a mistake in round t implies

$$\tilde{y}_t \langle \tilde{w}_t, \tilde{x}_t \rangle \leq 0. \tag{3.4}$$

We now consider the effect of the update on $\langle \tilde{w}^\star, \tilde{w}_{t+1} \rangle$ and $\|\tilde{w}_{t+1}\|^2$ relative to $\langle \tilde{w}^\star, \tilde{w}_t \rangle$ and $\|\tilde{w}_t\|^2$. First, we have

$$\begin{aligned} \langle \tilde{w}^\star, \tilde{w}_{t+1} \rangle &= \langle \tilde{w}^\star, \tilde{w}_t + \tilde{y}_t \tilde{x}_t \rangle \\ &= \langle \tilde{w}^\star, \tilde{w}_t \rangle + \tilde{y}_t \langle \tilde{w}^\star, \tilde{x}_t \rangle \\ &\geq \langle \tilde{w}^\star, \tilde{w}_t \rangle + \gamma \end{aligned}$$

where the inequality uses the γ-margin assumption. Next, by "expanding the square", we have

$$\begin{aligned} \|\tilde{w}_{t+1}\|^2 &= \|\tilde{w}_t + \tilde{y}_t \tilde{x}_t\|^2 \\ &= \|\tilde{w}_t\|^2 + 2\tilde{y}_t \langle \tilde{w}_t, \tilde{x}_t \rangle + \|\tilde{y}_t \tilde{x}_t\|^2 \\ &\leq \|\tilde{w}_t\|^2 + \|\tilde{y}_t \tilde{x}_t\|^2 \\ &\leq \|\tilde{w}_t\|^2 + 2 \end{aligned}$$

where the first inequality uses (3.4), and the second inequality uses the assumption $\|x_t\| \leq 1$ (which implies $\|\tilde{x}_t\|^2 \leq 2$).

Therefore, if there have been M_t mistakes after t rounds, then

$$\langle \tilde{w}^\star, \tilde{w}_{t+1} \rangle \geq M_t \gamma \quad \text{and} \quad \|\tilde{w}_{t+1}\|^2 \leq 2M_t.$$

By Cauchy-Schwarz and the assumption that $\|\tilde{w}^\star\| \le \sqrt{2}$,

$$\langle \tilde{w}^\star, \tilde{w}_{t+1} \rangle \le \|\tilde{w}^\star\| \|\tilde{w}_{t+1}\| \le \sqrt{2}\|\tilde{w}_{t+1}\|.$$

Combining these inequalities gives $M_t \le 4/\gamma^2$. □

Exercise 3.4. A new university president has assembled a committee of N experts to help her make yes/no decisions every day; for every decision she must make, she receives yes/no recommendations from all experts. The committee contains a special subcommittee of s experts (for some odd number $s \in [N]$) that is "good", meaning that the majority vote among this subcommittee always gives the correct yes/no decision. The president unfortunately does not know the identity or membership of this special subcommittee. Explain how the president can use Perceptron to make decisions—in roughly $O(N)$ time per decision—and what is the resulting mistake bound? Could she do better if she had more time per decision? ♢

Exercise 3.5. Consider the online concept learning scenario with the MCONJ_n-realizability assumption. Suppose Perceptron is used with the following embedding of $\{0,1\}^n$ into $\mathbb{R}^n$:

$$\{0,1\}^n \ni (x_1, \ldots, x_n) \mapsto \frac{1}{\sqrt{n}}(2x_1 - 1, \ldots, 2x_n - 1) \in \mathbb{R}^n.$$

Note that the embedding maps every element of $\{0,1\}^n$ into $\{x \in \mathbb{R}^n \mid \|x\| = 1\}$, the unit sphere of $\mathbb{R}^n$. What mistake bound (solely as a function of n) is implied by Theorem 3.5? ♢

A different algorithm called Winnow, due to Littlestone (1988), learns LTFs over $\mathbb{R}^n$ under a different kind of margin assumption (essentially replacing the Euclidean norms on inputs and weight vectors by ℓ^∞- and ℓ^1-norms).

3.4 Optimal mistake bounds

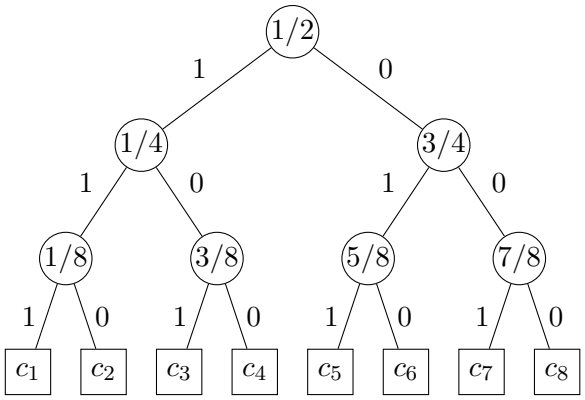
3.4.1 Mistake trees

The strategies we used to prove the mistake lower bounds in Theorem 1.3[4] and Theorem 3.4 share a common structure. We construct a rooted perfect binary tree of some depth d with each of the 2^d leaf nodes corresponding to some concept $c \in \mathcal{C}$, and each non-leaf node annotated with some input $x \in \mathcal{X}$. (The 2^d concepts will need to be distinct, but the inputs used to annotate the non-leaf nodes need not be distinct.) Each edge is annotated with either 0 or 1 so that each binary sequence in $\{0,1\}^d$ appears along the edges of some root-to-leaf path. If the root-to-leaf path ending at $c \in \mathcal{C}$ is annotated by the binary sequence $(y_1, \ldots, y_d) \in \{0,1\}^d$ (on the edges) and the input sequence $(x_1, \ldots, x_d)$ (on the non-leaf nodes), then it must be that $c(x_i) = y_i$ for all $i \in [d]$. A tree with this property is called a *(complete) mistake tree* for $\mathcal{C}$. See Figure 3.1 for an example.

The following theorem is immediate from the definition of a mistake tree.

Theorem 3.6 (Littlestone, 1988)**.** *Suppose $\mathcal{C}$ has a mistake tree of depth d. Then in the online concept learning scenario with concept class $\mathcal{C}$, for each learning algorithm, there is a sequence of d input/label-pairs consistent with some $c^\star \in \mathcal{C}$ on which the learner makes d mistakes.*

[4]Recall from Section 3.1.2 that the expert advice scenario can be simulated in the online concept learning scenario.



$c_1(x) := \mathbb{1}\{x > 0\}$
$c_2(x) := \mathbb{1}\{x > 1/8\}$
$c_3(x) := \mathbb{1}\{x > 1/4\}$
$c_4(x) := \mathbb{1}\{x > 3/8\}$
$c_5(x) := \mathbb{1}\{x > 1/2\}$
$c_6(x) := \mathbb{1}\{x > 5/8\}$
$c_7(x) := \mathbb{1}\{x > 3/4\}$
$c_8(x) := \mathbb{1}\{x > 7/8\}$

Figure 3.1: A depth $d = 3$ mistake tree for $\mathsf{LTF}_{\mathbb{R}^1}$.

Exercise 3.6. Recall the $\mathsf{EXPERTS}_N$ concept class. Let $N = 8$, and construct a mistake tree for $\mathsf{EXPERTS}_N$ of depth 3. Generalize the construction to any $N \in \mathbb{N}$ that is a power-of-two. ◊

Exercise 3.7. Let $n = 3$, and construct a mistake tree for MCONJ_n of depth 3. Generalize the construction to any $n \in \mathbb{N}$. ◊

3.4.2 Littlestone dimension

Theorem 3.6 shows that the existence of deep mistake trees puts lower bounds on the mistake bounds achievable by any learning algorithm. It turns out that there is an algorithm that can use the non-existence of a deep mistake tree to guarantee a mistake bound.

Define the *Littlestone dimension* of a non-empty concept class $\mathcal{C}$, denoted $\mathsf{Ldim}(\mathcal{C})$, to be the largest integer d such that there exists a mistake tree for $\mathcal{C}$ of depth d. If mistake trees for $\mathcal{C}$ of arbitrarily large d exist, then we say $\mathsf{Ldim}(\mathcal{C}) = \infty$. For convenience, it is also useful to define $\mathsf{Ldim}(\emptyset) = -1$.

The following facts about Ldim are easy to show.

- Ldim is monotone:
$$\mathcal{C}' \subseteq \mathcal{C} \implies \mathsf{Ldim}(\mathcal{C}') \leq \mathsf{Ldim}(\mathcal{C}).$$
- Only singleton sets have zero Ldim:
$$\mathsf{Ldim}(\mathcal{C}) = 0 \iff |\mathcal{C}| = 1.$$
- The construction in Figure 3.1 shows that $\mathsf{Ldim}(\mathsf{LTF}_{\mathbb{R}^1}) \geq 3$. In fact, the proof of Theorem 3.4 shows that $\mathsf{Ldim}(\mathsf{LTF}_{\mathbb{R}^1}) = \infty$.
- If there is a learning algorithm for the online concept learning scenario with concept class $\mathcal{C}$ and the realizability assumption with mistake bound M, then $\mathsf{Ldim}(\mathcal{C}) \leq M$. For example, we have the following:
  - $\mathsf{Ldim}(\mathcal{C}) \leq \log_2(|\mathcal{C}|)$ whenever $\mathcal{C}$ is finite (via Halving).
  - $\mathsf{Ldim}(\mathsf{CONJ}_n) \leq n + 1$ (via Attribute Elimination; see Exercise 3.3).

Exercise 3.8. Prove the facts about Ldim in the bullets above. $\Diamond$

Littlestone (1988) showed that the following algorithm, called the *Standard Optimal Algorithm (SOA)*, guarantees a mistake bound of $\mathsf{Ldim}(\mathcal{C})$, which is optimal by Theorem 3.6.

Algorithm 3.3: Standard Optimal Algorithm for the online concept learning scenario with concept class $\mathcal{C}$ and the realizability assumption.

- Initially, set $V_1 := \mathcal{C}$.
- In round t:
  - Observe $x_t \in \mathcal{X}$.
  - Let $V_t^y := \{c \in V_t \mid c(x_t) = y\}$ for each $y \in \{0, 1\}$.
  - Predict $a_t := \mathbb{1}\{\mathsf{Ldim}(V_t^1) > \mathsf{Ldim}(V_t^0)\}$.
  - Observe label $y_t \in \{0, 1\}$.
  - Update version space: $V_{t+1} := V_t^{y_t}$.

SOA maintains a *version space* based on input/label-pairs seen so far, i.e., $V_t \subseteq \mathcal{C}$ is the subset of concepts that are consistent with the first $t-1$ input/label-pairs. (To borrow a term from FTL and Halving, these are the concepts that are still "alive" at the start of round t.) In round t, the version space V_t is partitioned into two sets, V_t^0 and V_t^1, depending on how the concepts classify x_t. Unlike Halving, which just considers the cardinalities of the two sets to make its own prediction, SOA considers the Ldim of these two sets and follows the set with larger Ldim.

Theorem 3.7 (Littlestone, 1988)**.** *In the online concept learning scenario with the $\mathcal{C}$-realizability assumption, SOA makes at most $\mathsf{Ldim}(\mathcal{C})$ mistakes.*

Proof. We may assume without loss of generality that $\mathsf{Ldim}(\mathcal{C}) < \infty$. As mentioned above, for all $t \in \mathbb{N}$, we have that $V_t = \{c \in \mathcal{C} \mid c(x_i) = y_i \ \forall i \in \{1, \ldots, t-1\}\}$ is the version space at the start of round t. With the realizability assumption, V_t can never become empty, and hence $\mathsf{Ldim}(V_t) \geq 0$ for all $t \in \mathbb{N}$. If $\mathsf{Ldim}(V_t) = 0$, then V_t is a singleton, and the definition of Ldim and the realizability assumption ensure that SOA does not make a mistake.

Suppose SOA makes a mistake in round t. We claim that the updated version space, V_{t+1}, after the mistake satisfies

$$\mathsf{Ldim}(V_{t+1}) \leq \mathsf{Ldim}(V_t) - 1.$$

To show this, assume for sake of contradiction that $\mathsf{Ldim}(V_{t+1}) = \mathsf{Ldim}(V_t) =: d_t$. Since $V_{t+1} = V_t^{y_t}$, and by the monotonicity of Ldim and the choice of a_t,

$$\mathsf{Ldim}(V_t) \geq \mathsf{Ldim}(V_t^{a_t}) \geq \mathsf{Ldim}(V_t^{y_t}),$$

it must be that the above inequalities actually hold with equality; in particular,

$$\mathsf{Ldim}(V_t^0) = \mathsf{Ldim}(V_t^1).$$

We construct a mistake tree for V_t of depth $d_t + 1$ as follows: take depth d_t mistake trees for V_t^0 and V_t^1, and let their roots be the children of a new root node labeled by x_t. This is a mistake tree for

V_t and it has depth $d_t + 1$, which is a contradiction of the fact that $\mathsf{Ldim}(V_t) = d_t$. So we conclude that $\mathsf{Ldim}(V_{t+1}) \leq \mathsf{Ldim}(V_t) - 1$, i.e., every mistake leads to a reduction in Ldim by at least 1.

Since $\mathsf{Ldim}(V_1) = \mathsf{Ldim}(\mathcal{C})$ and $\mathsf{Ldim}(V_t) \geq 0$ for all $t \in \mathbb{N}$, it must be that the number of mistakes is at most $\mathsf{Ldim}(\mathcal{C})$. □

Exercise 3.9. Let $K_n = (V, E)$ denote the complete graph on $n \geq 4$ vertices.

(a) Let $\mathcal{C}_{K_n} := \{c_v \mid v \in V\}$ be the concept class on the domain E, where for each $v \in V$,

$$c_v(e) := \mathbb{1}\{v \in e\} \quad \forall e \in E.$$

Determine the value of $\mathsf{Ldim}(\mathcal{C}_{K_n})$ by exhibiting a mistake tree of the requisite depth and also giving a learning algorithm with the requisite mistake bound.

(b) If $\mathcal{C}$ is a concept class on the domain $\mathcal{X}$, then its *dual class* $\mathcal{C}^* := \{\mathsf{e}_x \mid x \in \mathcal{X}\}$ is a concept class on the domain $\mathcal{C}$, where for each $\mathsf{e}_x \in \mathcal{C}^*$ (called the *point evaluation function* for x),

$$\mathsf{e}_x(c) := c(x) \quad \forall c \in \mathcal{C}.$$

Determine the value of $\mathsf{Ldim}(\mathcal{C}^*_{K_n})$ (as above).

(c) Recall the $\mathsf{EXPERTS}_N$ concept class on the domain $\{0,1\}^N$. Describe the dual class $\mathsf{EXPERTS}^*_N$, and determine the value of $\mathsf{Ldim}(\mathsf{EXPERTS}^*_N)$ (as above). ◊

Topic 4: Statistical learning

4.1 Supervised machine learning

4.1.1 Machine learning as programming-by-example

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

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

4.1.2 Statistical framework

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

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

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

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

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

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

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

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

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

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

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

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

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

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

and

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

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

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

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

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

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

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

4.1.3 Probably Approximately Correct criterion

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

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

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

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

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

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

4.1.4 Role of test data

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

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

[2] "Efficient PAC learning" puts further requirements on the computational complexity of the learning algorithm; for now, we defer computational considerations until later in Section 7.1.

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

4.2 PAC learning from online concept learning

4.2.1 Batch versus online learning

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

4.2.2 From online to batch learning

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

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

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

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

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

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

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

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

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

where T is the number of training data.

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

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

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

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

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

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

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

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

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

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

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

and therefore

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

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

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

and the right-hand side is

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

with the second inequality coming from (4.1). □

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

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

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

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

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

□

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

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

□

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

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

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

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

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

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

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

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

4.3 Detour: Tools to bound the probability of large deviations

4.3.1 Chebyshev's inequality

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\Pr\big(|\mathbf{z} - \mathrm{E}(\mathbf{z})| > c\sqrt{n}\big) \leq \frac{9 - 5/n}{48c^4}.$$

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

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

4.3.2 Logarithmic moment generating function and the Cramér-Chernoff method

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

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

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

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

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

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

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

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

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

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

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

Indeed,

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

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

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

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

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

and

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

4.3.3 Rate function and Cramér's theorem

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

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

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

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

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

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

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

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

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

The claim now follows from Proposition 4.3. □

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

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

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

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

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

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

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

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

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

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

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

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

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

4.3.4 Subgaussianity and Hoeffding's inequality

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

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

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

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

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

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

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

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

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

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

Next, by Taylor's theorem with explicit remainder,

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

for some λ' between 0 and λ. Recall that

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

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

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

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

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

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

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

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

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

Proof. From Table 4.2, we have

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

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

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

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

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

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

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

4.3.5 Poisson heuristic and Bennett's inequality

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

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

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

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

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

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

By independence,

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

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

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

where h_{B} is defined in Table 4.2.

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

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

Therefore by Lemma 4.3,

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

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

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

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

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

(see Exercise 4.3). Therefore

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

Exercise 4.3. Prove (4.3). ◊

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

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

(b) Show that

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

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

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

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

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

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

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

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

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

4.4 PAC learning by minimizing training error rate

4.4.1 Union bound

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

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

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

4.4.2 Learning by finding a consistent hypothesis

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

4.4.3 Learning by finding a hypothesis with low training error rate

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

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

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

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

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

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

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

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

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

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

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

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

Henceforth we assume this event holds.

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

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

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

Then

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

Therefore

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

4.4.4 Occam's razor

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

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

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

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

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

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

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

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

□

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

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

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

Topic 5: Uniform convergence

5.1 Uniform convergence of training error rates

5.1.1 Finite hypothesis classes

Our approach to analyzing ERM in Theorem 4.4 was to establish *uniform convergence* of training error rates to population error rates for hypotheses from $\mathcal{H}$: for $\mathbf{S} \sim P^T$ for any probability distribution P over $\mathcal{X} \times \mathcal{Y}$,

$$\Pr\left(\forall h \in \mathcal{H},\ |\mathsf{err}(h;\mathbf{S}) - \mathsf{err}(h;P)| \leq \sqrt{\frac{\ln(2|\mathcal{H}|/\delta)}{T}}\right) \geq 1-\delta. \tag{5.1}$$

In the $1-\delta$ probability event from (5.1), a bound is established on the difference between training and population error rates that holds uniformly for all $h \in \mathcal{H}$. This should be contrasted with a statement of the following form:

$$\forall h \in \mathcal{H},\ \Pr\left(|\mathsf{err}(h;\mathbf{S}) - \mathsf{err}(h;P)| \leq \sqrt{\frac{\ln(2/\delta)}{T}}\right) \geq 1-\delta. \tag{5.2}$$

In each of the $1-\delta$ probability events from (5.2), the difference between training and population error rates is bounded only for a single hypothesis. Going from (5.2) to (5.1) was just a matter of applying the union bound, replacing δ in (5.2) with $\delta/|\mathcal{H}|$. However, this is only possible if $\mathcal{H}$ is finite; for infinite $\mathcal{H}$, the simple union bound does not work.

5.1.2 Impossibility for some infinite hypothesis classes

For some infinite hypothesis classes, there is no hope to establish a uniform bound on $|\mathsf{err}(h;\mathbf{S}) - \mathsf{err}(h;P)|$ that tends to zero with the number of training data. For example, suppose $\mathcal{X} = [0,1]$, $\mathcal{Y} = \{0,1\}$, and $\mathcal{H} = \{h_V \mid V \subseteq \mathcal{X}, |V| < \infty\}$, where $h_V(x) := \mathbb{1}\{x \in V\}$ for $x \in \mathcal{X}$. And suppose $(\mathbf{x},\mathbf{y}) \sim P$ has $\mathbf{x}$ uniformly distributed on $[0,1]$, and $\mathbf{y} = 0$ deterministically. Then for all $h_V \in \mathcal{H}$,

$$\mathsf{err}(h_V;P) = \Pr(h_V(\mathbf{x}) = 1) = \int_0^1 \mathbb{1}\{x \in V\}\,\mathrm{d}x = 0.$$

However, with $\mathbf{S} = ((\mathbf{x}_1,\mathbf{y}_1),\ldots,(\mathbf{x}_T,\mathbf{y}_T)) \sim P^T$ and $\mathbf{V} := \{\mathbf{x}_1,\ldots,\mathbf{x}_T\}$, we have

$$\mathsf{err}(h_{\mathbf{V}};\mathbf{S}) = 1.$$

Therefore, we always have

$$\exists h_V \in \mathcal{H} \text{ s.t. } |\mathsf{err}(h_V;\mathbf{S}) - \mathsf{err}(h_V;P)| = 1.$$

What makes the above hypothesis class unsuitable for uniform convergence? Are there infinite hypothesis classes for which a uniform convergence guarantee like (5.1) holds?

5.2 Vapnik-Chervonenkis theory

5.2.1 Empirical covers and empirical processes

Vapnik and Chervonenkis (1971) developed a theory to study algorithms for learning binary classifiers, and a key part of their theory is a technique for proving uniform convergence of training error rates for hypothesis classes. Their basic technique has two components. The first component characterizes which functions classes can be effectively replaced by a finite approximation. The relevant finite approximation is called an *empirical cover*, and there is a combinatorial property of a function class that determines if it always admits a non-trivial empirical cover.

The second component of the technique concerns how to effectively use a "union bound" over an empirical cover to establish uniform convergence of empirical means over an iid sample to the population means. The (random) probability distribution $\mathbf{P}_n$ that puts $1/n$ mass on each point in a size-n iid sample from P is called an *empirical distribution* (or *empirical measure*). (The average value of a function under $\mathbf{P}_n$ is called an *empirical average*.) The study of how $\mathbf{P}_n$ deviates from P is the subject of empirical process theory, and Vapnik and Chervonenkis drawn up techniques from that literature.

5.2.2 VC dimension

Let $\mathcal{F} \subseteq \{0,1\}^{\mathcal{Z}}$ be a class of $\{0,1\}$-valued functions defined on a domain $\mathcal{Z}$. The *behaviors* of $\mathcal{F}$ on a finite set of m points $\{z_1, \ldots, z_m\} \subseteq \mathcal{Z}$ is

$$\mathcal{F}[\{z_1, \ldots, z_m\}] := \{(f(z_1), \ldots, f(z_m)) \mid f \in \mathcal{F}\} \subseteq \{0,1\}^m.$$

Intuitively, we only care about the behaviors of $\mathcal{F}$ on a finite set of points when it comes to understanding the empirical averages of functions from $\mathcal{F}$. However, the number of behaviors, even on a finite set, can vary a lot depending on the function class. What about a function class determines whether the number of behaviors is small or large?

We say that $\mathcal{F}$ *shatters* the m points $\{z_1, \ldots, z_m\}$ if

$$\mathcal{F}[\{z_1, \ldots, z_m\}] = \{0,1\}^m;$$

that is, for each $(b_1, \ldots, b_m) \in \{0,1\}^m$, there exists some $f \in \mathcal{F}$ such that $f(z_i) = b_i$ for all $i \in [m]$. The *Vapnik-Chervonenkis (VC) dimension* of $\mathcal{F}$, denoted $\mathsf{VCdim}(\mathcal{F})$, is the largest value m such that there exists a set of m points from $\mathcal{Z}$ that is shattered by $\mathcal{F}$; if there are arbitrarily largest sets of point shattered by $\mathcal{F}$, then $\mathsf{VCdim}(\mathcal{F}) = \infty$.

Lemma 5.1. *For any $\mathcal{F} \subseteq \{0,1\}^{\mathcal{Z}}$, $\mathsf{VCdim}(\mathcal{F}) \leq \log_2(|\mathcal{F}|)$.*

Proof. If a set $\{z_1, \ldots, z_m\}$ of cardinality m is shattered by $\mathcal{F}$, then $|\mathcal{F}[\{z_1, \ldots, z_m\}]| = 2^m$. Now map every $f \in \mathcal{F}$ to $(f(z_1), \ldots, f(z_m)) \in \mathcal{F}[\{z_1, \ldots, z_m\}]$ to deduce $|\mathcal{F}| \geq 2^m$ as well. □

Example 5.1. Recall the class of monotone disjunctions MDISJ_n over the Boolean n-cube $\{0,1\}^n$:

$$\mathsf{MDISJ}_n := \{d_S \mid S \subseteq [n]\}$$

where $d_S(x_1, \ldots, x_n) = \bigvee_{j \in S} x_j$. We claim that $\mathsf{VCdim}(\mathsf{MDISJ}_n) = n$. We have $\mathsf{VCdim}(\mathsf{MDISJ}_n) \leq n$ by Lemma 5.1 since $|\mathsf{MDISJ}_n| = 2^n$. So it remains to show that $\mathsf{VCdim}(\mathsf{MDISJ}_n) \geq n$.

We claim that the set $T := \{e_1, \ldots, e_n\}$ is shattered by MDISJ_n; here, $e_i \in \{0,1\}^n$ is the vector with a 1 in the i-th position and 0s elsewhere. To show this, we must establish that every possible labeling of T can be realized by functions from MDISJ_n. Consider an arbitrary labeling $b \in \{0,1\}^n$, written $b = (b_1, \ldots, b_n)$; b_i is the desired label of e_i. Set $S := \{j \in [n] \mid b_j = 1\}$. Then for each $i \in [n]$,

$$d_S(e_i) = \bigvee_{j \in S} e_i[j] = \bigvee_{j \in S} \mathbb{1}\{i = j\} = \mathbb{1}\{i \in S\} = b_i.$$

Therefore the labeling b is realized by the monotone disjunction d_S. We conclude that T is shattered by MDISJ_n, and hence $\mathsf{VCdim}(\mathsf{MDISJ}_n) \geq |T| = n$. $\Diamond$

Example 5.2. Recall the class of linear threshold functions over Euclidean n-space $\mathbb{R}^n$:

$$\mathsf{LTF}_{\mathbb{R}^n} := \Big\{ f \colon \mathbb{R}^n \to \{0,1\} \;\Big|\; \exists (w, \theta) \in \mathbb{R}^n \times \mathbb{R} \text{ s.t. } f(x) = \mathbb{1}\{\langle w, x \rangle > \theta\} \Big\}.$$

We claim that $\mathsf{VCdim}(\mathsf{LTF}_{\mathbb{R}^n}) = n + 1$.

