Mathematics of Machine Learning

Arthur Conmy*

Part II, Lent Term 2021

These notes are based on lectures given (virtually) by Dr R. Shah in Lent term 2021. Credit is also due to Evan Chen for the style file for these notes[1].

Contents

*Please send any corrections and/or feedback to asc70@cam.ac.uk

[1] Available here: https://github.com/vEnhance/dotfiles/blob/master/texmf/tex/latex/evan/evan.sty.

§1 Introduction

The course will be divided into three parts:

- Statistical learning theory (including empirical risk minimization).
- Computation (including (stochastic) gradient descent).
- Popular methods in practise.

The course will move from a more theoretical background to practical things. There is a significant gap between the theory and the practise in machine learning, as will be seen in the course.

§1.1 Conditional Expectation

The results stated here are true subject to certain convergence conditions. II Stochastic Financial Models and II Probability and Measure deal with such things more formally. For our purposes, we will use the results as tools to develop theory.

Definition 1.1 (Conditional density)**.** Let Z and W be random variables with joint density $f(z, w)$, and let $f_W(w)$ be the marginal density of W (integrate over all z values). Then the **conditional density** of z given w is

[Conditional expectation] In the same notation as above, we define the *random variable* $\mathbb{E}[Z \mid W]$ as

$$\mathbb{E}[Z \mid W] = \int z f_{Z|W}(z, W) \mathrm{d}z. \tag{1}$$

Note that $\mathbb{E}[Z \mid W]$ is a function of a random variable (W), so is itself a random variable. It is not simply a number, as ordinary expectations are.

Remark 1.2 (Digression: design choice)**.** I have reorganised this section to include condtional expectation things before the initial setup for ERM.This is because I have found conditional expectation one of the most frustrating (though certainly necessary) parts of the course, so any familiarity now will help later, but total familiarity is probably not to be expected.

Remark 1.3. We can interpret $\mathbb{E}[Z|W]$ as 'the function of W that's our best guess for Z given only the information contained in W'[a].

[a] see https://dynalist.io/d/bx3GM7El5D_PsHTOvxgJlTyW

Remark 1.4. In this course, whenever we are conditioning over something, think of this as what is currently fixed.

Two examples from later in the course are in the definition of risk (6), we've chosen our hypothesis h and are now evaluating the expectation of the loss (risk); we are NOT considering some non-deterministic method for generating h for example

This is perhaps counterintuitive since of course $\mathbb{E}[Z|W]$ is a random *variable* that's a function of W however ...

Interpretation (1.3) gives some intuition behind the next result, which generalises what we did in IA.

Theorem 1.5 (General tower property)

Let $f : \mathbb{R}^d \to \mathbb{R}^m$ and Z and W be random variables. Then

$$\mathbb{E}[\mathbb{E}[Z \mid W] \mid f(W)] = \mathbb{E}[Z \mid f(W)]. \tag{2}$$

Theorem 1.6 (Taking out what is known)

Let f be real-valued. Then

$$\mathbb{E}[f(W)Z \mid W) = f(W)\mathbb{E}[Z \mid W] \tag{3}$$

As an example of how to use these tools,

Theorem 1.7 (Best least squares predictor)

The following holds:

$$\mathbb{E}\left[Z - f(W)\right]^2 = \mathbb{E}\left[Z - \mathbb{E}\left[Z \mid W\right]\right]^2 + \mathbb{E}\left[f(W) - \mathbb{E}\left[Z \mid W\right]\right]^2 \tag{4}$$

We will write out the extended details of how to apply our conditional expectation theory to get this result.

Proof. Initially, add the obvious term to the LHS:

$$\begin{aligned}\mathbb{E}\left[Z - f(W)\right]^2 &= \mathbb{E}\left[Z - \mathbb{E}\left[Z \mid W\right] + \mathbb{E}\left[Z \mid W\right] - f(W)\right]^2 \\ &= \mathbb{E}\left[Z - \mathbb{E}\left[Z \mid W\right]\right]^2 + \mathbb{E}\left[f(W) - \mathbb{E}\left[Z \mid W\right]\right]^2 \\ &\quad -2\mathbb{E}\left[Z - \mathbb{E}\left[Z \mid W\right]\right]\mathbb{E}\left[f(W) - \mathbb{E}\left[Z \mid W\right]\right].\end{aligned}$$

So we need to show that $\mathbb{E}\left[Z - \mathbb{E}\left[Z \mid W\right]\right]\mathbb{E}\left[f(W) - \mathbb{E}\left[Z \mid W\right]\right] = 0$.
We do this by using the tower property to insert a condition on W:

$$\begin{aligned}&\mathbb{E}\left[Z - \mathbb{E}\left[Z \mid W\right]\right]\mathbb{E}\left[f(W) - \mathbb{E}\left[Z \mid W\right]\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[Z - \mathbb{E}\left[Z \mid W\right]\right]\mathbb{E}\left[f(W) - \mathbb{E}\left[Z \mid W\right]\right] \mid W\right].\end{aligned}$$

From here we can pull out the latter expectation term, since it's a function of W, by (1.6). The former term left inside the expectation is 0, since

$$\begin{aligned}\mathbb{E}\left[Z - \mathbb{E}\left[Z \mid W\right] \mid W\right] &= \mathbb{E}\left[Z \mid W\right] - \mathbb{E}\left[\mathbb{E}\left[Z \mid W\right] \mid W\right] \\ &= \mathbb{E}\left[Z \mid W\right] - \mathbb{E}\left[Z \mid W\right] = 0\end{aligned}$$

using the tower property once more.

□

We'll introduce terminology in the next section that means that this result is saying that the hypothesis $h : \mathcal{X} \to \mathbb{R}$ minimising $R(h)$ under squared error loss is $h_0(x) = \mathbb{E}\left[Y \mid X = x\right]$, and nothing deeper than this (by considering $Z = Y$ and $X = W$).

Theorem 1.8 (Conditional Jensen)
Given convex $f : \mathbb{R} \to \mathbb{R}$,

$$\mathbb{E}\left[f(Z) \mid W\right] \geq f(\mathbb{E}\left[Z|W\right]). \tag{5}$$

To remember which way round the inequality goes, do a sanity check of W constant and $f(x) = x^2$ (variance is non-negative).

§1.2 Terminology

We will set up definitions in order to solve classification problems like MNIST or spam detection.

Consider a pair of random variables $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ with joint distribution P_0. We call X the **input** or **predictor** and Y the **output** or **response**.

Our goal is to predict Y from X. We do this via a **hypothesis**[4] $h : \mathcal{X} \to \mathcal{Y}$, and measure the quality of the prediction using a **loss function** $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$.

We can be in the **classification** setting where $\mathcal{Y} = \{-1, 1\}$, and typically ℓ is the **misclassification loss**, or '0-1 loss' $\ell(h(x), y) = \mathbb{1}\left[h(x) \neq y\right]$. Here, we refer to h as a **classifier**.

Alternatively, we can be in the **regression** setting, where $\mathcal{Y} = \mathbb{R}$ and typically ℓ is the squared error: $\ell(h(x), y) = (h(x) - y)^2$.

Our aim is to pick h with small **risk**

$$R(h) = \mathbb{E}[\ell(h(X), Y) \mid h]. \tag{6}$$

We have the conditioning over h as we consider the classifier to be fixed; h will be generally constructed from some random data, and this takes that into account.

A classifier h_0 that minimises the 0-1 risk is called a **Bayes classifier**. Its associated risk is the **Bayes risk**.

