Statistical Foundations of Learning[1]

Debarghya Ghoshdastidar
TUM Informatik
WS 2019/20

Updated on: October 15, 2019

[1]These notes should be used as guidelines to the contents covered in the lectures. Relevant materials in the textbooks should also be studied. The notes may contain errors. Please verify the mathematical details.

Contents

Course details

0.1 Focus of this course

The success of machine learning has been unprecedented in the past few years. It is natural to wonder why popular machine learning algorithms exhibit good performance on a wide range of problems. This course will take a foundational perspective on learning, and will describe the mathematical principles that are useful for theoretically analysing the performance of machine learning algorithms.

The first part of the course will focus on statistical learning theory that provides the foundation for a systematic study of supervised learning. We will cover fundamental topics such as the Vapnik-Chervonenkis (VC) theory and the Probably-Approximately-Correct (PAC) framework. The theories provide two perspective for analysing the goodness of supervised learning algorithms. We will introduce the notion of empirical risk minimisation and different loss functions. We will also cover concepts of algorithmic stability, regularisation and boosting as well as analysis of nearest neighbour classification and support vector machines, If time permits, online learning will be covered.

The second part of the course will briefly cover the theoretical foundations of unsupervised learning, Unlike the supervised setting, there is a lack of unified theory in the case of unsupervised learning. Most of the lecture will focus on clustering for which we will study axiomatic, probabilistic, information theoretic and approximation based approaches to quantify the goodness of clustering algorithms as well the solvability of clustering problems. We will extend the discussion to hierarchical clustering and graph clustering, and cover some theory for random graphs. If time permits, dimension reduction will be covered.

0.1.1 Lecture notes and reference

These lecture notes should be used as guidelines to the contents covered in the lectures. For the first part, there are excellent textbooks. The textbook of Shalev-Shwartz and Ben-

David [2018] is highly recommended and is freely available online.[1] Most of the contents in the lecture notes are aligned with this textbook. Additionally, some ideas have taken from courses on learning theory offered by Shivani Agarwal, Ruth Urner, Ilya Tolstikhin and Ambuj Tewari at different institutes.[2]

The second part has very few coherent material. We will refer to some parts of Shalev-Shwartz and Ben-David [2018] and Blum et al. [2020].[3] However, most of the remaining content will be on research papers that will be mentioned in corresponding chapters.

0.2 Course logistics

Course number/name: IN2378 (Statistical Foundations of Learning)
SWS: 3V + 1Ü
Credits: 5

0.2.1 Registration

Register for the course on TUM Online. We will use Moodle for announcements, discussions and assignments.

0.2.2 Meetings

Lectures will be held at TUM Informatik **MI Hörsaal 2** (Room: 00.04.011) on:

- **Tuesday 16:00-18:00**
- **Friday 14:00-16:00**

The teaching will be in the form of lectures and discussions, and techniques of inverted classroom will be used. Lectures will be taught on board, and sometimes, the lecture notes will be presented.

The bi-weekly schedule will be as follows. The first and every alternate week after that will have two 90-min lectures (on both days of the week). Starting from the second week, tentatively every alternate week will have a 90-minute lecture (on Tuesday) and a 90-minute tutorial session (on Fridays 25.10, 08.11, 22.11, 06.12, 20.12, 10.01, 24.01, 07.02).

[1] Online copy: https://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf

[2] The lecture notes of these courses are also available online. However, the approaches in the courses may differ, and could be confusing to read all of them in parallel.

[3] Online link for Blum's book: https://www.cs.cornell.edu/jeh/book.pdf

Every 90-min lecture will consist of two parts:

- In the second part, new concepts are introduced, key theorems are stated and their significance is discussed. Main proof techniques will be pointed out so that the students can study/derive the mathematical details at home.
- The first part could last between 0 to 60 min, where the mathematical details from the previous lecture will be discussed.

Every bi-weekly 90-min tutorial will discuss solutions to the bi-weekly homework, and also answer further difficulties in the material from the preceding week.

0.3 Exam and assignment

There will be a 90-minute written examination at the end of the semester, possibly on:

- **February 17, 2020 (Monday) 8:00 – 9:30**

Venue for the exam and schedule for the repeat exam will be announced later.

In addition, there will be bi-weekly assessments in the form of homework. The final grades will be based on the final written examination. However, exceptional homework scores (about top 10%) will lead to 0,3 additional note in the final exam.

The bi-weekly assignment can be submitted **individually or in teams of two**. Submissions must be made through Moodle in form of a single PDF file. In case of team submissions, only one submission should be made containing the names of both members. **You must use the same team for all the subsequent assignments.** If a team member drops out, the other member cannot form another group.

Late submission: Each team is eligible for a total of maximum 5 late days throughout the semester, and maximum 2 late days per assignment. If these limits are exceeded, late submissions will be ignored.

0.4 Further discussions

There will not be further office hours to discuss problems related to the course. If there is problem related to understanding the scientific content, everyone is advised to raise the question on the discussion forum on the course's Moodle page. Everyone is also encourage to answer / attempt to answer questions raised by others. If there is any other problem, talk to the lecturer during or at the end of the regular meetings.

Part I

Statistical Learning Theory

Chapter 1

Vapnik Chervonenkis (VC) Theory

In this chapter, we introduce a formal framework for supervised learning that encompasses classification, regression and related machine learning problems. We then study the notion of generalisation, that is, understanding how well can we generalise the prediction task from training samples to unforeseen test data. Our discussions in this chapter mainly focus on a part of learning theory known as the theory of generalisation or Vapnik-Chervonenkis (VC) theory, introduced by Vladimir Vapnik and Alexey Chervonenkis [Vapnik and Chervonenkis, 1971, Vapnik, 2013]. The material in this chapter aligns with Chapters 2, 4, 6 and partially 9 of Shalev-Shwartz and Ben-David [2018].

1.1 Framework of supervised learning

We first introduce our formal model of supervised learning. There is a set $\mathcal{X}$ that includes all instances that we would like to classify or more generally, predict. We often call $\mathcal{X}$ as the *domain* or *feature space*. For instance, $\mathcal{X}$ could be the set containing details of all participants in this course. There is also a set $\mathcal{Y}$, called the *label set*, that contains all possible outcomes of our prediction task. For instance, we could try to predict whether you like or hate football. This would be an example of binary classification with $\mathcal{Y} = \{0, 1\}$ or $\{-1, +1\}$. We could also try to predict the total duration of football that you have played in your lifetime, which would correspond to a regression task with $\mathcal{Y} = [0, \infty)$ or $\mathbb{R}$. In many cases, the data is represented by p-dimensional real vectors, that is, $\mathcal{X} \subseteq \mathbb{R}^p$. For most of this course, we will focus on the case of $\mathcal{X} = \mathbb{R}^p$ and $\mathcal{Y} = \{0, 1\}$.

The goal of machine learning is to produce a *predictor* $h : \mathcal{X} \to \mathcal{Y}$. In supervised learning,

we find such a predictor based on a *training sample*

$$S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\},$$

where $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ is an ordered pair of training data and corresponding label. We will denote the training sample size as m. A *learner* or *learning algorithm* $\mathcal{A}$ takes the training sample S as input, and outputs a predictor h. Formally, we write the learner as a function

$$\mathcal{A} : \bigcup_{i=1}^{\infty} (\mathcal{X} \times \mathcal{Y})^i \to \mathcal{Y}^{\mathcal{X}},$$

where $\mathcal{Y}^{\mathcal{X}}$ is the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$.[1]

1.1.1 Loss and risk

The key property of a good predictor is that it should have low error on *test data*, that is, instances whose labels were not observed at the time training / learning. We quantify the goodness of the predictor in terms of its *risk*, also called the *generalisation error* or the *expected loss*. To formalise the notion of risk, we define a loss function

$$\ell : \mathcal{Y} \times \mathcal{Y} \to [0, \infty),$$

that is, for any $x \in \mathcal{X}$ with true label/value $y \in \mathcal{Y}$ and for predictor h, we compare $h(x)$ and y and report a non-negative loss $\ell(h(x), y)$ that accounts for any deviation of the predicted value $h(x)$ from the true value y.[2]

For binary classification, the typical loss function is the 0-1 loss, $\ell(h(x), y) = \mathbf{1}\{h(x) \neq y\}$, where $\mathbf{1}\{\cdot\}$ is the indicator function. The same can also be extended to multi-class classification, but more general loss functions are also used in this setting. In regression, the most common loss function is the squared loss $\ell(h(x), y) = (h(x) - y)^2$.

We also require a probabilistic view of learning to define risk. For this, we assume that $\mathcal{D}$ is a probability distribution on $\mathcal{X} \times \mathcal{Y}$.

Definition 1.1 (Risk / generalisation error). *The* risk *of a predictor h is defined as*

$$L_{\mathcal{D}}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(h(x), y)], \tag{1.1}$$

which is the expected loss that the predictor h would incur when it has to predict for a test instance *sampled according to $\mathcal{D}$.*

[1] For most of the course, we will not refer to any randomised algorithm. Hence, given the training sample S, the output $\mathcal{A}(S)$ is fixed.

[2] In some texts, a loss function is defined as a function on $\ell : \mathcal{Y}^{\mathcal{X}} \times \mathcal{X} \times \mathcal{Y} \to [0, \infty)$, that is, it takes three inputs (h, x, y). This characterisation is same as ours, with h and x being written as two separate inputs instead of writing ℓ as a function $h(x)$.

In particular, for the 0-1 loss, one can verify that the risk is the probability of predicting an incorrect label.[3]

For this course, we will assume that the training samples $(x_1, y_1), \ldots, (x_m, y_m)$ are independent and identical distributed (iid) according to $\mathcal{D}$. We will denote this as $S \sim \mathcal{D}^m$. Both the iid assumption and the fact that the training and test data are sampled from the same distribution are natural in most scenarios.[4]

Example 1 (Bayes classifier). Observe that $\mathcal{D}$ is a joint distribution on $\mathcal{X} \times \mathcal{Y}$, and we can decompose it into $\mathcal{D}_{\mathcal{X}}$, the marginal distribution of $\mathcal{D}$ over $\mathcal{X}$, and $\mathbb{P}_{\mathcal{Y}|\mathcal{X}}(y|x)$, the conditional probability of observing the label y when the data instance is x. We restrict to binary classification, $\mathcal{Y} = \{0, 1\}$, and define $\eta(x) = \mathbb{P}_{\mathcal{Y}|\mathcal{X}}(y = 1|x)$. Note that the risk can be written as

$$L_{\mathcal{D}}(h) = \mathbb{E}_{\mathcal{D}_{\mathcal{X}}}\left[\mathbb{E}_{\mathcal{Y}|\mathcal{X}}[\mathbf{1}\{h(x) \neq y\} \,|x]\right] = \mathbb{E}_{\mathcal{D}_{\mathcal{X}}}\left[\mathbb{P}_{\mathcal{Y}|\mathcal{X}}(h(x) \neq y \,|x)\right].$$

Exercise 1.1. Write $\mathbb{P}_{\mathcal{Y}|\mathcal{X}}(h(x) \neq y \,|x)$ in terms of $\eta(x)$. Verify that that the risk is minimised by the classifier

$$h(x) = \begin{cases} 1 & \text{if } \eta(x) \geq 0.5 \\ 0 & \text{if } \eta(x) < 0.5 \end{cases}$$

Hint: Note that $h(x)$ is a deterministic function given x, and hence, $\mathbb{P}(h(x) = 0, y = 1 \,|x) = \mathbb{P}(y = 1|x)\mathbf{1}\{h(x) = 0\}$.

This is known as the *Bayes classifier*. More generally, the *Bayes decision rule* is given by

$$h(x) = \arg\max_{y \in \mathcal{Y}} \mathbb{P}_{\mathcal{Y}|\mathcal{X}}(y|x).$$

The Bayes decision rule results in the minimum possible risk for a given distribution $\mathcal{D}$, known as the *Bayes risk*,

$$L^*_{\mathcal{D}} = \min_{h \in \mathcal{Y}^{\mathcal{X}}} L_{\mathcal{D}}(h). \tag{1.2}$$

1.1.2 Empirical risk minimisation (ERM)

In context of the previous discussion, a 'good' predictor can be obtained if we find h which has a small risk (hopefully, the least among all possible h). Unfortunately, we cannot compute $L_{\mathcal{D}}(h)$ since we do not have access to $\mathcal{D}$, except for the training data $S \sim \mathcal{D}^m$.