We first show $\mathsf{VCdim}(\mathsf{LTF}_{\mathbb{R}^n}) \geq n + 1$ by exhibiting a set of $n + 1$ points that is shattered by $\mathsf{LTF}_{\mathbb{R}^n}$. We claim that the set $T := \{0, e_1, \ldots, e_n\}$, where e_i is the i-th coordinate basis vector in $\mathbb{R}^n$, is shattered by $\mathsf{LTF}_{\mathbb{R}^n}$. To show this, we must establish that every possible labeling of T can be realized by functions from $\mathsf{LTF}_{\mathbb{R}^n}$. Consider an arbitrary labeling $b \in \{0,1\}^{n+1}$, written $b = (b_0, b_1, \ldots, b_n)$; b_0 is the desired label of 0, and for each $i \in [n]$, b_i is the desired label of e_i. Let $\tilde{b} := 2b - 1 = (\tilde{b}_0, \tilde{b}_1, \ldots, \tilde{b}_n) \in \{-1, 1\}^{n+1}$ be the $\{-1,1\}$-version of b, and consider the weight vector w and threshold θ given by

$$w := 2\tilde{b}_1 e_1 + \cdots + 2\tilde{b}_n e_n, \quad \theta := -\tilde{b}_0$$

Then

$$\mathbb{1}\{\langle w, 0 \rangle > \theta\} = \mathbb{1}\{0 > -\tilde{b}_0\} = b_0,$$

and for each $i \in [n]$,

$$\mathbb{1}\{\langle w, e_i \rangle > \theta\} = \mathbb{1}\{2\tilde{b}_i > -\tilde{b}_0\} = \mathbb{1}\{\tilde{b}_i + \tilde{b}_0/2 > 0\} = b_i.$$

Therefore the labeling b is realized by the LTF $x \mapsto \mathbb{1}\{\langle w, x \rangle > \theta\}$. We conclude that T is shattered by $\mathsf{LTF}_{\mathbb{R}^n}$, and hence $\mathsf{VCdim}(\mathsf{LTF}_{\mathbb{R}^n}) \geq |T| = n + 1$.

It now remains to show that $\mathsf{VCdim}(\mathsf{LTF}_{\mathbb{R}^n}) < n + 2$. To do this, we use the following simple observation (recalling the discussion in Section 3.3.3 about Perceptron): if a set $T \subset \mathbb{R}^n$ is shattered by $\mathsf{LTF}_{\mathbb{R}^n}$, then $\tilde{T} := \{(x, -1) \mid x \in T\} \subset \mathbb{R}^{n+1}$ is shattered by $\mathsf{HLTF}_{\mathbb{R}^{n+1}}$. So it suffices to show that no set of $n + 2$ points is shattered by $\mathsf{HLTF}_{\mathbb{R}^{n+1}}$. Consider any $n + 2$ points $x_1, \ldots, x_{n+2} \in \mathbb{R}^{n+1}$. They must be linearly dependent, so we can write

$$c_1 x_1 + \cdots + c_{n+2} x_{n+2} = 0$$

for some $c_1, \ldots, c_{n+2} \in \mathbb{R}$ that are not all zero. In particular,

$$\langle w, c_1 x_1 + \cdots + c_{n+2} x_{n+2} \rangle = 0, \quad \forall w \in \mathbb{R}^{n+1}. \tag{5.3}$$

We may assume without loss of generality that $c_i > 0$ for some $i \in [n+2]$ (or else replace all c_i with $-c_i$). Define $b = (b_1, \ldots, b_{n+2}) \in \{0,1\}^{n+2}$ by $b_i := \mathbb{1}\{c_i > 0\}$ for each $i \in [n+2]$. Suppose, for sake

of contradiction, that the labeling b is realized by some HLTF, i.e., there exists $w \in \mathbb{R}^{n+1}$ such that, for all $i \in [n+2]$,

$$\langle w, x_i \rangle > 0 \iff c_i > 0.$$

This implies

$$\left\langle w, \sum_{i=1}^{n+2} c_i x_i \right\rangle = \sum_{i=1}^{n+2} c_i \langle w, x_i \rangle > 0,$$

which is a contradiction of (5.3). So the labeling b of $x_1, \ldots, x_{n+2}$ cannot be realized by any HLTF. Since this holds for any choice of $n+2$ points from $\mathbb{R}^{n+1}$, it follows that no set of $n+2$ points from $\mathbb{R}^{n+1}$ is shattered by $\mathsf{HLTF}_{\mathbb{R}^{n+1}}$. This means that no set of $n+2$ points from $\mathbb{R}^n$ is shattered by $\mathsf{LTF}_{\mathbb{R}^n}$, and hence $\mathsf{VCdim}(\mathsf{LTF}_{\mathbb{R}^n}) < n+2$. ◊

Exercise 5.1. Let $\mathcal{X} := \{0, 1, 2, \ldots, N-1\}$ for $N \in \mathbb{N}$, and

$$\mathcal{H} := \{h \colon \mathcal{X} \to \{0,1\} \mid \exists \theta \in \mathbb{R} \text{ s.t. } h(x) = \mathbb{1}\{x > \theta\} \ \forall x \in \mathcal{X}\}.$$

What are $\mathsf{Ldim}(\mathcal{H})$ and $\mathsf{VCdim}(\mathcal{H})$? ◊

Exercise 5.2. Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ denote a hypothesis class on the domain $\mathcal{X}$. Let $\mathcal{H}^* := \{\mathsf{e}_x \mid x \in \mathcal{X}\}$ be the dual class where

$$\mathsf{e}_x(h) := h(x) \quad \forall h \in \mathcal{H}.$$

(a) Show that

$$\mathsf{VCdim}(\mathcal{H}^*) \leq 2^{\mathsf{VCdim}(\mathcal{H})+1} - 1.$$

(b) Let $\mathcal{X} := \mathbb{R}^n$ and $\mathcal{H} := \{h \colon \mathbb{R}^n \to \{0,1\} \mid \exists w \in \mathbb{R}^n \text{ s.t. } h(x) = \mathbb{1}\{\langle w, x \rangle > 0\} \ \forall x \in \mathbb{R}^n\}$. Show that $\mathsf{VCdim}(\mathcal{H}^*) = \mathsf{VCdim}(\mathcal{H})$. ◊

5.2.3 Sauer's lemma

If $\mathcal{F}$ has VC dimension d, then (by definition) there are finite point sets $T \subset \mathcal{Z}$ of all sizes $|T| \leq d$ for which the behaviors of $\mathcal{F}$ on T is all behaviors: $|\mathcal{F}[T]| = 2^{|T|}$. However, if $d < \infty$, this exponential growth stops as soon as $|T| > d$, after which the number of behaviors grows at most polynomially.

Define the *growth function* $\Pi_{\mathcal{F}} \colon \mathbb{N} \to \mathbb{N}$ for the function class $\mathcal{F}$ by

$$\Pi_{\mathcal{F}}(n) := \max\big\{|\mathcal{F}[T]| \ \big| \ T \subseteq \mathcal{Z}, |T| = n\big\}, \quad \forall n \in \mathbb{N}.$$

Sauer's lemma, given in Lemma 5.2, implies that $\Pi_{\mathcal{F}}$ is asymptotically upper-bounded by a polynomial in n whenever $\mathsf{VCdim}(\mathcal{F}) < \infty$.

Lemma 5.2 (Sauer, 1972; Shelah, 1972)**.** *If* $d := \mathsf{VCdim}(\mathcal{F}) < \infty$*, then* $\Pi_{\mathcal{F}}(n) \leq \binom{n}{\leq d} := \sum_{k=0}^{d} \binom{n}{k}$.

There are a number of proofs of Lemma 5.2. The proof we give below is based on those of Alon (1983) and Frankl (1983) using a technique called shifting.

Lemma 5.2 concerns subsets of the Boolean n-cube $\{0,1\}^n$, so it is useful to specialize the notions of shattering and VC dimension to such subsets. For a subset $V \subseteq \{0,1\}^n$ and $J \subseteq [n]$, let $V|_J := \{(v_i)_{i \in J} \mid v \in V\}$. We say V shatters J if $V|_J = \{0,1\}^J$, and define $\mathsf{VCdim}(V)$ to be the largest value d such that there exists $J \subseteq [n]$ of cardinality $|J| = d$ such that V shatters T.

Note that each $v \in \{0,1\}^n$ can also be regarded as the characteristic vector for a subset of $[n]$; we will write v to also refer to that subset. With this overloaded notation, we have that $V \subseteq 2^{[n]}$ shatters $J \subseteq [n]$ if and only if $\{v \cap J \mid v \in V\} = 2^J$. When v is regarded this way, we also use the shorthand $v - i$ to mean $v \setminus \{i\}$ for any $i \in [n]$.

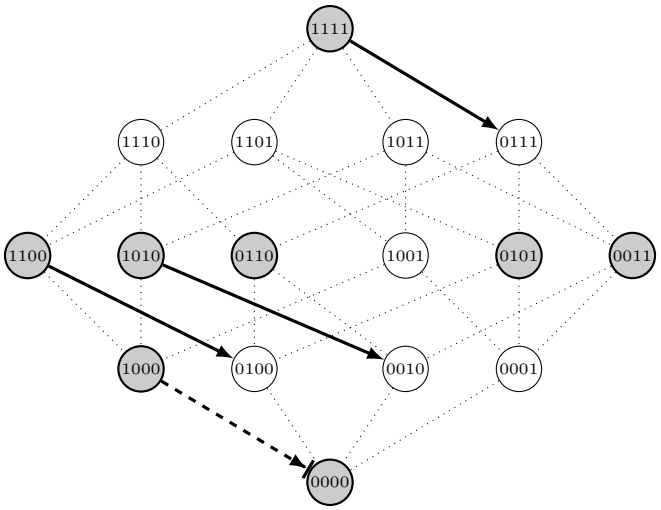


Figure 5.1: Shifting $V \subseteq \{0,1\}^n$ (for $n = 4$) along coordinate $i = 1$. The family V is indicated by the gray colored vertices. (Vertices corresponding to $v \in \{0,1\}^n \setminus V$ are also depicted but colored white.) The dark solid arrows indicate $v \in V$ that are successfully shifted (e.g., $\mathsf{S}_{1,V}((1,1,0,0)) = (0,1,0,0)$). Successful shifts move $v \in V$ down to lower levels of the Boolean n-cube. The dark dashed arrow indicates an unsuccessful attempt to shift $v = (1,0,0,0)$ that is blocked by the presence of $(0,0,0,0) \in V$.

Proposition 5.1. *For any finite set $T \subseteq \mathcal{Z}$, we have $\mathsf{VCdim}(\mathcal{F}[T]) \leq \mathsf{VCdim}(\mathcal{F})$.*

Exercise 5.3. Prove Proposition 5.1, and explain in what sense the inequality is tight. ◊

The goal of the proof of Lemma 5.2 is to perform some transformations to $V := \mathcal{F}[T]$ (for some $T \subseteq \mathcal{Z}$ with $|T| = n$) that (i) preserve the cardinality of V, and (ii) do not increase the VC dimension of V. After all of the transformations, the resulting family $W \subseteq 2^{[n]}$ will have some useful properties that make it very easy to bound $|W|$ (and hence also $|V|$).

The transformations to V are called *(down)shifting* along a coordinate $i \in [n]$. The shifting transformations are as follows.

$$\text{Shift } v \in V \text{ along coordinate } i \in [n]: \qquad \mathsf{S}_{i,V}(v) := \begin{cases} v - i & \text{if } i \in v \text{ and } v - i \notin V, \\ v & \text{otherwise.} \end{cases}$$

$$\text{Shift } V \text{ along coordinate } i \in [n]: \qquad \mathsf{S}_i(V) := \{\mathsf{S}_{i,V}(v) \mid v \in V\}.$$

If $\mathsf{S}_{i,V}(v) = v - i \neq v$, then we say that v is successfully shifted along coordinate i. See Figure 5.1 for an illustration. There are two possible reasons for an unsuccessful shift along coordinate i. First, it could be that $i \notin v$. Second, it could be that $i \in v$ but $v - i \in V$; in this case, the attempted shift is "blocked" by the presence of $v - i \in V$.

Exercise 5.4. For the $V \subseteq \{0,1\}^4$ from Figure 5.1, list the members of $\mathsf{S}_2(V)$. ◊

Proposition 5.2. *For any $V \subseteq 2^{[n]}$ and any $i \in [n]$:*

1. $|\mathsf{S}_i(V)| = |V|$.

2. *For any $J \subseteq [n]$, if J is shattered by $\mathsf{S}_i(V)$, then J is shattered by V. (Consequently, $\mathsf{VCdim}(\mathsf{S}_i(V)) \leq \mathsf{VCdim}(V)$.)*

Proof. Observe that $v \in V$ is successfully shifted by $\mathsf{S}_{i,V}$ if and only if $i \in v$ and $v - i \notin V$; so V and $\mathsf{S}_i(V)$ are in 1-to-1 correspondence (via $\mathsf{S}_{i,V}$). This proves the first item of the claim.

To prove the second item, consider any $J \subseteq [n]$ shattered by $\mathsf{S}_i(V)$, so $\{\mathsf{S}_{i,V}(v) \cap J \mid v \in V\} = 2^J$. We need to show that J is also shattered by V. There are two cases to consider.

- Case 1: $i \notin J$. Then for every $v \in V$, we have $\mathsf{S}_{i,V}(v) \cap J = v \cap J$ since $\mathsf{S}_{i,V}$ only (possibly) removes i from v. So $\{v \cap J \mid v \in V\} = \{\mathsf{S}_{i,V}(v) \cap J \mid v \in V\} = 2^J$, which shows that J is also shattered by V.

- Case 2: $i \in J$. Since $\{\mathsf{S}_{i,V}(v) \cap J \mid v \in V\} = 2^J$, for any choice of $K \in 2^J$, there exists $v \in V$ such that
$$\mathsf{S}_{i,V}(v) \cap J = K \cup \{i\}.$$
This means that $i \in \mathsf{S}_{i,V}(v)$, implying that $\mathsf{S}_{i,V}(v) = v$, an unsuccessful shift of $v \in V$ along coordinate i, blocked by the presence of $u := v - i$ in V. Therefore, we have both
$$u \cap J = K \setminus \{i\} \in V \quad \text{and}$$
$$v \cap J = K \cup \{i\} \in V.$$
Since this holds for all choices of $K \subseteq J$, it follows that V also shatters J. □

Exercise 5.5. Can there exist $V \subseteq \{0,1\}^n$ such that $\mathsf{VCdim}(\mathsf{S}_1(V)) < \mathsf{VCdim}(V)$? ◊

Suppose, starting with V, we repeatedly apply S_i to V for each $i \in [n]$. Whenever there is a successful shift, the "total weight" of V (i.e., $\sum_{v \in V} |v|$) decreases by at least 1. So this process must eventually stop with some set $W \subseteq 2^{[n]}$ with $|W| = |V|$ and $\mathsf{VCdim}(W) \leq \mathsf{VCdim}(V)$.

Proposition 5.3. *Suppose $W \subseteq 2^{[n]}$ satisfies $\mathsf{S}_i(W) = W$ for each $i \in [n]$. For each $v \in W$, we have $u \in W$ for all $u \subseteq v$ (i.e., W is* downward-closed*).*

Proof. Pick any nonempty $v \in W$ and any $i \in v$. If $v - i \notin W$, then $\mathsf{S}_{i,W}(v)$ would be a successful shift, a contradiction of the assumption that $\mathsf{S}_i(W) = W$. Therefore it must be that $v - i \in W$. Repeatedly applying this argument shows that every subset $u \subseteq v$ is also in W. □

Proposition 5.4. *Suppose $W \subseteq 2^{[n]}$ is downward-closed, and let $d := \mathsf{VCdim}(W)$. Then*
$$|W| \leq \binom{n}{\leq d}.$$

Proof. We first claim that every $v \in W$ has $|v| \leq d$. To see this, consider any $v \in W$, and let $J := v$. Since W is downward-closed, W shatters J. Since $\mathsf{VCdim}(W) = d$, it must be that $|J| \leq d$. This establishes $|v| \leq d$.

So, every $v \in W$ is a subset of $[n]$ of cardinality at most d; and there are at most $\binom{n}{\leq d}$ subsets of $[n]$ of cardinality at most d. □

Combining Propositions 5.1 to 5.4 (along with the above observation that the number of successful shifts is finite) completes the proof Lemma 5.2.

5.2.4 Population and empirical averages

Let P be a probability distribution over $\mathcal{Z}$ (which we regard as the "population"), and let $\mathbf{z}_{1:n} := (\mathbf{z}_1, \dots, \mathbf{z}_n) \sim P^n$ be a size-n iid sample from P. For a real-valued function $f\colon \mathcal{Z} \to \mathbb{R}$, we are concerned with two different "averages":

- the *population average of f*
$$\mathrm{E}[f(\mathbf{z})]$$
(where $\mathbf{z} \sim P$);
- the *empirical average of f*
$$\frac{1}{n}\sum_{i=1}^{n} f(\mathbf{z}_i).$$

Our goal is to show the following theorem.

Theorem 5.1. *Let $\mathcal{F} \subseteq \{0,1\}^{\mathcal{Z}}$ be a function class with $d := \mathsf{VCdim}(\mathcal{F}) < \infty$. Let P be a probability distribution over $\mathcal{Z}$ and $(\mathbf{z}_1, \dots, \mathbf{z}_n) \sim P^n$. Then*

$$\mathrm{E}\left[\sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n} f(\mathbf{z}_i) - \mathrm{E}[f(\mathbf{z})]\right|\right] \le 2\sqrt{\frac{2\ln(2\binom{n}{\le d})}{n}}.$$

A proof that has a lot of overlap with that of Theorem 5.1 establishes the following theorem.

Theorem 5.2 (Vapnik and Chervonenkis, 1971)**.** *Let $\mathcal{F} \subseteq \{0,1\}^{\mathcal{Z}}$ be a function class with $d := \mathsf{VCdim}(\mathcal{F}) < \infty$. Let P be a probability distribution over $\mathcal{Z}$ and $(\mathbf{z}_1, \dots, \mathbf{z}_n) \sim P^n$. Then for any $\delta \in (0,1)$, with probability at least $1-\delta$,*

$$\sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n} f(\mathbf{z}_i) - \mathrm{E}[f(\mathbf{z})]\right| \le O\left(\sqrt{\frac{\log(\binom{n}{\le d}/\delta)}{n}}\right).$$

It is also possible to derive Theorem 5.2 as an immediate corollary of Theorem 5.1 by applying a probability inequality called McDiarmid's inequality. We focus on Theorem 5.1 because its proof is more streamlined than the direct proof of Theorem 5.2 but retains most of the key ideas.

In Theorems 5.1 and 5.2, because $\mathcal{F}$ has finite VC dimension, we might hope that we can replace the supremum over $\mathcal{F}$ with a maximum over the set of "behaviors" of $\mathcal{F}$ on $\{\mathbf{z}_1, \dots, \mathbf{z}_n\}$. The thought would be that (at least for Theorem 5.2) because there are only $\binom{n}{\le d}$ behaviors, we can establish a probability inequality for each behavior separately, and then apply a union bound. This is almost how the proofs of Theorems 5.1 and 5.2 work, but some important issues arise that require careful attention.

The first issue is that, even though $\mathcal{F}$ has only finitely-many behaviors on the size-n iid sample—and hence finitely-many values of the empirical average $\frac{1}{n}\sum_{i=1}^{n} f(\mathbf{z}_i)$ as f ranges over $\mathcal{F}$—there may be infinitely-many values of the population average $\mathrm{E}[f(\mathbf{z})]$ as f ranges over $\mathcal{F}$.

A way to resolve this issue is based on the following observation: if $\mathbf{x}$ and $\mathbf{x}'$ are iid real-valued random variables, then $\mathrm{E}[\mathbf{x}] = \mathrm{E}[\mathbf{x}' \mid \mathbf{x}]$, which implies $\mathrm{E}[|\mathbf{x} - \mathrm{E}[\mathbf{x}]|] \le \mathrm{E}[|\mathbf{x} - \mathbf{x}'|]$.[1] This suggests

[1] In fact, if we considered squared differences instead of absolute differences, the left-hand side would be equal to half the right-hand side: $\mathrm{E}[(\mathbf{x} - \mathrm{E}[\mathbf{x}])^2] = \frac{1}{2}\mathrm{E}[(\mathbf{x} - \mathbf{x}')^2]$.

that, for the purposes of analysis only, it may be useful to consider a second size-n iid sample $\mathbf{z}'_{1:n} := (\mathbf{z}'_1, \dots, \mathbf{z}'_n) \sim P^n$ that is independent of—but identically distributed as—the original sample $\mathbf{z}_{1:n}$. We call this second sample the *ghost sample* 👻. If we could replace $\mathrm{E}[f(\mathbf{z})]$ with the empirical average of f on the ghost sample, then we could again hope to use the fact that the (ghost) empirical average takes only finitely-many values as f ranges over $\mathcal{F}$. (It is still not clear how to exactly take advantage of this, but intuitively this could seem like progress.)

Proposition 5.5 (Symmetrization by ghost sample)**.** *Let $\mathbf{z}_{1:n}$ and $\mathbf{z}'_{1:n}$ be independent size-n iid samples from a probability distribution P over $\mathcal{Z}$. For any $\mathcal{F} \subseteq \{0,1\}^{\mathcal{Z}}$,*

$$\mathrm{E}\left[\sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n} f(\mathbf{z}_i) - \mathrm{E}[f(\mathbf{z})]\right|\right] \le \mathrm{E}\left[\sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n} f(\mathbf{z}_i) - \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{z}'_i)\right|\right].$$

Proof. Define

$$\boldsymbol{\Delta} f := \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{z}_i) - \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{z}'_i)$$

to be the difference between the two empirical averages of $f\colon \mathcal{Z} \to \mathbb{R}$. Consider any $f \in \mathcal{F}$, and observe that

$$\mathrm{E}[f(\mathbf{z})] = \mathrm{E}\left[\frac{1}{n}\sum_{i=1}^{n} f(\mathbf{z}_i)\right] = \mathrm{E}\left[\frac{1}{n}\sum_{i=1}^{n} f(\mathbf{z}'_i) \,\middle|\, \mathbf{z}_{1:n}\right]$$

since $\mathbf{z}_{1:n}$ and $\mathbf{z}'_{1:n}$ are independent and identically distributed. Therefore,

$$\left|\frac{1}{n}\sum_{i=1}^{n} f(\mathbf{z}_i) - \mathrm{E}[f(\mathbf{z})]\right| = \left|\mathrm{E}\left[\boldsymbol{\Delta} f \,\middle|\, \mathbf{z}_{1:n}\right]\right| \le \mathrm{E}\left[|\boldsymbol{\Delta} f| \,\middle|\, \mathbf{z}_{1:n}\right]$$

where the inequality follows by Jensen's inequality ($|\mathrm{E}[\mathbf{x}]| \le \mathrm{E}[|\mathbf{x}|]$). We also have

$$|\boldsymbol{\Delta} f| \le \sup_{f'\in\mathcal{F}} |\boldsymbol{\Delta} f'|,$$

and therefore

$$\mathrm{E}\left[|\boldsymbol{\Delta} f| \,\middle|\, \mathbf{z}_{1:n}\right] \le \mathrm{E}\left[\sup_{f'\in\mathcal{F}} |\boldsymbol{\Delta} f'| \,\middle|\, \mathbf{z}_{1:n}\right].$$

Since $f \in \mathcal{F}$ was arbitrary, we have

$$\sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n} f(\mathbf{z}_i) - \mathrm{E}[f(\mathbf{z})]\right| \le \mathrm{E}\left[\sup_{f'\in\mathcal{F}} |\boldsymbol{\Delta} f'| \,\middle|\, \mathbf{z}_{1:n}\right].$$

Taking expectation of both sides gives the claim. □

Exercise 5.6. In the same setting as Proposition 5.5, show that

$$\mathrm{E}\left[\sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n} f(\mathbf{z}_i) - \mathrm{E}[f(\mathbf{z})]\right|\right] \ge \frac{1}{2}\,\mathrm{E}\left[\sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n} f(\mathbf{z}_i) - \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{z}'_i)\right|\right]. \qquad \lozenge$$

Proposition 5.5 is called a "symmetrization" argument, because it effectively replaces each (possibly) asymmetric random variable $f(\mathbf{z}_i) - \mathrm{E}[f(\mathbf{z})]$ with a symmetric random variable $f(\mathbf{z}_i) - f(\mathbf{z}'_i)$. So, to prove Theorem 5.1, it suffices bound the right-hand side in Proposition 5.5. (In view of Exercise 5.6, at most a constant factor is lost.) But this raises the second issue: how to take advantage of the finitely-many behaviors of $\mathcal{F}$ on the (two) iid samples. Note that the set of behaviors is itself a random subset of $\{0,1\}^{2n}$; what leverage is gotten by fixing this set of behaviors?

The expression on the right-hand side in Proposition 5.5 concerns two independent iid samples, $\mathbf{z}_{1:n}$ and $\mathbf{z}'_{1:n}$, which we can imagine are derived by starting with a single size-$2n$ iid sample, say, $(\mathbf{z}^a_1, \mathbf{z}^b_1, \ldots, \mathbf{z}^a_n, \mathbf{z}^b_n) \sim P^{2n}$, and then, for each $i \in [n]$, assigning one of $\mathbf{z}^a_i$ and $\mathbf{z}^b_i$ to the original sample, and then assigning the other to the ghost sample. Any way of making this assignment is valid as long as it does not depend on the size-$2n$ iid sample. This flexibility turns out to be key.

Proposition 5.6. *Let $\mathbf{z}_{1:n}$ and $\mathbf{z}'_{1:n}$ be independent size-n iid samples from a probability distribution P over $\mathcal{Z}$. For any $\mathcal{F} \subseteq \{0,1\}^{\mathcal{Z}}$ and any $(\sigma_1, \ldots, \sigma_n) \in \{-1,1\}^n$,*

$$\mathrm{E}\left[\sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n}(f(\mathbf{z}_i) - f(\mathbf{z}'_i))\right|\right] \le 2\,\mathrm{E}\left[\sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n}\sigma_i f(\mathbf{z}_i)\right|\right].$$

Proof. Let $(\mathbf{z}^a_1, \mathbf{z}^b_1, \ldots, \mathbf{z}^a_n, \mathbf{z}^b_n) \sim P^{2n}$ be a size-$2n$ iid sample from P. We determine $\mathbf{z}_{1:n}$ and $\mathbf{z}'_{1:n}$ as follows. For each $i \in [n]$:

- If $\sigma_i = +1$, then $\mathbf{z}_i := \mathbf{z}^a_i$ and $\mathbf{z}'_i := \mathbf{z}^b_i$.
- If $\sigma_i = -1$, then $\mathbf{z}_i := \mathbf{z}^b_i$ and $\mathbf{z}'_i := \mathbf{z}^a_i$.