Define the **regression function** η as

$$\eta(x) = \mathbb{P}(Y = 1 \mid X = x). \tag{7}$$

Note that in practise we don't have 'access' to η, since to know η we need to know the joint distribution to evaluate it. When minimising Bayes risk, we're essentially minimising *population-wide* risk. We'll go on to study empiricial risk minimisation (ERM), where we minimise risk with respect to our training data, or data we observe empirically.

[4] the use of the term is unlike the use in 'hypothesis testing' from statistics.

Theorem 1.9

A Bayes Classifier is given by

$$h_0(x) = \begin{cases} 1 & \text{if } \eta(x) > 1/2 \\ -1 & \text{otherwise.} \end{cases} \tag{8}$$

Proof. This is not a deep result. The 0-1 risk is just the dumb function '0 if we were right, 1 if we were wrong' so the risk, given $X = x$ is

$$\begin{aligned} R(h(x)) &= \mathbb{1}[h(x) = 1]\,\mathbb{P}[Y = -1 \mid X = x] + \mathbb{1}[h(x) = -1]\,\mathbb{P}[Y = 1 \mid X = x] \\ &= \mathbb{1}[h(x) = 1](1 - \eta(x)) + \mathbb{1}[h(x) = -1]\,\eta(x). \end{aligned}$$

Now we want to minimise this, so just do casework on $\eta(x) < \frac{1}{2}$, $\eta(x) = \frac{1}{2}$ and $\eta(x) > \frac{1}{2}$. In each case our Bayes classifier will be optimal. □

In reality, we have **training data** that is a set of iid copies (X_1, Y_1), (X_2, Y_2), $\cdots$, (X_n, Y_n) of (X, Y). We want to use this to construct $\hat{h}$ minimising $R(\hat{h})$. Therefore the conditioning in (6) will be on this training data, when considering $R(\hat{h})$.

The classical statistical approach is to model P_0 using a parametric family. In this approach, we need estimate these unknown parameters.

The machine learning approach is that we're given a class $\mathcal{H}$ from which we will then pick $\hat{h}$. In this approach, we will need an algorithm for picking $\hat{h}$.

Definition 1.10 (Sign)**.** sgn is the sign (± 1) of a real, and we define $\operatorname{sgn}(0) = -1$ in this course.

For this reason, in the two-class classification problem, we will label the classes with $+1$ and -1 rather than 0 and 1, as seen elsewhere.

Example 1.11 (Examples of $\mathcal{H}$)

$$\mathcal{H} = \{x \mapsto \operatorname{sgn}(\mu + x^T\beta)\} \tag{9}$$

where $\mu \in \mathbb{R}$ and $\beta \in \mathbb{R}^p$ is one example of a class. Another is

$$\mathcal{H} = \left\{ x \mapsto \operatorname{sgn}\left(\sum_j \phi_j(x) w_j \right) \right\} \tag{10}$$

where $w \in \mathbb{R}^d$ and $\phi_j \in \mathbb{B}$ for a given class of functions $\mathbb{B} = \{f : \mathcal{X} \to \mathbb{R}\}$. Note that these classes have, in general, many degrees of freedom.

§2 Statistical Learning Theory

As alluded to earlier, minimising population-wide risk is generally not practical nor possible when applying ML methods. So almost all ML methods aim to minimise **empirical risk**:

Definition 2.1 (Empirical Risk)**.** Empirical risk or **training error** is the expectation of the loss $\ell(h(X), Y)$ where (X, Y) follows the *empirical* distribution (which will generally (always?) be some number of iid samples from P_0):

$$\hat{R}(h) = \frac{1}{n}\sum_i \ell(h(X_i), Y_i) \tag{11}$$

Given some class $\mathcal{H}$ of hypotheses, the argmin over this set with respect to this empirical risk is called the empirical risk minimiser.

We've used a lot of words, but there are immediate examples that this setup allows us to describe:

Example 2.2 (Least squares regression from IB Statistics)

In this familiar setting, our class of hypotheses is

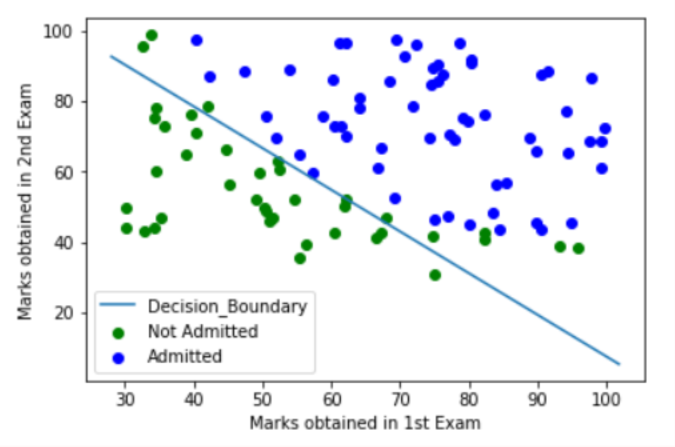
$$\mathcal{H} = \{x \mapsto (\mu + x^T\beta)\} \tag{12}$$

where $\mu \in \mathbb{R}$ and $\beta \in \mathbb{R}^p$, and our loss function is squared error

$$\ell(X_i, Y_i) = (Y_i - \mu - X_i^T\beta)^2. \tag{13}$$

Example 2.3 (0-1 classification)

If $\mathcal{Y} = \{-1, +1\}$, $\mathcal{H} = \{x \mapsto \text{sgn}\mu + x^T\beta\}$ (see (9)) and our loss is 0-1 loss, then what we're doing here is finding a hyperplane that divides space into two regions, one for each of -1 and 1, and the empirical loss will be the number of 1s in the -1 region plus the number of -1s in the 1 region, all over n. Visually:



The hyperplane in this case is known as the *decision boundary*.

Definition 2.4 (Aitch, aitch and aitch)**.** Let $\hat{h} \in \mathcal{H}$ be the hypothesis minimising empirical risk.

Let $h^* \in \mathcal{H}$ be the hypothesis minimising (population) risk over all $\mathcal{H}$.

Let h_0 be the function minimising (population) risk over all functions $h : \mathcal{X} \mapsto \mathcal{Y}$.

Note that h_0, the 'God hypothesis' won't be a perfect predictor, since our data comes from P_0, an inherently random distribution. We can now consider

$$R(\hat{h}) - R(h_0) = \underbrace{R(\hat{h}) - R(h^*)}_{\substack{\textbf{Excess risk} \text{ from our} \\ \text{choice within hypothesis class.}}} + \underbrace{R(h^*) - R(h_0)}_{\substack{\textbf{Appoximation error} \text{ from our} \\ \text{choice of hypothesis class.}}} \tag{14}$$

where the terms have the given interpretations. Furthermore, the first term can be seen as a measure of how much we overfit to the training data, and the latter term how much we underfit the underlying distribution by choosing too restrictive a $\mathcal{H}$.

Write

$$\begin{aligned} R(\hat{h}) - R(h^*) &= R(\hat{h}) - \hat{R}(\hat{h}) + \underbrace{\hat{R}(\hat{h}) - \hat{R}(h^*)}_{\leq 0} + \hat{R}(h^*) - R(h^*) \quad (15) \\ &\leq R(\hat{h}) - \hat{R}(\hat{h}) + \hat{R}(h^*) - R(h^*). \quad (16) \end{aligned}$$

where the middle term is non-positive since we chose $\hat{h}$ to be optimal on the training data.

We've interpreted all these terms already, except $\hat{R}(h^*)$. This is the risk of the optimal classifier with respect to all $\mathcal{H}$, but when it can only see the training data. So it is another measure of how much we are overfitting.

§2.1 Sub-Gaussianity and Hoeffding's Inequality

Recall Markov's inequality from 1A Probability. Note that this is a simple inequality, but has a nice consequence (on an example sheet from that course).