[3] The general convention is to use capital letters to denote random variables $(X, Y) \sim \mathcal{D}$, while an instance is denoted by small letters (x, y). This is not strictly followed in machine learning, and we will mostly use small letters to denote both random variables and instances.

[4] Many practical problems violate one of the above two assumptions. There exist theoretical results when these assumptions, which we will not study in this course.

Hence, we cannot directly minimise the risk $L_{\mathcal{D}}(h)$ over all h. To overcome this limitation, we instead minimise the empirical risk defined as

$$L_S(h) = \frac{1}{m}\sum_{i=1}^{m} \ell(h(x_i), y_i). \tag{1.3}$$

Exercise 1.2. Verify that the empirical risk of h is an unbiased estimated of its true risk, that is, $\mathbb{E}_S\left[L_S(h)\right] = L_{\mathcal{D}}(h)$. Does it hold if S is not iid? Under what conditions on S or h, will the equality fail?

Note that minimisation of $L_S(h)$ over all $h \in \mathcal{Y}^{\mathcal{X}}$ could be computationally intractable since the set of all functions $\mathcal{Y}^{\mathcal{X}}$ is infinite in most practical cases. In this chapter, we ignore this computational aspect and focus only on the statistical part, that is, we assume that we have finite training sample S but an infinite computation power.

1.1.3 Hypothesis class

One of the main questions that we will study is the following:

> *Assume that our learner $\mathcal{A}$ (for example, ERM) provides a predictor $\widehat{h} = \mathcal{A}(S)$. What can we say about $L_{\mathcal{D}}(\widehat{h})$?*

Typically, we are interested in showing that a learning algorithm is 'good' (that is, $L_{\mathcal{D}}(\widehat{h})$ is small) and hence, we focus on deriving an upper bound on $L_{\mathcal{D}}(\widehat{h})$.[5] The above question can be asked in several ways. For instance, since we know that the ERM predictor has a small empirical risk, one could ask the following question:

> *Let $\widehat{h} = \mathcal{A}(S)$. How large can $L_{\mathcal{D}}(\widehat{h}) - L_S(\widehat{h})$ be?*[6]

This question is the topic of the present chapter. We now show that the ERM, as described earlier, can result in a high difference between true and empirical risk.

Example 2 (A naive approach to ERM). Consider a binary classification problem. Let S be the training sample. Consider the following classifier

$$h_S(x) = \begin{cases} 1 & \text{if } (x,1) \in S, \\ 0 & \text{otherwise (if } (x,0) \in S \text{ and for every } x \text{ we have not observed in } S). \end{cases}$$

It follows that $L_S(h_S) = 0$ for every S. However, if $\mathcal{D}$ is such that $\eta(x) = \mathbb{P}_{\mathcal{Y}|\mathcal{X}}(y = 1\,|x)$ has infinite support, then $L_{\mathcal{D}}(h_S) > 0$ with probability 1 for every S (since h_S labels at most m instances as 1, and is incorrect on many infinitely many instances).

[5] We will later see a negative result, called no free-lunch theorem, which states no learner can be good for all possible data. To prove this theorem, we will derive a lower bound for $L_{\mathcal{D}}(\widehat{h})$.

[6] In some texts, the difference between true and empirical risk is called *generalisation error*. However, we will use the term generalisation error for $L_{\mathcal{D}}(\widehat{h})$ and not the difference.

In particular, consider the simplified setting where $\mathcal{X} = \mathbb{R}$ and $\eta(x) = 1$ for all x. In this case $h_S(x) = 1$ for all x observed in the training sample, and incorrectly labels all other x as 0. Hence, $L_{\mathcal{D}}(h_S) = 1$ while $L_S(h_S) = 0$.
Exercise 1.3. Does the above example violate the relation $\mathbb{E}_S[L_S(h)] = L_{\mathcal{D}}(h)$? Why?

The above example shows that ERM does not generalise well if the predictors are allowed to be arbitrarily complex (that is, if we optimise over $\mathcal{Y}^{\mathcal{X}}$). One can avoid this issue by restricting the minimisation over some set $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, called a *hypothesis class*. Henceforth, we will only consider the following ERM learner:

$$\text{ERM:} \qquad \widehat{h} = \arg\min_{h \in \mathcal{H}} \; L_S(h) \tag{1.4}$$

which is restricted to the hypothesis class $\mathcal{H}$. The above learner is called *ERM with inductive bias* since the resulting predictor is biased towards certain types of functions.

Here are some simple, yet common, hypothesis / function classes.
Example 3 (Decision stumps). Let $\mathcal{X} = \mathbb{R}$ and $\mathcal{Y} = \{0, 1\}$. A *decision stump* is a one-level decision tree of the following form. For some $t \in \mathcal{X}$ and $b \in \{0, 1\}$,

$$h_{t,b}(x) = \begin{cases} b & \text{if } x \leq t, \\ 1 - b & \text{if } x > t. \end{cases}$$

The set $\mathcal{H} = \big\{h_{t,b} : t \in \mathbb{R}, b \in \{0, 1\}\big\} \subset \{0, 1\}^{\mathbb{R}}$ is the hypothesis class of decision stumps. We will later see that ERM for this class is efficiently solvable.
Example 4 (Linear classifiers). For binary classification in $\mathbb{R}^p$, one of the popular classes in the set of linear classifiers

$$\mathcal{H} = \big\{h_{w,b} = \text{sign}(\langle w, x\rangle + b) \;:\; w \in \mathbb{R}^p, b \in \mathbb{R}\big\}.$$

Here, $\langle w, x\rangle = w^T x$. We define the label set as $\mathcal{Y} = \{-1, +1\}$ for convenience.

In the next sections, we will see that for 'nice' hypothesis classes, one can derive upper bounds on the deviation $L_{\mathcal{D}}(\widehat{h}) - L_S(\widehat{h})$. Before going into this discussion, we conclude this section with some remarks on the effect of $\mathcal{H}$ on the generalisation error $L_{\mathcal{D}}(h)$.

1.1.4 Underfitting and overfitting

Recall that the Bayes risk is the least possible risk that can be achieved by any predictor, and in fact, obtained by the Bayes decision rule. Hence, it is natural to compare the performance of a learned predictor with the Bayes risk $L^*_{\mathcal{D}} = \min_{h \in \mathcal{Y}^{\mathcal{X}}} L_{\mathcal{D}}(h)$. In particular, one may ask:

Let $\widehat{h} = \mathcal{A}(S)$. How large can $L_{\mathcal{D}}(\widehat{h}) - L^*_{\mathcal{D}}$ be?

Consider the ERM learner (1.4). Then it is obvious that $\widehat{h}$ can only be as good as the best possible predictor in the class $\mathcal{H}$, that is, $L_{\mathcal{D}}(\widehat{h}) \geq \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h)$. We may write

$$L_{\mathcal{D}}(\widehat{h}) - L^*_{\mathcal{D}} = \underbrace{L_{\mathcal{D}}(\widehat{h}) - \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h)}_{\text{estimation error}} + \underbrace{\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) - L^*_{\mathcal{D}}}_{\text{approximation error}}.$$

The *estimation error* quantifies how bad is the output of the learner with respect to the best possible predictor in the class, whereas *approximation error* quantifies the error induced because we restricted the search to the class $\mathcal{H}$. In Example 2, we eliminated the approximation error by minimising over $\mathcal{Y}^{\mathcal{X}}$, but in this case, the ERM predictor could still have a high estimation error. On the other hand, we may consider a trivial $\mathcal{H} = \{h\}$ containing a single predictor, which will lead to zero estimation error, but possible a high approximation error (unless h is the Bayes rule).

Thus the size of $\mathcal{H}$ often controls both the approximation and estimation errors. A small $\mathcal{H}$, a simpler hypothesis class, is easier to estimate but may provide poor approximation of the optimal decision. This leads to the notion of *underfitting* — given training samples, the predicted model cannot achieve a low training error, and hence, may lead to poor test error as well. A larger $\mathcal{H}$ provides better approximation, but can be difficult to estimate correctly. In other words, we may fit the training data too well, but the predictor is not able to generalise well to unseen test data. This trade-off is also known as the *bias-variance tradeoff*, which we will discuss at a later stage.

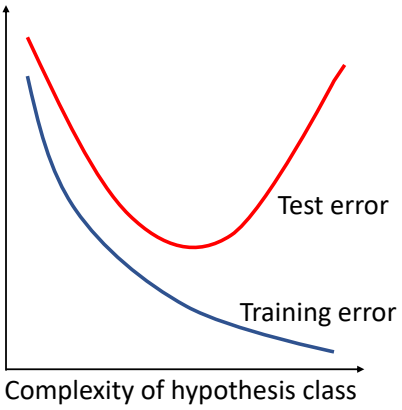

Figure 1.1: Typical behaviour of training error (empirical risk) and test error (generalisation error) for varying model complexity

1.2 Uniform convergence

We now study a way to derive bounds on the difference $L_{\mathcal{D}}(\widehat{h}) - L_S(\widehat{h})$. We restrict the discussion to classification, or more precisely, the 0-1 loss function. However, the technique can be easily used in a more general setting of bounded loss functions, that is, when $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, B]$ for some $B < \infty$. We need the following concentration inequality.

Theorem 1.2 (Hoeffding's inequality). *Let $X_1, \ldots, X_n$ be independent random variables such that $X_i \in [a_i, b_i]$ almost surely. For any $\epsilon > 0$, the following statements hold:*

$$\mathbb{P}\left(\sum_{i=1}^n X_i - \sum_{i=1}^n \mathbb{E}[X_i] > \epsilon\right) \leq \exp\left(-\frac{2\epsilon^2}{\sum_i (b_i - a_i)^2}\right),$$

$$\mathbb{P}\left(\sum_{i=1}^n X_i - \sum_{i=1}^n \mathbb{E}[X_i] < -\epsilon\right) \leq \exp\left(-\frac{2\epsilon^2}{\sum_i (b_i - a_i)^2}\right),$$

and

$$\mathbb{P}\left(\left|\sum_{i=1}^n X_i - \sum_{i=1}^n \mathbb{E}[X_i]\right| > \epsilon\right) \leq 2\exp\left(-\frac{2\epsilon^2}{\sum_i (b_i - a_i)^2}\right),$$

Proof. See Wikipedia pages for Hoeffding's inequality and Hoeffding's lemma. □

Exercise 1.4. Let $h \in \{-1, 1\}^{\mathcal{X}}$ be a fixed hypothesis and $S \sim \mathcal{D}^m$ be a training sample. Show that

$$\mathbb{P}_S\left(\left|L_{\mathcal{D}}(h) - L_S(h)\right| > \epsilon\right) \leq 2\exp\left(-2m\epsilon^2\right).$$

Let $\widehat{h} = \mathcal{A}(S)$ for some learner $\mathcal{A}$. The above statement does **not** hold for $\widehat{h}$ (except in some trivial cases). Why?

The above exercise states that we can bound the generalisation error for a fixed h using Hoeffding's inequality, but it does not immediately provide a bound on $L_{\mathcal{D}}(\widehat{h})$. To obtain such a bound, we note that if $\widehat{h} \in \mathcal{H}$, then

$$\left|L_{\mathcal{D}}(\widehat{h}) - L_S(\widehat{h})\right| \leq \sup_{h \in \mathcal{H}} \left|L_{\mathcal{D}}(h) - L_S(h)\right|,$$

which leads to the conclusion

$$\mathbb{P}_S\left(\left|L_{\mathcal{D}}(\widehat{h}) - L_S(\widehat{h})\right| > \epsilon\right) \leq \mathbb{P}_S\left(\sup_{h \in \mathcal{H}} \left|L_{\mathcal{D}}(h) - L_S(h)\right| > \epsilon\right). \tag{1.5}$$

Instead of bounding only the difference for $\widehat{h}$, we derive a bound that holds uniformly for all $h \in \mathcal{H}$.

1.2.1 Generalisation error bound for finite hypothesis class

We first conisder the simple case where $\mathcal{H}$ contains finitely many hypotheses, that is, $\mathcal{H} = \{h_1, h_2, \ldots, h_{|\mathcal{H}|}\}$ with $|\mathcal{H}|$ being the cardinality of class $\mathcal{H}$. We can bound the probability in (1.5) using the following result, which follows from the union bound for probabilities.