Then $\mathbf{z}_{1:n}$ and $\mathbf{z}'_{1:n}$ are independent and each is a size-n iid sample from P. Moreover,

$$\sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n}(f(\mathbf{z}_i) - f(\mathbf{z}'_i))\right| = \sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n}\sigma_i(f(\mathbf{z}^a_i) - f(\mathbf{z}^b_i))\right|. \tag{5.4}$$

By the triangle inequality,

$$\sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n}\sigma_i(f(\mathbf{z}^a_i) - f(\mathbf{z}^b_i))\right| \le \sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n}\sigma_i f(\mathbf{z}^a_i)\right| + \sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n}\sigma_i f(\mathbf{z}^b_i)\right|. \tag{5.5}$$

Therefore, combining (5.4) and (5.5) and taking expectations, we get

$$\begin{aligned}\mathrm{E}\left[\sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n}(f(\mathbf{z}_i) - f(\mathbf{z}'_i))\right|\right] &\le \mathrm{E}\left[\sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n}\sigma_i f(\mathbf{z}^a_i)\right| + \sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n}\sigma_i f(\mathbf{z}^b_i)\right|\right] \\ &= 2\,\mathrm{E}\left[\sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n}\sigma_i f(\mathbf{z}_i)\right|\right]\end{aligned}$$

where the last step uses linearity of expectation and the fact that $(\mathbf{z}_1, \ldots, \mathbf{z}_n)$, $(\mathbf{z}^a_1, \ldots, \mathbf{z}^a_n)$, and $(\mathbf{z}^b_1, \ldots, \mathbf{z}^b_n)$ have the same distribution. □

At face value, it may seem that the only purpose of Proposition 5.6 is to bound a quantity involving two size-n iid samples (or a size-$2n$ iid sample) with a quantity involving only one

size-n iid sample. This is helpful, but not the main point. In fact, for any fixed assignment $(\sigma_1, \ldots, \sigma_n) \in \{-1, 1\}^n$, Proposition 5.6 is a "de-symmetrization", as it moves from the symmetric random variable $f(\mathbf{z}_i) - f(\mathbf{z}'_i)$ to the (possibly) asymmetric random variable $\sigma_i f(\mathbf{z}_i)$!

The key is that the bound in Proposition 5.6 holds for every choice of the assignment $(\sigma_1, \ldots, \sigma_n)$, so the right-hand side (without the factor of 2) can be replaced by

$$\frac{1}{2^n} \sum_{\sigma \in \{-1,1\}^n} \mathrm{E}\left[\sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^n \sigma_i f(\mathbf{z}_i) \right| \right].$$

If we introduce a random sign vector $(\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_n)$ that is (i) distributed uniformly over $\{-1, 1\}^n$, and (ii) regarded as independent of $\mathbf{z}_{1:n}$, then the above expression is identical to

$$\mathrm{E}\left[\sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^n \boldsymbol{\sigma}_i f(\mathbf{z}_i) \right| \right].$$

The $\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_n$ are called iid *Rademacher random variables*. Combining Propositions 5.5 and 5.6 immediately gives the following proposition.

Proposition 5.7 (Symmetrization by random signs)**.** *Let $\mathbf{z}_{1:n}$ a size-n iid sample from a probability distribution P over $\mathcal{Z}$. Let $(\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_n)$ be n iid Rademacher random variables, independent of $\mathbf{z}_{1:n}$. For any $\mathcal{F} \subseteq \{0, 1\}^{\mathcal{Z}}$,*

$$\mathrm{E}\left[\sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^n f(\mathbf{z}_i) - \mathrm{E}[f(\mathbf{z})] \right| \right] \leq 2\,\mathrm{E}\left[\sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^n \boldsymbol{\sigma}_i f(\mathbf{z}_i) \right| \right].$$

The last step is to use the tower property of conditional expectations:

$$\mathrm{E}\left[\sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^n \boldsymbol{\sigma}_i f(\mathbf{z}_i) \right| \right] = \mathrm{E}\left[\mathrm{E}\left[\sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^n \boldsymbol{\sigma}_i f(\mathbf{z}_i) \right| \,\middle|\, \mathbf{z}_{1:n} \right]\right], \tag{5.6}$$

and now focus on the inner conditional expectation. In this conditional expectation, the iid sample $\mathbf{z}_{1:n}$ is fixed. We now (finally!) regard the supremum over $\mathcal{F}$ as a maximum over the behaviors $\mathcal{F}[\{\mathbf{z}_1, \ldots, \mathbf{z}_n\}] \subseteq \{0, 1\}^n$, and we apply the following lemma.

Lemma 5.3 (Massart's finite class lemma)**.** *Let $(\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_n)$ be n iid Rademacher random variables. For any finite subset $V \subseteq \mathbb{R}^n$ with $\|v\| \leq \sqrt{n}$ for all $v \in V$,*

$$\mathrm{E}\left[\max_{(v_1,\ldots,v_n) \in V} \left| \frac{1}{n} \sum_{i=1}^n \boldsymbol{\sigma}_i v_i \right| \right] \leq \sqrt{\frac{2 \ln(2|V|)}{n}}.$$

Proof. For each $v = (v_1, \ldots, v_n) \in V' := V \cup (-V)$, let $\mathbf{x}_v := \frac{1}{n} \sum_{i=1}^n \boldsymbol{\sigma}_i v_i$. Then

$$\max_{(v_1,\ldots,v_n) \in V} \left| \frac{1}{n} \sum_{i=1}^n \boldsymbol{\sigma}_i v_i \right| = \max_{v \in V'} \mathbf{x}_v.$$

For each $v = (v_1, \ldots, v_n) \in V'$, the random variables $(v_1/n)\boldsymbol{\sigma}_1, \ldots, (v_n/n)\boldsymbol{\sigma}_n$ are independent, and $(v_i/n)\boldsymbol{\sigma}_i$ is $(v_i/n)^2$-subgaussian (by Lemma 4.1), so $\mathbf{x}_v$ is $\|v\|^2/n^2$-subgaussian:

$$\mathsf{K}_{\mathbf{x}_v}(\lambda) \leq \frac{\|v\|^2 \lambda^2}{2n^2} \leq \frac{\lambda^2}{2n}.$$

Fix $\lambda > 0$. Then by Jensen's inequality,

$$\begin{aligned}
\mathrm{E}\left[\max_{v\in V'} \mathbf{x}_v\right] &\le \frac{1}{\lambda} \ln \mathrm{E}\left[\exp\left(\lambda \max_{v\in V'} \mathbf{x}_v\right)\right] \\
&= \frac{1}{\lambda} \ln \mathrm{E}\left[\max_{v\in V'} \exp(\lambda \mathbf{x}_v)\right] \\
&\le \frac{1}{\lambda} \ln \mathrm{E}\left[\sum_{v\in V'} \exp(\lambda \mathbf{x}_v)\right] \\
&= \frac{1}{\lambda} \ln\left(\sum_{v\in V'} \exp(\mathsf{K}_{\mathbf{x}_v}(\lambda))\right).
\end{aligned}$$

Now applying the bound on $\mathsf{K}_{\mathbf{x}_v}(\lambda)$ for each $v \in V'$, we obtain

$$\mathrm{E}\left[\max_{v\in V'} \mathbf{x}_v\right] \le \frac{\ln|V'|}{\lambda} + \frac{1}{\lambda} \cdot \frac{\lambda^2}{2n}. = \frac{\ln|V'|}{\lambda} + \frac{\lambda}{2n}.$$

Plugging in $\lambda = \sqrt{2n \ln|V'|}$ and using the fact $|V'| \le 2|V|$ gives the claim. □

Lemma 5.3 gives the following bound on the conditional expectation from (5.6):

$$\mathrm{E}\left[\sup_{f\in\mathcal{F}} \left|\frac{1}{n}\sum_{i=1}^n \boldsymbol{\sigma}_i f(\mathbf{z}_i)\right| \,\middle|\, \mathbf{z}_{1:n}\right] \le \sqrt{\frac{2\ln(2|\mathcal{F}[\{\mathbf{z}_1,\ldots,\mathbf{z}_n\}]|)}{n}}.$$

Now we appeal to Lemma 5.2 to bound $|\mathcal{F}[\{\mathbf{z}_1,\ldots,\mathbf{z}_n\}]|$ by $\binom{n}{\le d}$, where $d = \mathsf{VCdim}(\mathcal{F})$. Putting this together with Proposition 5.7 proves Theorem 5.1.

Exercise 5.7. In the proof of Proposition 5.6, if we had taken expectations after (5.4), and then replaced the assignment $(\sigma_1,\ldots,\sigma_n)$ with a random assignment $(\boldsymbol{\sigma}_1,\ldots,\boldsymbol{\sigma}_n)$ (as in Proposition 5.7), we would have gotten the following variation on Proposition 5.7:

$$\mathrm{E}\left[\sup_{f\in\mathcal{F}} \left|\frac{1}{n}\sum_{i=1}^n f(\mathbf{z}_i) - \mathrm{E}[f(\mathbf{z})]\right|\right] \le \mathrm{E}\left[\sup_{f\in\mathcal{F}} \left|\frac{1}{n}\sum_{i=1}^n \boldsymbol{\sigma}_i(f(\mathbf{z}_i^a) - f(\mathbf{z}_i^b))\right|\right].$$

(a) Explain how to use Massart's finite class lemma (Lemma 5.3) and Sauer's lemma (Lemma 5.2) directly with the right-hand side of the above expression, without passing through Proposition 5.7. How does the result compare to Theorem 5.1?

(b) Show that

$$\begin{aligned}
\mathrm{E}\left[\sup_{f\in\mathcal{F}} \left|\frac{1}{n}\sum_{i=1}^n \boldsymbol{\sigma}_i(f(\mathbf{z}_i^a) - f(\mathbf{z}_i^b))\right|\right] &\ge \mathrm{E}\left[\sup_{f\in\mathcal{F}} \left|\frac{1}{n}\sum_{i=1}^n \boldsymbol{\sigma}_i(f(\mathbf{z}_i) - \mathrm{E}[f(\mathbf{z})])\right|\right] \\
&\ge \mathrm{E}\left[\sup_{f\in\mathcal{F}} \left|\frac{1}{n}\sum_{i=1}^n \boldsymbol{\sigma}_i f(\mathbf{z}_i)\right|\right] - \frac{\sup_{f\in\mathcal{F}}|\mathrm{E}[f(\mathbf{z})]|}{\sqrt{n}}.
\end{aligned}$$ ◊

5.2.5 Zero-one loss class

How can Theorem 5.1 be used to understand the convergence of training error rates to population error rates for hypotheses in a hypothesis class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$? The key is simply to define a function class $\mathcal{F}_{\mathcal{H}} := \{f_h \mid h \in \mathcal{H}\}$ over the domain $\mathcal{Z} := \mathcal{X} \times \{0,1\}$, where $f_h(x,y) = \mathbb{1}\{h(x) \neq y\}$. This function class $\mathcal{F}_{\mathcal{H}}$ is the *zero-one loss class* associated with $\mathcal{H}$. If P is a probability distribution on $\mathcal{Z} = \mathcal{X} \times \{0,1\}$ and we consider an iid sample from P, then the population averages of $f_h \in \mathcal{F}_{\mathcal{H}}$ is the population error rate of h, and the empirical average of f_h is the training error rate of h.

Proposition 5.8. $\mathsf{VCdim}(\mathcal{F}_{\mathcal{H}}) = \mathsf{VCdim}(\mathcal{H})$.

Proof. We first show that $\mathsf{VCdim}(\mathcal{F}_{\mathcal{H}}) \geq \mathsf{VCdim}(\mathcal{H})$. Take any $T := \{x_1, \ldots, x_d\} \subseteq \mathcal{X}$ shattered by $\mathcal{H}$, i.e., $\mathcal{H}[T] = \{0,1\}^d$. Now define $T' := \{(x_1, 0), \ldots, (x_d, 0)\} \subseteq \mathcal{X} \times \{0,1\}$. For each $h \in \mathcal{H}$, we have $f_h(x_i, 0) = \mathbb{1}\{h(x_i) \neq 0\} = h(x_i)$. So $\mathcal{F}_{\mathcal{H}}[T'] = \mathcal{H}[T] = \{0,1\}^d$, and hence T' is shattered by $\mathcal{F}_{\mathcal{H}}$. This implies $\mathsf{VCdim}(\mathcal{F}_{\mathcal{H}}) \geq \mathsf{VCdim}(\mathcal{H})$.

Now we show that $\mathsf{VCdim}(\mathcal{F}_{\mathcal{H}}) \leq \mathsf{VCdim}(\mathcal{H})$. Take any $T' := \{(x_1, y_1), \ldots, (x_d, y_d)\} \subseteq \mathcal{X} \times \{0,1\}$ of d distinct points shattered by $\mathcal{F}_{\mathcal{H}}$. We want to show that $T := \{x_1, \ldots, x_d\}$ is a set of d points shattered by $\mathcal{H}$.

We first show that $x_1, \ldots, x_d$ are distinct points. To see this, suppose for sake of contradiction that we have $x_i = x_j$ for some $i \neq j$. Then it must be that $y_i \neq y_j$. But for any $h \in \mathcal{H}$, we have

$$f_h((x_i, y_i)) = \mathbb{1}\{h(x_i) \neq y_i\} = 1 - \mathbb{1}\{h(x_i) \neq y_j\} = 1 - \mathbb{1}\{h(x_j) \neq y_j\} = 1 - f_h((x_j, y_j)).$$

This means that no $f_h \in \mathcal{F}_{\mathcal{H}}$ can assign value 1 to both (x_i, y_i) and (x_j, y_j), and hence T' is not shattered by $\mathcal{F}_{\mathcal{H}}$, a contradiction. So, $x_1, \ldots, x_d$ are distinct, and $|T| = d$.

Now we show that T is shattered by $\mathcal{H}$. To see this, consider any labeling $(b_1, \ldots, b_d) \in \{0,1\}^d$. Since T is shattered by $\mathcal{F}_{\mathcal{H}}$, there exists $f_h \in \mathcal{F}_{\mathcal{H}}$ such that $f_h(x_i, y_i) = \mathbb{1}\{b_i \neq y_i\}$ for each $i \in [d]$. But for each $i \in [d]$, we have $f_h(x_i, y_i) = \mathbb{1}\{h(x_i) \neq y_i\}$. Therefore $h(x_i) = b_i$ for each $i \in [d]$. So T is shattered by $\mathcal{H}$. This implies $\mathsf{VCdim}(\mathcal{F}_{\mathcal{H}}) \leq \mathsf{VCdim}(\mathcal{H})$. □

5.2.6 Lower bound for uniform convergence

Exercises 5.6 and 5.7 shows that Proposition 5.7 is tight up to constant factors and an additive $O(1/\sqrt{n})$ term. But what of the bound obtained via Massart's finite class lemma (Lemma 5.3)? This turns out to be almost tight—up to roughly a $\log(n)$ factor—for a worst-case choice of P.

Proposition 5.9. *For any $\mathcal{F} \subseteq \{0,1\}^{\mathcal{Z}}$, there is a probability distribution P over $\mathcal{Z}$ such that, if $\mathbf{z}_{1:n}$ is a size-n iid sample from P, and $(\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_n)$ are n iid Rademacher random variables regarded as independent of $\mathbf{z}_{1:n}$, then*

$$\mathrm{E}\left[\sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\sigma}_i f(\mathbf{z}_i) \right| \right] \geq \frac{1}{4} \min\left\{1, \sqrt{\frac{\mathsf{VCdim}(\mathcal{F})}{n}}\right\}.$$

Proof. Let $d := \min\{\mathsf{VCdim}(\mathcal{F}), n\}$. Let $T := \{t_1, \ldots, t_d\}$ be a set of d points shattered by $\mathcal{F}$, and let P be the uniform distribution on T. Define $\mathbf{S}_j := \{i \in [n] \mid \mathbf{z}_i = t_j\}$ and $\mathbf{N}_j := |\mathbf{S}_j|$ for each $j \in [d]$. Then for each $f \in \mathcal{F}$,

$$\frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\sigma}_i f(\mathbf{z}_i) = \frac{1}{n} \sum_{j=1}^{d} \sum_{i \in \mathbf{S}_j} \boldsymbol{\sigma}_i f(t_j).$$

Since T is shattered by $\mathcal{F}$, there exists $\mathbf{f}_{\mathrm{o}} \in \mathcal{F}$ such that $\mathbf{f}_{\mathrm{o}}(t_j) = \mathbb{1}\{\sum_{i \in \mathbf{S}_j} \boldsymbol{\sigma}_i > 0\}$ for each $j \in [d]$. Therefore

$$\sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\sigma}_i f(\mathbf{z}_i) \right| \geq \left| \frac{1}{n} \sum_{j=1}^{d} \sum_{i \in \mathbf{S}_j} \boldsymbol{\sigma}_i \mathbf{f}_{\mathrm{o}}(t_j) \right| = \frac{1}{n} \sum_{j=1}^{d} \max\left\{ 0, \sum_{i \in \mathbf{S}_j} \boldsymbol{\sigma}_i \right\},$$

and taking expectations,

$$\mathrm{E}\left[\sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\sigma}_i f(\mathbf{z}_i) \right| \right] \geq \frac{1}{n} \sum_{j=1}^{d} \mathrm{E}\left[\max\left\{ 0, \sum_{i \in \mathbf{S}_j} \boldsymbol{\sigma}_i \right\} \right].$$

By the symmetry of the distribution of the Rademacher random variables, we have

$$\mathrm{E}\left[\max\left\{ 0, \sum_{i \in \mathbf{S}_j} \boldsymbol{\sigma}_i \right\} \,\middle|\, \mathbf{z}_{1:n} \right] = \frac{1}{2} \mathrm{E}\left[\left| \sum_{i \in \mathbf{S}_j} \boldsymbol{\sigma}_i \right| \,\middle|\, \mathbf{z}_{1:n} \right]$$

A tedious calculation (or Khintchine's inequality) shows that

$$\mathrm{E}\left[\left| \sum_{i \in \mathbf{S}_j} \boldsymbol{\sigma}_i \right| \,\middle|\, \mathbf{z}_{1:n} \right] = \mathbf{N}_j \binom{\mathbf{N}_j - 1}{\lfloor (\mathbf{N}_j - 1)/2 \rfloor} 2^{1 - \mathbf{N}_j} \geq \sqrt{\frac{\mathbf{N}_j}{2}}.$$

Therefore

$$\mathrm{E}\left[\sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\sigma}_i f(\mathbf{z}_i) \right| \right] \geq \frac{1}{2\sqrt{2}n} \mathrm{E}\left[\sum_{j=1}^{d} \sqrt{\mathbf{N}_j} \right].$$

By Hölder's inequality, we have

$$n = \sum_{j=1}^{d} \mathbf{N}_j = \sum_{j=1}^{d} \mathbf{N}_j^{1/3} \mathbf{N}_j^{2/3} \leq \left(\sum_{j=1}^{d} \mathbf{N}_j^{1/2} \right)^{2/3} \left(\sum_{j=1}^{d} \mathbf{N}_j^{2} \right)^{1/3},$$

and hence

$$\sum_{j=1}^{d} \sqrt{\mathbf{N}_j} \geq \frac{n^{3/2}}{\sqrt{\sum_{j=1}^{d} \mathbf{N}_j^2}}.$$

By Jensen's inequality,

$$\mathrm{E}\left[\frac{n^{3/2}}{\sqrt{\sum_{j=1}^{d} \mathbf{N}_j^2}} \right] \geq \frac{n^{3/2}}{\sqrt{\mathrm{E}\left[\sum_{j=1}^{d} \mathbf{N}_j^2\right]}} = \frac{n^{3/2}}{\sqrt{\sum_{j=1}^{d} \mathrm{E}\left[\mathbf{N}_j^2\right]}}.$$

The random variables $(\mathbf{N}_1, \ldots, \mathbf{N}_d)$ are the occupancies of d bins after throwing n balls independently into uniformly chosen bins, so

$$\mathrm{E}\big[\mathbf{N}_j^2\big] = \mathrm{var}(\mathbf{N}_j) + \mathrm{E}[\mathbf{N}_j]^2 = \frac{n}{d}\left(1 - \frac{1}{d}\right) + \left(\frac{n}{d}\right)^2 = \frac{n}{d} + \frac{n(n-1)}{d^2}.$$

We conclude

$$\mathrm{E}\left[\sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\sigma}_i f(\mathbf{z}_i) \right| \right] \geq \frac{1}{2\sqrt{2}n} \cdot \frac{n^{3/2}}{\sqrt{n + n(n-1)/d}} = \frac{1}{2\sqrt{2}} \sqrt{\frac{d}{d + n - 1}} \geq \frac{1}{4}\sqrt{\frac{d}{n}}. \qquad \square$$

Topic 6: Optimal sample complexity

6.1 Comparing sample complexity upper bounds for PAC learning

Using the VC Theorem (Theorem 5.2, combined with Proposition 5.8), we conclude that for any concept class $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$, ERM with $\mathcal{H} = \mathcal{C}$ has the following guarantee: for any probability distribution P over $\mathcal{X} \times \{0,1\}$, if the training data is a size-n iid sample from P, then for any $\delta \in (0,1)$, with probability at least $1-\delta$, the hypothesis $\mathbf{h}$ returned by ERM satisfies

$$\mathsf{err}(\mathbf{h}; P) \leq \inf_{c \in \mathcal{C}} \mathsf{err}(c; P) + O\left(\sqrt{\frac{d \log(n/d) + \log(1/\delta)}{n}}\right)$$

where $d := \mathsf{VCdim}(\mathcal{C})$. Under $\mathcal{C}$-realizability, the first term on the right-hand side is zero, and the second term is at most $\epsilon \in (0,1)$ when n is at least

$$O\left(\frac{d \log(1/\epsilon)}{\epsilon^2} + \frac{\log(1/\delta)}{\epsilon^2}\right).$$

The quantity on the right-hand side is therefore an upper bound on the optimal (i.e., lowest achievable) sample complexity for (ϵ, δ)-PAC learning of $\mathcal{C}$, since it is the sample complexity achieved by a particular learning algorithm (ERM). In fact, using an improved version of the VC Theorem (analogous to the improvement of Bennett's inequality over Hoeffding's inequality), the sample complexity achieved by ERM can be improved to the following.

Theorem 6.1. *For any $\epsilon, \delta \in (0,1)$, if the number of training data n is at least*

$$O\left(\frac{d \log(1/\epsilon)}{\epsilon} + \frac{\log(1/\delta)}{\epsilon}\right) \tag{6.1}$$

where $d := \mathsf{VCdim}(\mathcal{H})$, then with probability at least $1-\delta$, every hypothesis in $\mathcal{H}$ consistent with an iid sample of size-n from P has error rate on P at most ϵ. Consequently, ERM (Algorithm 4.3 with hypothesis class $\mathcal{H} = \mathcal{C}$) (ϵ, δ)-PAC learns $\mathcal{C}$ with sample complexity as in (6.1) where $d = \mathsf{VCdim}(\mathcal{C})$.

The sample complexity upper bound in Theorem 6.1 is an important improvement over the sample complexity upper bound implied by Theorem 4.2, namely

$$\frac{\ln(|\mathcal{C}|) + \ln(1/\delta)}{\epsilon},$$

which is only applicable to finite concept classes. Similar to Theorem 4.2, the sample complexity depends only polynomially on $1/\epsilon$ and $\log(1/\delta)$ (though the precise dependence on $1/\epsilon$ has gotten a little worse); it replaces the dependence on $\ln(|\mathcal{C}|)$ with a dependence on $\mathsf{VCdim}(\mathcal{C})$.

We make two final remarks on upper bounds. First, the guarantee is said to be *distribution-free* because it applies to any probability distribution P satisfying $\mathcal{C}$-realizability; no assumptions are made on the marginal distribution of P over $\mathcal{X}$. It may be possible to achieve a better sample complexity if one only has to consider specific (classes of) marginal distributions. Second, we note that sample complexity does not take into account computational complexity. In fact, it is not always computationally tractable to implement ERM. Therefore, sample complexity is only part of the story in understanding the difficulty of learning.

Example 6.1. From Example 5.2, the VC dimension of $\mathsf{LTF}_{\mathbb{R}^d}$ is $d+1$. (The same is true for LTF_d over the Boolean d-cube.) Therefore, ERM (or CH) can be used to PAC learn $\mathsf{LTF}_{\mathbb{R}^d}$ with sample complexity

$$O\left(\frac{d\log(1/\epsilon)}{\epsilon} + \frac{\log(1/\delta)}{\epsilon}\right).$$

CH has a computationally-efficient implementation via linear programming: to find a LTF consistent with training data, it suffices to solve a linear feasibility problem, which can be done efficiently via algorithms for linear programming. If the training data is $S = ((x_1, y_1), \ldots, (x_n, y_n)) \in (\mathbb{R}^d \times \{0,1\})^n$, then the system of linear inequalities (in variables $w \in \mathbb{R}^d$ and $\theta \in \mathbb{R}$) to satisfy is

$$\begin{aligned} w^\mathsf{T} x_i - \theta > 0 \quad &\text{for all } i \in [n] \text{ such that } y_i = 1, \\ w^\mathsf{T} x_i - \theta \leq 0 \quad &\text{for all } i \in [n] \text{ such that } y_i = 0. \end{aligned}$$

The system of inequalities is guaranteed to be feasible for training data that is sampled from a $\mathsf{LTF}_{\mathbb{R}^d}$-realizable distribution. $\lozenge$

6.2 Sample complexity lower bound for PAC learning

6.2.1 Statement of the lower bound

A lower bound on the optimal sample complexity of (ϵ, δ)-PAC learning a concept class $\mathcal{C}$ complements the upper bound implied by Theorem 6.1. Since upper bounds are (typically) proved by finding specific algorithms that achieve a certain sample complexity, a lower bound shows how much room for improvement there is (in terms of sample complexity) by designing better algorithms.

Theorem 6.2. *For any concept class $\mathcal{C} \subseteq \{0,1\}^\mathcal{X}$ with $d := \mathsf{VCdim}(\mathcal{C})$, any learning algorithm A, and any $\epsilon \in (0, 1/4)$, there is a probability distribution P over $\mathcal{X} \times \{0,1\}$ that satisfies $\mathcal{C}$-realizability such that, if A receives an iid sample of size-n with*

$$n < \frac{d-1}{16\epsilon},$$

then with probability at least $1/4$, the output $\mathbf{h}$ of A satisfies $\mathsf{err}(\mathbf{h}; P) > \epsilon$.

Theorem 6.2 implies a lower bound of

$$\Omega\left(\frac{d}{\epsilon}\right)$$

on the optimal sample complexity of $(\epsilon, 1/4)$-PAC learning $\mathcal{C}$. A simple modification yields a lower bound for (ϵ, δ)-PAC learning of

$$\Omega\left(\frac{d + \log(1/\delta)}{\epsilon}\right).$$

Up to the $\log(1/\epsilon)$ factor and hidden constants, it matches the upper bound from Theorem 6.1.[1] In particular, it establishes the VC dimension as the key property of a concept class that determines if it is PAC-learnable or not.

[1] In fact, the $\log(1/\epsilon)$ can be removed from the upper bound by using a different learning algorithm (Hanneke, 2016).