Theorem 2.5 (Chernoff Bound)

Let W be a random variable and $\alpha > 0$. Then

$$\mathbb{P}\left[W \geq t\right] \leq e^{-\alpha t}\mathbb{E}\left[e^{\alpha W}\right] \tag{17}$$

Proof. For any increasing function $\phi : \mathbb{R} \to [0, \infty)$, $W \geq t$ implies $\phi(W) \geq \phi(t)$, so in this case

$$\mathbb{P}\left[W \geq t\right] \leq \mathbb{P}\left[e^{\alpha W} \geq e^{\alpha t}\right] \leq e^{-\alpha t}\mathbb{E}\left[e^{\alpha W}\right] \tag{18}$$

by Markov's inequality, as required. □

This is important since this has introduced an MGF.

Example 2.6 (A tail bound for Gaussian random variables)

Suppose $W \sim N(0, \sigma^2)$, so has MGF $\mathbb{E}\left[e^{\alpha W}\right] = \exp\left(\frac{1}{2}\alpha^2\sigma^2\right)$. Then for $t > 0$,

$$\mathbb{P}\left[W \geq t\right] \leq \exp\left(-\alpha t + \frac{1}{2}\alpha^2\sigma^2\right). \tag{19}$$

Taking the infinimum over all $\alpha > 0$, we get

$$\mathbb{P}\left[W \geq t\right] \leq \exp\left(-t/2\sigma^2\right). \tag{20}$$

This derivation essentially uses bounding with MGFs. This motivates

Definition 2.7 (Sub-Gaussian)**.** A random variable W is **sub-Gaussian** with parameter $\sigma > 0$ if

$$\mathbb{E}\left[\exp\left(\alpha(W - \mathbb{E}\left[W\right])\right] \leq \exp\left(\frac{1}{2}\alpha^2\sigma^2\right) \tag{21}$$

for every $\alpha \in \mathbb{R}$.

i.e. after normalising to mean 0, the MGF of W is always less than the MGF of a variance $N(0, \sigma^2)$ random variable.

Example 2.8 (Basic properties of sub-Gaussian random variables)

Let W be a sub-Gaussian random variable with parameter $\sigma > 0$. Then

- W is sub-Gaussian for all $\sigma' \geq \sigma$.
- $-W$ is also sub-Gaussian.
- $\mathbb{P}\left[W - \mathbb{E}\left[W\right] \geq t\right] \leq \exp\left(-t^2/2\sigma^2\right)$.
- $\mathbb{P}\left[|W - \mathbb{E}\left[W\right]| \geq t\right] \leq 2\exp\left(-t^2/2\sigma^2\right)$

Proof. The first two remarks are immediate. To see the third, apply the Chernoff bound since it directly introduces an MGF. The fourth is then a corollary of the third, using the second.

The third property is called the *sub-Gaussian tail bound*, and we will repeatedly use it. □

Definition 2.9. A *Rademacher random variable* ε takes values ±1 with equal probability.

Theorem 2.10

A Rademacher random variable is sub-Gaussian with $\sigma = 1$.

Proof. Directly compute

$$\begin{aligned}\mathbb{E}\left[e^{\alpha\varepsilon}\right] = \frac{1}{2}(e^{\alpha} + e^{-\alpha}) &= \sum_{k=0}^{\infty} \frac{\alpha^{2k}}{(2k)!} \\ &\leq \sum_{k=0}^{\infty} \left(\frac{\alpha^2}{2}\right)^k \frac{1}{k!} \\ &= e^{\alpha^2/2}.\end{aligned}$$

□

How does this relate to ERM? This brings us onto the first deeper result of the course.

Theorem 2.11 (Hoeffding's Lemma)

If W takes values in an interval $[a, b]$, then W is sub-Gaussian with parameter $(b-a)/2$.

Proof. We will prove the weaker result with $\sigma = b - a$, using a technique called *symmetrisation*. The full result will be on the example sheet, using a *change of measure* argument[a].

To check whether a random variable is sub-Gaussian we normalize so the mean is zero, so WLOG assume $\mathbb{E}[W] = 0$.

We want to use the results already established for Rademacher random variables, and to do this we cook up W' an independent copy of W so that the random variable $W - W'$ is symmetric about 0, meaning that $\varepsilon(W - W')$ and $W - W'$ have the same distribution (written $W - W' \overset{d}{=} \varepsilon(W - W')$). We write

$$\mathbb{E}\left[e^{\alpha\varepsilon(W-W')} \mid W, W'\right] \le \exp\left(\alpha(W-W')^2/2\right) \le \exp\left(\alpha^2(b-a)^2/8\right), \tag{22}$$

where conditioning on W, W' effectively fixes these random variables, allowing us to use the Rademacher result.

We relate this back to W by noting that the familiar linearity of expectation property

$$\mathbb{E}[X + Y \mid Z] = \mathbb{E}[X \mid Z] + \mathbb{E}[Y \mid Z] \tag{23}$$

(where all the expectations are defined) holds, so we have that

$$\alpha(W - \mathbb{E}[W]) = \mathbb{E}\left[\alpha(W - W') \mid W\right]. \tag{24}$$

Therefore

$$\begin{aligned}\mathbb{E}[\exp(\alpha W)] &= \mathbb{E}\left[\exp\left(\alpha(W - \mathbb{E}\left[W'\right])\right)\right] && (25)\\ &= \mathbb{E}\left[\exp\left(\mathbb{E}\left[\alpha(W-W') \mid W\right]\right)\right] && (26)\\ &\le \mathbb{E}\left[\exp\left(\alpha(W-W')\right)\right]. && (27)\end{aligned}$$

where in the last step we apply conditional Jensen, and then simplify via the tower property in the degenerate case conditioning on some constant random variable, to go from to expectation signs to just one.

Finally, we can use the tower property once more to write the equality

$$\mathbb{E}\left[\exp\left(\alpha(W-W')\right)\right] = \mathbb{E}\left[\mathbb{E}\left[\exp\left(\alpha\varepsilon(W-W')\right) \mid W, W'\right]\right] \tag{28}$$

where to be specific, we choose the f in the tower property statement to be some degenerate (e.g constant) function to complete the argument (in essence we omit something like '... | 42]' at the end of both of the above expectation expressions). We've recovered (22), so we're done.

□

[a] see [2], also.

Theorem 2.12

Suppose $W_1, \dots , W_n$ are independent and each W_i is sub-Gaussian with parameters $\sigma_i > 0$. Then for all $\gamma \in \mathbb{R}^n$, $\gamma^T W$ is sub-Gaussian with parameter

$$\sqrt{\sum_i \gamma_i^2 \sigma_i^2}. \tag{29}$$

Proof. As before, WLOG $\mathbb{E}[W_i] = 0$. Then consider the MGF of $\gamma^T W$, which factorises via independence.

□

Theorem 2.13 (Hoeffding's Inequality)

Suppose $W_1, \dots, W_n$ are independent and bounded random variables, with $a_i \le W_i \le b_i$.

$$\mathbb{P}\left[\frac{1}{n}\sum_i (W_i - \mathbb{E}[W_i]) \ge t\right] \le \exp\left(\frac{-2n^2t^2}{\sum_i (a_i - b_i)^2}\right) \tag{30}$$

Proof. This is in the form of (2.12), so apply (2.8). □

Theorem 2.14 (Upper bound on sub-Gaussian, mean zero random variables)

Suppose $W_1, \dots, W_d$ are sub-Gaussian random variables with mean 0 and parameter $\sigma > 0$. Then

$$\mathbb{E}[\max W] \le \sigma\sqrt{2\log d}. \tag{31}$$

Proof. We see sub-Gaussian things, so we gravitate towards an MGF. For $\alpha > 0$,

$$\begin{aligned}\mathbb{E}[\max(\exp(\alpha W))] &= \mathbb{E}[\exp(\alpha \max W)] && (32)\\ &\ge \exp(\alpha \mathbb{E}[\max W]) && (33)\end{aligned}$$

by applying Jensen. So we can concern ourselves with that initial expression, and crudely bound the maximum with a sum (when we can't make further progress because we make almost no distributional assumptions in this course, we will often do this 'union bound' trick):

$$\mathbb{E}[\max(\exp(\alpha W))] \le \sum \mathbb{E}[\exp(\alpha W)] \le d\exp\left(\alpha^2\sigma^2/2\right) \tag{34}$$

now this works for any α, and after rearranging and finding the α that gives us the sharpest inequality, we get the result.