Exercise 1.5. Recall the union bound (see Boole's inequality on Wikipedia). Use this to prove the following. Let $X_1, \ldots, X_n$ be random variables (could also be dependent). Then

$$\mathbb{P}\left(\max_{1\le i\le n} X_i > \epsilon\right) \le \sum_{i=1}^{n} \mathbb{P}(X_i > \epsilon)$$

We now bound (1.5) as

$$\begin{aligned}\mathbb{P}_S\left(\left|L_{\mathcal{D}}(\widehat{h}) - L_S(\widehat{h})\right| > \epsilon\right) &\le \mathbb{P}_S\left(\sup_{h\in\mathcal{H}}\left|L_{\mathcal{D}}(h) - L_S(h)\right| > \epsilon\right) \\ &\le 2|\mathcal{H}|\exp\left(-2m\epsilon^2\right), \end{aligned} \tag{1.6}$$

where the last step uses both the above union bound, and the inequality for each h.

One can state the bound in (1.6) in the following way, which provides an explicit bound on the generalisation error. Given $\delta \in (0,1)$. For $\widehat{h} \in \mathcal{H}$, with probability $1-\delta$,

$$L_S(\widehat{h}) - \sqrt{\frac{\ln(|\mathcal{H}|) + \ln\left(\frac{2}{\delta}\right)}{2m}} < L_{\mathcal{D}}(\widehat{h}) < L_S(\widehat{h}) + \sqrt{\frac{\ln(|\mathcal{H}|) + \ln\left(\frac{2}{\delta}\right)}{2m}} \tag{1.7}$$

Here, ln denotes the natural logarithm. Typically, we are interested in the upper bound on $L_{\mathcal{D}}(\widehat{h})$. Also, we focus less on the constants or the term $\ln(\frac{2}{\delta})$, and are interested in the dependence on size of $\mathcal{H}$ (here, logarithmic) and the training sample size m.

1.2.2 Uniform convergence for infinite hypothesis class

Most hypothesis classes have infinite cardinality (for instance, linear classifiers, decision stumps). Obviously, the error bound (1.7) has no meaning in such cases. We will see that the bound can be improved so that $\mathcal{H}$ is replaced by the notion of *growth function* defined below. Note that the following discussion is restricted to binary classification, $\mathcal{Y} = \{-1, +1\}$.

Consider a sequence $C = (x_1, \ldots, x_m) \in \mathcal{X}^m$. We define the *restriction* of a hypothesis class $\mathcal{H} \subseteq \{-1,+1\}^{\mathcal{X}}$ to C as

$$\mathcal{H}_{|C} = \big\{(h(x_1), \ldots, h(x_m)) \ : \ h \in \mathcal{H}\big\},$$

which is the set of all possible labelling of the m data points in C.

Definition 1.3 (Growth function). *For binary function classes $\mathcal{H} \subseteq \{-1,+1\}^{\mathcal{X}}$, the growth function $\tau_{\mathcal{H}}(m) : \mathbb{N} \to \mathbb{N}$ is given by*

$$\tau_{\mathcal{H}}(m) = \max_{C \subseteq \mathcal{X}: |C|=m} \left|\mathcal{H}_{|C}\right|.$$

It is the maximum number of possible binary labelling for any m instances in $\mathcal{X}$.

Exercise 1.6. Verify that $\tau_{\mathcal{H}}(m) \leq \min\{|\mathcal{H}|, 2^m\}$.

We now state the main result of this section, which is an equivalent for (1.6)–(1.7) for infinite hypothesis classes.

Theorem 1.4 (Uniform convergence for infinite $\mathcal{H}$). *If $m \geq \frac{2\ln 4}{\epsilon^2}$ and $\mathcal{H} \subseteq \{-1,+1\}^{\mathcal{X}}$ has a growth function $\tau_{\mathcal{H}}(\cdot)$, then we have*

$$\mathbb{P}_S\left(\sup_{h\in\mathcal{H}} \left|L_{\mathcal{D}}(h) - L_S(h)\right| > \epsilon\right) \leq 4\tau_{\mathcal{H}}(2m)\exp\left(-m\epsilon^2/8\right),$$

Hence, for any $\delta \in (0,1)$, the generalisation error of the a learned predictor $\widehat{h} = \mathcal{A}(S) \in \mathcal{H}$ satisfies

$$L_{\mathcal{D}}(\widehat{h}) < L_S(\widehat{h}) + \sqrt{\frac{8\left(\ln\left(\tau_{\mathcal{H}}(2m)\right) + \ln\left(\frac{4}{\delta}\right)\right)}{m}} \qquad \textit{with probability } 1-\delta.$$

Note that the above bound holds for any learner, and not necessarily ERM.

Exercise 1.7. The second statement in the theorem does not impose any assumption on m. Verify that the condition $m\epsilon^2 > 2\ln 4$ holds for the second statement.

Before proving the above theorem, we make some remarks about the proof, highlighting the reasons that allow us to replace $|\mathcal{H}|$ by the growth function (ignoring different constant factors, this is the only difference between the bounds in (1.6) and Theorem 1.4). The following discussion are informal (even partly incorrect), and may make more sense after going through the proof of the theorem. The proof of (1.6) has two parts:

- union bound gives $\mathbb{P}_S(\sup_h |L_{\mathcal{D}}(h) - L_S(h)| > \epsilon) \leq |\mathcal{H}| \sup_h \mathbb{P}_S(|L_{\mathcal{D}}(h) - L_S(h)| > \epsilon)$
- Hoeffding's inequality provides bound on $\mathbb{P}_S(|L_{\mathcal{D}}(h) - L_S(h)| > \epsilon)$

Hoeffding's inequality will be used at the last stage of the proof of Theorem 1.4. However, it is obvious that union bound, as stated above, cannot be used for infinite $\mathcal{H}$. To work around this, the proof requires the following steps:

- **Symmetrisation by introducing independent copy of S**:
  Observe that for any h and any training set S (of size m), $L_S(h)$ can take only finitely many values (verify this). One may partition $\mathcal{H}$ into groups, where the empirical risk

is same for all h in a group. Hence, for every S, when we through the lens of L_S, there are much less than $|\mathcal{H}|$ different hypotheses. However, this is not the case for $|L_{\mathcal{D}}(h) - L_S(h)|$ since $L_{\mathcal{D}}(\cdot)$, being an expectation, may take infinitely many values. Hence, it is easier if we get rid of $L_{\mathcal{D}}(\cdot)$.

This is achieved through symmetrisation. Let S' be another training sample of size m, independent of S. Small $|L_{\mathcal{D}}(h) - L_S(h)|$ would mean that $|L_S(h) - L_{S'}(h)|$ is also small. Formally, we use symmetrisation to modify the problem into bounding the probability $\mathbb{P}_{S,S'}\left(\sup_h |L_S(h) - L_{S'}(h)| > \epsilon'\right)$ for some ϵ'. To control this, we can now exploit the fact that $|L_S(\cdot) - L_{S'}(\cdot)|$ takes only finitely many values, and hence, our previous informal argument based on partitioning $\mathcal{H}$ would be possible.

- **Rademacher symmetrisation (swapping permutations)**:
  There still remains a major hurdle in the application of union bound — it can be applied only when the events under consideration are fixed. For instance, given a h, the event $\left\{|L_{\mathcal{D}}(h) - L_S(h)| > \epsilon\right\}$ is fixed and only its occurrence is random. On the other hand, when we partition $\mathcal{H}$ based on the values of $|L_S(\cdot) - L_{S'}(\cdot)|$, then the partition itself is random (depends on (S, S')). To work around this, we use the technique of Rademacher symmetrisation, which introduces Rademacher random variables, that is, random variables taking values $\{-1, +1\}$ each with probability $\frac{1}{2}$.

  We postpone the mathematical details to the next section, but provide an intuitive explanation (that may seem quite different from the mathematical statements). Observe that $S \cup S'$ can be viewed at a set of $2m$ i.i.d. training instances. If we randomly swap the i-th instances of S and S' for every i, and again compute the absolute difference in the two empirical means, then its distribution should not differ. This is because, irrespective of the swapping permutations, the quantity $|L_S(\cdot) - L_{S'}(\cdot)|$ remains as a function of i.i.d. samples. Now, we may fix (condition on) S, S' and only consider the random swapping, in which case we need not consider supremum of all $\mathcal{H}$, but only over $\mathcal{H}_{|S \cup S'}$, a finite quantity.

- **Union bound and Hoeffding's inequality**:
  The above step loosely ensures that we need to care about only $|\mathcal{H}_{|S \cup S'}| \leq \tau_{\mathcal{H}}(2m)$ types of functions. We can now apply union bound resulting in a multiplicative factor of at most $\tau_{\mathcal{H}}(2m)$, and finish the proof with the help of Hoeffding's inequality.

1.2.3 Proof of uniform convergence

We now convert the above discussion into a proof. We begin with following lemma.

Lemma 1.5 (Symmetrisation by introducing independent copy of S). *Let $S, S' \sim$*

$\mathcal{D}^m$ be two independent training sets, each of size m. For $m\epsilon^2 > 2\ln 4$,

$$\mathbb{P}_S\left(\sup_{h\in\mathcal{H}}|L_S(h) - L_{\mathcal{D}}(h)| > \epsilon\right) \leq 2\mathbb{P}_{S,S'}\left(\sup_{h\in\mathcal{H}}|L_S(h) - L_{S'}(h)| > \frac{\epsilon}{2}\right).$$

Proof. For each S, define h_S^{bad} as a function for which $\left|L_S(h_S^{bad}) - L_{\mathcal{D}}(h_S^{bad})\right| > \epsilon$, if such a function exists.[7]

Exercise 1.8. Verify that the following two events are equal

$$\left\{\left|L_S(h_S^{bad}) - L_{\mathcal{D}}(h_S^{bad})\right| > \epsilon\right\} = \left\{\sup_{h\in\mathcal{H}}|L_S(h) - L_{\mathcal{D}}(h)| > \epsilon\right\}. \tag{1.8}$$

We use the fact $A \subset B \implies \mathbb{P}(A) \leq \mathbb{P}(B)$ to write

$$\begin{aligned}
&\mathbb{P}_{S,S'}\left(\sup_{h\in\mathcal{H}}|L_S(h) - L_{S'}(h)| > \frac{\epsilon}{2}\right)\\
&\geq \mathbb{P}_{S,S'}\left(\left|L_S(h_S^{bad}) - L_{S'}(h_S^{bad})\right| > \frac{\epsilon}{2}\right)\\
&\geq \mathbb{P}_{S,S'}\left(\left\{\left|L_S(h_S^{bad}) - L_{\mathcal{D}}(h_S^{bad})\right| > \epsilon\right\}\bigcap\left\{\left|L_{\mathcal{D}}(h_S^{bad}) - L_{S'}(h_S^{bad})\right| \leq \frac{\epsilon}{2}\right\}\right)\\
&= \mathbb{E}_{S,S'}\left[\mathbf{1}\left\{\left|L_S(h_S^{bad}) - L_{\mathcal{D}}(h_S^{bad})\right| > \epsilon\right\}\mathbf{1}\left\{\left|L_{\mathcal{D}}(h_S^{bad}) - L_{S'}(h_S^{bad})\right| \leq \frac{\epsilon}{2}\right\}\right]\\
&= \mathbb{E}_S\left[\mathbf{1}\left\{\left|L_S(h_S^{bad}) - L_{\mathcal{D}}(h_S^{bad})\right| > \epsilon\right\}\mathbb{P}_{S'|S}\left(\left|L_{\mathcal{D}}(h_S^{bad}) - L_{S'}(h_S^{bad})\right| \leq \frac{\epsilon}{2}\right)\right]
\end{aligned}$$

Note that, conditioned on S, the function h_S^{bad} is a deterministic function. So we can apply Hoeffding's inequality (Exercise 1.4) to write

$$\mathbb{P}_{S'|S}\left(\left|L_{\mathcal{D}}(h_S^{bad}) - L_{S'}(h_S^{bad})\right| > \frac{\epsilon}{2}\right) \leq 2\exp\left(-\frac{m\epsilon^2}{2}\right) \leq \frac{1}{2}$$

for $m\epsilon^2 \geq 2\ln 4$. Plugging this into the previous derivation, we have

$$\begin{aligned}
\mathbb{P}_{S,S'}\left(\sup_{h\in\mathcal{H}}|L_S(h) - L_{S'}(h)| > \frac{\epsilon}{2}\right) &\geq \frac{1}{2}\mathbb{E}_S\left[\mathbf{1}\left\{\left|L_S(h_S^{bad}) - L_{\mathcal{D}}(h_S^{bad})\right| > \epsilon\right\}\right]\\
&= \frac{1}{2}\mathbb{P}_S\left(\left|L_S(h_S^{bad}) - L_{\mathcal{D}}(h_S^{bad})\right| > \epsilon\right)\\
&= \frac{1}{2}\mathbb{P}_S\left(\sup_{h\in\mathcal{H}}|L_S(h) - L_{\mathcal{D}}(h)| > \epsilon\right),
\end{aligned}$$

where the last step is due to (1.8). This proves the lemma. □

[7] We need the h_S^{bad} because there may not be any $h \in \mathcal{H}$ for which $|L_S(h) - L_{\mathcal{D}}(h)|$ achieves the supremum value (typical problem with supremum).