Similar to the lower bounds we had given in online learning (e.g., Theorems 1.5, 2.4 and 3.6), the proof of Theorem 6.2 is based on an adversary argument. For each learning algorithm, the adversary chooses an algorithm-specific distribution P that will cause that specific algorithm to fail. The adversary has freedom to choose P in any way as long as it satisfies $\mathcal{C}$-realizability.[2]

6.2.2 Heuristic argument for constant ϵ

Before giving the proof of Theorem 6.2, we give some of ideas in a heuristic proof of a weaker claim: if $n < K \cdot d$ for some constant $K > 0$, then for any learning algorithm, there is a $\mathcal{C}$-realizable distribution P on which the learning algorithm will typically output a hypothesis $\mathbf{h}$ with $\mathsf{err}(\mathbf{h}; P) > 1/(2e^K)$. This will (heuristically) imply that ϵ-learning of $\mathcal{C}$ for $\epsilon = 1/(2e^K)$ has sample complexity more than $K \cdot d$. Here, $1/(2e^K)$ should just be thought of as another positive constant; its particular dependence on K is not important.

The proof must make use of the VC dimension of $\mathcal{C}$, and in particular, the existence of a set of cardinality d that is shattered by $\mathcal{C}$. So let $T := \{t_1, \ldots, t_d\} \subseteq \mathcal{X}$ be such a shattered set. This means that $\mathcal{C}[T] = \{0,1\}^d$, so all 2^d possible ways of labeling the points in T are achievable by concepts from $\mathcal{C}$.

We can think of the iid sample of size-n from P in two parts. The first part is the unlabeled data $\mathbf{x}_1, \ldots, \mathbf{x}_n$, which comprise an iid sample of size-n from the marginal of P over $\mathcal{X}$. Let us assume the marginal distribution of P over $\mathcal{X}$ is the uniform distribution on T. In this case, the iid sample $\mathbf{x}_1, \ldots, \mathbf{x}_n$ can be regarded as "sampling with replacement" from T. The second part is the collection of labels provided the target concept $c^\star$ on the points of $T \cap \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. In particular, the training data does not reveal the values of $c^\star$ on $T \setminus \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$.

What is the expected number of points from T that are seen in the iid sample $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$? This is an elementary question in probability whose answer is

$$d\left(1 - \left(1 - \frac{1}{d}\right)^n\right).$$

For large d (and using the fact $n < Kd$), this quantity is less than $d(1 - 1/e^K)$.

Thus, the labels assigned by $c^\star$ on more than d/e^K points in T are never seen by the learner. Since all ways to label T are possible under $\mathcal{C}$-realizability, the best that the learner can do is to randomly guess these unseen labels, getting only about half of them correct. So $\mathbf{h}$ errs on more than $d/(2e^K)$ points, i.e., $\mathsf{err}(\mathbf{h}; P) > 1/(2e^K)$.

6.2.3 Heuristic argument for all $\epsilon > 0$

The heuristic argument above forced the learner's hypothesis to have a constant error rate. To prove Theorem 6.2, we only need to force the learner's hypothesis to have error rate more than ϵ, which—in the asymptotic $\epsilon \to 0$—will eventually be smaller than any given constant. We can therefore afford to realize the scenario from the previous heuristic argument on just an $O(\epsilon)$ proportion of the distribution. To achieve this, we have the marginal distribution of P spread out $O(\epsilon)$ probability mass uniformly over the first $d-1$ points $t_1, \ldots, t_{d-1}$ in T. The remaining $1 - O(\epsilon)$ mass is left to t_d, whose only role is to make most of the training data "useless".

[2] PAC learning requires a learning algorithm to succeed under all distributions P satisfying $\mathcal{C}$-realizability, so the adversary has freedom to any marginal distribution of P over $\mathcal{X}$.

In a size-n iid sample from this marginal distribution, only every one in about $\Omega(1/\epsilon)$ of the draws will be among $t_1, \ldots, t_{d-1}$; most of the sample is comprised of the "useless" point t_d. Hence, following the previous heuristic argument, if n is less than $\Omega((d-1)/\epsilon)$, then the learner never sees (the labels of) a constant fraction of $t_1, \ldots, t_{d-1}$, and hence can at best guess their labels. Ultimately, the learner's returned hypothesis is likely to have error rate

$$\mathsf{err}(\mathbf{h}; P) > \Omega(1) \times (\#\text{ missed points among } t_1, \ldots, t_{d-1}) \times (\text{probability mass of each } t_i)$$

which is at least ϵ when all of the constants are set just right.

6.2.4 Rigorous proof for all $\epsilon > 0$

We now turn the heuristic arguments into a rigorous proof of Theorem 6.2.

Proof of Theorem 6.2. The proof uses the probabilistic method to show the existence of the $\mathcal{C}$-realizable P that causes the given learning algorithm to fail. Here, failure means that the hypothesis $\mathbf{h}$ returned by the learning algorithm A has $\mathsf{err}(\mathbf{h}; P) > \epsilon$ with probability at least $1/4$.

To use the probabilistic method, we define a random probability distribution $\mathbf{P}$, specifically one that is uniformly distributed over the set $\{P_b \mid b \in \{0,1\}^d\}$. The members of this set of distributions are defined as follows.

- Each of the distributions P_b for $b \in \{0,1\}^d$ has the same marginal distribution Q over $\mathcal{X}$, where
$$Q(t_1) = \cdots = Q(t_{d-1}) := \frac{4\epsilon}{d-1} \qquad \text{and} \qquad Q(t_d) := 1 - 4\epsilon.$$
Here, $T := \{t_1, \ldots, t_d\}$ is a set of d points shattered by $\mathcal{C}$. Call $t_1, \ldots, t_{d-1}$ the "informative points".

- For $(\mathbf{x}, \mathbf{y}) \sim P_b$, we have
$$\mathbf{x} = t_i \implies \mathbf{y} = b_i, \quad \forall i \in [d].$$
That is, under P_b, the labels are given by the concept $c \in \mathcal{C}$ with the behavior
$$(c(t_1), \ldots, c(t_d)) = (b_1, \ldots, b_d).$$

The random distribution $\mathbf{P}$ is generated as follows. Let $\mathbf{b} = (\mathbf{b}_1, \ldots, \mathbf{b}_d)$ be uniformly distributed on $\{0,1\}^d$, so $\mathbf{b}_1, \ldots, \mathbf{b}_d$ are iid Bernoulli$(1/2)$ random variables. Then set $\mathbf{P} := P_{\mathbf{b}}$.

Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be a size-n iid sample from Q, with $n < (d-1)/(16\epsilon)$. The learning algorithm A receives the labels $\mathbf{b}_i$ corresponding to the $t_i \in \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. It may also receive the label $\mathbf{b}_d$ if $t_d \in \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, but the subsequent analysis will completely ignore t_d and $\mathbf{b}_d$. What is important is that A does not receive the labels $\mathbf{b}_i$ for $t_i \notin \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$.

Let
$$\mathbf{R} := \{i \in [d-1] \mid t_i \in \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}\}$$
be (the indices of) the informative points seen in the sample (i.e., the "revealed" points). By Bernoulli's inequality (i.e., $(1-x)^n \geq 1 - nx$ for all $x \in [0,1]$ and $n \in \mathbb{N}$),

$$\mathrm{E}[|\mathbf{R}|] = (d-1)\left(1 - \left(1 - \frac{4\epsilon}{d-1}\right)^n\right) \leq 4\epsilon n.$$

By Markov's inequality (Proposition 4.1) and the assumption $n < (d-1)/(16\epsilon)$,

$$\Pr\left(|\mathbf{R}| > \frac{d-1}{2}\right) \le \Pr(|\mathbf{R}| \ge 8\epsilon n) \le \frac{\mathrm{E}[|\mathbf{R}|]}{8\epsilon n} \le \frac{1}{2}.$$

So, letting $\mathbf{U} := \{i \in [d-1] \mid t_i \notin \{\mathbf{x}_1, \dots, \mathbf{x}_n\}\} = [d-1] \setminus \mathbf{R}$ be the (indices of the) informative points not seen among $\mathbf{x}_1, \dots, \mathbf{x}_n$ (i.e., the "unrevealed" points), we have

$$\Pr\left(|\mathbf{U}| > \frac{d-1}{2}\right) \ge \frac{1}{2}.$$

Define

$$\mathbf{W} := |\{i \in \mathbf{U} \mid \mathbf{h}(t_i) \ne \mathbf{b}_i\}|$$

to be the number of informative points not seen by A and misclassified by A's hypothesis $\mathbf{h}$. The output $\mathbf{h}$ of A is independent of the "unrevealed" labels $\{\mathbf{b}_i \mid i \in \mathbf{U}\}$, so

$$\mathbf{W} \mid (\mathbf{h}, \mathbf{x}_{1:n}) \sim \mathrm{Binomial}(|\mathbf{U}|, 1/2).$$

The median of $\mathrm{Binomial}(|\mathbf{U}|, 1/2)$ is $|\mathbf{U}|/2$, so

$$\Pr\left(\mathbf{W} \ge \frac{|\mathbf{U}|}{2} \,\middle|\, (\mathbf{h}, \mathbf{x}_{1:n})\right) \ge \frac{1}{2}.$$

In the event that $|\mathbf{U}| > (d-1)/2$ and $\mathbf{W} \ge |\mathbf{U}|/2$, it follows that

$$\begin{aligned}
\mathsf{err}(\mathbf{h}; P_{\mathbf{b}}) &\ge \sum_{i \in \mathbf{U}} \frac{4\epsilon}{d-1} \cdot \mathbb{1}\{\mathbf{h}(t_i) \ne \mathbf{b}_i\} \\
&= \frac{4\epsilon}{d-1} \cdot \mathbf{W} \\
&> \frac{4\epsilon}{d-1} \cdot \frac{d-1}{4} = \epsilon.
\end{aligned}$$

Therefore, we have shown that

$$\Pr(\mathsf{err}(\mathbf{h}; P_{\mathbf{b}}) > \epsilon) \ge \Pr\left(|\mathbf{U}| > \frac{d-1}{2} \wedge \mathbf{W} \ge \frac{|\mathbf{U}|}{2}\right) \ge \frac{1}{4}.$$

By the tower property of conditional expectations,

$$\Pr(\mathsf{err}(\mathbf{h}; P_{\mathbf{b}}) > \epsilon) = \mathrm{E}\Big[\Pr(\mathsf{err}(\mathbf{h}; P_{\mathbf{b}}) > \epsilon \mid \mathbf{b})\Big] \ge \frac{1}{4}.$$

Therefore, there exists a choice of $b \in \{0,1\}^d$ such that

$$\Pr(\mathsf{err}(\mathbf{h}; P_b) > \epsilon) \ge \mathrm{E}\Big[\Pr(\mathsf{err}(\mathbf{h}; P_{\mathbf{b}}) > \epsilon \mid \mathbf{b})\Big] \ge \frac{1}{4}. \qquad \square$$

The proof of Theorem 6.2 actually considers a so-called Bayesian setting in which the data generating mechanism is, itself, random. Let $L(A, b) := \Pr(\mathsf{err}(A(\mathbf{S}); P_b) > \epsilon)$ where $\mathbf{S} \sim P_b^n$ and $A(S)$ is the hypothesis returned by learning algorithm A on input $S \in (\mathcal{X} \times \{0,1\})^n$. Then, the proof shows that if $n < (d-1)/(16\epsilon)$,

$$\min_A \mathrm{E}[L(A, \mathbf{b})] \ge \frac{1}{4},$$

where $\mathbf{b} \sim \mathrm{Uniform}(\{0,1\}^d)$, and the minimization is taken over all learning algorithms A.

Topic 7: Boosting

7.1 Weak versus strong PAC learning

The problem of boosting originated in the context of PAC learning. Recall that a learning algorithm A is a PAC learner for concept class $\mathcal{C} \subseteq \{-1,1\}^{\mathcal{X}}$ if it has the following property. For $\epsilon, \delta \in (0,1)$, and any data distribution P that is $\mathcal{C}$-realizable, if A is provided as input an iid sample of size-n from P (where n is the sample complexity of A, which may depend on ϵ, δ, $\mathcal{C}$, etc.), then with probability at least $1-\delta$, it returns a hypothesis $\mathbf{h}\colon \mathcal{X} \to \{-1,1\}$ such that $\mathsf{err}(\mathbf{h}; P) \leq \epsilon$. If A is also a computationally-efficient algorithm (in a suitable sense that can been made precise), then A is an *efficient PAC learner for $\mathcal{C}$*. To distinguish PAC learning from what we are about to describe, we will also call such learning algorithms *strong (PAC) learners for $\mathcal{C}$*.

We say a learning algorithm A is a *weak (PAC) learner for $\mathcal{C}$* if it has all of the requirements of being a strong learner for $\mathcal{C}$, except for the requirement that it works for all ϵ. Instead, it is only required to succeed for a particular $\epsilon = \epsilon_{\mathrm{o}}$, where $\epsilon_{\mathrm{o}} \in (0, 1/2)$. Here, it is important that ϵ_{o} is strictly bounded away from $1/2$, since achieving error rate $1/2$ is trivial for binary label spaces.

A question asked by Kearns and Valiant (1989) is: does the existence of an efficient weak learner imply the existence of an efficient strong learner? That is, is strong learning a concept class as "easy" (in the computational complexity sense of polynomial-time equivalence) as weak learning the same concept class? Schapire (1990) answered the question affirmatively by designing a *boosting algorithm*: an algorithmic reduction from strong learning to weak learning. The boosting algorithm achieves the strong learning guarantee for a concept class $\mathcal{C}$ by using repeatedly invoking a weak learner for $\mathcal{C}$ as a subroutine. After Schapire's original proof was found, several other boosting algorithms have been developed, with the AdaBoost algorithm of Freund and Schapire (1997) probably being the most famous.

The key idea in boosting algorithms is to exploit the requirement that weak learners succeed under any marginal distribution over $\mathcal{X}$. All boosting algorithms work by invoking the weak learner under different distributions, and then use (or combine) the outputs of the weak learners (called "weak hypotheses") in some way.

Exercise 7.1. The original definition of weak learner, in fact, has two exceptions: the algorithm only needs to work for $\epsilon = \epsilon_{\mathrm{o}}$ for some $\epsilon_{\mathrm{o}} \in (0, 1/2)$, and also only for $\delta = \delta_{\mathrm{o}}$ for some $\delta_{\mathrm{o}} \in (0,1)$. Suppose you have such an efficient weak learner for $\mathcal{C}$. Explain how to construct an efficient learning algorithm that $(\epsilon_{\mathrm{o}} + \varepsilon, \delta)$-PAC learns $\mathcal{C}$ for any $\varepsilon \in (0,1)$ and $\delta \in (0,1)$. What is the sample complexity of your algorithm? $\Diamond$

7.2 In-sample boosting

7.2.1 In-sample weak learning

It is a little simpler to think about weak versus strong learning in the context of a fixed training data set—i.e., "in-sample" weak versus strong learning—and then worry only later about how to get "out-of-sample" guarantees. To facilitate this discussion, we use a modified definition of weak learners that is tailored for a fixed training data set.

Let $S := ((x_1, y_1), \ldots, (x_n, y_n)) \in (\mathcal{X} \times \{-1, 1\})^n$ be size-n training data set. (For now, we do not need to assume that it is an iid sample.) An *(in-sample) weak learner* for S is an algorithm that, given any *(re-)weighting* $p = (p_1, \ldots, p_n) \in \Delta^{n-1}$ of the training examples, returns a hypothesis $h \colon \mathcal{X} \to \{-1, 1\}$ with *p-weighted (empirical) accuracy*

$$\mathsf{acc}(h; S, p) := \sum_{i=1}^{n} p_i \, \mathbb{1}\{h(x_i) = y_i\}$$

satisfying

$$\mathsf{acc}(h; S, p) \geq \frac{1 + \gamma}{2} \tag{7.1}$$

for some $\gamma > 0$. The number γ is called the *advantage* of h over random guessing (with respect to the re-weighting p), and $(1 + \gamma)/2$ corresponds to what we previously called $1 - \epsilon_{\mathrm{o}}$. The ability to deal with any re-weighting of the training data set is analogous to the requirement that weak learners of the flavor from Section 7.1 be able to work with any marginal distribution over $\mathcal{X}$.[1] The assumption that there is a weak learner for the dataset with advantage γ is called the *γ-weak learning assumption (γ-WLA)*.

Example 7.1. Recall Exercise 3.4. For an odd number $s \in [d]$, consider the concept class $\mathsf{SUBCOM}_{d,s} \subseteq \{-1, 1\}^{\mathcal{X}}$ over the input space $\mathcal{X} := \{-1, 1\}^d$, defined by

$$\mathsf{SUBCOM}_{d,s} := \{c_T \mid T \subseteq [d], |T| = s\}$$

where

$$c_T(x) := \operatorname{sign}\left(\sum_{j \in T} x_j\right).$$

(It can be checked the $\mathsf{VCdim}(\mathsf{SUBCOM}_{d,s}) = \Theta(s \log(d/s))$ whenever $s \leq d/2$.)

We claim that the following algorithm is an in-sample weak learner for any data set $S := ((x_1, y_1), \ldots, (x_n, y_n)) \in (\mathcal{X} \times \{-1, 1\})^n$ labeled by some $c_T \in \mathsf{SUBCOM}_{d,s}$:

> Given a weighting $p = (p_1, \ldots, p_n) \in \Delta^{n-1}$, determine
>
> $$j^* := \operatorname*{arg\,max}_{j \in [d]} \sum_{i=1}^{n} p_i \, y_i x_{i,j},$$
>
> and return $h(x) := x_{j^*}$.

Observe that for each $i \in [n]$, we have

$$y_i \sum_{j \in T} x_{i,j} \geq 1,$$

and therefore

$$\sum_{i=1}^{n} p_i y_i x_{i,j^*} \geq \max_{j \in T} \sum_{i=1}^{n} p_i \, y_i x_{i,j} \geq \frac{1}{s} \sum_{j \in T} \sum_{i=1}^{n} p_i \, y_i x_{i,j} \geq \frac{1}{s}.$$

[1] Given a weak learner for a concept class $\mathcal{C}$ of the type from Section 7.1, it is possible to design an efficient in-sample weak learner for any training data set whose labels are provided by some concept from $\mathcal{C}$.

This implies

$$\mathsf{acc}(h; S, p) = \sum_{i=1}^{n} p_i \, \mathbb{1}\{h(x_i) = y_i\} = \sum_{i=1}^{n} p_i \, \frac{1 + y_i x_{i,j^*}}{2} = \frac{1 + \sum_{i=1}^{n} p_i \, y_i x_{i,j^*}}{2} \geq \frac{1 + 1/s}{2}.$$

So the γ-WLA holds for S with $\gamma = 1/s$. $\Diamond$

7.2.2 An alternative characterization of the weak learning assumption

Before we describe the boosting algorithm (i.e., the algorithmic reduction from strong learning to weak learning), we first give an alternative characterization of the γ-WLA, which will in turn motivate the boosting algorithm.

Consider all possible hypotheses that the weak learner could output, and call this set $\mathcal{H}$. For simplicity, assume $\mathcal{H}$ is finite, say, $\mathcal{H} = \{h^{(1)}, \ldots, h^{(d)}\}$. Define an $n \times d$ matrix A by

$$A_{i,j} := \mathbb{1}\{h^{(j)}(x_i) = y_i\}.$$

The γ-WLA says that for any $p \in \Delta^{n-1}$, there is some $j \in [d]$ such that

$$p^\mathsf{T} A e_j \geq \frac{1 + \gamma}{2},$$

where e_j is the j-th coordinate basis vector in $\mathbb{R}^d$. (The above expression is equivalent to (7.1) for $h = h^{(j)}$.) In other words,

$$\min_{p \in \Delta^{n-1}} \max_{q \in \{e_1, \ldots, e_d\}} p^\mathsf{T} A q \geq \frac{1 + \gamma}{2}. \tag{7.2}$$

Since linear functions over the probability simplex Δ^{d-1} are extremized at the vertices of the simplex, (7.2) is equivalent to

$$\min_{p \in \Delta^{n-1}} \max_{q \in \Delta^{d-1}} p^\mathsf{T} A q \geq \frac{1 + \gamma}{2}. \tag{7.3}$$

By the Von Neumann's minimax theorem (Theorem 2.3),

$$\min_{p \in \Delta^{n-1}} \max_{q \in \Delta^{d-1}} p^\mathsf{T} A q = \max_{q \in \Delta^{d-1}} \min_{p \in \Delta^{n-1}} p^\mathsf{T} A q,$$

so (7.2) (or (7.3)) is equivalent to

$$\max_{q \in \Delta^{d-1}} \min_{p \in \Delta^{n-1}} p^\mathsf{T} A q \geq \frac{1 + \gamma}{2}. \tag{7.4}$$

Again, since linear functions over Δ^{n-1} are extremized at the vertices, (7.4) is equivalent to

$$\max_{q \in \Delta^{d-1}} \min_{i \in [n]} e_i^\mathsf{T} A q \geq \frac{1 + \gamma}{2}. \tag{7.5}$$

This means that there exists some $q = (q_1, \ldots, q_d) \in \Delta^{d-1}$ such that, for all $i \in [n]$,

$$\begin{aligned} \sum_{j=1}^{d} q_j \, \mathbb{1}\{h^{(j)}(x_i) = y_i\} &= \sum_{j=1}^{d} q_j \frac{1 + y_i h^{(j)}(x_i)}{2} \\ &= \frac{1 + y_i \sum_{j=1}^{d} q_j h^{(j)}(x_i)}{2} \\ &\geq \frac{1 + \gamma}{2}. \end{aligned}$$

This is equivalent to the existence of $q \in \Delta^{d-1}$ such that

$$y_i \sum_{j=1}^{d} q_j h^{(j)}(x_i) \geq \gamma \quad \forall i \in [n]. \tag{7.6}$$

So, in the "feature space" $\{-1,1\}^d$ where the feature vector for $x \in \mathcal{X}$ is $\phi(x) := (h^{(1)}(x), \ldots, h^{(d)}(x))$, the γ-WLA implies *linearly separability* of the training data set, i.e., the existence of a linear threshold function that classifies all of the training examples correctly.

Since γ-WLA implies linear separability, one possible approach to strong learning would be to simply apply the feature map ϕ to all of the training examples, i.e.,

$$S' := ((\phi(x_1), y_1), \ldots, (\phi(x_n), y_n)) \in (\{-1,1\}^d \times \{-1,1\})^n,$$

and then apply a strong learner for LTF_d to this transformed training data set S', such as that described in Example 6.1. This gives a strong PAC learner with sample complexity and time complexity polynomial in $1/\epsilon$, $1/\delta$, and d.

There are at least two potential issues with the above approach to strong learning.

1. The first issue is computational. It is possible that $|\mathcal{H}| = d$ is very large, and yet the weak learner is able to efficiently find a suitable hypothesis in $\mathcal{H}$ in $o(d)$ time. Yet the approach to strong learning we described seems to require $\Omega(d)$ time; for instance, it applies the feature map $\phi\colon \mathcal{X} \to \{-1,1\}^d$ to the training data.

2. The second issue statistical. The weak learner uses a hypothesis class $\mathcal{H}$ with $\mathsf{VCdim}(\mathcal{H}) \leq \log_2(d)$. However, the proposed strong learner uses a hypothesis class LTF_d, which has $\mathsf{VCdim}(\mathsf{LTF}_d) = d + 1$.

Both issues may be even more pronounced if $|\mathcal{H}| = \infty$ but $\mathsf{VCdim}(\mathcal{H}) < \infty$. Fortunately, both issues can be overcome by taking advantage of (7.6) and how we arrived at this alternative characterization.

7.2.3 Approximately solving the game using Hedge

Recall that the Von Neumann minimax theorem (Theorem 2.3) has an algorithmic proof based on executing Hedge (Algorithm 2.2) in a particular instance of the Online Allocation problem. From the execution of Hedge in this context (Proposition 2.4), there is a way to obtain (in the terminology of Theorem 2.3) $p \in \Delta^{n-1}$ and $q \in \Delta^{d-1}$ that approximately achieve the optimal value of the game (which is at least $(1+\gamma)/2$ in our case). This suggests a simple algorithm based on Hedge (Freund and Schapire, 1997), given as Algorithm 7.1 below. The difference between Algorithm 7.1 and the algorithm described in Proposition 2.4 is that in Algorithm 7.1, WL does not necessarily provide a "best response" in each round; it may only give a "good enough response" for the purposes of satisfying the γ-WLA. Nevertheless, we can still follow the analysis of Hedge to analyze Algorithm 7.1.

Algorithm 7.1: EtaBoost for in-sample boosting.

Hyperparameters: $\eta > 0$ and $T \in \mathbb{N}$.
Input: data set $S := ((x_1, y_1), \ldots, (x_n, y_n)) \in (\mathcal{X} \times \{-1, 1\})^n$; in-sample weak learner WL for S.

- Initially, set $p_1 := (1/n, \ldots, 1/n) \in \Delta^{n-1}$.
- For $t = 1, 2, \ldots, T$:
  - Invoke WL for data set S with weighting $p_t := (p_{t,1}, \ldots, p_{t,n}) \in \Delta^{n-1}$ to obtain hypothesis $h_t \colon \mathcal{X} \to \{-1, 1\}$.
  - Define loss vector $\ell_t := (\ell_{t,1}, \ldots, \ell_{t,n}) \in \{0, 1\}^n$ by
  $$\ell_{t,i} := \mathbb{1}\{h_t(x_i) = y_i\}.$$
  - Compute new weighting $p_{t+1} := (p_{t+1,1}, \ldots, p_{t+1,n}) \in \Delta^{n-1}$:
  $$p_{t+1,i} := \frac{p_{t,i} \exp(-\eta \ell_{t,i})}{Z_{t+1}} \quad \text{for all } i \in [n],$$
  where
  $$Z_{t+1} := \sum_{i=1}^{n} p_{t,i} \exp(-\eta \ell_{t,i}).$$
- Return: final hypothesis $\hat{h}(x) := \operatorname{sign}(\hat{g}(x))$ where $\hat{g}(x) := \frac{1}{T} \sum_{t=1}^{T} h_t(x)$.

The analysis of Algorithm 7.1 is much like the proof of Proposition 2.4; we simply have to interpret the guarantees for Hedge (Algorithm 2.2) in this setting.