□

It's worth noting that while this result makes no distributional assumptions on the W_i, it is 'more striking' in some sense when the W_i are independent; where the variables are well correlated, the max won't differ too much from, say, W_1.

§2.2 Finite Hypotheses Classes

We have developed enough theory to prove an important bound on the risk of the ERM.

Theorem 2.15 (Finite hypothesis class excess risk bound)

Suppose $\mathcal{H}$ is finite and ℓ takes values in $[0, M]$. Then with probability at least $1-\delta$, the ERM $\hat{h}$ satisfies

$$R(\hat{h}) - R(h^*) \leq M\sqrt{\frac{2(\log|\mathcal{H}| + \log(1/\delta))}{n}}. \tag{35}$$

Proof. In the notation of Hoeffding's inequality (2.13), we have $a_i = 0$ and $b_i = M$, and the RHS of the bound of that inequality is therefore $\exp\left(-2nt^2/M^2\right)$.

Recall we can write

$$R(\hat{h}) - R(h^*) = R(\hat{h}) - \hat{R}(\hat{h}) + \underbrace{\hat{R}(\hat{h}) - \hat{R}(h^*)}_{\leq 0} + \hat{R}(h^*) - R(h^*) \tag{36}$$

(where the inequality holds since $\hat{h}$ is optimal over the training data).

Let $t > 0$. Then

$$\mathbb{P}\left[R(\hat{h}) - R(h^*) > t\right] = \mathbb{P}\left[R(\hat{h}) - R(h^*) > t, \hat{h} \neq h^*\right] \tag{37}$$

$$\leq \mathbb{P}\left[R(\hat{h}) - \hat{R}(h^*) > t/2, \hat{h} \neq h^*\right] + \mathbb{P}\left[\hat{R}(h^*) - R(h^*) > t/2\right] \tag{38}$$

since the two last events imply the former event. The latter term is actually pretty much in the form of Hoeffding's inequality, specifically

$$\mathbb{P}\left[\hat{R}(h^*) - R(h^*) > t/2\right] = \mathbb{P}\left[\frac{1}{n}\sum \ell(h^*(X_i), Y_i) - \mathbb{E}\left[\ell(h^*(X_i), Y_i)\right] > t/2\right] \tag{39}$$

$$\leq \exp\left(-\frac{nt^2}{2M^2}\right) \tag{40}$$

by that result. For the first term, we have to get a bit messy (as we have already done by excluding h^* as we did) to be able to get a clean final form, and introduce $\mathcal{H}^- = \mathcal{H} \setminus \{h^*\}$. When $\hat{h} \in \mathcal{H}^-$, $R(\hat{h}) - \hat{R}(\hat{h}) \leq \max_{h\in\mathcal{H}^-} R(h) - \hat{R}(h)$. Now we know nothing about h^* and therefore have to use crude union bounding once more:

$$\mathbb{P}\left[R(\hat{h}) - \hat{R}(h^*) > t/2, \hat{h} \neq h^*\right] \leq \mathbb{P}\left[\max_{h\in\mathcal{H}^-} R(h) - \hat{R}(h) > t/2\right] \tag{41}$$

$$= \mathbb{P}\left[\bigcup_{h\in\mathcal{H}^-} \{R(h) - \hat{R}(h) > t/2\}\right] \tag{42}$$

$$\leq \sum_{h\in\mathcal{H}^-} \mathbb{P}\left[R(h) - \hat{R}(h) > t/2\right] \tag{43}$$

$$\leq (|\mathcal{H}| - 1)\exp\left(-\frac{nt^2}{2M^2}\right) \tag{44}$$

So

$$\mathbb{P}\left[R(\hat{h}) - R(h^*) > t\right] \leq |\mathcal{H}|\exp\left(-nt^2/2M^2\right) = \delta \tag{45}$$

and we get the result by rearranging this for t. □

Note that a similar bound could be found by applying the central limit theorem, since our setup involves a bunch of iid random variables. However, our result is not asymptotic, unlike the limit theorems we've seen before.

This result is something to be positive about: in loose terms, even if $\mathcal{H}$ is pretty large, we don't need that much training data to ensure our ERM has low excess risk.

Example 2.16 (The histogram classifier)

Consider the classification setting $\mathcal{X} = [0,1)^2$. Divide $[0,1)^2$ into m^2 disjoint squares R_0, ... , R_{m^2-1} where

$$R_{im+j} = \left[\frac{i}{m}, \frac{i+1}{m}\right) \times \left[\frac{j}{m}, \frac{j+1}{m}\right). \tag{46}$$

Also let

$$\bar{Y}_j = \operatorname{sgn} \sum_{i:X_i \in R_j} Y_i \tag{47}$$

i.e. these output what the majority of the points in each square are (there's annoying mismatch between -1 and 1, and 0 and 1 here). And finally

$$\hat{h}^{\text{hist}}(x) = \sum_{j=0}^{m^2-1} \hat{Y}_j \mathbb{1}\left[x \in R_j\right] \tag{48}$$

i.e. classify based on plurality of training data in the region that the test data lands in.

With $\mathcal{H}$ being the size 2^{m^2} set of hypotheses classifying based on which $1/m$ square a point falls in, we can bound the risk with (2.15).

It can be shown that we approach the Bayes classifier in this scenario, as we increase m (in some limit scenario).

§2.3 Infinite Hypotheses Classes

To handle the case where the hypothesis class is infinite, we turn maxima into suprema: let

$$G(Z_1, ..., Z_n) = \sup_{h \in \mathcal{H}} R(h) - \hat{R}(h) \tag{49}$$

where $Z_i = (X_i, Y_i)$. This is a direct generalisation of our previous work, but what is more problematic when $|\mathcal{H}| = \infty$ is that our usual union bounding trick will fail miserably.

However a key property that we used before, when G was an average, is that when we have a bunch of iid random variables $Z_1, ..., Z_n \in \mathcal{X} \times \mathcal{Y}$, each of these does not contribute greatly to the the average.

Motivated by this, let $\varepsilon > 0$ let $\tilde{h} \in \mathcal{H}$ be an 'ε-good' hypothesis in the sense that

$$G(z_1, ..., z_n) < R(\tilde{h}) - \hat{R}(\tilde{h}) + \varepsilon \tag{50}$$

Consider perturbing WLOG the first argument of G. Then (writing out everything...)

$$G(z_1, ..., z_n) - G(z_1', z_2, ..., z_n) < \frac{1}{n}\{\ell(y_1', \tilde{h}(x_1')) - \ell(y_1, \tilde{h}(x_1))\} + \varepsilon \tag{51}$$

which is a formalisation of the intuitively clear idea that individual data points do not affect the 'global' gap in excess risk too much (due to the factor of $1/n$).

In fact if the loss takes values in $[0, M]$, and $\varepsilon > 0$ is arbitrary,

$$|G(z_1, ..., z_n) - G(z_1, ..., z_i', ..., z_n)| \leq \frac{M}{n}. \tag{52}$$

for an perturbed i. Such an inequality is called a **bounded differences property**.