Bounding via Rademacher symmetrisation. We define $\sigma = (\sigma_1, \ldots, \sigma_m)$, where $\sigma_1, \ldots, \sigma_m$ are independent Rademacher variables, that is $\mathbb{P}(\sigma_i = +1) = \mathbb{P}(\sigma_i = -1) = \frac{1}{2}$. Let (x_i', y_i') be the i-th instance in S', and define

$$Y_\sigma \equiv Y_\sigma(S, S', h) := \frac{1}{m} \sum_{i=1}^{m} \sigma_i \big(\mathbf{1}\left\{h(x_i) \neq y_i\right\} - \mathbf{1}\left\{h(x_i') \neq y_i'\right\} \big).$$

Observe that $Y_{(1,\ldots,1)} = L_S(h) - L_{S'}(h)$.

Exercise 1.9. Verify that Y_σ has the same distribution for every σ. (Hint: Earlier discussion on Rademacher symmetrisation in terms of swapping permutations may be useful.)

The above fact implies that $\sup_h |Y_\sigma|$ has same distribution for every σ. So we write

$$\begin{aligned}
\mathbb{P}_{S,S'}\left(\sup_{h\in\mathcal{H}} |L_S(h) - L_{S'}(h)| > \frac{\epsilon}{2} \right) &= \mathbb{P}_{S,S'}\left(\sup_{h\in\mathcal{H}} |Y_{(1,\ldots,1)}| > \frac{\epsilon}{2} \right) \\
&= \frac{1}{2^m} \sum_{\sigma\in\{-1,+1\}^m} \mathbb{E}_{S,S'}\left[\mathbf{1}\left\{ \sup_{h\in\mathcal{H}} |Y_\sigma| > \frac{\epsilon}{2} \right\} \right] \\
&= \mathbb{E}_{S,S'}\left[\mathbb{P}_{\sigma|S,S'}\left(\sup_{h\in\mathcal{H}} |Y_\sigma| > \frac{\epsilon}{2} \right) \right]
\end{aligned}$$

We now bound the conditional probability. Since we have conditioned on S, S', we may focus only on $\mathcal{H}_{|S\cup S'}$ (instead of $\mathcal{H}$) since the possible values Y_σ can take for any σ depends only elements of $\mathcal{H}_{|S\cup S'}$. So

$$\begin{aligned}
\mathbb{P}_{\sigma|S,S'}\left(\sup_{h\in\mathcal{H}} |Y_\sigma| > \frac{\epsilon}{2} \right) &= \mathbb{P}_{\sigma|S,S'}\left(\sup_{\mathbf{h}\in\mathcal{H}_{|S\cup S'}} |Y_\sigma| > \frac{\epsilon}{2} \right) \\
&\leq \sum_{\mathbf{h}\in\mathcal{H}_{|S\cup S'}} \mathbb{P}_{\sigma|S,S'}\left(|Y_\sigma| > \frac{\epsilon}{2} \right) \\
&\leq |\mathcal{H}_{|S\cup S'}| \sup_{h\in\mathcal{H}} \mathbb{P}_{\sigma|S,S'}\left(|Y_\sigma| > \frac{\epsilon}{2} \right) \qquad (1.9)
\end{aligned}$$

where the first inequality is due to union bound. Recall the definition of Y_σ and note that, conditioned on S, S', it is the average of m independent zero-mean random variables each taking values in $\{-1, 0, +1\}$. We apply Hoeffding's inequality to bound the conditional probability by $2\exp\left(-\frac{m\epsilon^2}{8}\right)$. Finally recall that $|\mathcal{H}_{|S\cup S'}| \leq \tau_{\mathcal{H}}(2m)$ from the definition of growth function. Substituting these bounds in (1.9) and combining with Lemma 1.5 leads to the claim of Theorem 1.4.

1.3 VC dimension

Recall that the growth function satisfies $\tau_{\mathcal{H}}(2m) \leq 2^{2m}$. Plugging this in Theorem 1.4 immediately leads to the bound

$$L_{\mathcal{D}}(\widehat{h}) \leq L_S(\widehat{h}) + \sqrt{16 \ln 2 + \frac{8 \ln\left(\frac{4}{\delta}\right)}{m}}$$

with probability $1 - \delta$. This is quite weak since trivially $L_{\mathcal{D}}(\widehat{h}) \leq 1$. Thus, the uniform convergence bound of Theorem 1.4 has little significance unless we are able to show that $\tau_{\mathcal{H}}$ grows slowly for some infinite function classes. This is the topic of the present section.

Definition 1.6 (Shattering). *Let $\mathcal{H} \subseteq \{-1, +1\}^{\mathcal{X}}$ and $C = (x_1, \ldots, x_m) \in \mathcal{X}^m$. We say that C is shattered by $\mathcal{H}$ if $|\mathcal{H}_{|C}| = 2^m$.*
In other words, for every possible labelling $s \in \{-1, +1\}^m$ of instances in C, there is a $h_s \in \mathcal{H}$ such that that $h_s(x_i) = s_i$ for $i = 1, \ldots, m$.

Definition 1.7 (VC dimension). *The Vapnik Chervonenkis (VC) dimension of a non-empty $\mathcal{H} \subseteq \{-1, +1\}^{\mathcal{X}}$ is the cardinality of the largest possible subset of $\mathcal{X}$ that can be shattered by $\mathcal{H}$, that is,*

$$\mathrm{VCdim}(\mathcal{H}) = \max\{m \in \mathbb{N} \ : \ \tau_{\mathcal{H}}(m) = 2^m\}.$$

If $\mathcal{H}$ can shatter arbitrarily large sets, then $VCdim(\mathcal{H}) = \infty$.

Exercise 1.10. For non-empty $\mathcal{H}$, show that $\mathrm{VCdim}(\mathcal{H}) = 0$ if and only if $|\mathcal{H}| = 1$.
Hint: Verify that if $|\mathcal{H}| \geq 2$, then there is a point that $\mathcal{H}$ can shatter.

We will see the VC dimension of few function classes later, but we first demonstrate the role of VC dimension in the context of uniform convergence.

1.3.1 Sauer's lemma

For $\mathcal{H}$ with finite VC dimension, the Sauer's lemma provides a bound on the growth function in terms of $\mathrm{VCdim}(\mathcal{H})$.

Theorem 1.8 (Sauer's lemma). *Let $\mathcal{H} \subseteq \{-1, +1\}^{\mathcal{X}}$ be non-empty with $\mathrm{VCdim}(\mathcal{H}) = d < \infty$. For all $m \in \mathbb{N}$,*

$$\tau_{\mathcal{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i}.$$

A simpler bound on growth function is often used, which holds for all $m \geq d \geq 1$,

$$\tau_{\mathcal{H}}(m) \leq \left(\frac{em}{d}\right)^d.$$

Proof. The proof is by induction on m and d.

Base case: Note that there are two base cases: $d = 0, m \geq 1$ and $m = 1, d \geq 1$. For $d = 0, m \geq 1$: From Exercise 1.10, $d = 0 \implies |\mathcal{H}| = 1$ and $\tau_{\mathcal{H}}(m) = 1 = \binom{m}{0}$. For $d \geq 1, m = 1$: Again $d \geq 1 \implies |\mathcal{H}| \geq 2$. Hence, there is $x \in \mathcal{X}$ such that $|\mathcal{H}_{|\{x\}}| = 2$. So $\tau_{\mathcal{H}}(m) = 2 = \binom{m}{0} + \binom{m}{1}$ for $m = 1$.

Induction step: Let $m > 1$ and $d > 0$. We assume that the hypothesis $\tau_{\mathcal{H}}(m') \leq \sum_{i=0}^{d'} \binom{m'}{i}$ holds for all $m' < m$ **or** $d' < d$. Particularly, we assume it holds for $(m-1, d-1)$ and $(m-1, d)$.

Let $C = (x_1, x_2, \ldots, x_m) \in \mathcal{X}^m$ and define $C' = (x_2, \ldots, x_m)$. For every $(y_2, \ldots, y_m) \in \mathcal{H}_{|C'}$ there can be only two possibilities:

- both $(-1, y_2, \ldots, y_m)$ and $(+1, y_2, \ldots, y_m)$ are in $\mathcal{H}_{|C}$
- either $(-1, y_2, \ldots, y_m) \in \mathcal{H}_{|C}$ or $(+1, y_2, \ldots, y_m) \in \mathcal{H}_{|C}$

Let $Y = \big\{(y_2, \ldots, y_m) \in \mathcal{H}_{|C'} \ : \ (-1, y_2, \ldots, y_m), (+1, y_2, \ldots, y_m) \in \mathcal{H}_{|C}\big\}$. Verify that

$$|\mathcal{H}_{|C}| = |\mathcal{H}_{|C'}| + |Y|.$$

Since $\text{VCdim}(\mathcal{H}) = d$, by our induction hypothesis $|\mathcal{H}_{|C'}| \leq \tau_{\mathcal{H}}(m-1) \leq \sum_{i=0}^{d} \binom{m-1}{i}$. On the other hand, we may view Y as a function class $Y \subseteq \{-1, +1\}^{C'}$ and claim that

$$\text{VCdim}(Y) \leq d - 1.$$

The claim trivially holds if $m \leq d$. For $m > d$, this can be proved by contradiction. If $\text{VCdim}(Y) = d$, then there exists $C'' \subset C'$ of cardinality d such that C'' is shattered by Y. By construction, this would imply that $C'' \cup \{x_1\}$ is shattered by $\mathcal{H}_{|C}$, or more generally $\mathcal{H}$. Hence, $\text{VCdim}(\mathcal{H}) \geq d + 1$, which is a contradiction.

By inductive hypothesis, $\text{VCdim}(Y) \leq d - 1 \implies |Y| \leq \sum_{i=0}^{d-1} \binom{m-1}{i}$. Hence,

$$\begin{aligned}
|\mathcal{H}_{|C}| &\leq \sum_{i=0}^{d} \binom{m-1}{i} + \sum_{i=0}^{d-1} \binom{m-1}{i} \\
&= \binom{m}{0} + \sum_{i=1}^{d} \left(\binom{m-1}{i} + \binom{m-1}{i-1} \right) \\
&= \sum_{i=0}^{d} \binom{m}{i}
\end{aligned}$$

since $\binom{m}{i} = \binom{m-1}{i} + \binom{m-1}{i-1}$ (verify). Since the above is true for every $C \in \mathcal{X}^m$, the bound also holds for $\tau_{\mathcal{H}}(m)$.

The simpler bound is derived in the following way

$$\begin{aligned}
\tau_{\mathcal{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i} &\leq \sum_{i=0}^{d} \binom{m}{i} \left(\frac{m}{d}\right)^{d-i} && \text{we assume } m \geq d \\
&= \left(\frac{m}{d}\right)^{d} \sum_{i=0}^{d} \binom{m}{i} \left(\frac{d}{m}\right)^{i} 1^{d-i} \\
&\leq \left(\frac{m}{d}\right)^{d} \left(1 + \frac{d}{m}\right)^{m} \\
&\leq \left(\frac{em}{d}\right)^{d} && \text{since } \left(1 + \frac{x}{n}\right)^{n} \leq e^{x}
\end{aligned}$$

□

Shalev-Shwartz and Ben-David [2018, Section 6.5.1] provides a different proof, where induction happens only on m. Sauer's lemma leads to an useful uniform convergence bound when combined with Theorem 1.4.

Corollary 1.9 (Uniform convergence for finite VC dimension). *Let $\mathcal{H} \subseteq \{-1, +1\}^{\mathcal{X}}$ has $\mathrm{VCdim}(\mathcal{H}) = d < \infty$, then for any $\delta \in (0, 1)$, the generalisation error of any $h \in \mathcal{H}$ satisfies*

$$L_{\mathcal{D}}(h) < L_S(h) + \sqrt{\frac{8\left(d \ln\left(\frac{em}{d}\right) + \ln\left(\frac{4}{\delta}\right)\right)}{m}} \qquad \textit{with probability } 1 - \delta.$$

1.3.2 VC dimension of some function classes

We now compute the VC dimension of some classes. The first few examples (exercises) can be found in Shalev-Shwartz and Ben-David [2018].