Theorem 7.1. *Suppose the γ-Weak Learning Assumption for $S := ((x_1, y_1), \ldots, (x_n, y_n)) \in (\mathcal{X} \times \{-1, 1\})^n$ holds, and $\hat{h} \colon \mathcal{X} \to \{-1, 1\}$ is the output of Algorithm 7.1 when executed with S and an in-sample weak learner for S. If the hyperparameters satisfy $\eta := 2\gamma$ and $T > 2\ln(n)/\gamma^2$, then $\hat{h}(x_i) = y_i$ for all $i \in [n]$.*

Proof. The weightings p_t are computed as in Hedge, with loss vectors ℓ_t determined by the hypotheses h_t returned by the weak learner. Theorem 2.2 implies the following: for any $u \in \Delta^{n-1}$,

$$\frac{1}{T} \sum_{t=1}^{T} \langle p_t, \ell_t \rangle \leq \frac{1}{T} \sum_{t=1}^{T} \langle u, \ell_t \rangle + \frac{\ln(n)}{\eta T} + \frac{\eta}{8}. \tag{7.7}$$

Let us interpret the inequality in (7.7). First, the left-hand side satisfies

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^{T} \langle p_t, \ell_t \rangle &= \frac{1}{T} \sum_{t=1}^{T} \left(\sum_{i=1}^{n} p_{t,i} \, \mathbb{1}\{h_t(x_i) = y_i\} \right) \\ &= \frac{1}{T} \sum_{t=1}^{T} \mathsf{acc}(h_t; S, p_t) \\ &\geq \frac{1 + \gamma}{2} \end{aligned}$$

where the inequality follows from the γ-WLA. Now suppose $u = e_i$ for some $i \in [n]$. Then

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\langle e_i, \ell_t\rangle &= \frac{1}{T}\sum_{t=1}^{T}\mathbb{1}\{h_t(x_i) = y_i\} \\ &= \frac{1}{T}\sum_{t=1}^{T}\frac{1 + y_i h_t(x_i)}{2} \\ &= \frac{1 + y_i \hat{g}(x_i)}{2}.\end{aligned}$$

Therefore, via (7.7), we have

$$\frac{1 + y_i \hat{g}(x_i)}{2} \geq \frac{1+\gamma}{2} - \frac{\ln(n)}{\eta T} - \frac{\eta}{8} \quad \text{for all } i \in [n].$$

This is equivalent to

$$y_i \hat{g}(x_i) \geq \gamma - \frac{2\ln(n)}{\eta T} - \frac{\eta}{4} \quad \text{for all } i \in [n].$$

Thus, the choice of the hyperparameters η and T ensure that $y_i \hat{g}(x_i) > 0$ for all $i \in [n]$, which means that $\hat{h}(x_i) = y_i$ for all $i \in [n]$. □

A slight variation of the proof of Theorem 7.1 shows how the error rate on S decreases with T.

Theorem 7.2. *In the same setting as Theorem 7.1, if $\eta := 2\gamma$ and $M := \{i \in [n] \mid y_i \hat{g}(x_i) \leq 0\}$ (where $\hat{g}$ is from Algorithm 7.1), then*

$$|M| \leq n \exp\left(-\frac{\gamma^2 T}{2}\right).$$

Proof. By Theorem 2.2, we have

$$\frac{1}{T}\sum_{t=1}^{T}\langle p_t, \ell_t\rangle \leq \frac{1}{T}\sum_{t=1}^{T}\langle u, \ell_t\rangle + \frac{\mathsf{RE}(u, p_1)}{\eta T} + \frac{\eta}{8}. \tag{7.8}$$

for all $u \in \Delta^{n-1}$. Let u be the uniform distribution over M. Then

$$\mathsf{RE}(u, p_1) = \sum_{i=1}^{n} u_i \ln\left(\frac{u_i}{1/n}\right) = -\ln(|M|/n).$$

Then, as in the proof of Theorem 7.1 (using the γ-WLA),

$$\frac{1}{T}\sum_{t=1}^{T}\langle p_t, \ell_t\rangle \geq \frac{1+\gamma}{2},$$

and

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\langle u, \ell_t\rangle &= \sum_{i=1}^{n} u_i \cdot \frac{1}{T}\sum_{t=1}^{T}\langle e_i, \ell_t\rangle \\ &= \sum_{i=1}^{n} u_i \cdot \frac{1 + y_i\hat{g}(x_i)}{2} \\ &= \frac{1}{2} + \frac{1}{2}\sum_{i=1}^{n} u_i \cdot y_i\hat{g}(x_i) \\ &\leq \frac{1}{2}.\end{aligned}$$

So, from (7.8), we obtain

$$\frac{1+\gamma}{2} \leq \frac{1}{2} - \frac{\ln(|M|/n)}{\eta T} + \frac{\eta}{8}.$$

Plugging in the choice $\eta = 2\gamma$ and re-arranging gives the claimed upper bound on $|M|$. □

Theorems 7.1 and 7.2 show that the γ-WLA for a data set S not only implies the linear separability of S, but also the existence of an efficient algorithm for finding a LTF that correctly classifies all (or approximately all) of the examples in S. This addresses the computational issue raised at the end of Section 7.2.2.

The AdaBoost algorithm of Freund and Schapire (1997) is an improved version of Algorithm 7.1 that removes the hyperparameter η. Instead, it performs an update in each round t similar to that of Algorithm 7.1, except that the update is based on the actual advantage γ_t of h_t, rather than the lower bound γ guaranteed by the γ-WLA. Another difference is that the final hypothesis given by AdaBoost is a weighted majority vote, where the weight of h_t is based on the actual advantage of h_t. These are important modifications that make AdaBoost a very practical and successful machine learning algorithm.

7.3 Out-of-sample performance of boosting

7.3.1 VC dimension of majority votes over hypotheses

Having shown how the γ-WLA leads to an efficient algorithm for finding a hypothesis that classifies all (or approximately all) of the examples "in-sample", it remains to understand how that hypothesis performs "out-of-sample". That is, if Algorithm 7.1 is executed on an iid sample $\mathbf{S} = ((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)) \sim P^n$ for some $\mathcal{C}$-realizable distribution P, what is the error rate of the final hypothesis on P?

A first step towards answering this question is to consider the VC dimension of the hypothesis class being used by the boosting algorithm. If the weak learner uses hypothesis class $\mathcal{H} \subseteq \{-1, 1\}^{\mathcal{X}}$, then the boosting algorithm uses the class of *majority votes over T hypotheses from $\mathcal{H}$*:

$$\mathsf{MAJ}_{\mathcal{H},T} := \left\{ x \mapsto \operatorname{sign}\left(\frac{1}{T}\sum_{t=1}^{T} h_t(x)\right) \;\middle|\; (h_1, \ldots, h_T) \in \mathcal{H}^T \right\}.$$

(When $\mathcal{H}$ is finite, the class $\mathsf{MAJ}_{\mathcal{H},T}$ can be viewed as a subclass of $\mathsf{LTF}_{|\mathcal{H}|}$ when inputs are first transformed with the feature map $x \mapsto (h(x))_{h\in\mathcal{H}}$.)

Proposition 7.1. $\mathsf{VCdim}(\mathsf{MAJ}_{\mathcal{H},T}) \leq 2d_{\mathsf{WL}}T(\log_2(eT) + 0.1)$ *where* $d_{\mathsf{WL}} := \mathsf{VCdim}(\mathcal{H})$.

Proof. Suppose $S := \{x_1, \ldots, x_m\}$ is a set of cardinality m shattered by $\mathsf{MAJ}_{\mathcal{H},T}$. Then it suffices to show that $m \leq 2d_{\mathsf{WL}}T(\log_2(eT) + 0.1)$. Every $v \in \mathsf{MAJ}_{\mathcal{H},T}[S]$ is determined by a choice of T behaviors from $\mathcal{H}[S]$, and hence

$$|\mathsf{MAJ}_{\mathcal{H},T}[S]| \leq |\mathcal{H}[S]|^T.$$

By Sauer's lemma (Lemma 5.2), we have

$$|\mathcal{H}[S]|^T \leq \binom{m}{\leq d}^T \leq \left(\frac{em}{d_{\mathsf{WL}}}\right)^{d_{\mathsf{WL}}T}.$$

On the other hand, since S is shattered by $\mathsf{MAJ}_{\mathcal{H},T}$,

$$|\mathsf{MAJ}_{\mathcal{H},T}[S]| = 2^m.$$

So we have

$$m \leq d_{\mathsf{WL}}T\log_2\left(\frac{em}{d_{\mathsf{WL}}}\right).$$

Letting $u := m/(d_{\mathsf{WL}}T)$, the inequality above becomes

$$u \leq \log_2(eT) + \log_2(u).$$

It can be checked that $\log_2(u) \leq u/2 + 0.1$ for all $u > 0$, so

$$u \leq 2(\log_2(eT) + 0.1).$$

Hence $m \leq 2d_{\mathsf{WL}}T(\log_2(eT) + 0.1)$. □

7.3.2 Generalization guarantee for boosting

We can now give a generalization guarantee for boosting under the γ-WLA. The main idea is to leverage uniform convergence of training error rates for $\mathsf{MAJ}_{\mathcal{H},T}$ (which makes use of Proposition 7.1), along with the training error rate guarantee of boosting the γ-WLA from Theorem 7.2.

Theorem 7.3. *Let P be a probability distribution over $\mathcal{X} \times \{-1, 1\}$ such that $\mathbf{S} \sim P^n$ always satisfies the γ-WLA, and let $\delta \in (0, 1)$. If $\hat{\mathbf{h}}$ is the output of Algorithm 7.1 (with hyperparameters $\eta := 2\gamma$ and T) on input $\mathbf{S}$ and an in-sample weak learner for $\mathbf{S}$ with advantage γ that always outputs hypotheses from $\mathcal{H}$, then with probability at least $1 - \delta$,*

$$\mathsf{err}(\hat{\mathbf{h}}; P) \leq \exp\left(-\frac{\gamma^2 T}{2}\right) + O\left(\sqrt{\frac{V_{d_{\mathsf{WL}},T}\log(n/V_{d_{\mathsf{WL}},T}) + \log(1/\delta)}{n}}\right),$$

where $d_{\mathsf{WL}} = \mathsf{VCdim}(\mathcal{H})$ and $V_{d_{\mathsf{WL}},T} = O(d_{\mathsf{WL}}T\log(T))$.

Using $T := 2\ln(2/\epsilon)/\gamma^2$, it follows that the error rate bound is at most ϵ provided that the number of training data n is at least

$$O\left(\frac{d_{\mathsf{WL}}}{\epsilon^2\gamma^2}\log^2\left(\frac{1}{\epsilon}\right)\log\left(\frac{1}{\gamma^2}\log\frac{1}{\epsilon}\right) + \frac{1}{\epsilon^2}\log\left(\frac{1}{\delta}\right)\right).$$

Each appearance of ϵ^2 can be be replaced by ϵ by using the improved version of the VC theorem alluded to in the context of Theorem 6.1. Also, some logarithmic factors can be removed by directly using the growth function for $\mathsf{MAJ}_{\mathcal{H},T}$ instead of its VC dimension. But even without these improvements, the quantity is at most a polynomial in d_{WL}, $1/\gamma$, $1/\epsilon$, and $1/\delta$.

7.4 Margins theory

7.4.1 An empirical puzzle

In Theorem 7.3, the error rate bound has two terms: one that decreases with T, and another that increases with T. This suggests a trade-off that should be optimized by carefully choosing T (as done in the previous section). However, in practice, it is occasionally observed with boosting algorithms (like AdaBoost, for which T is the only hyperparameter) that the population error rate only decreases with T, even beyond the point at which the training error rate is zero. That is, the learning algorithm seemed to be "resistant to over-fitting". To provide a theoretical explanation of this phenomenon, Schapire et al. (1998) gave a different analysis of boosting based on what they call the margins theory.

Recall that the output of Algorithm 7.1 is $\hat{h} = \operatorname{sign} \circ \hat{g}$ where $\hat{g}(x) = \frac{1}{T}\sum_{t=1}^{T} h_t(x)$. In particular, $\hat{g}$ is a *convex combination* of hypotheses from $\mathcal{H}$; the set of such convex combintations is called the *convex hull* of $\mathcal{H}$, written

$$\operatorname{conv}(\mathcal{H}) := \left\{ \sum_{i=1}^{d} \alpha_i h_i \;\middle|\; d \in \mathbb{N},\ \alpha = (\alpha_1, \ldots, \alpha_d) \in \Delta^{d-1},\ (h_1, \ldots, h_d) \in \mathcal{H}^d \right\}.$$

Since $h(x) \in \{-1, 1\}$ for each $h \in \mathcal{H}$, it follows that $\hat{g}(x) \in [-1, 1]$. Although only the sign of $\hat{g}(x)$ is used for prediction, the magnitude of $\hat{g}(x)$ can be regarded as a measure of confidence in the prediction. Let us call $y\hat{g}(x)$ the *margin* of $\hat{g}$ on $(x, y) \in \mathcal{X} \times \{-1, 1\}$.

Schapire et al. (1998) found that, in several experiments, running AdaBoost with T larger than that sufficient to achieve zero training error rate resulted in $\hat{g}$ with larger margins on larger fractions of the training data set. For example, the following was observed in one such experiment using AdaBoost with a decision tree learner as the Weak Learning algorithm:

	T		
	5	100	1000
Training error rate	0.0%	0.0%	0.0%
Test error rate	8.4%	3.3%	3.1%
% margins below 0.5	7.7%	0.0%	0.0%
Minimum margin	0.14	0.52	0.55

Moreover, in these experiments, it appeared that larger margins on the training data set correlated with better out-of-sample performance. Schapire et al. (1998) sought to explain this phenomenon with their margins theory.

7.4.2 Margin errors

For a data set $S := ((x_1, y_1), \ldots, (x_n, y_n)) \in (\mathcal{X} \times \{-1, 1\})^n$, $f \colon \mathcal{X} \to [-1, 1]$, and $\theta > 0$, define the *θ-margin error rate* by

$$\mathsf{err}_\theta(f; S) := \frac{1}{n}\sum_{i=1}^{n} \ell_\theta(y f(x)),$$

where $\ell_\theta \colon \mathbb{R} \to \mathbb{R}_+$ is the *θ-margin error (or loss) function*, defined by

$$\ell_\theta(s) := \mathbb{1}\{s \le \theta\}.$$

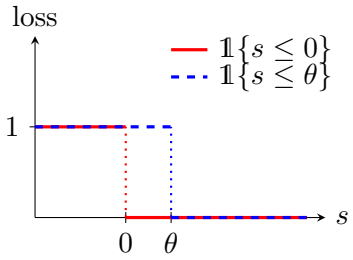


Figure 7.1: Comparison of zero-one loss $\ell_{0/1}(s) = \mathbb{1}\{s \le 0\}$ and θ-margin loss $\ell_\theta(s) = \mathbb{1}\{s \le \theta\}$.

See Figure 7.1 for a comparison of the θ-margin loss function to the zero-one loss function. Observe that

$$\mathbb{1}\{\text{sign}(f(x)) \neq y\} \le \ell_\theta(yf(x)),$$

so

$$\mathsf{err}(\text{sign} \circ f; S) \le \mathsf{err}_\theta(f; S).$$

Similarly, for a probability distribution P on $\mathcal{X} \times \{-1, 1\}$, define

$$\mathsf{err}_\theta(f; P) := \mathrm{E}[\ell_\theta(\mathbf{y}f(\mathbf{x}))]$$

for $(\mathbf{x}, \mathbf{y}) \sim P$, so we also have

$$\mathsf{err}(\text{sign} \circ f; P) \le \mathsf{err}_\theta(f; P).$$

One can show that Algorithm 7.1, with appropriate choice of hyperparameter η, finds $\hat{g}$ with $\mathsf{err}_\theta(\hat{g}; S) = 0$ for $\theta = \Omega(\gamma)$ under the γ-WLA. (Schapire et al. (1998) do the same for AdaBoost.) The main question is how this can be used to bound $\mathsf{err}(\text{sign} \circ \hat{g}; P)$.

Exercise 7.2. How should the hyperparameters η and T be set so that Algorithm 7.1 achieves $\mathsf{err}_{\gamma/3}(\hat{g}; S) = 0$ under the γ-WLA? ◊

7.4.3 Uniform covers

For any $K \in \mathbb{N}$, define

$$\text{avg}_K(\mathcal{H}) := \left\{ \frac{1}{K} \sum_{t=1}^{K} h_t \;\middle|\; (h_1, \ldots, h_K) \in \mathcal{H}^K \right\}$$

to be the set of all *uniform averages* of K hypotheses from $\mathcal{H}$. We can now state the centerpiece of the margins theory.

Theorem 7.4 (Schapire, Freund, Bartlett, and Lee, 1998)**.** *Let $\mathcal{H} \subseteq \{-1, 1\}^{\mathcal{X}}$ be a hypothesis class over $\mathcal{X}$, let P be a probability distribution over $\mathcal{X} \times \{-1, 1\}$, and let $S \in (\mathcal{X} \times \{-1, 1\})^n$. For every $K \in \mathbb{N}$, $\theta > 0$, and $f \in \text{conv}(\mathcal{H})$,*

$$\mathsf{err}(\text{sign} \circ f; P) \le \mathsf{err}_\theta(f; S) + \max_{g \in \text{avg}_K(\mathcal{H})} \left(\mathsf{err}_{\theta/2}(g; P) - \mathsf{err}_{\theta/2}(g; S) \right) + 4 \exp\left(-\frac{\theta^2 K}{8} \right).$$

The last term on the right-hand side in Theorem 7.4 is $O(1/n)$ for $K = \Theta(\log(n)/\theta^2)$. (There is nothing special about $O(1/n)$; see Exercise 7.3.) When S is replaced by $\mathbf{S} \sim P^n$, the max term on the right-hand side can be bounded (in expectation, or with high probability) by proving uniform convergence of $\mathsf{err}_{\theta/2}(g; \mathbf{S})$ to $\mathsf{err}_{\theta/2}(g; P)$ for all $g \in \mathrm{avg}_K(\mathcal{H})$.

Corollary 7.1. *Let $\mathcal{H} \subseteq \{-1,1\}^{\mathcal{X}}$ be a finite hypothesis class over $\mathcal{X}$, let P be a probability distribution over $\mathcal{X} \times \{-1,1\}$, and let $\mathbf{S} \sim P^n$. For any $\theta \in (0,1]$ and $\delta \in (0,1)$, with probability at least $1-\delta$, every $f \in \mathrm{conv}(\mathcal{H})$ satisfies*

$$\mathsf{err}(\mathrm{sign} \circ f; P) \leq \mathsf{err}_\theta(f; S) + \sqrt{\frac{4\ln(4n)\ln(|\mathcal{H}|)}{\theta^2 n} + \frac{\ln(1/\delta)}{2n}} + \frac{1}{n}.$$

Exercise 7.3. Improve Corollary 7.1 by choosing K to balance the terms in Theorem 7.4 in a better way. ♢

The key to proving Theorem 7.4 is to show that every $f \in \mathrm{conv}(\mathcal{H})$ can be approximated by some $g_f \in \mathrm{avg}_K(\mathcal{H})$ in the following sense.

- The functions in $\mathrm{conv}(\mathcal{H})$ and $\mathrm{avg}_K(\mathcal{H})$ are $[-1,1]$-valued, so the sort of approximation we care about is in terms of the output values being close—say, within some small value $\varepsilon > 0$.
- Since Theorem 7.4 compares 0-margin errors (on the left-hand side) to θ-margin errors (on the right-hand side), it will be enough to ensure that every $f \in \mathrm{conv}(\mathcal{H})$ is approximated in output value by some $g_f \in \mathrm{avg}_K(\mathcal{H})$ up to error $\varepsilon := \theta/2$.
- We will allow the $\theta/2$-closeness in output values between f and g_f to fail for some portion of the data distribution (or training data set).

This is done in the following proposition.

Proposition 7.2. *For any $\mathcal{H} \subseteq \{-1,1\}^{\mathcal{X}}$, any $\mathcal{X}$-valued random variable $\mathbf{z}$, any $K \in \mathbb{N}$, any $\varepsilon > 0$, and any $f \in \mathrm{conv}(\mathcal{H})$, there exists $g_f \in \mathrm{avg}_K(\mathcal{H})$ such that*

$$\Pr(|f(\mathbf{z}) - g_f(\mathbf{z})| > \varepsilon) \leq 2\exp\left(-\frac{\varepsilon^2 K}{2}\right).$$

The proof of Proposition 7.2 is based on a technique attributed to B. Maurey (Pisier, 1981) and uses the probabilistic method. We define a random function $\mathbf{G}$ from $\mathrm{avg}_K(\mathcal{H})$ that is meant to approximate f. We first show that, for any fixed $x \in \mathcal{X}$, $\mathbf{G}(x)$ is close to $f(x)$ for most realizations of the random function $\mathbf{G}$. Then, we replace x with $\mathbf{z}$ and take expectation, so $\mathbf{G}(\mathbf{z})$ is close to $f(\mathbf{z})$ in an average-case sense. Switching the order of expectations, we conclude that there is some $g_f \in \mathrm{avg}_K(\mathcal{H})$ such that $g_f(\mathbf{z})$ is also close to $f(\mathbf{z})$ in an average-case sense.

Proof of Proposition 7.2. Since $f \in \mathrm{conv}(\mathcal{H})$, we can write $f = \sum_{i=1}^d \alpha_i h_i$ for some $d \in \mathbb{N}$, $\alpha = (\alpha_1, \ldots, \alpha_d) \in \Delta^{d-1}$, and $(h_1, \ldots, h_d) \in \mathcal{H}^d$. Let $\mathbf{J}_1, \ldots, \mathbf{J}_K$ be iid random variables (independent of $\mathbf{z}$) with

$$\Pr(\mathbf{J}_1 = j) = \alpha_j \quad \forall j \in [d],$$

and define the random function $\mathbf{G}$ taking values in $\mathrm{avg}_K(\mathcal{H})$ by

$$\mathbf{G} := \frac{1}{K}\sum_{t=1}^K h_{\mathbf{J}_t}.$$

Fix $x \in \mathcal{X}$. Then $K(\mathbf{G}(x) - f(x)) = \sum_{t=1}^{K}(h_{\mathbf{J}_t}(x) - f(x))$ is the sum of K iid random variables, each taking values in $\{-1 - f(x), 1 - f(x)\}$ and each with mean zero. By Hoeffding's inequality (Corollary 4.4),

$$\Pr(\mathbf{G}(x) - f(x) > \varepsilon) \le \exp\left(-\frac{\varepsilon^2 K}{2}\right).$$

Similarly, the same probability bound applies to the event $f(x) - \mathbf{G}(x) > \varepsilon$. By a union bound,

$$\Pr(|f(x) - \mathbf{G}(x)| > \varepsilon) \le 2\exp\left(-\frac{\varepsilon^2 K}{2}\right).$$

Since the above inequality holds for every $x \in \mathcal{X}$, we can replace x with the random variable $\mathbf{z}$, and the same inequality holds in expectation:

$$\Pr\Big(|f(\mathbf{z}) - \mathbf{G}(\mathbf{z})| > \varepsilon\Big) = \mathrm{E}\Big[\mathrm{E}\big[\mathbb{1}\{|f(\mathbf{z}) - \mathbf{G}(\mathbf{z})| > \varepsilon\} \,\big|\, \mathbf{z}\big]\Big] \le 2\exp\left(-\frac{\varepsilon^2 K}{2}\right).$$

The order of the expectations can be reversed:

$$\Pr\Big(|f(\mathbf{z}) - \mathbf{G}(\mathbf{z})| > \varepsilon\Big) = \mathrm{E}\Big[\mathrm{E}\big[\mathbb{1}\{|f(\mathbf{z}) - \mathbf{G}(\mathbf{z})| > \varepsilon\} \,\big|\, \mathbf{G}\big]\Big].$$

So there exists $g_f \in \mathrm{avg}_K(\mathcal{H})$ such that

$$\Pr\Big(|f(\mathbf{z}) - g_f(\mathbf{z})| > \varepsilon\Big) = \mathrm{E}\big[\mathbb{1}\{|f(\mathbf{z}) - \mathbf{G}(\mathbf{z})| > \varepsilon\} \,\big|\, \mathbf{G} = g_f\big] \le 2\exp\left(-\frac{\varepsilon^2 K}{2}\right). \qquad \square$$

Proof of Theorem 7.4. Let $(\mathbf{x}, \mathbf{y}) \sim P$. Let $S := ((x_1, y_1), \ldots, (x_n, y_n))$. Define the $\mathcal{X}$-valued random variable $\mathbf{z}$ as follows.

- Toss a fair coin.
- If the coin comes up heads, set $\mathbf{z} := \mathbf{x}$.
- If the coin comes up tails, randomly pick $\mathbf{I} \sim \mathrm{Uniform}([n])$, and set $\mathbf{z} := x_{\mathbf{I}}$.