§2.4 Bounded Differences Inequality

For our next result, we will need the following notation and definition:

For a sequence $a_s, a_{s+1}, ...$ write $a_{j:k}$ for the **subsequence** $a_j, ..., a_k$.

Definition 2.17 (Martingale difference sequence)**.** A sequence of random variables $D_1, ..., D_n$ is a **martingale difference sequence** with respect to another sequence of random variables $W_0, ..., W_n$ if, for $1 \leq i \leq n$,

1. $\mathbb{E}\left[|D_i|\right] < \infty$.
2. D_i is a function of $W_{0:i}$.
3. $\mathbb{E}\left[D_i \mid W_{0:i-1}\right] = 0$.

Example 2.18

If $D_1, ..., D_n$ are independent and mean zero and satisfy the first property, then they are a martingale difference sequence with respect to $c, D_1, ..., D_n$ where c is a deterministic constant.

We will first need two preliminary results to prove the bounded differences inequality

Theorem 2.19 (2.12 for martingale random variables)

Let $D_1, ..., D_n$ be a martingale difference sequence with respect to $W_0, ..., W_n$ such that

$$\mathbb{E}\left[\exp\left(\alpha D_i\right) \mid W_{0:i-1}\right] \leq \exp\left(\frac{1}{2}\alpha^2\sigma_i^2\right) \tag{53}$$

holds for all α and all i. Also let $\gamma \in \mathbb{R}^n$. Then $\sum_i \gamma_i D_i$ is sub-Gaussian with parameter $\sqrt{\sum_i \sigma_i^2\gamma_i^2}$.

Proof. By the tower property

$$\mathbb{E}\left[\exp\left(\alpha\sum_{i=1}^{n}\gamma_i D_i\right)\right] = \mathbb{E}\left[\mathbb{E}\left[\exp\left(\alpha\sum_{i=1}^{n-1}\gamma_i D_i\right)\exp\left(\alpha\gamma_n D_n\right) \mid W_{0:n-1}\right]\right] \tag{54}$$

and now we can use taking out what is known since $D_{0:n-1}$ is a function of $W_{0:n-1}$, which will leave the last term in the inner expectation as something our assumptions gave us control over:

$$\leq \mathbb{E}\left[\exp\left(\alpha\sum_{i=1}^{n-1}\gamma_i D_i\right)\exp\left(\frac{1}{2}\alpha^2\gamma_n^2\sigma_n^2\right)\right] \tag{55}$$

and at this point we can pull out that second factor and apply the same tower trick a further $n-1$ times to get

$$\leq \prod_{i=1}^{n}\exp\left(\frac{1}{2}\alpha^2\gamma_i^2\sigma_i^2\right) = \exp\left(\frac{1}{2}\alpha^2\sum_{i=1}^{n}\gamma_i^2\sigma_i^2\right). \tag{56}$$

□

Theorem 2.20 (Azuma-Hoeffding)

Let $D_1, ..., D_n$ be a martingale difference sequence with respect to $W_0, ..., W_n$. Suppose also that for each i we have random variables A_i and B_i that are bounds for D_i ($A_i \leq D_i \leq B_i$) that differ by at most L_i where the L_is are constant, and that A_i and B_i are also functions of $W_{0:i-1}$. Then for $t \geq 0$,

$$\mathbb{P}\left[\sum_{i=1}^{n} D_i \geq t\right] \leq \exp\left(-2t^2 \middle/ \sum_i L_i^2\right). \tag{57}$$

Proof. Conditional on $W_{0:i-1}$, A_i and B_i are fixed, so D_i is a bounded random variable and hence we're in the setting where we can apply Hoeffding's lemma with this conditioning. Therefore we have the moment generating function bound

$$\mathbb{E}\left[e^{\alpha D_i} \mid W_{0:i-1}\right] \leq \exp\left(\frac{1}{2}\alpha^2 \left(\frac{L_i}{2}\right)^2\right) \tag{58}$$

which means we can apply (2.19) to deduce that $\sum_{i=1}^{n} D_i$ is sub-Gaussian with parameter $\frac{1}{2}\sqrt{\sum_i L_i^2}$. So we can apply the sub-Gaussian tail bound and we get the desired result.

□

Theorem 2.21 (Bounded differences inequality)

Let $f : \mathcal{Z}_1 \times ... \times \mathcal{Z}_n \to \mathbb{R}$ satisfy a bounded differences property

$$f(w_{1:n}) - f(w_{1:i-1}, w_i', w_{i+1:n}) \leq L_i \tag{59}$$

$\forall w_1 \in \mathcal{Z}_1, ..., w_i' \in \mathcal{Z}_i, ..., w_n \in \mathcal{Z}_n$ where $1 \leq i \leq n$. Suppose random variables $W_1 \in \mathcal{Z}_1, ..., W_n \in \mathcal{Z}_n$ are independent. Then

$$\mathbb{P}\left[f(W_{1:n}) - \mathbb{E}\left[f(W_{1:n})\right] \geq t\right] \leq \exp\left(-2t^2 \Big/ \sum_i L_i^2\right). \tag{60}$$

Proof. Introduce the deterministic random variable $W_0 = c$ (c arbitrary and constant). Then we can turn our expression into a sum of a bunch of random variables as follows:

$$f(W_{1:n}) - \mathbb{E}\left[f(W_{1:n})\right] = \sum_i \underbrace{\mathbb{E}\left[f(W_{1:n}) \mid W_{0:i}\right] - \mathbb{E}\left[f(W_{1:n}) \mid W_{0:i-1}\right]}_{D_i}. \tag{61}$$

Since this will telescope, and the first term comes out through conditional expectation. Now D_i is a martingale difference sequence with respect to $W_0, ..., W_n$, since checking the definition,

1. f satisfies the bounded differences inequality so itself must be bounded (it will vary by at most $\sum L_i$).
2. Is clear.
3. Follows from the tower property, since $W_{0:i-1}$ is a function of itself, and also a function of $W_{0:i}$.

We now want to cook up A_i and B_i in the notation of Azuma-Hoeffding. Let $\mathcal{Z}_0 = \{c\}$. Define

$$F_i : \mathcal{Z}_0 \times ... \times \mathcal{Z}_i \to \mathbb{R} \tag{62}$$

by $(w_0, ..., w_i) \mapsto \mathbb{E}\left[f(W_{0:n}) \mid W_{0:i} = w_{0:i}\right]$ and so we have

$$D_i = F_i(W_{0:i}) - F_{i-1}(W_{0:i}). \tag{63}$$

Now force $A_i \leq D_i \leq B_i$ in the most blatant way possible:

$$A_i = \inf_{w_i} F_i(W_{0:i-1}, w_i) - F_{i-1}(W_{0:i-1}) \tag{64}$$

and let B_i be the corresponding supremum (recall these are supposed to be functions of $W_{0:i-1}$!). We now have to do a careful check that $B_i - A_i \leq L_i$ by a lot of symbol pushing: to begin note that

$$B_i - A_i = \sup_{w_i, w_i'} F_i(W_{0:i-1}, w_i) - F_i(W_{0:i-1}, w_i') \tag{65}$$

and writing the F expressions out and bringing the expectations together (using independence of all the $W_{0:i}$ as well as the Martingale independence of the future $W_{i+1:n}$ from the past, we get a quantity that's always at most L_i. □

This is a generalisation of Hoeffding's inequality, by taking $\mathcal{Z}_i = [a_i, b_i]$ and $f(W_{1:n}) = \frac{1}{n}\sum W_i$, where the boundedness of the random variables in Hoeffding's inequality give the bounded differences property.