Exercise 1.11 (Finite class). Let $\mathcal{H}$ be a finite class. Show that $\mathrm{VCdim}(\mathcal{H}) \leq \log_2(|\mathcal{H}|)$. Hint: This follows from the definition of VC dimension.

For infinite $\mathcal{H}$, the general trick for computing VC dimension is based on the observation that $\mathrm{VCdim}(\mathcal{H}) = d \leq \infty$ if:

- there exists some set $C \in \mathcal{X}^d$ that can be shattered by $\mathcal{H}$
- no set of cardinality $d + 1$ can be shattered by $\mathcal{H}$

Exercise 1.12 (Threshold functions). Let $\mathcal{H} \subseteq \{-1,1\}^{\mathbb{R}}$ be the class of all *threshold functions* of the following form. For some $t \in \mathcal{X}$,

$$h_t(x) = \begin{cases} -1 & \text{if } x \leq t, \\ +1 & \text{if } x > t. \end{cases}$$

Show that $\text{VCdim}(\mathcal{H}) = 1$.

Exercise 1.13 (Decision stumps). Recall the class $\mathcal{H} \subseteq \{-1,1\}^{\mathbb{R}}$ of one-dimensional *decision stumps*, which are of the following form. For some $t \in \mathcal{X}$ and $b \in \{-1,+1\}$,

$$h_{t,b}(x) = \begin{cases} b & \text{if } x \leq t, \\ -b & \text{if } x > t. \end{cases}$$

Show that $\text{VCdim}(\mathcal{H}) = 2$.

Exercise 1.14 (Intervals). Let $\mathcal{H} \subseteq \{-1,1\}^{\mathbb{R}}$ be the class of *intervals* of the following form. For some $a, b \in \mathcal{X}$ with $a < b$,

$$h_{a,b}(x) = \begin{cases} +1 & \text{if } a \leq x \leq b, \\ -1 & \text{otherwise.} \end{cases}$$

Show that $\text{VCdim}(\mathcal{H}) = 2$.

Axis-parallel rectangles in $\mathbb{R}^2$.

These are 2-dimensional generalisation of intervals of the following form. For $a < b$ and $c < d$,

$$h_{a,b,c,d}\left(x^{(1)}, x^{(2)}\right) = \begin{cases} +1 & \text{if } a \leq x^{(1)} \leq b \text{ and } a \leq x^{(1)} \leq b, \\ -1 & \text{otherwise.} \end{cases}$$

The VC dimension of the class of axis-aligned rectangles is 4. To see this, verify that the following four points $\{(1,0),(0,1),(-1,0),(0,-1)\}$ are shattered. For any set C of 5 points,

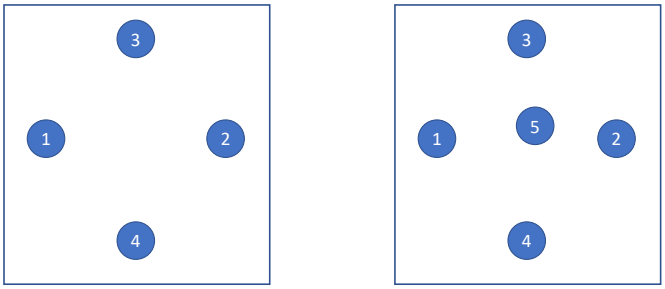

Figure 1.2: (Left) Four points that can be shattered by axis-parallel rectangles; (Right) Typical example of 5 points, where the central point cannot be labelled 0 by axis-parallel rectangles when all outer points are 1.

let $x_1 \in C$ be a left-most point (no point has lower value in first axis), $x_2 \in C \backslash \{x_1\}$ be a right-most point (none of other 3 points have higher value in first axis), $x_3 \in C \backslash \{x_1, x_2\}$ be top-most point and $x_4 \in C \backslash \{x_1, x_2, x_3\}$ be bottom-most point. No function in $\mathcal{H}$ can achieve the labelling $(1, 1, 1, 1, 0)$.

Exercise 1.15 (Axis-parallel bounding boxes in $\mathbb{R}^p$). Extend the definition of axis-parallel rectangles to define the class of axis-parallel bounding boxes in $\mathbb{R}^p$. Use the above discussion to show that the VC dimension of this class is $2p$.

Convex polygons in $\mathbb{R}^2$. Now consider the class $\mathcal{H} \subseteq \{-1, +1\}^{\mathbb{R}^2}$ of binary functions defined by convex polygons with no restriction on the number of edges, that is, for every convex polygon c in $\mathbb{R}^2$, there is $h_c \in \mathcal{H}$ soh that $h_c(x) = 1$ if $x \in c$, and -1 otherwise.

We claim that $\text{VCdim}(\mathcal{H}) = \infty$. To see this, consider any m points on a circle. For a labelling of the points, consider the convex hull c of the points labelled as $+1$. Observe that h_c correctly labels all points. Hence, $\mathcal{H}$ shatters every set of m points on a circle for every m. Thus $\text{VCdim}(\mathcal{H}) = \infty$.

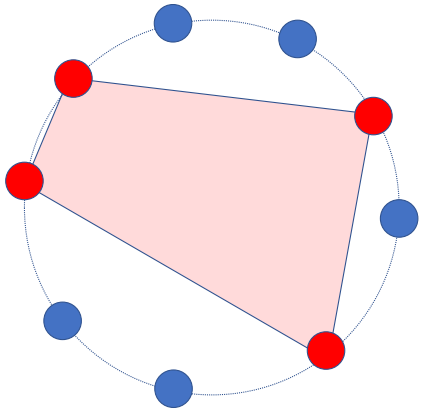

Figure 1.3: Convex polygons with unbounded number of edges can shatter any m points lying on a circle.

Theorem 1.10 (VC dimension of Linear classifiers / Halfspaces). *Recall the function class of halfspaces in $\mathbb{R}^p$*

$$\mathcal{H} = \big\{\text{sign}(\langle w, x\rangle + b) \ : \ w \in \mathbb{R}^p, b \in \mathbb{R}\big\}.$$

$\text{VCdim}(\mathcal{H}) = p + 1$.

Proof. To see that $\mathcal{H}$ shatters some set of $(p+1)$ points, consider the points $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_p, \mathbf{0} \in \mathbb{R}^p$, where $\mathbf{e}_i$ is the i-th standard basis vector. Check that for every labelling $(y_1, \ldots, y_{p+1}) \in \{-1, +1\}^{p+1}$ of the $(p + 1)$ points, we can achieve this labelling with $w = (y_1, \ldots, y_p)$ and $b = \frac{1}{2} y_{p+1}$.

We prove that $\mathcal{H}$ cannot shatter any set of $(p+2)$ points by contradiction. Assume that the points $x_1, x_2, \ldots, x_{p+2} \in \mathbb{R}^p$ can be shattered. Consider the set of $p+1$ linear equations

$$\begin{pmatrix} x_1 & x_2 & \cdots & x_{p+2} \\ 1 & 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_{p+2} \end{pmatrix} = \mathbf{0} \, .$$

Since there are $p+2$ variables and $p+1$ equations, there is a solution $(a_1, \ldots, a_{p+2}) \neq \mathbf{0}$ for the above linear system.[8] Let $I_+ = \{i : a_i > 0\}$ and $I_- = \{i : a_i < 0\}$. Verify that

$$\sum_{i \in I_+} a_i = \sum_{i \in I_-} |a_i| \qquad \text{and} \qquad \sum_{i \in I_+} a_i x_i = \sum_{i \in I_-} |a_i| x_i$$

Under the assumption that the points are shattered, there exists some $h_{w,b} \in \mathcal{H}$ that can label $(x_i)_{i \in I_+}$ by $+1$ and $(x_i)_{i \in I_-}$ by -1, that is,

$$\langle w, x_i \rangle + b \begin{cases} > 0 & \text{for } i \in I_+ \\ < 0 & \text{for } i \in I_- \end{cases}$$

This implies

$$\begin{aligned} 0 < \sum_{i \in I_+} a_i (\langle w, x_i \rangle + b) &= \left\langle w, \sum_{i \in I_+} a_i x_i \right\rangle + b \sum_{i \in I_+} a_i \\ &= \left\langle w, \sum_{i \in I_-} |a_i| x_i \right\rangle + b \sum_{i \in I_-} |a_i| = \sum_{i \in I_-} |a_i| (\langle w, x_i \rangle + b) < 0 \end{aligned}$$

which is a contradiction. Hence, $\mathcal{H}$ cannot shatter $p+2$ points. □

Theorem 1.11 (VC dimension of 2-layer neural networks with binary activation). *Consider the following class of 2-layer networks. The input $x \in \mathbb{R}^p$. There are N units in the hidden layer each corresponding to a function*

$$f_i(x) = \operatorname{sign}(\langle w_i, x \rangle + b_i), \qquad i = 1, \ldots, N.$$

Let $f(x) = (f_1(x), \ldots, f_N(x)) \in \{\pm 1\}^N$. The output layer contains a single node, which returns

$$h(x) = \operatorname{sign}(\langle w, f(x) \rangle + b).$$

Let $\mathcal{H}$ be the class of all such 2-layer networks parameterised by $w \in \mathbb{R}^N$, $w_1, \ldots, w_N \in \mathbb{R}^p$ and $b, b_1, \ldots, b_N \in \mathbb{R}$. Then $\operatorname{VCdim}(\mathcal{H}) = O(pN \log_2(pN))$.

[8] Another way to look at this: The vectors $\binom{x_1}{1}, \ldots, \binom{x_{p+2}}{1}$ are $p+2$ vectors in $\mathbb{R}^{p+1}$. So they must be linearly dependent, that is, there must be $a_1, \ldots, a_{p+2}$ not all zero such that $\sum_i a_i \binom{x_i}{1} = 0$.

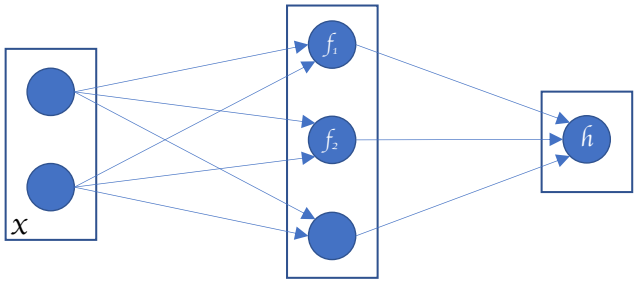

Figure 1.4: 2-layer neural network.

Proof. We begin with two claims.

Claim. *Let* $\mathcal{G}' \subseteq \mathcal{Y}'^{\mathcal{X}}$ *and* $\mathcal{G}'' \subseteq \mathcal{Y}''^{\mathcal{X}}$ *be two classes. Define* $\mathcal{G} = \mathcal{G}' \times \mathcal{G}'' \subseteq (\mathcal{Y}' \times \mathcal{Y}'')^{\mathcal{X}}$ *as*

$$\mathcal{G} = \{(g'(\cdot), g''(\cdot)) \ : \ g' \in \mathcal{G}', g'' \in \mathcal{G}''\}.$$

The growth functions satisfy $\tau_{\mathcal{G}}(m) \leq \tau_{\mathcal{G}'}(m)\tau_{\mathcal{G}''}(m)$.

To prove this, consider a sequence C of size m. Then

$$\mathcal{G}_{|C} = \{(\mathbf{g}', \mathbf{g}'') \ : \ \mathbf{g}' \in \mathcal{G}'_{|C}, \mathbf{g}'' \in \mathcal{G}''_{|C}\}.$$

Hence, $|\mathcal{G}_{|C}| = |\mathcal{G}'_{|C}| \cdot |\mathcal{G}''_{|C}| \leq \tau_{\mathcal{G}'}(m)\tau_{\mathcal{G}''}(m)$.

Claim. *Let* $\mathcal{G}' \subseteq \mathcal{Y}^{\mathcal{X}}$ *and* $\mathcal{G}'' \subseteq \mathcal{Z}^{\mathcal{Y}}$ *be two classes. Define* $\mathcal{G} = \mathcal{G}'' \circ \mathcal{G}' \subseteq \mathcal{Z}^{\mathcal{X}}$ *as*

$$\mathcal{G} = \{g''(g'(\cdot)) \ : \ g' \in \mathcal{G}', g'' \in \mathcal{G}''\}.$$

The growth functions satisfy $\tau_{\mathcal{G}}(m) \leq \tau_{\mathcal{G}'}(m)\tau_{\mathcal{G}''}(m)$.