Applying Proposition 7.2 to $\mathbf{z}$ (with $\varepsilon = \theta/2$) gives the following: for every $f \in \mathrm{conv}(\mathcal{H})$, there exists $g_f \in \mathrm{avg}_K(\mathcal{H})$ such that

$$\frac{1}{2}\left(\Pr\Big(|f(\mathbf{x}) - g_f(\mathbf{x})| > \theta/2\Big) + \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}\{|f(x_i) - g_f(x_i)| > \theta/2\}\right) \le 2\exp\left(-\frac{\theta^2 K}{8}\right). \tag{7.9}$$

We next reason about how classification errors relate to $\theta/2$-margin errors, and how $\theta/2$-margin errors relate to θ-margin errors. If $yf(x) \le 0$ and $|f(x) - g_f(x)| \not> \theta/2$, then $yg_f(x) \le \theta/2$. This can be expressed using indicator functions as follows:

$$\mathbb{1}\{yf(x) \le 0\} \le \mathbb{1}\{yg_f(x) \le \theta/2\} + \mathbb{1}\{|f(x) - g_f(x)| > \theta/2\}.$$

Similarly, if $yg_f(x) \le \theta/2$ and $|f(x) - g_f(x)| \not> \theta/2$, then $yf(x) \le \theta$:

$$\mathbb{1}\{yg_f(x) \le \theta/2\} \le \mathbb{1}\{yf(x) \le \theta\} + \mathbb{1}\{|f(x) - g_f(x)| > \theta/2\}.$$

Therefore

$$\Pr(\mathbf{y}f(\mathbf{x}) \le 0) \le \Pr(\mathbf{y}g_f(\mathbf{x}) \le \theta/2) + \Pr(|f(\mathbf{x}) - g_f(\mathbf{x})| > \theta/2) \tag{7.10}$$

and

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{y_i g_f(x_i) \le \theta/2\} \le \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{y_i f(x_i) \le \theta\} + \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{|f(x_i) - g_f(x_i)| > \theta/2\}. \tag{7.11}$$

Putting (7.9)–(7.11) together gives

$$\begin{aligned}
\Pr(\mathbf{y}f(\mathbf{x}) \le 0) &\le \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{y_i f(x_i) \le \theta\} \\
&\qquad + \Pr(\mathbf{y}g_f(\mathbf{x}) \le \theta/2) - \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{y_i g_f(x_i) \le \theta/2\} \\
&\qquad + \Pr(|f(\mathbf{x}) - g_f(\mathbf{x})| > \theta/2) + \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{|f(x_i) - g_f(x_i)| > \theta/2\} \\
&\le \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{y_i f(x_i) \le \theta\} \\
&\qquad + \max_{g \in \mathrm{avg}_K(\mathcal{H})} \left(\Pr(\mathbf{y}g(\mathbf{x}) \le \theta/2) - \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{y_i g(x_i) \le \theta/2\} \right) \\
&\qquad + 4\exp\left(-\frac{\theta^2 K}{8}\right).
\end{aligned}$$

□

Exercise 7.4. Let $\mathcal{H} \subseteq \{-1,1\}^{\mathcal{X}}$ be a hypothesis class over $\mathcal{X}$ with VC dimension $d < \infty$. Suppose $\mathbf{S} \sim P^n$ for a probability distribution P over $\mathcal{X} \times \{-1,1\}$. Prove a bound on

$$\mathrm{E}\left[\sup_{g \in \mathrm{avg}_K(\mathcal{H})} \mathsf{err}_{\theta/2}(g; P) - \mathsf{err}_{\theta/2}(g; \mathbf{S}) \right]$$

in terms of n, d, and $K \in \mathbb{N}$. The bound should hold for every choice of $\theta \ge 0$. ◊

7.4.4 Boosting as an optimization algorithm

Many training algorithms for machine learning can be viewed as an algorithm for minimizing a particular objective function, and the boosting algorithm from Algorithm 7.1 is no exception. Recall that when the weak learner uses a finite hypothesis class $\mathcal{H} := \{h^{(1)}, \dots, h^{(d)}\}$, then we can regard boosting as an algorithm for learning a linear threshold function on the d-dimensional feature space given by the feature map $\phi(x) := (h^{(1)}(x), \dots, h^{(d)}(x))$. That is, the final hypothesis returned by boosting is

$$\hat{h}(x) = \mathrm{sign}(\langle \phi(x), \lambda \rangle), \quad x \in \mathcal{X}$$

for some coefficient vector $\lambda = (\lambda^{(1)}, \dots, \lambda^{(d)}) \in \mathbb{R}^d$. In fact, Algorithm 7.1 can be seen as a method for finding λ that (approximately) minimizes the sum of exponential losses

$$L(\lambda) := \sum_{i=1}^{n} \ell_{\exp}(y_i \langle \phi(x_i), \lambda \rangle) = \sum_{i=1}^{n} \exp(-y_i \langle \phi(x_i), \lambda \rangle)$$

where $\ell_{\exp}(s) := \exp(-s)$. The search strategy employed is *coordinate descent*: repeatedly pick one of the optimization variables to adjust (in a way that reduces the objective value), holding all other variables fixed.

In coordinate descent, the choice of the optimization variable to adjust is guided by the partial derivatives of the objective function L at the current setting of the variables. The partial derivative of L with respect to $\lambda^{(j)}$ is

$$\frac{\partial L}{\partial \lambda^{(j)}}(\lambda) = -\sum_{i=1}^{n} \exp(-y_i \langle \phi(x_i), \lambda \rangle) y_i h^{(j)}(x_i).$$

If the partial derivative with respect to $\lambda^{(j)}$ is negative, then slightly increasing $\lambda^{(j)}$ will decrease the objective value.

Suppose $\lambda_t = (\lambda_t^{(1)}, \ldots, \lambda_t^{(d)})$ is the setting of the optimization variables in iteration t. (We initialize with $\lambda_1 = (0, \ldots, 0) \in \mathbb{R}^d$.) Define $p_t = (p_{t,1}, \ldots, p_{t,n}) \in \Delta^{n-1}$ by

$$p_{t,i} := \frac{\exp(-y_i \langle \phi(x_i), \lambda_t \rangle)}{L_t} \quad \forall i \in [n], \quad \text{where} \quad L_t := \sum_{i=1}^{n} \exp(-y_i \langle \phi(x_i), \lambda_t \rangle) = L(\lambda_t). \tag{7.12}$$

Then for each $j \in [d]$, we have

$$\begin{aligned} \frac{\partial L}{\partial \lambda^{(j)}}(\lambda_t) &= -L_t \sum_{i=1}^{n} p_{t,i}\ y_i h^{(j)}(x_i) \\ &= -2L_t \left(\mathsf{acc}(h^{(j)}; S, p_t) - \frac{1}{2} \right). \end{aligned}$$

The γ-WLA implies that there is some $j_t \in [d]$ for which $\mathsf{acc}(h^{(j_t)}; S, p_t) \geq (1+\gamma)/2$, and hence

$$\frac{\partial L}{\partial \lambda^{(j_t)}}(\lambda_t) \leq -L_t \gamma < 0.$$

So under the γ-WLA, the weak learner discovers an optimization variable $\lambda^{(j_t)}$ that is suitable for updating in coordinate descent. (In Algorithm 7.1, this is the j_t such that $h_t = h^{(j_t)}$.) The update is

$$\lambda_{t+1}^{(j)} := \begin{cases} \lambda_t^{(j)} + \eta/2 & \text{if } j = j_t, \\ \lambda_t^{(j)} & \text{if } j \neq j_t \end{cases}$$

for some $\eta > 0$. This update from λ_t to λ_{t+1} also implies the following update from p_t to p_{t+1} (defined in a manner analogous to (7.12)):

$$\begin{aligned} p_{t+1,i} &\propto \exp(-y_i \langle \phi(x_i), \lambda_{t+1} \rangle) \\ &= \exp(-y_i \langle \phi(x_i), \lambda_t \rangle) \exp(-(\eta/2) y_i h^{(j_t)}(x_i)) \\ &\propto p_{t,i} \exp(-(\eta/2) y_i h^{(j_t)}(x_i)) \\ &\propto p_{t,i} \exp(-\eta \mathbb{1}\{h^{(j_t)}(x_i) = y_i\}), \end{aligned}$$

which is the same as the update from p_t to p_{t+1} in Algorithm 7.1.

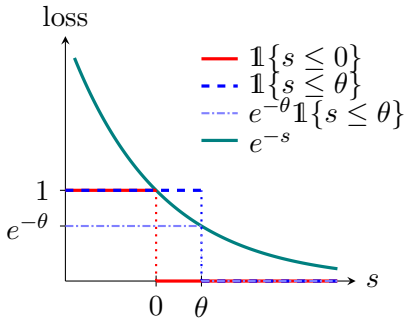


Figure 7.2: Comparison of zero-one loss $\ell_{0/1}(s) = \mathbb{1}\{s \le 0\}$, θ-margin loss $\ell_\theta(s) = \mathbb{1}\{s \le \theta\}$ (and a scaled version of it), and exponential loss $\ell_{\exp}(s) = \exp(-s)$.

Exercise 7.5. Theorems 7.1 and 7.2 suggest the hyperparameter setting of $\eta := 2\gamma$ under the γ-WLA. Show that under the γ-WLA,

$$L(\lambda_{t+1}) \le \exp(-\gamma^2/2) L(\lambda_t)$$

with this choice of η. *Hint: use Lemma 4.1 (Hoeffding's Lemma).* $\lozenge$

Why is the sum of exponential losses a meaningful objective to try to minimize? First, the exponential loss is an upper bound on the zero-one loss: $\ell_{\exp}(s) \ge \ell_{0/1}(s)$ for all s. So if the sum of exponential losses is small, then so is the number of classification mistakes on the training data. Secondly, the exponential loss also upper-bounds re-scaled θ-margin loss for all $\theta > 0$: $\ell_{\exp}(s) \ge e^{-\theta}\ell_\theta(s)$ for all s. See Figure 7.2 for an illustration. One important caveat here is that the θ-margin loss $\ell_\theta(s)$ is primarily meaningful when $|s| \le 1$. However, λ is not explicitly constrained in a way to ensure $|y_i \langle \phi(x_i), \lambda \rangle| \le 1$. So the fact that Algorithm 7.1 seeks to minimize the sum of exponential losses is not sole reason that it can be resistant to over-fitting.

Topic 8: Linear learning

8.1 Linear learning paradigm

Boosting as we saw in Chapter 7 is an example of a common paradigm in machine learning, which we call *linear learning*[1]:

- Design a feature map $\phi\colon \mathcal{X} \to \mathbb{R}^d$ (perhaps implicitly, such as via the choice of a weak learning algorithm as in boosting).
- Use training data $S = ((x_1, y_1), \ldots, (x_T, y_T)) \in (\mathcal{X} \times \{-1, 1\})^T$ and an optimization algorithm to find a vector of coefficients (a.k.a. *weight vector*) $w \in \mathbb{R}^d$ that (approximately) minimizes

$$\frac{1}{n} \sum_{i=1}^{n} \ell(y_i \langle \phi(x_i), w \rangle)$$

  for some loss function $\ell\colon \mathbb{R} \to \mathbb{R}_+$ (such as the exponential loss $\ell_{\exp}$).
- Return the linear threshold function $x \mapsto \operatorname{sign}(\langle \phi(x), w \rangle)$.

8.1.1 Computational intractability

With boosting, under the γ-WLA, the linear learning playbook is carried out efficiently (even when the dimension d of the feature map is infinite!). However, the γ-WLA implies that the training data set S is linearly separable. If the data distribution P does not guarantee S is linearly separable, we might still hope to find an LTF that compares well to the best error rate achievable by LTFs. This was called PAC agnostic learning in Section 4.4.3, and implementing ERM with $\mathcal{H} = \mathsf{LTF}_{\mathbb{R}^d}$ would be one way to achieve this. Unfortunately, this is computationally intractable in general.

Theorem 8.1 (Johnson and Preparata, 1978)**.** *The following problem is NP-hard:*

> *Given $S \subset \{0,1\}^d \times \{-1,1\}$ and an integer k, decide if there is a linear threshold function that misclassifies at most k examples in S.*

Proof. We give an efficient reduction from the Vertex Cover problem, which is well-known to be NP-complete (Karp, 1972). The Vertex Cover problem is as follows: given an undirected graph $G = (V, E)$ and an integer k, is there a vertex cover $U \subseteq V$ of cardinality at most k? A vertex cover for a graph is a subset of the vertices that contains, for every edge $\{u, v\}$ in the graph, at least one of u and v.

Let $G = (V, E)$ and k be an instance of the Vertex Cover problem. Denote the vertices by $V = \{1, 2, \ldots, n\}$, and let m be the number of edges. We construct a dataset $S \subset \{0,1\}^d \times \{-1,1\}$ with $d = 2n$ and $|S| = n + 2m$ based on the graph G. Let $e_1, \ldots, e_{2n}$ be the standard coordinate basis vectors for $\mathbb{R}^{2n}$. For any distinct $i, j \in [2n]$, define the vector $x_{i,j} := e_i + e_j \in \mathbb{R}^{2n}$. We construct the dataset S as follows:

[1] There are some other examples of boosting algorithms that fall outside of the linear learning paradigm, such as those that return a branching program (e.g., Mansour and McAllester, 2002; Kalai and Servedio, 2005).

- For each $i \in V$, add the labeled example $(x_{i,n+i}, +1)$ to S. These are the positive examples, one per vertex.
- For each edge $\{i, j\} \in E$, add the labeled examples $(x_{i,j}, -1)$ and $(x_{n+i,n+j}, -1)$ to S. These are the negative examples, two per edge.

We show the following:

There is a linear threshold function $h_{w,b} \in \mathsf{LTF}_{\mathbb{R}^d}$ that misclassifies at most k examples in S.
$\Leftrightarrow$ There is a vertex cover for G of cardinality at most k.

(Here, we use the convention $h_{w,b}(x) = 1$ iff $\langle w, x\rangle + b \geq 0$.)

($\Rightarrow$) Suppose there is a linear threshold function $h_{w,b} \in \mathsf{LTF}_{\mathbb{R}^d}$ that misclassifies at most k examples in S. Let $U \subseteq V$ be defined as follows:

- For each vertex $i \in V$, if $h_{w,b}(x_{i,n+i}) = -1$ (a mistake), then add i to U.
- For each edge $\{i, j\} \in E$, if $h_{w,b}(x_{i,j}) = +1$ (a mistake) or $h_{w,b}(x_{n+i,n+j}) = +1$ (also a mistake), then add exactly one of i and j to U.

Since $h_{w,b}$ misclassifies at most k examples in S, this set U has cardinality at most k. We claim that U is a vertex cover for G. Consider any edge $\{i, j\} \in E$. If $h_{w,b}$ misclassifies $x_{i,n+i}$ or $x_{j,n+j}$, then $i \in U$ or $j \in U$. Now instead suppose $h_{w,b}$ correctly classifies both $x_{i,n+i}$ and $x_{j,n+j}$. Writing $w = (w_1, \ldots, w_{2n})$, we must have

$$\begin{aligned} & w_i + w_{n+i} + b \geq 0 \\ \text{and} \qquad & w_j + w_{n+j} + b \geq 0, \end{aligned}$$

which implies

$$(w_i + w_j + b) + (w_{n+i} + w_{n+j} + b) \geq 0.$$

This means that at least one of $x_{i,j}$ and $x_{n+i,n+j}$ is misclassified, so we include one of i and j in U. We conclude that U is a vertex cover for G.

($\Leftarrow$) Now instead suppose G has a vertex cover U of cardinality at most k. Define the linear threshold function $h_{w,b}$ as follows:

- Set $b := -1$.
- For each $i \in V$, set

$$w_i := w_{n+i} := \begin{cases} -1 & \text{if } i \in U, \\ +1 & \text{if } i \notin U. \end{cases}$$

We claim that $h_{w,b}$ misclassifies at most k examples. Consider any edge $\{i, j\} \in E$ and the corresponding negative examples $x_{i,j}$ and $x_{n+i,n+j}$. Then $\langle x_{i,j}, w\rangle = w_i + w_j \leq 0$ since at least one of i and j is in U. Therefore $h_{w,b}(x_{i,j}) = -1$. Similarly, $h_{w,b}(x_{n+i,n+j}) = -1$. So $h_{w,b}$ correctly classifies all negative examples. Now consider any $i \in V$ and the corresponding positive example $x_{i,n+i}$. Then $\langle x_{i,n+i}, w\rangle = w_i + w_{n+i} < 1$ iff $i \in U$. So $h_{w,b}(x_{i,n+i}) = -1$ iff $i \in U$. Since $|U| \leq k$, it follows that $h_{w,b}$ misclassifies at most k (positive) examples. $\square$

Theorem 8.1 implies that (unless P=NP) it is computationally intractable to find an LTF that minimizes the training error rate on an arbitrary data set. Similar forms of computational intractability hold for many other hypothesis classes. What can we do about this? Here are some possibilities (which may not be mutually exclusive nor individually sufficient):

- Settle for weaker guarantees, such as comparing to a weaker benchmark (e.g., the best achievable θ-margin error rate).
- Make assumptions about the data set/distribution, e.g., P is $\mathcal{C}$-realizable, $P_{\mathbf{x}}$ is Gaussian.
- Change the learning model, e.g., assume query access to the target function.

Two common algorithmic strategies that are particularly effective with linear learning are:

- Using a convex surrogate loss function (e.g., $\ell_{\exp}$) instead of the zero-one loss.
- Enlarging the hypothesis class to make optimization easier.

8.1.2 Convex surrogate losses

Fix a probability distribution P over $\mathcal{X} \times \{-1, 1\}$, and let $(\mathbf{x}, \mathbf{y}) \sim P$. For any loss function $\ell\colon \mathbb{R} \to \mathbb{R}$ and $f\colon \mathcal{X} \to \mathbb{R}$, define

$$\mathsf{R}_\ell(f; P) := \mathrm{E}[\ell(\mathbf{y} f(\mathbf{x}))],$$
$$\mathsf{R}^\star_\ell(P) := \inf_{f\colon \mathcal{X}\to\mathbb{R}} \mathsf{R}_\ell(f; P).$$

Here, $\mathsf{R}_\ell(f; P)$ is the *ℓ-risk* of f, and $\mathsf{R}^\star_\ell(P)$ is the *optimal ℓ-risk*. In the special case of zero-one loss $\ell_{0/1}$, we simply write

$$\mathsf{R}(f; P) := \mathrm{E}\big[\ell_{0/1}(\mathbf{y} f(\mathbf{x}))\big],$$
$$\mathsf{R}^\star(P) := \inf_{f\colon \mathcal{X}\to\mathbb{R}} \mathsf{R}(f; P).$$

(If we assume $f(\mathbf{x}) \neq 0$, then $\mathsf{R}(f; P) = \mathsf{err}(\operatorname{sign} \circ f; P)$.) We call $\mathsf{R}(f; P)$ the *(classification) risk* of f, and $\mathsf{R}^\star(P)$ the *optimal (or Bayes) risk*. The difference between $\mathsf{R}_\ell(f; P)$ and $\mathsf{R}^\star_\ell(P)$ (respectively, $\mathsf{R}(f; P)$ and $\mathsf{R}^\star(P)$) is the *excess ℓ-risk* (respectively, excess (classification) risk) of f.

If f has small excess ℓ-risk, what can be said about its excess risk? Note that if $\mathsf{R}^\star_\ell(P) = \mathsf{R}^\star(P) = 0$ (which may be the case if we assume realizability), then for many loss functions ℓ, we can exploit a relationship between ℓ and $\ell_{0/1}$ to answer this question. In particular, $\ell_{\exp} \geq \ell_{0/1}$ point-wise, so small $\ell_{\exp}$-risk implies small risk. But it is also possible that $\mathsf{R}^\star_\ell(P) \neq 0$ and $\mathsf{R}^\star(P) \neq 0$, for example if $\mathbf{y}$ is not a deterministic function of $\mathbf{x}$, in which case the function $f^\star_\ell$ achieving the optimal ℓ-risk can be very different from a function $f^\star$ achieving the optimal classification risk.

Proposition 8.1. $\mathsf{R}^\star(P) = \mathsf{R}(f^\star; P) = \mathrm{E}[\min\{p(\mathbf{x}), 1 - p(\mathbf{x})\}]$, *where*

$$p(x) := \Pr(\mathbf{y} = 1 \mid \mathbf{x} = x), \qquad f^\star(x) := \operatorname{sign}(p(x) - 1/2), \qquad x \in \mathcal{X}.$$

Proposition 8.2. *Let* $\ell = \ell_{\exp}$. *Then* $\mathsf{R}^\star_\ell(P) = \mathsf{R}_\ell(f^\star_\ell; P) = 2\,\mathrm{E}[\sqrt{p(\mathbf{x})(1 - p(\mathbf{x}))}]$, *where*

$$p(x) := \Pr(\mathbf{y} = 1 \mid \mathbf{x} = x), \qquad f^\star_\ell(x) := \frac{1}{2} \ln \frac{p(x)}{1 - p(x)}, \qquad x \in \mathcal{X}.$$

(Here, $f^\star_\ell$ may take values in $\{-\infty, \infty\}$.)

On the other hand, we see from Propositions 8.1 and 8.2 that the optimal risk and optimal $\ell_{\exp}$-risk are quite related, and this allows us to show the following.

Proposition 8.3. *Let $\ell = \ell_{\exp}$. For any $f\colon \mathcal{X} \to \mathbb{R}$,*

$$\mathsf{R}(f;P) - \mathsf{R}^\star(P) \le \sqrt{2(\mathsf{R}_\ell(f;P) - \mathsf{R}^\star_\ell(P))}.$$

Proposition 8.3 is good motivation for seeking $f\colon \mathcal{X} \to \mathbb{R}$ with small excess $\ell_{\exp}$-risk. The convexity of $\ell_{\exp}$ is useful when f has an affine/linear parameterization, e.g., $f(x) = \langle \phi(x), w\rangle$ for weight vector parameter $w \in \mathbb{R}^d$ because the composition of a convex function and an affine function is convex, and hence $w \mapsto \ell_{\exp}(y\langle \phi(x), w\rangle)$ is convex. Since a non-negative scaling of a convex function is convex, and so is the sum (or integral) of convex functions, it follows that objectives $J\colon \mathbb{R}^d \to \mathbb{R}$ of the form

$$J(w) := \mathrm{E}[\ell_{\exp}(\mathbf{y}\langle \phi(\mathbf{x}), w\rangle)], \quad (\mathbf{x},\mathbf{y}) \sim P$$

are convex functions. We will see in Section 8.2 that convex objectives can be efficiently minimized using very simple algorithms.

There are many other convex surrogate loss functions that have (qualitatively) similar properties as $\ell_{\exp}$. Some of the most well-known ones are as follows.

- *Hinge loss*: $\ell_{\text{hinge}}(s) := \max\{0, 1-s\}$.
- *Square loss*: $\ell_{\text{square}}(s) := (1-s)^2$.
- *Logistic loss*: $\ell_{\text{logistic}}(s) := \ln(1+e^{-s})$.

For each of the losses $\ell \in \{\ell_{\exp}, \ell_{\text{hinge}}, \ell_{\text{square}}, \ell_{\text{logistic}}\}$, Zhang (2004) shows that excess risk

$$\mathsf{R}(f;P) - \mathsf{R}^\star(P) \le C(\mathsf{R}_\ell(f;P) - \mathsf{R}^\star_\ell(P))^{1/c}$$

for some $C > 0$ and $c \ge 1$ (both depending only on ℓ). For example, for ℓ_{hinge}, $C = 1$ and $c = 1$. A more general theory about such excess risks and ℓ-risks was developed by Bartlett et al. (2006).

For hinge loss, note that $\mathsf{R}(f;P) \le \mathsf{R}_{\ell_{\text{hinge}}}(f;P)$, but this loses a factor of two for functions that are $\{-1,1\}$-valued. That is, for $g\colon \mathcal{X} \to \{-1,1\}$ and $(x,y) \in \mathcal{X} \times \{-1,1\}$, we have $\ell_{\text{hinge}}(yg(x)) = 2 \times \mathbb{1}\{g(x) \ne y\}$. This factor of two can be avoided by using a linear threshold function with a random threshold (or an optimally chosen threshold).

Proposition 8.4. *For $f\colon \mathcal{X} \to \mathbb{R}$ and $\tau \in \mathbb{R}$, let $h_{f,\tau}\colon \mathcal{X} \to \{-1,1\}$ be the classifier with $h_{f,\tau}(x) = 1$ iff $f(x) > \tau$. If $\boldsymbol{\tau} \sim \text{Uniform}([-1,1])$ is independent of $(\mathbf{x},\mathbf{y}) \sim P$ for a probability distribution P over $\mathcal{X} \times \{-1,1\}$, then*

$$\mathrm{E}[\mathsf{err}(h_{f,\boldsymbol{\tau}};P)] \le \frac{1}{2}\mathsf{R}_{\ell_{\text{hinge}}}(f;P).$$

In particular, there exists $\tau \in [-1,1]$ such that $\mathsf{err}(h_{f,\tau};P) \le (1/2)\mathsf{R}_{\ell_{\text{hinge}}}(f;P)$.

Exercise 8.1. Prove Proposition 8.4. ◊

There is one important wrinkle when considering the use of convex surrogate losses in the linear learning paradigm. The notion of optimal ℓ-risk is defined by an infimum over all functions $f\colon \mathcal{X} \to \mathbb{R}$. If the functions under consideration are of the form $f(x) = \langle \phi(x), w\rangle$ for some $w \in \mathbb{R}^d$, we need to choose the feature map ϕ in a way so that $f^\star_\ell$ (or another function with small ℓ-risk) can be represented (or at least well-approximated) by $\langle \phi(x), w^\star\rangle$ for some $w^\star \in \mathbb{R}^d$. This is discussed further in Section 8.3.

8.2 Stochastic gradient descent

8.2.1 Stochastic convex optimization

Let P be a probability distribution on a domain $\mathcal{Z}$, and let $J\colon \mathbb{R}^d \to \mathbb{R}$ be an objective function of the form

$$J(w) := \mathrm{E}[L(w; \mathbf{z})]$$

where $\mathbf{z} \sim P$, and $L\colon \mathbb{R}^d \times \mathcal{Z} \to \mathbb{R}$ is a convex function of its first argument. The stochastic optimization problem is to find $w \in \mathbb{R}^d$ that approximately minimizes J.

Write $L_z := L(\cdot; z)$ for $z \in \mathcal{Z}$, and let $\nabla L_z\colon \mathbb{R}^d \to \mathbb{R}^d$ be a *subgradient map* such that $\nabla L_z(w) \in \partial L_z(w)$ for all $w \in \mathbb{R}^d$. We assume that there exists $G \geq 0$ such that $\mathrm{E}[\|\nabla L_{\mathbf{z}}(w)\|^2] \leq G^2$ for all $w \in \mathbb{R}^d$. For example, suppose $\mathcal{Z} = \mathcal{X} \times \{-1, 1\}$ and $\phi\colon \mathcal{X} \to \mathbb{R}^d$ is such that $\mathrm{E}[\|\phi(\mathbf{x})\|^2] \leq G^2$ for $(\mathbf{x}, \mathbf{y}) \sim P$. Suppose $L_{(x,y)}(w) := \ell_{\mathrm{hinge}}(y\langle \phi(x), w\rangle)$. Letting

$$\nabla L_{(x,y)}(w) := \begin{cases} 0 & \text{if } L_{(x,y)}(w) = 0, \\ -y\phi(x) & \text{if } L_{(x,y)}(w) > 0, \end{cases}$$

we have $\nabla L_{(x,y)}(w) \in \partial L_{(x,y)}(w)$ and $\mathrm{E}[\|\nabla L_{(\mathbf{x},\mathbf{y})}(w)\|] \leq G$.

8.2.2 Stochastic gradients

To solve the stochastic optimization problem, we assume access to the subgradient maps $\nabla L_{\mathbf{z}_t}$ for all $t \in [T]$, where $(\mathbf{z}_1, \ldots, \mathbf{z}_T) \sim P^T$ is a size-T iid sample from P. The outputs of $\nabla L_{\mathbf{z}_t}$ are regarded as *stochastic gradients* for the objective function J, which is justified by the following proposition.

Proposition 8.5. *Let $\mathbf{z} \sim P$. For any $w_{\mathrm{o}} \in \mathbb{R}^d$,*

$$\mathrm{E}[\nabla L_{\mathbf{z}}(w_{\mathrm{o}})] \in \partial J(w_{\mathrm{o}}).$$

Proof. Fix any $w, w_{\mathrm{o}} \in \mathbb{R}^d$. Then

$$\begin{aligned} J(w) &= \mathrm{E}[L_{\mathbf{z}}(w)] \\ &\geq \mathrm{E}[L_{\mathbf{z}}(w_{\mathrm{o}}) + \langle \nabla L_{\mathbf{z}}(w_{\mathrm{o}}), w - w_{\mathrm{o}}\rangle] && \text{since } \nabla L_{\mathbf{z}}(w_{\mathrm{o}}) \in \partial L_{\mathbf{z}}(w_{\mathrm{o}}) \\ &= J(w_{\mathrm{o}}) + \langle \mathrm{E}[\nabla L_{\mathbf{z}}(w_{\mathrm{o}})], w - w_{\mathrm{o}}\rangle && \text{by linearity of expectation.} \end{aligned}$$

Therefore $\mathrm{E}[\nabla L_{\mathbf{z}}(w_{\mathrm{o}})] \in \partial J(w_{\mathrm{o}})$. □

8.2.3 Stochastic (sub)gradient descent algorithm

The *stochastic (sub)gradient descent (SGD)* algorithm is given in Algorithm 8.1.