§2.5 Rademacher Complexity

We still want to develop some more tools in the $|\mathcal{H}| = \infty$ case.

Recall the inequality (14). We can now introduce the supremum quantity G (49) to further write

$$R(\hat{h}) - R(h^*) \leq R(\hat{h}) - \hat{R}(\hat{h}) + \hat{R}(h^*) - R(h^*) \tag{66}$$

$$\leq \underbrace{G - \mathbb{E}[G]}_{\text{controllable by bounded differences (2.21)}} + \underbrace{\hat{R}(h^*) - R(h^*)}_{\text{controllable by Hoeffding's (2.13)}} + \mathbb{E}[G]. \tag{67}$$

so we need to figure out how to control $\mathbb{E}[G]$.

Now define the set of functions (indexed by hypotheses)

$$\mathcal{F} = \{(x, y) \overset{f}{\mapsto} -\ell(h(x), y) | h \in \mathcal{H}\} \tag{68}$$

(check the order of terms in G to verify that we do need a minus sign here) so that we can rewrite

$$G = \sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} f(Z_i) - \mathbb{E}[f(Z_i)]. \tag{69}$$

We're going to develop a result that controls terms like that in the summand.

For each f we get a mean zero random variable, and we're taking a supremum over these random variables. We've already established a result about the maximum of some mean zero *sub-Gaussian* random variables (2.14); $\mathbb{E}[\max W_i] \leq \sigma\sqrt{2\log d}$. But here d is infinite, so we can't (yet) apply it. So, more technology!

Definition 2.22 (More Rademacher terminology)**.** Let $\mathcal{F}$ be a class of functions $f : \mathcal{Z} \to \mathbb{R}$ and let $z_1, ..., z_n \in \mathcal{Z}$.

Then

$$\mathcal{F}(Z_{1:n}) = \{(f(Z_1), ..., f(Z_n)) | f \in \mathcal{F}\} \tag{70}$$

is the set of 'behaviours' that can be exhibited on $Z_{1:n}$. Then define the **empirical Rademacher complexity** as

$$\hat{\mathcal{R}}(\mathcal{F}(z_{1:n})) = \mathbb{E}\left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i f(z_i)\right]. \tag{71}$$

Given iid $Z_{1:n}$, we sometimes view the empirical Rademacher complexity as a random variable via conditioning:

$$\hat{\mathcal{R}}(\mathcal{F}(Z_{1:n})) = \mathbb{E}\left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i f(Z_i) \mid Z_{1:n}\right]. \tag{72}$$

Finally, define the **Rademacher complexity** of $\mathcal{F}$ as $\mathcal{R}_n(\mathcal{F}) = \mathbb{E}\left[\hat{R}(\mathcal{F}(Z_{1:n}))\right]$.

Theorem 2.23

Let $\mathcal{F}$ be a class of functions $f : \mathcal{Z} \to \mathbb{R}$ and let $Z_1, ..., Z_n \in \mathcal{Z}$ be iid random variables. Then

$$\mathbb{E}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} f(Z_i) - \mathbb{E}\left[f(Z_i)\right]\right] \le 2\mathcal{R}_n(\mathcal{F}). \tag{73}$$

Proof. Symmetrization.

Let $Z'_{1:n}$ be an independent copy of $Z_{1:n}$ (I don't think there is any assumption that the Z are all distributed in the same way (as training data probably would be). This is a more general situation). We can use this to 'bring the expectation out' of the inner term being expected over as follows, and crucially we end up with a symmetric distribution within the expectation:

$$\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} f(Z_i) - \mathbb{E}\left[f(Z_i)\right] = \sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\left[f(Z_i) - f(Z'_i)|Z_{1:n}\right] \tag{74}$$

$$\le \mathbb{E}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} f(Z_i) - f(Z'_i)|Z_{1:n}\right] \tag{75}$$

where in the first equality we use that $\mathbb{E}\left[X|X\right] = X$ and $\mathbb{E}\left[X|Y\right] = \mathbb{E}\left[X\right]$ where X and Y are independent, and in the inequality we use $\sup \mathbb{E}\left[-\right] \le \mathbb{E}\left[\sup -\right]$ (because each individual expectation is at most the expectation of the sup, so then we can take sup).

Let $\varepsilon_{1:n}$ be iid Rademachers. Magically,

$$\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} f(Z_i) - f(Z'_i) =^d \sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i(f(Z_i) - f(Z'_i)). \tag{76}$$

Now take expectation of the above inequality, and break up the sup as follows:

$$\le \mathbb{E}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i f(Z_i) + \sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i f(Z'_i)\right] \le 2\mathcal{R}_n(\mathcal{F}) \tag{77}$$

where we can turn the negative in positive due to the ε_i being symmetric. □

we can now state and prove the bound for the excess risk that this part of the course has been working towards:

Theorem 2.24

Let $\mathcal{F} = \{(x, y) \mapsto \ell(h(x), y) | h \in \mathcal{H}\}$ and suppose ℓ takes values in $[0, M]$. Then with probability at least $1 - \delta$,

$$R(\hat{h}) - R(h^*) \leq 2\mathcal{R}_n(\mathcal{F}) + M\sqrt{2\log(2/\delta)/n}. \tag{78}$$

Proof. Let $G = \sup_{h \in \mathcal{H}} R(h) - \hat{R}(h)$. We have our old friend, the bound

$$R(\hat{h}) - R(h^*) \leq G - \mathbb{E}[G] + \hat{R}(h^*) - R(h^*) + \mathbb{E}[G] \tag{79}$$

and now we know that that last term is at most $2\mathcal{R}_n(\mathcal{F})$. Now viewing G as a function of all the Z_i, it satisfies a bounded differences property with $L_i = M/n$ (52). So we can apply the bounded differences inequality directly:

$$\mathbb{P}[G - \mathbb{E}[G] \geq t/2] \leq \exp\left(-\frac{nt^2}{2M^2}\right). \tag{80}$$

Applying Hoeffding (or the generalisation; bounded differences with the average function) gives that

$$\mathbb{P}\left[\hat{R}(h^*) - R(h^*) \geq t/2\right] \leq \exp\left(-\frac{nt^2}{2M^2}\right). \tag{81}$$

Now we can pick the t values such that

$$\delta = 2\exp\left(-\frac{nt^2}{2M^2}\right) \tag{82}$$

and then the result falls out.

□

Recall the inequality (78), that involves the Rademacher complexity $\mathcal{R}_n(\mathcal{F})$.

The Rademacher complexity is computed by assigning uniformly random ±1 variables to all the empirical losses and then taking this sup, and hence loosely is a measure of how 'wild' the various ±1 strings we can get from $(f(z_1), ..., f(z_n))$ are (since if these are all similar, this expected value will be lower).

Theorem 2.25

$$\mathcal{R}_n(\mathcal{F}) = \mathbb{E}\left[\sup_{f \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i f(Z_i)\right] \tag{83}$$

$$\mathcal{R}_n(\mathcal{F}) = \mathbb{E}\left[\hat{R}(\mathcal{F}(Z_{1:n}))\right] = \mathbb{E}\left[\underbrace{\mathbb{E}\left[\sup_{f \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i f(Z_i) | Z_{1:n}\right]}_{\hat{R}(\mathcal{F}(Z_{1:n}))}\right] = \mathbb{E}\left[\sup_{f \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i f(Z_i)\right] \tag{84}$$

by using the degenerate case of the tower property, and (72).