To prove the claim, observe that for any sequence $C = (x_1, \ldots, x_m)$,

$$\begin{aligned}
|\mathcal{G}_{|C}| &= \left|\left\{(g''(g'(x_1)), \ldots, g''(g'(x_m))) \ : \ g' \in \mathcal{G}', g'' \in \mathcal{G}''\right\}\right| \\
&= \left|\left\{(g''(\mathbf{g}'_1), \ldots, g''(\mathbf{g}'_m)) \ : \ \mathbf{g}' \in \mathcal{G}'_{|C}, g'' \in \mathcal{G}''_{|C}\right\}\right| \\
&= \left|\bigcup_{\mathbf{g}' \in \mathcal{G}'_{|C}} \left\{(g''(\mathbf{g}'_1), \ldots, g''(\mathbf{g}'_m)) \ : \ g'' \in \mathcal{G}''_{|C}\right\}\right| \\
&\leq \sum_{\mathbf{g}' \in \mathcal{G}'_{|C}} \left|\mathcal{G}''_{|\mathbf{g}'}\right| \leq |\mathcal{G}'_{|C}|\tau_{\mathcal{G}''}(m) \leq \tau_{\mathcal{G}'}(m)\tau_{\mathcal{G}''}(m).
\end{aligned}$$

We now prove the theorem. Let $\mathcal{H}_i \subseteq \{\pm 1\}^{\mathbb{R}^p}$ denote the class of halfspaces corresponding to i-th unit of hidden layer, and $\mathcal{H}' \subseteq \{\pm 1\}^{\mathbb{R}^N}$ be the function class of halfspaces corresponding to the output layer. Hence, $\mathcal{H} = \mathcal{H}' \circ (\mathcal{H}_1 \times \dots \mathcal{H}_N)$.

From Theorem 1.10, we have $\text{VCdim}(\mathcal{H}_i) = p+1$ for all i and $\text{VCdim}(\mathcal{H}') = N+1$. Using Sauer's lemma,

$$\tau_{\mathcal{H}_i}(m) \leq \left(\frac{em}{p+1}\right)^{p+1} < (me)^{p+1}, \quad \text{and.} \quad \tau_{\mathcal{H}'}(m) \leq \left(\frac{em}{N+1}\right)^{p+1} < (me)^{N+1}.$$

Putting this the previous claims,

$$\tau_{\mathcal{H}}(m) < (me)^{N(p+1)+N+1} < m^{6pN}$$

for $m > e$. We use the fact $p \geq 1$ to simplify the relation. Recall if $\text{VCdim}(\mathcal{H}) = d$, then $2^d = \tau_{\mathcal{H}}(d)$ and $\tau_{\mathcal{H}}(m) < 2^m$ for all $m > d$.

Claim. *For any $c > 0$, $x \geq 2$ and $m = 3cx \log_2 x$, we have $2^m > m^{cx}$.*

Verify the inequality in logarithmic form, that is, $m > cx \log_2 m$. Do this by plugging the value of m in $cx \log_2 x$ and then simplifying.

From the above claim it follows that for $m \geq 18pN \log_2(pN)$, $\tau_{\mathcal{H}}(m) < 2^m$. Hence, $\text{VCdim}(\mathcal{H}) = O(pN \log_2(pN))$. □

Chapter 2

Probably Approximately Correct (PAC) learning

In this chapter, we take a different perspective to the theory of generalisation, where the focus is on how well we can learn a hypothesis class $\mathcal{H}$. The main concept that we will discuss is the notion of *learnability*, that is, whether we can learn the best possible predictor in $\mathcal{H}$ upto some error. The Probably Approximately Correct (PAC) framework was proposed by Leslie Valiant [Valiant, 1984].

2.1 Learnability and connection to VC theory

The focus of Chapter 1 was on the difference between the empirical and true risks. We derived uniform convergence (Theorem 1.4) — a bound on $\sup_{h\in\mathcal{H}} \left|L_S(h) - L_{\mathcal{D}}(h)\right|$ that holds uniformly for all $\mathcal{D}$. In simple words, the consequence of Theorem 1.4 is that given the empirical risk of a learned predictor $L_S(\widehat{h})$, we can guess what its true risk would be. In PAC, we are interested in knowing how large is the true risk $L_{\mathcal{D}}(\widehat{h})$ compared to the best possible predictor in $\mathcal{H}$, that is, $\inf_{h\in\mathcal{H}} L_{\mathcal{D}}(h)$.

Two cases are sometimes considered in PAC learning.

Definition 2.1 (Realisable and agnostic settings). *A distribution $\mathcal{D}$ on $\mathcal{X}\times\mathcal{Y}$ is said to be* realisable *with respect to a hypothesis class $\mathcal{H}\subseteq\mathcal{Y}^{\mathcal{X}}$ if there exists a predictor $h^*\in\mathcal{H}$ such that $L_{\mathcal{D}}(h^*)=0$. In the* agnostic PAC model*, there may not exist any $h\in\mathcal{H}$ such that $L_{\mathcal{D}}(h)=0$.*

To put it simply, $\mathcal{H}$ contains an ideal predictor h^* in the realisable case, whereas under the agnostic model, we do not care about the ideal predictor, but would focus only on the best

possible $h \in \mathcal{H}$ which achieves the smallest risk $\inf_{h\in\mathcal{H}} L_{\mathcal{D}}(h)$. Except some later discussions, we will not distinguish between the two cases.

Exercise 2.1. Assume $\mathcal{Y} = \{\pm 1\}$. Recall that we may decompose $\mathcal{D}$ into $\mathcal{D}_{\mathcal{X}}$, marginal distribution over $\mathcal{X}$, and $\mathbb{P}_{\mathcal{Y}|\mathcal{X}}(y|x)$, the conditional probability for the labels.
Show that if $\mathbb{P}_{\mathcal{Y}|\mathcal{X}}(y|x)$ is non-degenerate, $\mathcal{D}$ cannot be realisable with respect to any $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$. Give a complete characterisation of $\mathcal{D}$ so that it is realisable with respect to $\mathcal{H}$.

Before introducing the notion of learnability, let us see the consequences of VC theory (Theorem 1.4) in deriving a bound on true risk of ERM. For convenience, define

$$L_{\mathcal{D}}(\mathcal{H}) = \inf_{h\in\mathcal{H}} L_{\mathcal{D}}(h).$$

Corollary 2.2 (Error bound for ERM). *Let $m \geq \frac{2\ln 4}{\epsilon^2}$ and $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$ has a growth function $\tau_{\mathcal{H}}(\cdot)$. Assume that the ERM learner $\mathcal{A}$ returns the predictor $\widehat{h} = \mathcal{A}(S) \in \mathcal{H}$. We have*

$$\mathbb{P}_S\left(L_{\mathcal{D}}(\widehat{h}) > L_{\mathcal{D}}(\mathcal{H}) + 2\epsilon\right) \leq 4\tau_{\mathcal{H}}(2m)\exp\left(-m\epsilon^2/8\right),$$

Hence, for any $\delta \in (0,1)$, the risk of $\widehat{h}$ satisfies

$$L_{\mathcal{D}}(\widehat{h}) < L_{\mathcal{D}}(\mathcal{H}) + 2\sqrt{\frac{8\left(\ln\left(\tau_{\mathcal{H}}(2m)\right) + \ln\left(\frac{4}{\delta}\right)\right)}{m}} \qquad \textit{with probability } 1-\delta.$$

Proof. The proof is left as an exercise. Use Theorem 1.4, and the fact $L_S(\widehat{h}) = \min_{h\in\mathcal{H}} L_S(h)$ since $\widehat{h}$ is the solution of ERM. Recall that Theorem 1.4 can also be used to derive a lower bound for $L_{\mathcal{D}}(\widehat{h})$. Is this possible in the case of Corollary 2.2? Why? □

Exercise 2.2. Show that the bound in Corollary 2.2 can be slightly improved to

$$L_{\mathcal{D}}(\widehat{h}) < L_{\mathcal{D}}(\mathcal{H}) + \sqrt{\frac{8\left(\ln\left(\tau_{\mathcal{H}}(2m)\right) + \ln\left(\frac{4}{\delta}\right)\right)}{m}} + \sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2m}} \qquad \text{with probability } 1-\delta.$$

Corollary 2.2 shows that, $\tau_{\mathcal{H}}$ grows slowly, we can use ERM to learn predictors that are nearly optimal for the class $\mathcal{H}$. This brings us to the notion of learnability.

Definition 2.3 (Learnability). *A hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ is* agnostic PAC learnable *if there exists a function $m_{\mathcal{H}} : (0,1)\times(0,1) \to \mathbb{N}$ and a learner $\mathcal{A}$ such that following holds. For every $\epsilon, \delta \in (0,1)$ and every $\mathcal{D}$ on $\mathcal{X}\times\mathcal{Y}$, when $\mathcal{A}$ runs on $m \geq m_{\mathcal{H}}(\epsilon,\delta)$ i.i.d. samples from $\mathcal{D}$, then the output $\widehat{h} = \mathcal{A}(\cdot)$ satisfies*

$$L_{\mathcal{D}}(\widehat{h}) \leq L_{\mathcal{D}}(\mathcal{H}) + \epsilon$$

with probability $1-\delta$, where the probability is over the training sample of size m.
If the above statement holds for every $\mathcal{D}$ that is realisable with respect to $\mathcal{H}$, then the class

$\mathcal{H}$ is said to be PAC learnable.[1]
The function $m_{\mathcal{H}}(\epsilon, \delta)$ is called the sample complexity *of learning the class $\mathcal{H}$.*

Note that agnostic PAC learnability also implies PAC learnability since the latter focusses on a restricted set of distributions. Definition 2.3, along with Corollary 2.2, leads to the following result on learnability of finite VC dimension classes.

Corollary 2.4 (Finite VC dimension classes are PAC learnable). *Every hypothesis class $\mathcal{H}$ with a finite VC dimension is agnostic PAC learnable, where ERM is the learner.*

2.1.1 No free lunch theorem

The previous discussions suggest that ERM is successful for most function classes. However, it cannot provide exceptional accuracies in all cases. This is shown in the following *no free lunch* theorem, which essentially states that there is no universal algorithm that can learn all possible function classes using a finite training sample. More precisely, it states that for every learner and every sample size m, there is a distribution under which the learner will fail given m training samples.

Theorem 2.5 (No free lunch theorem). *Let $\mathcal{A}$ be a learner for binary classification over $\mathcal{X}$. Assume that the training sample size $m < \frac{|\mathcal{X}|}{2}$. There exists a distribution $\mathcal{D}$ on $\mathcal{X} \times \{\pm 1\}$ such that there is a function $h \in \{\pm 1\}^{\mathcal{X}}$ with $L_{\mathcal{D}}(h) = 0$, and*

$$\mathbb{P}_{S \sim \mathcal{D}^m} \left(L_{\mathcal{D}}(\mathcal{A}(S)) \geq \frac{1}{8} \right) > \frac{1}{7} \, .$$

Proof. Let C be a set of $2m$ samples in $\mathcal{X}$. There are $T = 2^{2m}$ functions from C to $\{\pm 1\}$. Call them as $h_1, \ldots, h_T$. For each h_i, define the distribution $\mathcal{D}_i$ on $\mathcal{X} \times \{\pm 1\}$ as

$$\mathcal{D}_i(x, y) = \begin{cases} \frac{1}{2m} & \text{if } x \in C, y = h_i(x) \\ 0 & \text{otherwise.} \end{cases}$$

By construction, $S \sim \mathcal{D}_i^m$ will sample only pairs of the form $(x, h_i(x))$ for $x \in C$. No sample outside C will be observed. Note that $L_{\mathcal{D}_i}(h_i) = 0$ for every i.

We now claim that for every learner $\mathcal{A}$ that receives $S \in (C \times \{\pm 1\})^m$ and returns $\mathcal{A}(S) : C \to \{\pm 1\}$, it holds that

$$\max_{i \in [T]} \mathbb{E}_{S \sim \mathcal{D}_i^m} \left[L_{\mathcal{D}_i}(\mathcal{A}(S)) \right] \geq \frac{1}{4} \, . \tag{2.1}$$

[1] Note on the name: The solution of Exercise 2.1 would show that $\mathcal{D}$ is realisable if there is an $h^* \in \mathcal{H}$ such that $(x, y) \sim \mathcal{D}$ corresponds to $x \in \mathcal{D}_{\mathcal{X}}$ and $y = h^*(x)$, that is, there is no randomness in y given x. $\mathcal{H}$ is PAC learnable implies that there exists learner $\mathcal{A}$ and function $m_{\mathcal{H}}(\epsilon, \delta)$ so that the obtained predictor $\widehat{h} = \mathcal{A}(\cdot)$ is *probably* (with probability $1 - \delta$) *approximately correct* (close to h^* up to an excess risk of ϵ).