Algorithm 8.1: Stochastic (sub)gradient descent (SGD).

Hyperparameter: $\eta > 0$.
Input: subgradient maps $\nabla L_{\mathbf{z}_t}$ for all $t \in [T]$, where $(\mathbf{z}_1, \dots, \mathbf{z}_T) \sim P^T$.

- Initially, $\mathbf{w}_1 := 0$.
- For $t = 1, 2, \dots, T$:
  - Let $\mathbf{g}_t := \nabla L_{\mathbf{z}_t}(\mathbf{w}_t)$, and compute $\mathbf{w}_{t+1} := \mathbf{w}_t - \eta \mathbf{g}_t$.
- Return $\bar{\mathbf{w}} := \frac{1}{T} \sum_{t=1}^{T} \mathbf{w}_t$.

Theorem 8.2. *Assume that there exists $G > 0$ such that, for all $w \in \mathbb{R}^d$, we have $\mathrm{E}[\|\nabla L_{\mathbf{z}}(w)\|^2] \le G^2$ where $\mathbf{z} \sim P$. Then the output $\bar{\mathbf{w}}$ of SGD (Algorithm 8.1) satisfies, for any $u \in \mathbb{R}^d$,*

$$\mathrm{E}[J(\bar{\mathbf{w}})] \le J(u) + \frac{\|u\|^2}{2\eta T} + \frac{\eta G^2}{2}.$$

Proof. For each $t \in [T]$, observe that $\mathbf{w}_t$ depends only on $\mathbf{z}_1, \dots, \mathbf{z}_{t-1}$. By Proposition 8.5,

$$\mathrm{E}[\nabla L_{\mathbf{z}_t}(\mathbf{w}_t) \mid \mathbf{z}_1, \dots, \mathbf{z}_{t-1}] \in \partial J(\mathbf{w}_t).$$

Therefore, since J is convex,

$$J(u) \ge J(\mathbf{w}_t) + \langle \mathrm{E}[\nabla L_{\mathbf{z}_t}(\mathbf{w}_t) \mid \mathbf{z}_1, \dots, \mathbf{z}_{t-1}], u - \mathbf{w}_t \rangle.$$

Re-arranging, averaging over all $t \in [T]$, and appealing to linearity of expectation gives

$$\mathrm{E}\left[\frac{1}{T}\sum_{t=1}^{T} J(\mathbf{w}_t)\right] - J(u) \le \mathrm{E}\left[\frac{1}{T}\sum_{t=1}^{T} \langle \nabla L_{\mathbf{z}_t}(\mathbf{w}_t), \mathbf{w}_t - u \rangle\right] = \mathrm{E}\left[\frac{1}{T}\sum_{t=1}^{T} \langle \mathbf{g}_t, \mathbf{w}_t - u \rangle\right]. \tag{8.1}$$

Now consider any $t \in [T]$; the update $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \mathbf{g}_t$ implies

$$\frac{1}{2\eta}\|\mathbf{w}_{t+1} - u\|^2 - \frac{1}{2\eta}\|\mathbf{w}_t - u\|^2 = -\langle \mathbf{g}_t, \mathbf{w}_t - u \rangle + \frac{\eta}{2}\|\mathbf{g}_t\|^2.$$

Re-arranging and averaging over all $t \in [T]$ gives

$$\frac{1}{T}\sum_{t=1}^{T} \langle \mathbf{g}_t, \mathbf{w}_t - u \rangle \le \frac{\|\mathbf{w}_1 - u\|^2 - \|\mathbf{w}_{T+1} - u\|^2}{2\eta T} + \frac{\eta}{2T}\sum_{t=1}^{T}\|\mathbf{g}_t\|^2 \le \frac{\|u\|^2}{2\eta T} + \frac{\eta}{2T}\sum_{t=1}^{T}\|\mathbf{g}_t\|^2.$$

Taking expectation of both sides and using $\mathrm{E}[\|\mathbf{g}_t\|^2] \le G^2$ gives

$$\mathrm{E}\left[\frac{1}{T}\sum_{t=1}^{T} \langle \mathbf{g}_t, \mathbf{w}_t - u \rangle\right] \le \frac{\|u\|^2}{2\eta T} + \frac{\eta G^2}{2}. \tag{8.2}$$

Finally, by Jensen's inequality and convexity of J,

$$J(\bar{\mathbf{w}}) = J\left(\frac{1}{T}\sum_{t=1}^{T} \mathbf{w}_t\right) \le \frac{1}{T}\sum_{t=1}^{T} J(\mathbf{w}_t).$$

Combining this with (8.1) and (8.2) proves the claim. □

Much of the proof of Theorem 8.2 is reminiscent of the analysis of Hedge (Algorithm 2.2). In fact, the core of the analysis applies to an algorithm for online convex optimization (Section 2.5) called *online gradient descent (OGD)*, which has exactly the same mechanics as SGD but couched in the framework of online convex optimization where an adversary always reveals a subgradient of the (adversarially) chosen convex loss function at the learner's chosen weight vector. The setting differs from that in Section 2.5 only in terms of the domain of the weight vectors (called allocation vectors in Section 2.5).

Corollary 8.1. *In the setting of Theorem 8.2, if the hyperparameter*

$$\eta := \frac{W}{G\sqrt{T}}, \quad W > 0$$

is used in SGD (Algorithm 8.1), then the output $\bar{\mathbf{w}}$ *of SGD satisfies*

$$\mathrm{E}[J(\bar{\mathbf{w}})] \leq \inf_{w \in \mathbb{R}^d : \|w\| \leq W} J(w) + \frac{WG}{\sqrt{T}}.$$

Corollary 8.1 shows that, for any $W \geq 0$, an appropriate setting of the hyperparameter η and sufficiently many iid training data ensure that the output of SGD is competitive against every weight vector u of norm at most W. The sample complexity for achieving the guarantee of ϵ-suboptimality in objective value is at most

$$\left(\frac{WG}{\epsilon}\right)^2.$$

The time complexity is also similar: it is linear in the sample complexity, scaled by the cost of evaluating the subgradient map and updating the weight vector parameter (typically $O(d)$).

8.3 Approximation theory

8.3.1 Linear learning with convex surrogates and SGD

Let us instantiate Corollary 8.1 for linear learning with the ℓ-risk for $\ell = \ell_{\text{hinge}}$.

Corollary 8.2. *Let* P *be a probability distribution over* $\mathcal{X} \times \{-1, 1\}$*, and let* $((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_T, \mathbf{y}_T)) \sim P^T$*. Let* $\phi \colon \mathcal{X} \to \mathbb{R}^d$ *satisfy* $\mathrm{E}[\|\phi(\mathbf{x})\|^2] \leq G^2$ *for some* $G \geq 0$*. The output* $\bar{\mathbf{w}}$ *of SGD (Algorithm 8.1) with* $L_{\mathbf{z}_t}(w) := \ell_{\text{hinge}}(\mathbf{y}_t \langle \phi(\mathbf{x}_t), w \rangle)$ *for all* $t \in [T]$ *and hyperparameter*

$$\eta := \frac{W}{G\sqrt{T}}, \quad W > 0$$

satisfies

$$\mathrm{E}[\mathsf{err}(\mathrm{sign}(\langle \phi(\cdot), \bar{\mathbf{w}} \rangle); P)] \leq \mathrm{E}\big[\mathsf{R}_{\ell_{\text{hinge}}}(\langle \phi(\cdot), \bar{\mathbf{w}} \rangle; P)\big] \leq \min_{w \in \mathbb{R}^d : \|w\| \leq W} \mathsf{R}_{\ell_{\text{hinge}}}(\langle \phi(\cdot), w \rangle; P) + \frac{WG}{\sqrt{T}}.$$

The meaning of the expected error rate bound in Corollary 8.2 is unclear because it depends on our choice of the feature map ϕ. In the linear learning paradigm, we may make assumptions about the population P that imply the existence of a "nice" function $g \colon \mathcal{X} \to \mathbb{R}$ such that $\mathsf{R}_{\ell_{\text{hinge}}}(g; P)$ is small. This nice function g is not known to the learner, but we use the assumed "niceness"

properties to guide the choice of the feature map ϕ such that $\mathrm{E}[|\langle \phi(\mathbf{x}), w\rangle - g(\mathbf{x})|]$ is small for some $w \in \mathbb{R}^d$ (with $\|w\|$ not too large). One can check that

$$\min_{w\in\mathbb{R}^d:\|w\|\le W} \mathsf{R}_{\ell_{\mathrm{hinge}}}(\langle\phi(\cdot), w\rangle; P) \le \min_{\substack{g\colon \mathcal{X}\to\mathbb{R},\\ w\in\mathbb{R}^d:\|w\|\le W}} \Big(\mathsf{R}_{\ell_{\mathrm{hinge}}}(g; P) + \mathrm{E}[|\langle\phi(\mathbf{x}), w\rangle - g(\mathbf{x})|]\Big).$$

One assumption that could be made about the population P is that there exists a Lipschitz function with small ℓ_{hinge}-risk. To make this concrete, consider a Euclidean input space $\mathcal{X} := \mathbb{R}^n$; a function $g\colon \mathbb{R}^n \to \mathbb{R}$ is said to be *λ-Lipschitz* (with respect to the Euclidean metric) if

$$|g(x) - g(x')| \le \lambda\|x - x'\|, \quad x, x' \in \mathbb{R}^n$$

for some $\lambda \ge 0$. We assume that there is a λ-Lipschitz function $g\colon \mathbb{R}^n \to \mathbb{R}$ with $\mathsf{R}_{\ell_{\mathrm{hinge}}}(g; P) \le \epsilon$. Then, to take advantage of this assumption in the linear learning paradigm, we seek a feature map $\phi\colon \mathbb{R}^n \to \mathbb{R}^d$ such that every λ-Lipschitz function can be well-approximated by $x \mapsto \langle\phi(x), w\rangle$ for some $w \in \mathbb{R}^d$.

8.3.2 Approximation of Lipschitz functions by polynomials

A classical approach to function approximation is based on polynomials. The Weierstrass approximation theorem states that every continuous function on a compact set can be uniformly approximated (arbitrarily well) by a polynomial. A simple quantitative version of this theorem (just for Lipschitz functions) can be easily obtained using Bernstein's constructive proof of the Weierstrass theorem (Bernstein, 1912). We first show the result for functions on $[0, 1] \subset \mathbb{R}^1$.

Theorem 8.3. *For each $m \in \mathbb{N}$ and $k \in \{0, 1, \ldots, m\}$, define the polynomial $b_{m,k}$ by*

$$b_{m,k}(x) := \binom{m}{k} x^k (1 - x)^{m-k}.$$

(The polynomials $\{b_{m,k}\}_{k=0}^m$ are called the Bernstein polynomials of order m.) For any λ-Lipschitz function $g\colon [0, 1] \to \mathbb{R}$, the degree-$m$ polynomial p defined by

$$p(x) := \sum_{k=0}^{m} g(k/m) b_{m,k}(x)$$

satisfies

$$|p(x) - g(x)| \le \frac{\lambda}{2\sqrt{m}}, \quad \forall x \in [0, 1].$$

Proof. Fix $x \in [0, 1]$, and let $\mathbf{s} \sim \mathrm{Binomial}(m, x)$. Then

$$\begin{aligned}\mathrm{E}[g(\mathbf{s}/m)] &= \sum_{k=0}^{m} g(k/m)\binom{m}{k} x^k (1 - x)^{m-k} \\ &= \sum_{k=0}^{m} g(k/m) b_{m,k}(x) = p(x).\end{aligned}$$

Therefore

$$\begin{aligned}
|p(x) - g(x)| &= |\mathrm{E}[g(\mathbf{s}/m) - g(x)]| \\
&\leq \mathrm{E}[|g(\mathbf{s}/m) - g(x)|] && \text{(by Jensen's inequality)} \\
&\leq \lambda\, \mathrm{E}[|\mathbf{s}/m - x|] && \text{(because } g \text{ is } \lambda\text{-Lipschitz)} \\
&\leq \lambda \sqrt{\mathrm{E}\Big[(\mathbf{s}/m - x)^2\Big]} && \text{(by Jensen's inequality)} \\
&= \lambda \sqrt{\frac{x(1-x)}{m}} \\
&\leq \frac{\lambda}{2\sqrt{m}}. && \square
\end{aligned}$$

How does Theorem 8.3 motivate a feature map? Again, consider just $n = 1$; we define the degree-m polynomial feature map $\phi\colon \mathbb{R} \to \mathbb{R}^{m+1}$ by

$$\phi(x) := (1, x, x^2, \ldots, x^m).$$

Then, for any degree-m univariate polynomial $p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_m x^m$, consider the weight vector $w \in \mathbb{R}^{m+1}$ defined by

$$w := (a_0, a_1, \ldots, a_m).$$

It is easy to see that

$$\langle \phi(x), w \rangle = p(x).$$

This means that for every λ-Lipschitz function $g\colon [0,1] \to \mathbb{R}$, there exists $w \in \mathbb{R}^{m+1}$ such that

$$|\langle \phi(x), w \rangle - g(x)| \leq \frac{\lambda}{2\sqrt{m}}, \quad \forall x \in [0,1].$$

The catch, however, is that for functions on higher-dimensional Euclidean spaces, the situation is not so good. The generalization of Theorem 8.3 is the following.

Theorem 8.4. *For each $m \in \mathbb{N}$ and $\alpha \in \{0, 1, \ldots, m\}^n$, define the polynomial $b_{m,\alpha}$ by*

$$b_{m,\alpha}(x) := \prod_{i=1}^{n} \binom{m}{\alpha_i} x_i^{\alpha_i} (1 - x_i)^{m - \alpha_i}.$$

For any λ-Lipschitz function $g\colon [0,1]^n \to \mathbb{R}$, the polynomial p defined by

$$p(x) := \sum_{\alpha \in \{0,1,\ldots,m\}^n} g(\alpha/m) b_{m,\alpha}(x)$$

satisfies

$$|p(x) - g(x)| \leq \frac{\lambda}{2} \sqrt{\frac{n}{m}}, \quad \forall x \in [0,1]^n.$$

Exercise 8.2. Prove Theorem 8.4. ◊

The error bound in Theorem 8.4 is non-trivial only when $m = \Omega(\lambda^2 n)$. However, the polynomials $\{b_{m,\alpha}\}_{\alpha \in \{0,1,\ldots,m\}^n}$ span a vector space of dimension $(m+1)^n$, so the natural feature map to represent these polynomials will have dimension at least exponential in n. This is a manifestation of the "curse of dimensionality". The restriction to Lipschitz functions, as it turns out, is not enough for the linear learning paradigm to be effective.

Example 8.1. Fix any $b\colon \{0,1\}^n \to \{-1,1\}$, and define the function $g\colon [0,1]^n \to \mathbb{R}$ by

$$g(x) := \min\{b(v) + 2\|x - v\| \mid v \in \{0,1\}^n\}.$$

The function g is a 2-Lipschitz extension of b to $[0,1]^n$ (McShane, 1934). That is, g is 2-Lipschitz, and $g(x) = b(x)$ for all $x \in \{0,1\}^n$. Using this observation, it is easy to show the following proposition.

Proposition 8.6. *Let $\mathcal{F} := \{f\colon [0,1]^n \to \mathbb{R} \mid f$ is 2-Lipschitz with respect to $\|\cdot\|\}$. For every learning algorithm A and every T, there is a probability distribution P on $[0,1]^n \times \{-1,1\}$ such that:*

- $\inf_{f \in \mathcal{F}} \mathsf{R}_{\ell_{\text{hinge}}}(f; P) = 0$;
- *if $\hat{\mathbf{f}}$ is the output of A on input $((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_T, \mathbf{y}_T)) \sim P^T$, then*

$$\mathrm{E}[\mathsf{R}_{\ell_{\text{hinge}}}(\hat{\mathbf{f}}; P)] \geq 1 - \frac{T}{2^n}.$$

Thus, all methods are subject to the curse of dimensionality when faced with the entire class of 2-Lipschitz functions; the difficulty is not restricted to the linear learning paradigm or even computationally-efficient algorithms. ◊

8.3.3 Approximation of Lipschitz ridge functions by polynomials

To avoid the curse of dimensionality, we may make additional structural assumptions about the functions we seek to approximate. One possible structural assumption is that the function is inherently low-dimensional. This is the idea behind *ridge functions*, which are functions $g\colon \mathbb{R}^n \to \mathbb{R}$ of the form

$$g(x) = \sigma(\langle x, v \rangle), \quad \sigma\colon \mathbb{R} \to \mathbb{R}, \quad v \in \mathbb{R}^n.$$

Here, σ is called the *profile function* or *link function*; the ridge function g only depends on x through its projection to the span of v. In statistics, regression models based on ridge functions are called *single-index models*, where v is called the *index vector*.

Let $B^n := \{x \in \mathbb{R}^n \mid \|x\| \leq 1\}$ be the unit ball in $\mathbb{R}^n$, and suppose $g(x) = \sigma(\langle x, v \rangle)$ is a ridge function, with $v \in B^n$ and $\sigma\colon [-1,1] \to \mathbb{R}$ being λ-Lipschitz. If the vector v were known, then it would suffice to approximate σ with a low-degree polynomial on the compact domain $[-1,1]$ (since $\langle x, v \rangle \in [-1,1]$ for $x, v \in B^n$ by Cauchy-Schwarz), which can be done without any curse of dimensionality (Theorem 8.3). Do low-degree (multivariate) polynomials still suffice if v is unknown? We see that the answer is yes in Theorem 8.5.

Exercise 8.3. Fix $\lambda > 0$, and consider the profile function

$$\sigma(z) := \begin{cases} +1 & \text{if } z > 1/\lambda \\ \lambda z & \text{if } -1/\lambda \leq z \leq 1/\lambda \\ -1 & \text{if } z < -1/\lambda \end{cases}$$

and define $g(x) := \sigma(\langle x, v\rangle)$ for some unit vector v. Verify that σ is λ-Lipschitz, and show that

$$\frac{1}{2}\mathsf{R}_{\ell_{\text{hinge}}}(g; P) \le \mathsf{err}_\theta(\langle \cdot, v\rangle)$$

for $\theta := 1/\lambda$. (This shows that the assumption of a linear function with small margin error rate implies the existence of a Lipschitz ridge function with small hinge loss risk.) $\lozenge$

Theorem 8.5. *Fix $m, n \in \mathbb{N}$. There is a feature map $\phi\colon B^n \to \mathbb{R}^d$ for $d := \sum_{k=0}^m n^k$, whose components are n-variate monomials of degree at most m, with the following properties:*

1. *$\|\phi(x)\| \le \sqrt{m+1}$ for all $x \in B^n$.*
2. *For all $g\colon B^n \to \mathbb{R}$ of the form $g(x) = \sigma(\langle x, v\rangle)$ where $\sigma\colon [-1,1] \to \mathbb{R}$ is λ-Lipschitz and $v \in B^n$, there exists $w \in \mathbb{R}^d$ with $\|w\| \le \|\sigma\|_\infty 2^m$ satisfying*

$$|g(x) - \langle \phi(x), w\rangle| \le \frac{\lambda}{\sqrt{m}} \quad \forall x \in B^n.$$

(Above, $\|\sigma\|_\infty := \max_{z\in[-1,1]}|\sigma(z)|$.)

To prove Theorem 8.5, we need to strengthen Theorem 8.3 by additionally bounding the size of the polynomial coefficients.

Lemma 8.1. *Fix any $m \in \mathbb{N}$. For any λ-Lipschitz $\sigma\colon [-1,1] \to \mathbb{R}$, there is a univariate polynomial*

$$p(z) = a_0 + a_1 z + a_2 z^2 + \cdots + a_m z^m$$

of degree at most m satisfying:

1. *$|\sigma(z) - p(z)| \le \frac{\lambda}{\sqrt{m}}$ for all $z \in [-1,1]$;*
2. *$\sum_{k=0}^m |a_k| \le \|\sigma\|_\infty 2^m$.*

Proof. The construction of $p(z)$ uses the linear combination of Bernstein polynomials

$$p(z) := \sum_{k=0}^m \sigma\left(\frac{2k}{m} - 1\right) b_{m,k}\left(\frac{1+z}{2}\right)$$

as in the proof of Theorem 8.3; the changes are just to accommodate the domain $[-1,1]$ instead of $[0,1]$. The proof of the first item (i.e., the error bound) is the same as in Theorem 8.3.

Now we prove the second item. For a polynomial $q(z)$ written in the monomial basis, $q(z) = b_0 + b_1 z + b_2 z^2 + \cdots$, define $\|q(z)\|_{\mathrm{m}} := \sum_{k\ge 0}|b_k|$. We show $\|p(z)\|_{\mathrm{m}} \le \|\sigma\|_\infty 2^m$ for the polynomial $p(z)$ constructed above.

It can be checked that $\|\cdot\|_{\mathrm{m}}$ is a norm on the space of univariate polynomials. We will crucially need its homogeneity and triangle inequality properties. We additionally show that $\|\cdot\|_{\mathrm{m}}$ is sub-multiplicative. For any two polynomials $q(z) = b_0 + b_1 z + b_2 z^2 + \cdots + b_m z^m$ and $r(z) = c_0 + c_1 z + c_2 z^2 + \cdots + c_m z^m$, observe that

$$q(z)r(z) = \sum_{k=0}^{2m}\left(\sum_{i+j=k} b_i c_j\right) z^k.$$

By the triangle inequality,

$$\|q(z)r(z)\|_{\mathrm{m}} = \sum_{k=0}^{2m}\left|\sum_{i+j=k} b_i c_j\right| \le \sum_{k=0}^{2m}\sum_{i+j=k}|b_i c_j| = \left(\sum_{i=0}^{m}|b_i|\right)\left(\sum_{j=0}^{m}|c_j|\right) = \|q(z)\|_{\mathrm{m}}\|r(z)\|_{\mathrm{m}}.$$

This shows that $\|\cdot\|_{\mathrm{m}}$ is sub-multiplicative.

We now bound the norm of $p(z)$ constructed above. Using the homogeneity and the sub-multiplicative properties of $\|\cdot\|_{\mathrm{m}}$, we have

$$\begin{aligned}
\left\|b_{m,k}\left(\frac{1+z}{2}\right)\right\|_{\mathrm{m}} &= \left\|\binom{m}{k}\left(\frac{1+z}{2}\right)^k\left(\frac{1-z}{2}\right)^{m-k}\right\|_{\mathrm{m}} \\
&= \binom{m}{k}\left\|\left(\frac{1+z}{2}\right)^k\left(\frac{1-z}{2}\right)^{m-k}\right\|_{\mathrm{m}} \\
&\le \binom{m}{k}\left\|\frac{1+z}{2}\right\|_{\mathrm{m}}^k\left\|\frac{1-z}{2}\right\|_{\mathrm{m}}^{m-k} \\
&= \binom{m}{k}.
\end{aligned}$$

By the triangle inequality and homogeneity,

$$\begin{aligned}
\|p(z)\|_{\mathrm{m}} = \left\|\sum_{k=0}^{m}\sigma\left(\frac{2k}{m}-1\right)b_{m,k}\left(\frac{1+z}{2}\right)\right\|_{\mathrm{m}} &\le \sum_{k=0}^{m}\left|\sigma\left(\frac{2k}{m}-1\right)\right|\left\|b_{m,k}\left(\frac{1+z}{2}\right)\right\|_{\mathrm{m}} \\
&\le \|\sigma\|_\infty \sum_{k=0}^{m}\binom{m}{k} \\
&= \|\sigma\|_\infty 2^m.
\end{aligned}$$

□

Before giving the proof of Theorem 8.5, we need to define the *k-fold Kronecker power* of a vector. For $x \in \mathbb{R}^n$ and $k \in \mathbb{N}$, the k-fold Kronecker power of x, written $x^{\otimes k} \in \mathbb{R}^{n^k}$, is the k-fold Kronecker product of x with itself. The components of $x^{\otimes k}$ are indexed by $[n]^k$: the component corresponding to $(i_1, \ldots, i_k) \in [n]^k$ is $\prod_{j=1}^k x_{i_j}$. For example, $x^{\otimes 2}$ is xx^T reshaped into a vector in $\mathbb{R}^{n^2}$. Observe that for any $x, y \in \mathbb{R}^n$ and $k \in \mathbb{N}$, we have

$$\begin{aligned}
\langle x^{\otimes k}, y^{\otimes k}\rangle &= \sum_{(i_1,\ldots,i_k)\in[n]^k}\prod_{j=1}^{k} x_{i_j}y_{i_j} \\
&= \left(\sum_{i=1}^{n} x_i y_i\right)^k \\
&= \langle x, y\rangle^k.
\end{aligned}$$

In particular, $\|x^{\otimes k}\| = \|x\|^k$. For simplicity, we write $x^{\otimes 0} = 1$.

Proof of Theorem 8.5. Define $\phi(x) := (x^{\otimes 0}, x^{\otimes 1}, x^{\otimes 2}, \ldots, x^{\otimes m}) \in \mathbb{R}^d$ for $d = \sum_{k=0}^m n^k$. For any $x \in B^n$,

$$\|\phi(x)\|^2 = \sum_{k=0}^{m}\|x^{\otimes k}\|^2 = \sum_{k=0}^{m}\|x\|^{2k} \le m+1.$$

Now consider $g(x) = \sigma(\langle x, v\rangle)$ for λ-Lipschitz $\sigma\colon [-1,1] \to \mathbb{R}$ and $v \in B^n$. Apply Lemma 8.1 to obtain a polynomial $p(z) = \sum_{k=0}^m a_k z^k$ with $\sum_{k=0}^m |a_k| \le \|\sigma\|_\infty 2^m$ and

$$|\sigma(z) - p(z)| \le \frac{\lambda}{\sqrt{m}} \quad \forall z \in [-1,1].$$

Define $w := (a_0 v^{\otimes 0}, a_1 v^{\otimes 1}, a_2 v^{\otimes 2}, \dots, a_m v^{\otimes m}) \in \mathbb{R}^d$. Then, for any $x \in B^n$,

$$\langle \phi(x), w\rangle = \sum_{k=0}^m \langle x^{\otimes k}, a_k v^{\otimes k}\rangle = \sum_{k=0}^m a_k \langle x, v\rangle^k = p(\langle x, v\rangle).$$

By Cauchy-Schwarz, we have $\langle x, v\rangle \in [-1,1]$ for all $x \in B^n$, and therefore

$$|g(x) - \langle \phi(x), w\rangle| = |\sigma(\langle x, v\rangle) - p(\langle x, v\rangle)| \le \frac{\lambda}{\sqrt{m}} \quad \forall x \in B^n.$$

Finally,

$$\|w\|^2 = \sum_{k=0}^m a_k^2 \|v^{\otimes k}\|^2 = \sum_{k=0}^m a_k^2 \|v\|^{2k} \le \sum_{k=0}^m a_k^2 \le \left(\sum_{k=0}^m |a_k|\right)^2 \le (\|\sigma\|_\infty 2^m)^2. \qquad \square$$

Now we can combine Theorem 8.5 and Corollary 8.1 to give a linear learning guarantee.