□

Recall that if $W_{1:d}$ are mean zero σ-sub-Gaussian random variables then $\mathbb{E}[\max W] \leq$

$\sigma\sqrt{\log d}$. We can apply that to the inner expectation since the conditioning essentially fixes the $f(Z_i)$, and then we simply have a linear combination of Rademachers, which is sub-Gaussian by (2.12). Having done this conditioning, we are essentially concerned with bounding the size of the set of ±1 strings mentioned above.

§2.6 VC Dimension

Consider the classification setting with $0-1$ loss. We aim to bound $\mathcal{R}_n(\mathcal{F})$, where the function class is the set of loss maps again (68).

Note that if $x_{1:n} \in \mathcal{X}^n$ and class labels $y_{1:n} \in \{-1,+1\}^n$ and $z_i = (x_i, y_i)$ then $|\mathcal{F}(z_{1:n})| = |\mathcal{H}(z_{1:n})|$, because there's a bijection

$$(\ell(h(x_i), y_i))_{i=1}^n \leftrightarrow (h(x_i))_{i=1}^n \tag{85}$$

since the y_is are fixed and so we can just cross check these terms. This is essentially a bijection of multisets however, since two different hypotheses could be identical on the training data.

Theorem 2.26

$$\hat{\mathcal{R}}(\mathcal{F}(z_{1:n})) \leq \sqrt{\frac{2\log|\mathcal{F}(z_{1:n})|}{n}}. \tag{86}$$

Let $|\mathcal{F}(z_{1:n})| = d$ and let $\mathcal{F}' = \{f_1, ..., f_d\} \subseteq \mathcal{F}$ be the set of *distinct* behaviours. Given Rademacher random variables $\varepsilon_{1:n}$ set $W_j = \frac{1}{n}\sum_{i=1}^n \varepsilon_i f_j(z_i)$. Then $\hat{\mathcal{R}}(\mathcal{F}(z_{1:n})) = \mathbb{E}\left[\max_j W_j\right]$.

Each ε_i is sub-Gaussian with parameter 1, so we can apply the linear combination of sub-Gaussians result; W_j is sub-Gaussian with parameter $\sqrt{\frac{1}{n^2}\sum_{i=1}^n f_j^2(z_i)} \leq \frac{1}{\sqrt{n}}$ (the bound because we're dealing with $0-1$ loss). Also $\mathbb{E}[W_j] = 0$, and so by (2.14) we're done. □

Essentially, we've reduced the complicated world of probability to the more beautiful world of combinatorics, since this bound involves the size of a set.

How can we bound this count of the number of behaviours?

Definition 2.27. Let $\mathcal{F}$ be a class of functions $f : \mathcal{X} \to \{a, b\}$ and $|\mathcal{F}| \geq 2$. We say that f **shatters** $x_{1:n}$ if $|\mathcal{F}(x_{1:n})| = 2^n$ i.e every possibility of assigment occurs for some $f \in \mathcal{F}$.

Also define the **shattering coefficient** s of a class of functions by

$$s(\mathcal{F}, n) = \max_{x_{1:n}\in\mathcal{X}^n} |\mathcal{F}(x_{1:n})|. \tag{87}$$

Note that this must be bounded above by 2^n.

How do we interpret these? Shattering seems to mean our hypotheses leave every possibility open. The shattering coefficient asks us to find data points that lead to the wildest set of behaviours when we apply $\mathcal{F}$ to them.

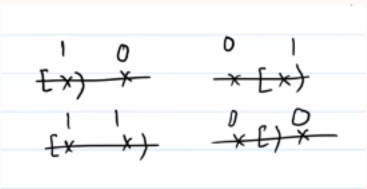
We need $s(\mathcal{F}, n)$ to be bounded by something better than 2^n since plugged into (2.26) this would give an n-independent bound.

Definition 2.28. Define the **VC dimension** $VC(\mathcal{F})$ to be the largest n such that there exists $x_{1:n}$ shattered by $\mathcal{F}$, or if no such n exists define $VC(\mathcal{F}) = +\infty$.

Example 2.29 (Shattering coefficient bound and VC dimension where $|\mathcal{H}| = \infty$)
Suppose $\mathcal{F} = \{\mathbb{1}\,[[a,b)]\,|a \leq b \in \mathbb{R}\}$, the set of $[a,b)$ windows, an infinite function class.

To compute the shattering coefficient of this class, consider $x_1 < x_2 < ... < x_n$ which divide the real line into $n+1$ intervals. Then our length n binary strings will turn out to be the same iff a and a' are in the same interval, and b and b' are in the same interval. So $s(\mathcal{F}, n) \leq \binom{n+1}{2} + 1$, where the $+1$ comes from choosing some really dumb interval that gives us an all zero string.

We can shatter all two points:



But can't shatter distinct $x_1 < x_2 < x_3$ since the binary string $\{1, 0, 1\}$ can't be generated by our simple interval indicator class. So $VC(\mathcal{F}) = 2$.

The lesson here is that despite being an infinite hypothesis class, $\mathcal{F}$ has a asymptotically sharp shattering coefficient bound.

Actually, we have the surprising result that shattering coefficients grow *polynomially* with n, which is strongly constrasted with the trivial 2^n bound.

Theorem 2.30 (Sauer-Shelah)
Let $\mathcal{F}$ have finite VC dimension d. Then

$$s(\mathcal{F},n) \leq \sum_{i=0}^{d} \binom{n}{i} \leq (n+1)^d. \tag{88}$$

Proof. Non-examinable.
First pick any $x_{1:n}$. We claim that there are at least $|\mathcal{F}(x_{1:n})|-1$ non-empty subsets of indices of $1:n$ such that $\mathcal{F}$ shatters these indices.

The result follows from this since we can specialise to the $x_{1:n}$ that achieves the shattering coefficient bound; where we'll have $|\mathcal{F}(x_{1:n})| = s(\mathcal{F},n)$, and then the binomial coefficent falls out since we can't shatter any size $> d$ subsets by definition of VC dimension.

The claim is true since we can induct on $k := |\mathcal{F}(x_{1:n})|$ (note we may assume the result holds for any n with this style of induction). The base case of size $k=1$ is vacuous. Assume we have the result for all $k' \leq k$. We'll show we get the result with $k+1$.

The idea is that since we assumed way back that $|\mathcal{F}| \geq 2$ always we can find an x_j such that we can decompose $\mathcal{F} = \mathcal{F}_+ \sqcup \mathcal{F}_-$ (here, $\mathcal{F} = \mathcal{F}(x_{1:n})$ has size $k+1$) into non-empty subsets such that the first always classifies x_j as 1, and the latter always as -1. Then the non-empty condition means we are now able to apply an induction procedure. We just do clever counting.

$|\mathcal{F}| = |\mathcal{F}_+| + |\mathcal{F}_-|$ and hence we have two positive integers summing to $k+1$ and hence the inductive hypothesis gives that the sets $\mathcal{X}_+$ and $\mathcal{X}_-$ of vectors shattered the two subsets have sizes summing to at least $k-1$ (we can't do better due to the -1 in hypothesis). Then for everything in the intersection of $\mathcal{X}_+$ and $\mathcal{X}_-$, we can add x_j to these subsets of $x_{1:n}$ and get a genuinely new shattered vector. Finally the singleton (x_j) can be shattered by the non-empty condition. We can put this together to get at least

$$1 + |\mathcal{X}_+ \cap \mathcal{X}_-| + |\mathcal{X}_+ \cup \mathcal{X}_-| = |\mathcal{X}_+| + |\mathcal{X}_-| + 1 \geq (k-1) + 1 \geq k \tag{89}$$

subsets shattered by $\mathcal{F}$. So we're done.

□

Corollary 2.31

$$\mathcal{R}_n(\mathcal{F}) \leq \sqrt{\frac{2VC(\mathcal{F})\log(n+1)}{n}} \tag{90}$$

(recall (2.26) and shattering definition).