Before proving (2.1), we discuss its role in proving Theorem 2.5. The maximum in (2.1) exceeding $\frac{1}{4}$ implies that there is a $\mathcal{D}_i$ for which the expectation exceeds $\frac{1}{4}$. Thus, for every learner $\mathcal{A}$, there exists h_i and $\mathcal{D}_i$ so that $L_{\mathcal{D}_i}(h_i) = 0$ and

$$\mathbb{E}_{S\sim\mathcal{D}_i^m}\left[L_{\mathcal{D}_i}(\mathcal{A}(S))\right] \geq \frac{1}{4}\,.$$

We now use the following inequality: If Z is a random variable that takes values in $[0,1]$, then

$$\mathbb{P}(Z \geq a) > \frac{\mathbb{E}[Z]-a}{1-a}$$

for any $a \in (0,1)$. One can prove this using Markov's inequality (try to prove it or see Lemma B.1 in Shalev-Shwartz and Ben-David [2018]). Using this inequality, we have

$$\mathbb{P}_{S\sim\mathcal{D}_i^m}\left(L_{\mathcal{D}}(\mathcal{A}(S)) \geq \frac{1}{8}\right) > \frac{\mathbb{E}_{S\sim\mathcal{D}_i^m}\left[L_{\mathcal{D}_i}(\mathcal{A}(S))\right]-\frac{1}{8}}{1-\frac{1}{8}} \geq \frac{\frac{1}{4}-\frac{1}{8}}{1-\frac{1}{8}} = \frac{1}{7}$$

which prove the theorem.

We now prove (2.1). Note that there are $Q = (2^{2m})^m$ possible ways to sample m examples from C. For each such sequence of m examples, there T possible labelling. Hence, for $i \in [Q]$ and $j \in [T]$, define

$$S_{i,j} = \{(x_1, f_i(x_1)), \ldots, (x_m, f_i(x_m)\}$$

where $X_j = (x_1, \ldots, x_m) \in C^m$ denotes the j-th possible way to sample m examples from C. Verify that $S \sim \mathcal{D}_i^m$ uniformly chooses one of Q possible training sequences $S_{i,1}, \ldots, S_{i,Q}$ and so,

$$\mathbb{E}_{S\sim\mathcal{D}_i^m}\left[L_{\mathcal{D}_i}(\mathcal{A}(S))\right] = \frac{1}{Q}\sum_{j=1}^{Q} L_{\mathcal{D}_i}(\mathcal{A}(S_{i,j}))\,.$$

Since the average lies between maximum and minimum, we write

$$\begin{aligned}
\max_{i\in[T]}\ \mathbb{E}_{S\sim\mathcal{D}_i^m}\left[L_{\mathcal{D}_i}(\mathcal{A}(S))\right] &\geq \frac{1}{T}\sum_{i=1}^{T}\frac{1}{Q}\sum_{j=1}^{Q} L_{\mathcal{D}_i}(\mathcal{A}(S_{i,j})) \\
&= \frac{1}{Q}\sum_{j=1}^{Q}\frac{1}{T}\sum_{i=1}^{T} L_{\mathcal{D}_i}(\mathcal{A}(S_{i,j})) \\
&\geq \min_{j\in[Q]}\frac{1}{T}\sum_{i=1}^{T} L_{\mathcal{D}_i}(\mathcal{A}(S_{i,j}))
\end{aligned}$$

Note that the last term fixes a sequence $X_j = (x_1, \ldots, x_m) \in C^m$ and then computes the average over all $\mathcal{D}_i$. However, for every such sequence X_j, there a set $C_j = \{v_1, \ldots, v_p\} \subset C$

of examples that do not appear in X_j. Verify that $|C_j| = p \geq m$. For any function $h : C \to \{\pm 1\}$,

$$L_{\mathcal{D}_i}(h) = \frac{1}{2m} \sum_{x \in C} \mathbf{1}\{h(x) \neq h_i(x)\} \geq \frac{1}{2m} \sum_{k=1}^{p} \mathbf{1}\{h(v_k) \neq h_i(v_k)\}$$
$$\geq \frac{1}{2p} \sum_{k=1}^{p} \mathbf{1}\{h(v_k) \neq h_i(v_k)\} \ .$$

Averaging over all $i \in [T]$,

$$\frac{1}{T} \sum_{i=1}^{T} L_{\mathcal{D}_i}(\mathcal{A}(S_{i,j})) \geq \frac{1}{T} \sum_{i=1}^{T} \frac{1}{2p} \sum_{k=1}^{p} \mathbf{1}\{\mathcal{A}(S_{i,j})(v_k) \neq h_i(v_k)\}$$
$$= \frac{1}{2} \cdot \frac{1}{p} \sum_{k=1}^{p} \frac{1}{T} \sum_{i=1}^{T} \mathbf{1}\{\mathcal{A}(S_{i,j})(v_k) \neq h_i(v_k)\}$$
$$\geq \frac{1}{2} \cdot \min_{k \in [p]} \frac{1}{T} \sum_{i=1}^{T} \mathbf{1}\{\mathcal{A}(S_{i,j})(v_k) \neq h_i(v_k)\} \ .$$

Note that for every v_k, we can group the T functions $\{h_1, \ldots, h_T\}$ into $T/2$ pairs such that every pair $(h_i, h_{i'})$ satisfy

$$h_i(v_k) \neq h_{i'}(v_k) \text{ and } h_i(x) = h_{i'}(x) \ \forall x \in C \backslash \{v_k\}.$$

Since $v_k \notin X_j$, the above implies that $S_{i,j} = S_{i',j}$, and so, $\mathcal{A}(S_{i,j}) = \mathcal{A}(S_{i',j})$ and

$$\mathbf{1}\{\mathcal{A}(S_{i,j})(v_k) \neq h_i(v_k)\} + \mathbf{1}\left\{\mathcal{A}(S_{i',j})(v_k) \neq h_{i'}(v_k)\right\} = 1,$$

which results in

$$\frac{1}{T} \sum_{i=1}^{T} \mathbf{1}\{\mathcal{A}(S_{i,j})(v_k) \neq h_i(v_k)\} = \frac{1}{2}$$

for every $k \in [p]$. Combining all the above steps, we get (2.1). □

Let us now look at the implications of Theorem 2.5. One of the straightforward consequences is the following.

Exercise 2.3. Let $\mathcal{X}$ be an infinite domain. Prove that $\{\pm 1\}^{\mathcal{X}}$ is *not PAC-learnable*.
Hint: Use Theorem 2.5 to argue that $m_{\mathcal{H}}\left(\frac{1}{8}, \frac{1}{7}\right)$ is infinite.

The result can also be extended to function classes $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$ that have infinite VC dimension.

Corollary 2.6 (Infinite VC dimension classes are not PAC learnable). *Let $\mathcal{X}$ be an infinite domain and let $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$ with $\mathrm{VCdim}(\mathcal{H}) = \infty$. Let $\mathcal{A}$ be a learner such that outputs function from $\mathcal{H}$. For every sample size m, there exists a distribution $\mathcal{D}$ realisable with respect to $\mathcal{H}$ such that*

$$\mathbb{P}_{S\sim\mathcal{D}^m}\left(L_{\mathcal{D}}(\mathcal{A}(S)) \geq \frac{1}{8}\right) > \frac{1}{7}\,.$$

Proof. (Exercise) Modify the proof of Theorem 2.5 by starting with a C of size $2m$ that is shattered by $\mathcal{H}$. If $\mathrm{VCdim}(\mathcal{H}) = \infty$, this can be done for arbitrarily large C. □

Remark (No free lunch vs learnability of finite VC dimension classes). Note that Theorem 2.5 does not contradict the previous conclusion about learnability of finite VC classes (Corollary 2.4). Assume $\epsilon < \frac{1}{8}$ and $\delta < \frac{1}{7}$. In the agnostic case, Corollary 2.4 states that some learner (ERM), when trained on $m_{\mathcal{H}}(\epsilon, \delta)$ samples, ensures that for every $\mathcal{D}$,

$$L_{\mathcal{D}}(\widehat{h}) \leq \inf_{h\in\mathcal{H}} L_{\mathcal{D}}(h) + \epsilon \qquad \text{with probability } 1-\delta.$$

On the other hand, Theorem 2.5 states that no learner (including ERM) can achieve

$$L_{\mathcal{D}}(\widehat{h}) \leq \epsilon \qquad \text{with probability } 1-\delta$$

for every $\mathcal{D}$. The results together assert that an approximation error, in the form of $L_{\mathcal{D}}(\mathcal{H}) = \inf_{h\in\mathcal{H}} L_{\mathcal{D}}(h)$, is possibly unavoidable in learning. Note that this can be avoided if $L_{\mathcal{D}}(\mathcal{H}) = 0$ (for instance, using linear classifiers in linearly separable problems).

2.1.2 Fundamental theorem of statistical learning

No free lunch theorem and uniform convergence results, taken together, leads to the conclusion that finite VC dimension is necessary and sufficient for PAC learnability. This is part of the *fundamental theorem of statistical learning*. The theorem, as stated in Shalev-Shwartz and Ben-David [2018], touches upon a related concept known as *uniform convergence property*. For the sake of completeness, we define this property before stating the fundamental theorem.

Definition 2.7 (Uniform convergence property). *A hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ is said to have the* uniform convergence property *(with respect to a loss ℓ) if there exists a function $m^{uc}_{\mathcal{H}} : (0,1)\times(0,1) \to \mathbb{N}$ such that for every $\epsilon, \delta \in (0,1)$ and every distribution $\mathcal{D}$ on $\mathcal{X}\times\mathcal{Y}$, then any sample $S \sim \mathcal{D}^m$ with $m \geq m_{\mathcal{H}}(\epsilon,\delta)$ satisfies*

$$\sup_{h\in\mathcal{H}} |L_S(h) - L_{\mathcal{D}}(h)| \leq \epsilon \qquad \textit{with probability } 1-\delta.$$

Observe that the uniform convergence property is another way of stating that generalisation error bounds in Chapter 1 can be made smaller than ϵ by controlling the training sample size.

Theorem 2.8 (Fundamental theorem of statistical learning (binary classification)). *Let $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$ and let the loss function be the 0-1 loss. Then the following are equivalent:*

1. *$\mathcal{H}$ has finite VC dimension.*
2. *$\mathcal{H}$ has uniform convergence property.*
3. *Any ERM rule is a successful agnostic PAC learner for $\mathcal{H}$.*
4. *$\mathcal{H}$ is agnostic PAC learnable.*
5. *$\mathcal{H}$ is PAC learnable.*
6. *Any ERM rule is a successful PAC learner for $\mathcal{H}$.*

Proof. (Exercise) Combine all previous theorems. You can prove $1 \implies 2 \implies 3$, $3 \implies 4 \implies 5$, and $3 \implies 6 \implies 5$. Finally, $5 \implies 1$ can be proved by contradiction based on Corollary 2.6. □

The fundamental theorem of statistical learning also holds for some other learning problems (for instance, regression). This requires generalisation of VC dimension for such problems. We will not cover this.

There is a quantitative version of the fundamental theorem (see Theorem 6.8 in Shalev-Shwartz and Ben-David [2018]). We look at a part of this result below. The following theorem shows both an upper and a lower bound for the sample complexity $m_{\mathcal{H}}(\epsilon, \delta)$.

Theorem 2.9 (Sample complexity for finite VC dimension classes). *Let $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$ have $\mathrm{VCdim}(\mathcal{H}) = d < \infty$. Then there exists constants $C_1, C_2 > 0$ such that $\mathcal{H}$ is agnostic PAC learnable with sample complexity*

$$C_1 \frac{d + \log_2(\frac{1}{\delta})}{\epsilon^2} \leq m_{\mathcal{H}}(\epsilon, \delta) \leq C_2 \frac{d \log_2(\frac{d}{\epsilon}) + \ln(\frac{1}{\delta})}{\epsilon^2} .$$

Proof. The upper bound follows from Corollary 2.2, and is left as an exercise. The lower bound may be viewed as a quantitative version of no free lunch theorem. See the proof is Section 28.2 of Shalev-Shwartz and Ben-David [2018]. □

2.2 Computational aspect of learning

So far, we have been only concerned about whether we can learn $\mathcal{H}$ from a finite number of samples. The conclusion was that, as long as $\text{VCdim}(\mathcal{H}) < \infty$, the class $\mathcal{H}$ can be successfully learned using $m_{\mathcal{H}}(\epsilon, \delta)$ number of samples. However, the learning problem could be computationally hard, that is, may have an exponential time complexity. A valid question would be *exponential in which parameter?* This is slightly confusing in comparison to measuring complexity of typical algorithms such as sorting. Note that the key parameters are ϵ, δ, which control the level of accuracy. There could be two sources of exponential computational complexity.