Corollary 8.3. *Fix any $m \in \mathbb{N}$, $\lambda \ge 0$, $\sigma_{\max} \ge 0$. Let P be a probability distribution over $B^n \times \{-1,1\}$, and let $((\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_T, \mathbf{y}_T)) \sim P^T$. Let the feature map $\phi\colon B^n \to \mathbb{R}^d$ for $d := \sum_{k=0}^m n^k$ be given by $\phi(x) := (x^{\otimes 0}, x^{\otimes 1}, x^{\otimes 2}, \dots, x^{\otimes m})$. The output $\bar{\mathbf{w}}$ of SGD (Algorithm 8.1) with $L_{\mathbf{z}_t}(w) := \ell_{\text{hinge}}(\mathbf{y}_t \langle \phi(\mathbf{x}_t), w\rangle)$ for all $t \in [T]$ and hyperparameter*

$$\eta := \frac{\sigma_{\max} 2^m}{\sqrt{(m+1)T}}$$

satisfies

$$\mathrm{E}[\mathsf{err}(\mathrm{sign}(\langle \phi(\cdot), \bar{\mathbf{w}}\rangle); P)] \le \mathrm{E}\big[\mathsf{R}_{\ell_{\text{hinge}}}(\langle \phi(\cdot), \bar{\mathbf{w}}\rangle; P)\big] \le \min_g \mathsf{R}_{\ell_{\text{hinge}}}(g; P) + \frac{\lambda}{\sqrt{m}} + \frac{\sigma_{\max} 2^m \sqrt{m+1}}{\sqrt{T}}$$

where the minimization is taken over ridge functions $g\colon B^n \to \mathbb{R}$ of the form $g(x) := \sigma(\langle x, v\rangle)$ where $\sigma\colon [-1,1] \to \mathbb{R}$ is λ-Lipschitz with $\|\sigma\|_\infty \le \sigma_{\max}$ and $v \in B^n$.

In order for the last two terms in the bound from Corollary 8.3 to be at most ϵ, we should use $m := 4(\lambda/\epsilon)^2$, and require a sample size

$$T \ge 4\left(\frac{\sigma_{\max}}{\epsilon}\right)^2 \left(4\left(\frac{\lambda}{\epsilon}\right)^2 + 1\right) 2^{8(\lambda/\epsilon)^2}.$$

Although the sample complexity has an exponential dependence on $(\lambda/\epsilon)^2$, the redeeming quality of this result is the lack of any explicit dependence on the dimension n of the input space. The time complexity is $O(Td) = O(T(n+1)^{4(\lambda/\epsilon)^2})$, which is at most a polynomial function of n.

8.4 Limitations of linear learning

8.4.1 PAC learning by linear learning

For which concept classes is linear learning a viable approach for efficient PAC learning? For simplicity, let us only consider finite concept classes $\mathcal{C} \subseteq \{-1,1\}^{\mathcal{X}}$ defined on finite input domains $\mathcal{X}$; let $D := |\mathcal{X}|$ and $N := |\mathcal{C}|$. For efficient PAC learning, it suffices to consider $\mathcal{C}$-realizable distributions P and an efficient implementation of the Consistent Hypothesis algorithm or a Weak Learner. In linear learning, the hypothesis class $\mathcal{H}$ used with these methods is the composition of a (homogeneous) linear threshold function $\mathrm{sign}(\langle \cdot, w \rangle) \in \mathsf{HLTF}_{\mathbb{R}^d}$ and a feature map $\phi\colon \mathcal{X} \to \mathbb{R}^d$. The feature map ϕ should be designed so that, for each concept $c \in \mathcal{C}$, there is a weight vector $w \in \mathbb{R}^d$ such that $c(x) = \mathrm{sign}(\langle \phi(x), w \rangle)$ for all $x \in \mathcal{X}$. This motivates the following definition: we say the pair (ϕ, ψ) is a *d-dimensional representation for $\mathcal{C}$* if $\phi\colon \mathcal{X} \to \mathbb{R}^d$ and $\psi\colon \mathcal{C} \to \mathbb{R}^d$ satisfy

$$c(x)\langle \phi(x), \psi(c) \rangle \geq 1 \quad \forall (x,c) \in \mathcal{X} \times \mathcal{C}.$$

When $\mathcal{C}$ has a d-dimensional representation (ϕ, ψ) (and ϕ can be evaluated in $\mathrm{poly}(d)$ time), the Consistent Hypothesis can be implemented by linear programming (Example 6.1); its running time is $\mathrm{poly}(d, T)$, where $T = \mathrm{poly}(d, 1/\epsilon, 1/\delta)$ is the sample complexity for (ϵ, δ)-PAC learning. Therefore, whether or not this is efficient depends on the dimension d of the representation. Ideally, we would like $d = \mathrm{poly}(\log(D), \log(N))$, where $\log_2(D)$ and $\log_2(N)$ are, respectively, the representation sizes of inputs from $\mathcal{X}$ and concepts from $\mathcal{C}$.

We further say (ϕ, ψ) is a *(W,G)-representation for $\mathcal{C}$* if $\max_{c \in \mathcal{C}} \|\psi(c)\| \leq W$ and $\max_{x \in \mathcal{X}} \|\phi(x)\| \leq G$. The product WG appears in the sample complexity of SGD with hinge loss (Corollary 8.2), so if $\mathcal{C}$ has a d-dimensional representation that is also a (W,G)-representation with $\max\{d, WG\} = \mathrm{poly}(\log(D), \log(N))$ (and ϕ can be evaluated in $\mathrm{poly}(d)$ time), then even SGD can be used for efficient PAC learning of $\mathcal{C}$.

It turns out every concept class has a trivial d-dimensional representation, but the dimension d may be very large, and it may only be a (W,G)-representation with very large product WG. So our goal will be to understand (upper- and lower-) bounds on the achievable d and WG for any given concept class.

8.4.2 Some simple representations

As a baseline, we first consider a trivial representation available for every concept class.

Proposition 8.7. *Every concept class $\mathcal{C} \subseteq \{-1,1\}^{\mathcal{X}}$ with $D := |\mathcal{X}| < \infty$ has a $(\sqrt{D}, 1)$-representation in $\mathbb{R}^D$.*

Proof. Consider any enumeration of the inputs: $\mathcal{X} = \{x^{(1)}, \ldots, x^{(D)}\}$. Set $\phi(x^{(i)}) := e_i$ (the i-th coordinate basis vector), and for every $c \in \mathcal{C}$,

$$\psi(c) := \sum_{j=1}^{D} c(x^{(j)}) e_j.$$

We have $\|\phi(x^{(i)})\| = 1$, and $\|\psi(c)\| = \sqrt{D}$. The weight vector $\psi(c)$ simply enumerates the value of c on all inputs, and its inner product with $\phi(x^{(i)})$ simply looks up the entry corresponding to $x^{(i)}$:

$$\langle \phi(x^{(i)}), \psi(c) \rangle = \sum_{j=1}^{D} \langle e_i, c(x^{(j)}) e_j \rangle = c(x^{(i)}). \qquad \square$$

For some concept classes, we can do far better than the trivial representation.

Example 8.2. Consider the concept class of monotone conjunctions MCONJ_n on $\{0,1\}^n$ (so $D = 2^n$). Recall that $\mathsf{MCONJ}_n := \{c_S \mid S \subseteq [n]\}$, where for $x = (x_1, \ldots, x_n) \in \{0,1\}^n$,

$$c_S(x) := \begin{cases} 1 & \text{if } x_i = 1 \text{ for all } i \in S, \\ -1 & \text{otherwise.} \end{cases}$$

For each $x = (x_1, \ldots, x_n) \in \{0,1\}^n$ and $S \subseteq [n]$, set

$$\phi(x) := (2x_1 - 1, \ldots, 2x_n - 1, 1) \in \mathbb{R}^{n+1},$$

and

$$\psi(c_S) := \sum_{i \in S} e_i - (|S| - 1)e_{n+1}.$$

We have $\|\phi(x)\| = \sqrt{n+1}$, and $\|\psi(c_S)\| = \sqrt{|S| + (|S|-1)^2} \leq n$. For every $x \in \{0,1\}^n$ such that $c_S(x) = 1$, we have

$$\begin{aligned} \langle \phi(x_1, \ldots, x_n), \psi(c_S) \rangle &= \sum_{i \in S} (2x_i - 1) - (|S| - 1) \\ &= |S| - (|S| - 1) = 1. \end{aligned}$$

And for every $x \in \{0,1\}^n$ such that $c_S(x) = -1$, we have

$$\begin{aligned} \langle \phi(x_1, \ldots, x_n), \psi(c_S) \rangle &= \sum_{i \in S} (2x_i - 1) - (|S| - 1) \\ &\leq (|S| - 2) - (|S| - 1) = -1. \end{aligned}$$

So MCONJ_n has a $(n, \sqrt{n+1})$-representation in $\mathbb{R}^{n+1}$. ◊

8.4.3 Lower bounds

We first relate the linear learnability to some standard properties of the concept class.

Proposition 8.8. *If $\mathcal{C}$ has a d-dimensional representation, then $\mathsf{VCdim}(\mathcal{C}) \leq d + 1$.*

Exercise 8.4. Prove Proposition 8.8 ◊

Proposition 8.9. *If $\mathcal{C}$ has a (W, G)-representation, then $\mathsf{Ldim}(\mathcal{C}) \leq (WG)^2$.*

Proof. Let (ϕ, ψ) be a (W, G)-representation for $\mathcal{C}$. Then for all $x \in \mathcal{X}$ and $c \in \mathcal{C}$, we have

$$c(x)\langle \phi(x), \psi(c) \rangle \geq 1.$$

We show that $\mathcal{C}$ can be learned in the online concept learning scenario with mistake bound $(WG)^2$. This follows from the performance guarantee for Perceptron (Algorithm 3.2); in particular, we appeal to the remark following Theorem 3.5. Since $\|\phi(x)\| \leq G$ for all $x \in \mathcal{X}$ and $\|\psi(c)\| \leq W$ for all $c \in \mathcal{C}$, the conditions of the homogeneous γ-margin condition with norm bounds $X := G$ and W, and margin $\gamma := 1$ are satisfied. So running Perceptron on any sequence of inputs $x_1, x_2, \ldots$ (transformed into vectors by ϕ) with labels $y_1, y_2, \ldots$ that are consistent with some $c \in \mathcal{C}$ guarantees a mistake bound of $(XW)^2/\gamma^2 = (WG)^2$. By Theorem 3.6 and the definition of Ldim, it follows that $(WG)^2 \geq \mathsf{Ldim}(\mathcal{C})$. □

Example 8.3. Let $\mathcal{X} := [D]$ and $\mathcal{C} := \{c_j \mid j \in \{0, 1, \ldots, D\}\}$, where $c_j(x) = 1$ iff $x > j$. Then $\mathsf{Ldim}(\mathcal{C}) = \log_2(D)$ but $\mathsf{VCdim}(\mathcal{C}) = 1$. It is possible efficiently PAC learn $\mathcal{C}$ with sample complexity not depending on D at all, but $(WG)^2 \geq \log_2(D)$, so the sample complexity guarantee for SGD has an explicit dependence on D. $\lozenge$

The lower bound of $\log(D)$ on $(WG)^2$ is nowhere close to the $(WG)^2$-value achieved by the trivial representation. In fact, since $\mathsf{Ldim}(\mathcal{C}) \leq \log_2|\mathcal{C}|$, Proposition 8.9 cannot be used to prove any lower bound on $(WG)^2$ higher than $\log_2|\mathcal{C}| = \log_2(N)$.

8.4.4 Parity functions

We now give an example of a concept class that is efficiently PAC-learnable, but for which every (W, G)-representation has high dimension and high $(WG)^2$. The concept class is that of *parity functions* PARITY_n on $\{0,1\}^n$,

$$\mathsf{PARITY}_n := \{\chi_S \mid S \subseteq [n]\}$$

where, for any $S \subseteq [n]$ and $x = (x_1, \ldots, x_n) \in \{0,1\}^n$,

$$\chi_S(x) := \begin{cases} 1 & \text{if } \sum_{i \in S} x_i \text{ is even,} \\ -1 & \text{if } \sum_{i \in S} x_i \text{ is odd.} \end{cases}$$

In fact, $\chi_S(x)$ can be thought of as a linear function of x over the finite field $\mathbb{F}_2$ (i.e., integers modulo two). Regard $x = (x_1, \ldots, x_n)$ as an n-vector in $\mathbb{F}_2^n$, and associate χ_S with an n-vector $a = (a_1, \ldots, a_n) \in \mathbb{F}_2^n$ where $a_i = 1$ iff $i \in S$. Then

$$\chi_S(x) = 1 \iff x_1 a_1 + \cdots + x_n a_n = 0.$$

To find $a \in \mathbb{F}_2^n$ consistent with several labeled examples $(x, \chi_S(x))$, we just need to solve a system of linear equations (over $\mathbb{F}_2$), which can be done efficiently with Gaussian Elimination. The sample complexity is $O(\log_2(|\mathsf{PARITY}_n|)) = O(n)$ (suppressing the usual dependence on $1/\epsilon$ and $1/\delta$). So PARITY_n is efficiently PAC learnable.

Theorem 8.6 (Forster, 2002)**.** *Every d-dimensional representation for PARITY_n has dimension $d \geq 2^{n/2}$, and every (W, G)-representation for PARITY_n has $WG \geq 2^{n/2}$.*

We just prove the second part of Theorem 8.6, i.e., the lower bound on WG. The key idea is to study the sign matrix that represents PARITY_n, i.e., the $2^n \times 2^n$ matrix H_n whose rows are indexed by $x \in \{0,1\}^n$ and columns are indexed by $S \subseteq [n]$, with

$$H_n(x, S) := \chi_S(x).$$

The matrix H_n is called a *Hadamard matrix*. In the case $n = 1$, we have the $2^1 \times 2^1$ matrix H_1:

$$H_1 := \begin{matrix} \\ 0 \\ 1 \end{matrix} \begin{matrix} \begin{matrix} \emptyset & \{1\} \end{matrix} \\ \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix} \end{matrix}.$$

For $n \geq 2$, we can partition the $x \in \{0,1\}^n$ by whether $x_n = 0$ or $x_n = 1$, and then partition the $S \subseteq [n]$ by whether $n \notin S$ or $n \in S$. This gives us the $2^n \times 2^n$ matrix H_n in block form:

$$H_n := \begin{matrix} \\ x_n = 0 \\ x_n = 1 \end{matrix} \begin{matrix} \begin{matrix} S \not\ni n & S \ni n \end{matrix} \\ \left[\begin{array}{c|c} +H_{n-1} & +H_{n-1} \\ \hline +H_{n-1} & -H_{n-1} \end{array}\right] \end{matrix}.$$

Indeed, if $n \notin S$, then $\chi_S(x)$ ignores x_n, and the behavior is the same as in $n-1$ dimensions. If $n \in S$ and $x_n = 0$, then $\chi_S(x) = \chi_{S\setminus\{n\}}(x)$ (and $S \setminus \{n\} \subseteq [n-1]$); if $n \in S$ and $x_n = 1$, then $\chi_S(x) = -\chi_{S\setminus\{n\}}(x)$.

The matrix H_n is clearly symmetric. It turns out H_n also has orthogonal columns, and by symmetry, orthogonal rows. This is clearly true for $n = 1$:

$$H_1^\intercal H_1 = \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix} \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}.$$

For $n \geq 2$,

$$\begin{aligned} H_n^\intercal H_n &= \left[\begin{array}{c|c} +H_{n-1} & +H_{n-1} \\ \hline +H_{n-1} & -H_{n-1} \end{array}\right] \left[\begin{array}{c|c} +H_{n-1} & +H_{n-1} \\ \hline +H_{n-1} & -H_{n-1} \end{array}\right] \\ &= \left[\begin{array}{c|c} 2H_{n-1}^2 & H_{n-1}^2 - H_{n-1}^2 \\ \hline H_{n-1}^2 - H_{n-1}^2 & 2H_{n-1}^2 \end{array}\right] \\ &= \left[\begin{array}{c|c} 2^n I_{n-1} & 0 \\ \hline 0 & 2^n I_{n-1} \end{array}\right] = 2^n I_n, \end{aligned}$$

where we use $H_{n-1}^2 = H_{n-1}^\intercal H_{n-1} = 2^{n-1} I_{n-1}$ by induction.

Proof of second part of Theorem 8.6. Let $(\mathbf{x}, \mathbf{S}) \sim \text{Uniform}(\{0,1\}^n) \otimes \text{Uniform}(2^{[n]})$. Consider a (W, G)-representation (ϕ, ψ) for PARITY_n, so

$$\chi_{\mathbf{S}}(\mathbf{x}) \langle \phi(\mathbf{x}), \psi(\chi_{\mathbf{S}}) \rangle \geq 1.$$

Taking expectations, we obtain

$$\mathrm{E}[\chi_{\mathbf{S}}(\mathbf{x}) \langle \phi(\mathbf{x}), \psi(\chi_{\mathbf{S}}) \rangle] \geq 1.$$

$$\begin{aligned} \mathrm{E}[\chi_{\mathbf{S}}(\mathbf{x}) \langle \phi(\mathbf{x}), \psi(\chi_{\mathbf{S}}) \rangle] &= \mathrm{E}[\langle \chi_{\mathbf{S}}(\mathbf{x}) \phi(\mathbf{x}), \psi(\chi_{\mathbf{S}}) \rangle] \\ &= \mathrm{E}[\langle \mathrm{E}[\chi_{\mathbf{S}}(\mathbf{x}) \phi(\mathbf{x}) \mid \mathbf{S}], \psi(\chi_{\mathbf{S}}) \rangle] && \text{(tower property)} \\ &\leq \mathrm{E}[\|\mathrm{E}[\chi_{\mathbf{S}}(\mathbf{x}) \phi(\mathbf{x}) \mid \mathbf{S}]\| \|\psi(\chi_{\mathbf{S}})\|] && \text{(Cauchy-Schwarz inequality)} \\ &\leq W \, \mathrm{E}[\|\mathrm{E}[\chi_{\mathbf{S}}(\mathbf{x}) \phi(\mathbf{x}) \mid \mathbf{S}]\|] \\ &\leq W \sqrt{\mathrm{E}[\|\mathrm{E}[\chi_{\mathbf{S}}(\mathbf{x}) \phi(\mathbf{x}) \mid \mathbf{S}]\|^2]} && \text{(Jensen's inequality).} \end{aligned}$$

We now show that

$$\mathrm{E}\big[\|\mathrm{E}[\chi_{\mathbf{S}}(\mathbf{x}) \phi(\mathbf{x}) \mid \mathbf{S}]\|^2\big] = 2^{-n} \, \mathrm{E}\big[\|\phi(\mathbf{x})\|^2\big] \leq 2^{-n} G^2, \tag{8.3}$$

which suffices to complete the proof since it allows us to conclude

$$1 \leq W \sqrt{2^{-n} G^2} = 2^{-n/2} W G.$$

Now to prove (8.3), we can regard $\|\mathrm{E}[\chi_{\mathbf{S}}(\mathbf{x}) \phi(\mathbf{x}) \mid \mathbf{S}]\|^2$ as

$$\|\mathrm{E}[\chi_{\mathbf{S}}(\mathbf{x}) \phi(\mathbf{x}) \mid \mathbf{S}]\|^2 = \langle \mathrm{E}[\chi_{\mathbf{S}}(\mathbf{x}) \phi(\mathbf{x}) \mid \mathbf{S}], \mathrm{E}\big[\chi_{\mathbf{S}}(\mathbf{x}') \phi(\mathbf{x}') \,\big|\, \mathbf{S}\big] \rangle = \mathrm{E}\big[\langle \chi_{\mathbf{S}}(\mathbf{x}) \phi(\mathbf{x}), \chi_{\mathbf{S}}(\mathbf{x}') \phi(\mathbf{x}') \rangle \,\big|\, \mathbf{S}\big].$$

where $\mathbf{x}'$ is an independent copy of $\mathbf{x}$ (and also independent of $\mathbf{S}$). Therefore

$$\begin{aligned}
\mathrm{E}\big[\|\mathrm{E}[\chi_{\mathbf{S}}(\mathbf{x})\phi(\mathbf{x}) \mid \mathbf{S}]\|^2\big] &= \mathrm{E}\big[\langle \chi_{\mathbf{S}}(\mathbf{x})\phi(\mathbf{x}), \chi_{\mathbf{S}}(\mathbf{x}')\phi(\mathbf{x}')\rangle\big] \\
&= \Pr\big(\mathbf{x} = \mathbf{x}'\big)\,\mathrm{E}\big[\langle \chi_{\mathbf{S}}(\mathbf{x})\phi(\mathbf{x}), \chi_{\mathbf{S}}(\mathbf{x}')\phi(\mathbf{x}')\rangle \;\big|\; \mathbf{x} = \mathbf{x}'\big] \\
&\qquad + \Pr\big(\mathbf{x} \neq \mathbf{x}'\big)\,\mathrm{E}\big[\langle \chi_{\mathbf{S}}(\mathbf{x})\phi(\mathbf{x}), \chi_{\mathbf{S}}(\mathbf{x}')\phi(\mathbf{x}')\rangle \;\big|\; \mathbf{x} \neq \mathbf{x}'\big] \\
&= 2^{-n}\,\mathrm{E}\big[\|\phi(\mathbf{x})\|^2\big] \\
&\qquad + (1 - 2^{-n})\,\mathrm{E}\big[\mathrm{E}\big[\chi_{\mathbf{S}}(\mathbf{x})\chi_{\mathbf{S}}(\mathbf{x}') \;\big|\; \mathbf{x}, \mathbf{x}'; \mathbf{x} \neq \mathbf{x}'\big]\langle \phi(\mathbf{x}), \phi(\mathbf{x}')\rangle \;\big|\; \mathbf{x} \neq \mathbf{x}'\big] \\
&= 2^{-n}\,\mathrm{E}\big[\|\phi(\mathbf{x})\|^2\big]
\end{aligned}$$

since $\mathrm{E}[\chi_{\mathbf{S}}(\mathbf{x})\chi_{\mathbf{S}}(\mathbf{x}') \mid \mathbf{x}, \mathbf{x}'; \mathbf{x} \neq \mathbf{x}'] = 2^{-n}\sum_{S\subseteq[n]} \chi_S(\mathbf{x})\chi_S(\mathbf{x}') = 0$ by the orthogonality of the (distinct) rows of H_n corresponding to $\mathbf{x}$ and $\mathbf{x}'$. □

Bibliography

Noga Alon. On the density of sets of vectors. *Discrete Mathematics*, 46(2):199–202, 1983.

Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Jānis M Barzdin and Rūsiņš V Freivald. On the prediction of general recursive functions. *Soviet Math. Doklady*, 13:1224–1228, 1972.

Serge Bernstein. Démonstration du théorème de Weierstrass fondée sur le calcul des probabilités. *Communications de la Société Mathématique de Kharkow*, 13(1):1–2, 1912.

Thomas M Cover. Behavior of sequential predictors of binary sequences. *Transactions on Prague Conference on Information Theory Statistical Decision Functions, Random Processes*, pages 263–272, 1965.

Jürgen Forster. A linear lower bound on the unbounded error probabilistic communication complexity. *Journal of Computer and System Sciences*, 65(4):612–625, 2002.

Peter Frankl. On the trace of finite sets. *Journal of Combinatorial Theory, Series A*, 34(1):41–45, 1983.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

Yoav Freund and Robert E Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999.

Steve Hanneke. The optimal sample complexity of PAC learning. *Journal of Machine Learning Research*, 17(38):1–15, 2016.

David Haussler, Nick Littlestone, and Manfred K Warmuth. Predicting {0, 1}-functions on randomly drawn points. *Information and Computation*, 115(2):248–292, 1994.

David P Helmbold and Manfred K Warmuth. On weak learning. *Journal of Computer and System Sciences*, 50(3):551–573, 1995.

David S Johnson and Franco P Preparata. The densest hemisphere problem. *Theoretical Computer Science*, 6(1):93–107, 1978.

Adam Tauman Kalai and Rocco A Servedio. Boosting in the presence of noise. *Journal of Computer and System Sciences*, 71(3):266–290, 2005.

Richard M Karp. Reducibility among combinatorial problems. In R.E. Miller and J.W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

Michael Kearns and Leslie Valiant. Cryptographic limitations on learning boolean formulae and finite automata. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, pages 433–444, 1989.

Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.

Nick Littlestone. From on-line to batch learning. In *Second Annual Workshop on Computational Learning Theory*, pages 269–284, 1989.

Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

Wolfgang Maass and György Turán. How fast can a threshold gate learn? In Stephen José Hanson, George A Drastal, and Ronald L Rivest, editors, *Computational Learning Theory and Natural Learning System: Constraints and Prospects*, pages 381–414. MIT Press, 1994.

Yishay Mansour and David McAllester. Boosting using branching programs. *Journal of Computer and System Sciences*, 64(1):103–112, 2002.

Edward James McShane. Extension of range of functions. *Bulletin of the American Mathematical Society*, 40(12):837–842, 1934.

Marvin Minsky and Seymour Papert. *Perceptrons: An introduction to computational geometry*. MIT Press, 1969.

Saburo Muroga. *Threshold logic and its applications*. Wiley-Interscience, 1971.

Albert Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, volume 12, pages 615–622, 1962.

Gilles Pisier. Remarques sur un résultat non publié de b. maurey. *Séminaire Analyse fonctionnelle*, pages 1–12, 1981.

Leonard Pitt and Leslie G Valiant. Computational limitations on learning from examples. *Journal of the ACM (JACM)*, 35(4):965–984, 1988.

Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386, 1958.

Norbert Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13(1): 145–147, 1972.

Robert E Schapire. The strength of weak learnability. *Machine Learning*, 5:197–227, 1990.

Robert E Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, 1998.

Saharon Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41(1):247–261, 1972.

Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Vladimir N Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1985.

Vladimir N Vapnik and Alexey Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16:264–280, 1971.

John von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320, 1928.

Tong Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *The Annals of Statistics*, 32(1):56–85, 2004.

Yurii A Zuev. Asymptotics of the logarithm of the number of boolean threshold functions. *Dokl. Akad. Nauk SSSR*, 306:528–530, 1989.