The first possibility is that the sample complexity is exponential, that is, $m_{\mathcal{H}}(\epsilon, \delta)$ is exponential in $\frac{1}{\epsilon}$ or $\frac{1}{\delta}$. Theorem 2.9, or its improved variant, rules out this possibility, stating that $m_{\mathcal{H}}(\epsilon, \delta) = O\left(\frac{d+\ln(\frac{1}{\delta})}{\epsilon^2}\right)$. The other possibility is that a successful learner takes exponential time to return the output, that is, the learner runs in time exponential with respect to $\frac{1}{\epsilon}$ or $\frac{1}{\delta}$.

Exercise 2.4 (Worst-case computational time for ERM). Suppose that ERM is used to learn $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$ where $\text{VCdim}(\mathcal{H}) = d < \infty$. There exists an implementation of ERM that runs in time exponential in d, $\frac{1}{\epsilon}$ and polynomial in $\frac{1}{\delta}$.
Does the result change if the (PAC) learner is given $m \gg m_{\mathcal{H}}(\epsilon, \delta)$ training samples such that m is exponential in $\frac{1}{\epsilon}, \frac{1}{\delta}$.

In some cases, however, ERM (or other learners) can be implemented in polynomial time, particularly in the realisable case.

Example 5 (Decision stumps). Recall the class of decision stumps $\mathcal{H} \subseteq \{\pm 1\}^{\mathbb{R}}$,

$$\mathcal{H} = \{h_{t,b} \ : \ h_{t,b}(x) = b \text{ for } x \leq t, \text{ and } h_{t,b}(x) = -b \text{ for } x > t\}.$$

Consider the realisable setting, that is, there is a true function h_{t^*,b^*} such that the training data $S = \{(x_i, y_i)\}_{i=1}^m$ satisfies $y_i = h_{t^*,b^*}(x_i)$.
Verify that the following is an ERM solution in the realisable case.

1. Let $i_1 = \arg\min_i \ x_i$, and set $\widehat{b} = y_{i_1}$ ... takes $O(m)$ time
2. Let $i_2 = \arg\max_{i:y_i=\widehat{b}} \ x_i$, and $i_3 = \arg\min_{i:y_i=-\widehat{b}} \ x_i$ ... takes $O(m)$ time
3. Set $\widehat{t} = \frac{1}{2}(x_{i_2} + x_{i_3})$, and return the predictor $h_{\widehat{t},\widehat{b}} \in \mathcal{H}$.

Observe that the above procedure successfully PAC learns in poly $\left(\frac{1}{\epsilon}, \frac{1}{\delta}\right)$ time.[2]

Exercise 2.5. Derive an ERM procedure that is a successful agnostic PAC learner, and runs in worst case $O(m^2)$ time. Hint: $O(m^2)$ time is due to sorting $x_1, \ldots, x_m$.

[2] poly $(a, b, \ldots, z)$ means polynomial in each of the variables $a, \ldots, z$.

2.2.1 Efficient learning algorithm

The notion of *efficient learner* is introduced differently in various texts. We will consider a simplified version of the definition in Shalev-Shwartz and Ben-David [2018].

Definition 2.10 (Time complexity of learner). *Assume that the loss function is ℓ, and consider a learner $\mathcal{A}$.*

1. *Given a function $f : (0,1)^2 \to \mathbb{N}$ and a class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, we say that $\mathcal{A}$ learns the class $\mathcal{H}$ in $O(f)$ times if:*
   - *$\mathcal{A}$ is (agnostically) PAC learns $\mathcal{H}$, and*
   - *for every distribution $\mathcal{D}$ and every $\epsilon, \delta \in (0,1)$, given training set S sampled i.i.d. from $\mathcal{D}$, the output $\widehat{h} = \mathcal{A}(S)$ is determined in $O(f((\epsilon,\delta)))$ time and $\widehat{h}(\cdot)$ can also be computed in $O(f((\epsilon,\delta)))$ time.*
2. *Consider a sequence of classes $(\mathcal{H}_p)_{p=1}^{\infty}$ such that $\mathcal{H}_p \subseteq \mathcal{Y}^{\mathcal{X}_p}$. Given a function $g : \mathbb{N} \times (0,1)^2 \to \mathbb{N}$, we say that the runtime of $\mathcal{A}$ with respect to $(\mathcal{H}_p)_{p=1}^{\infty}$ is $O(g)$ if, for every $p \in \mathbb{N}$, we can define a function $f_p(\epsilon,\delta) = g(p,\epsilon,\delta)$ such that $\mathcal{A}$ learns $\mathcal{H}_p$ in $O(f_p)$ time.*
3. *We say $\mathcal{A}$ is* efficient *with respect to $(\mathcal{H}_p)_{p=1}^{\infty}$ if g is* poly $\left(p, \frac{1}{\epsilon}, \frac{1}{\delta}\right)$.

The introduction of a sequence of classes $(\mathcal{H}_p)_{p=1}^{\infty}$ is abrupt, and needs further explanation. The typical example is where p governs the dimension of the domain space, for instance, $\mathcal{X}_p = \mathbb{R}^p$. For instance, linear classifiers or SVMs can be used for binary classification for every $\mathbb{R}^p$, and hence, it is natural to consider the time complexity of a learner with respect to the problem dimension — an exponential dependence on dimension is not desirable, particularly for high-dimensional binary classification.

Exercise 2.6. Read Sections 8.2.2–8.2.4 in Shalev-Shwartz and Ben-David [2018] which discuss the time complexity of ERMs for learning axis-parallel rectangles and Boolean conjunctions.

Exercise 2.7 (Efficient ERM for multi-dimensional decision stumps). The class of decision stumps $\mathcal{H}_p \subseteq \{\pm 1\}^{\mathbb{R}^p}$ is the class of functions $\{h_{t,b,i} \ : \ t \in \mathbb{R}, b \in \{\pm 1\}, i \in [p]\}$ such that

$$h_{t,b,i}(x) = \begin{cases} b & \text{if } x^{(i)} \leq t \\ -b & \text{if } x^{(i)} > t, \end{cases}$$

where $x^{(i)}$ denotes the i-th coordinate of $x \in \mathbb{R}^p$.

Extend the algorithm of Exercise 2.5 to obtain a successful agnostic PAC learner that runs in $O(pm^2)$ time.

Hint: Scan along each dimension, and apply ERM to obtain p 1-dimensional decision stumps. Among the p solutions, choose the one with minimum empirical risk.

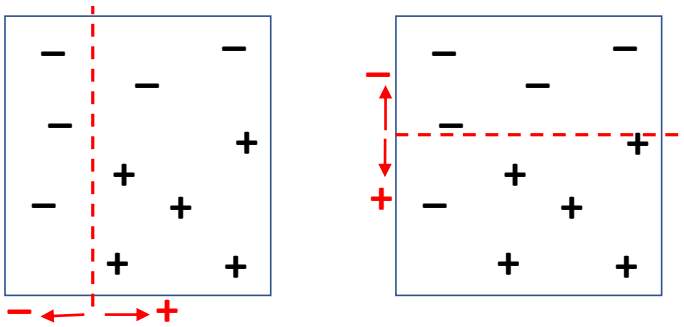

Figure 2.1: 2-dimensional decision stumps (red boundaries). Left one minimises empirical risk horizontally, and right one is minimiser along vertical direction.

2.2.2 Perceptron

Decision stumps in $\mathbb{R}^p$ correspond to axis-aligned linear classifiers. We now analyse the *batch perceptron* algorithm, which efficiently PAC learns the class of linear classifiers in $\mathbb{R}^p$

$$\mathcal{H} = \{h(x) = \text{sign}(\langle w, x\rangle + b) \ : \ w \in \mathbb{R}^p, b \in \mathbb{R}\},$$

in the *realisable case*. Recall that $\text{VCdim}(\mathcal{H}) = p + 1$. The algorithm is as follows.[3]

Algorithm 1: Batch perceptron

Input: Training samples $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$
Set $b_1 = 0, w_1 = \mathbf{0}$
for $t = 1, 2, \ldots$ **do**
  if $\exists (x_i, y_i) \in S$ *such that* $y_i(\langle w_t, x_i\rangle + b_t) \leq 0$ **then**
    $w_{t+1} \leftarrow w_t + y_i x_i$ and $b_{t+1} \leftarrow b_t + y_i$
  end
  return *Linear classifier* h_{w_t, b_t}
end

We analyse the convergence and time complexity of Algorithm 1 in the realisable case, that is, there is $w_\circ \in \mathbb{R}^p, b_circ \in \mathbb{R}$ such that for every (x, y)

$$y = \text{sign}(\langle w_\circ, x\rangle + b_\circ), \quad \text{or equivalently,} \quad y(\langle w_\circ, x\rangle + b_\circ) > 0.$$

When the above assumption holds, we say that *the data is separable*.

[3] The *perceptron* algorithm is typically used in *online learning*, where training samples arrive in a streamed manner. The presented version is called *batch perceptron* since, instead of data arriving in a stream, it is available as a batch. More generally, the setting that we have studied — $S \sim \mathcal{D}^m$ is available up front — is known as the *batch setting*.

Theorem 2.11 (Convergence of batch perceptron). *Assume that* $(x_1, y_1), \ldots, (x_m, y_m)$ *are separable. Define* $R = \max_i \|x_i\|_2$*, where* $\|\cdot\|_2$ *denotes the Euclidean norm. Let* $B = \min\left\{\sqrt{\|w\|_2^2 + b^2} \ : \ y_i(\langle w, x_i\rangle + b) \geq 1 \ \forall i \in [m]\right\}$*. Then Algorithm 1 converges in at most* $(RB)^2$ *iterations, and the solution achieves minimum empirical risk.*

Proof. The last part is straightforward. If Algorithm 1 converges, we have $y_i(\langle w, x_i\rangle + b) > 0$ for every $i \in [m]$, that is, it achieves zero training error, or equivalently, minimises the empirical risk.

To prove convergence, we first argue that if the data is separable, then one can find (w, b) such that $y_i(\langle w, x_i\rangle + b) \geq 1$ for all $i \in [m]$. To see this, let (w', b') satisfies the separability assumption $y_i(\langle w', x_i\rangle + b') > 0 \ \forall i$, and define $\gamma = \min_i y_i(\langle w', x_i\rangle + b')$. Verify that $y_i\left(\left\langle \frac{w'}{\gamma}, x_i \right\rangle + \frac{b'}{\gamma}\right) \geq 1$ for all $i \in [m]$.

For the rest of the proof, see proof of Theorem 9.1 in Shalev-Shwartz and Ben-David [2018]. Note that, in the book, the problem is transformed as follows: Map every $x \in \mathbb{R}^p$ to $\tilde{x} = (x, 1) \in \mathbb{R}^{p+1}$ and denote $\tilde{w} = (w, b)$. Then we can write $\mathcal{H}$ as $\mathcal{H} = \{\text{sign}(\langle \tilde{w}, \tilde{x}\rangle)\}$, the class of homogeneous linear classifiers in $\mathbb{R}^{p+1}$. In this case, we need to only update $\tilde{w}$, and analyse the convergence of these updates. □

We will possibly return to the perceptron algorithm later, if we cover online learning. Note that the batch perceptron may not converge in the agnostic case.

Chapter 3

Boosting

3.1 Weak learnability

3.2 AdaBoost

Chapter 4

Generalisation error for support vector machines

4.1 Support vector machine

4.2 Rademacher complexity

4.3 Analysis of SVM

Chapter 5

Analysis of k-Nearest Neighbour Rule

5.1 Universal consistency

5.2 Generalisation error for 1-NN

5.3 Universal consistency of k-NN

Chapter 6

Structural risk minimisation

6.1 Structural risk minisation

6.2 Regularisation

6.3 Stability

Part II

Theory of Unsupervised Learning

Bibliography

Avrim Blum, John Hopcroft, and Ravindran Kannan. *Foundations of data science*. Cambridge university press, 2020. URL https://www.cs.cornell.edu/jeh/book.pdf.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2018. URL https://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf.

Leslie G Valiant. A theory of the learnable. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 436–445. ACM, 1984.

Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

Vladimir N Vapnik and Alexey J Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264, 1971.