Example 2.32

Let $\mathcal{X} = \mathbb{R}^p$ and consider the class $\mathcal{F} = \{\mathbb{1}[A] \mid A \in \mathcal{A}\}$ where

$$\mathcal{A} = \left\{ \prod_{j=1}^{p} (-\infty, a_j] \mid a_1, ..., a_p \in \mathbb{R} \right\}. \tag{91}$$

We claim that $VC(\mathcal{F}) = p$.

After unpacking definitions, it is clear that we can shatter n points.

It's slightly less easy to see that we can't shatter $p+1$ points. After checking the small cases for n, it's clear that we have some point which is not 'extreme' in some direction (or at very least, is not *uniquely* extreme in some direction). Then we can cook up a binary string with 0 in the place of this entry, and 1 everywhere else, and we can't get this behaviour.

Theorem 2.33

Let $\mathcal{F}$ be a vector space of functions. Then we can consider the class of classifiers $\mathcal{H} = \{\text{sgn} \circ f \mid f \in \mathcal{F}\}$. Then

$$VC(\mathcal{H}) \leq \dim \mathcal{F}. \tag{92}$$

Proof. We can first note that this generalises the previous result (I think I could cook up things to make this work with some zero product thing).

Let $d = \dim \mathcal{F} + 1$ and take $x_{1:d} \in \mathcal{X}^d$. We need show that $x_{1:d}$ cannot be shattered by $\mathcal{H}$.

Consider the linear map $L : \mathcal{F} \to \mathbb{R}^d$ defined by $f \mapsto (f(x_1), ..., f(x_d))$. Then the image dimension is at most $d-1$ by rank-nullity. From here, take $\gamma \neq 0$ orthogonal to this image space. Then break γ down in to its positive and non-negative components; let $\gamma_i > 0$ for all $i \in I_+$ and $\gamma_i < 0$ for all $i \in I_-$. Then

$$\sum_{i \in I_+} \gamma_i f(x_i) + \sum_{i \in I_-} \gamma_i f(x_i) = 0 \tag{93}$$

holds for all $f \in \mathcal{F}$. Then the behaviour where $f(x_i) = \pm 1$ on $I_\pm$ cannot be observed since if so, the LHS of (93) would be positive. □

Example 2.34

Consider $\mathcal{X} = [0,1)^2$ and $\mathcal{F}$ to be the set of polynomials of degree at most d, and define $\mathcal{H}$ as the set of signs of these polynomials as seen in (2.33).

Then by stars and bars, $\dim \mathcal{F} = \binom{d+2}{2}$. So if $d = 5$ then $VC(\mathcal{H}) \leq 21$ and previous results (namely, (78) and the VC bound (2.6)) imply

$$R(\hat{h}) - R(h^*) \leq 2\sqrt{\frac{2 \times 42 \log(n+1)}{n}} + \sqrt{\frac{2\log(2/\delta)}{n}}. \tag{94}$$

Comparing to the histogram classifier (2.16), with the finite hypothesis class bound, we had

$$R(\hat{h}) - R(h^*) \leq \sqrt{\frac{2m^2 \log 2 + 2\log(1/\delta)}{n}} \tag{95}$$

where *importantly*, the two h^*s are different: we're dealing with two different hypothesis classes.

with probability at least $1 - \delta$.

and with that example over, we finish the most stats-heavy part of the course. Phew!

§3 Computation for ERM

It turns out that the discontinuity of 0-1 loss means computation of the ERM is computationally intractable. We will therefore adjust our theory to work with **convex** loss functions that will still have values in $[0, M]$. This will mean that our result (2.24) will hold, although we will need to figure out what Rademacher complexity become with this new loss function.

§3.1 Convex sets

Definition 3.1. $C \subset \mathbb{R}^d$ is **convex** if all lines segments lie entirely in C.

Example 3.2 (Basic properties of convex sets)

Intersections of convex sets are themselves convex.

Definition 3.3. The **convex hull** of $S \subset \mathbb{R}^d$, written conv$S$ is the intersection of all convex sets containing S. By the above, it is unsurprisingly convex.

Definition 3.4. $v \in \mathbb{R}^d$ is a **convex combination** of $v_1, ..., v_m \in \mathbb{R}^d$ if the α are non-negative and sum to 1 and

$$v = \alpha_1 v_1 + ... + \alpha_m v_m. \tag{96}$$

Lemma 3.5 (Convex combinations ... are what you think they are)
For $S \subset \mathbb{R}^d$, $v \in \text{conv}S$ iff v is a convex combination of some set of points in S.

Proof. Let D be the set of all convex combinations of points from S.

Then $\text{conv}S \subset D$ is on the example sheet.

To show $D \subset \text{conv}S$ induct on the number m of non-zero α terms appearing in (96). The case $m = 1$ is clear. Then for $m + 1$ non-zero terms WLOG making the indices nice,

$$v = \alpha_1 v_1 + ... + \alpha_{m+1} v_{m+1} = t\left(\frac{\alpha_1}{t} v_1 + ... + \frac{\alpha_m}{t} v_m\right) + (1 - t)v_{m+1}. \tag{97}$$

and terms here must indeed lie in $\text{conv}S$ by its convexity. □

Theorem 3.6
Let $S \subset \mathbb{R}^d$. For any linear map $L : \mathbb{R}^d \to \mathbb{R}^m$, $\text{conv}L(S) = L(\text{conv}S)$.

Proof. As before, intuitively clear. To formalise, use the convex combination characterisation and work both ways. □

§3.2 Convex functions

We define (strictly) convex functions as we did in IB Optimisation.

Definition 3.7. Let $C \subset \mathbb{R}^d$ be convex. Then $f : C \to \mathbb{R}$ is **convex** if

$$f(tx + (1 - t)y) \leq tf(x) + (1 - t)f(y) \tag{98}$$

$\forall t \in (0, 1)$, and $x \neq y \in C$. f is *strictly convex* if the inequality is strict.

Example 3.8 (Properties of convex functions)
Unsurprisingly, convex functions are closely related to convex sets. The following results are clear and/or on the second example sheet.

- If f is convex then $D = \{x \in C \mid f(x) \leq M\}$ is convex.
- The level sets of convex functions are convex sets.
- An important case of the first point is when the convex function is when applied to norm functions, which are always convex.
- If f is C^2 then f is convex iff its Hessian matrix is positive semi-definite. This is because of the supporting hyperplane (I think).

Definition 3.9. The **epigraph** of a convex function is the set of points

$$C = \{(z, y) \in \mathbb{R}^d \times \mathbb{R} : y \geq f(z)\}. \tag{99}$$

Why do we care about convex functions? It's because they allow *global* properties to be deduced from *local* properties; knowing $f(x)$ and $f(y)$ allows us to know a lot of things about all the values on the line segment joining x and y. Local maxima will also be global maxima, obviously important for machine learning.

§3.3 Convex surrogates

Consider $\mathcal{H} = \{x \mapsto \text{sgn}\beta^T x \mid \beta \in \mathbb{R}^p\}$. To compute the ERM we need minimise (over β)

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left[Y_i \neq \text{sgn}\beta^T X_i\right] \approx \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left[Y_i\beta^T X_i \in (-\infty, 0]\right] \tag{100}$$

where we have $\approx$ because of the annoying case where we defined $\text{sgn}0 = -1$.

If we replace the indicator function $\mathbb{1}\left[(-\infty, 0]\right]$ with a convex function, the resulting problem will become a convex optimisation problem.

Take the hypothesis class $\mathcal{H}$ to not be messy discontinuous functions but a family of real-valued functions. Then we could always recover a classifier post-composing with sgn.

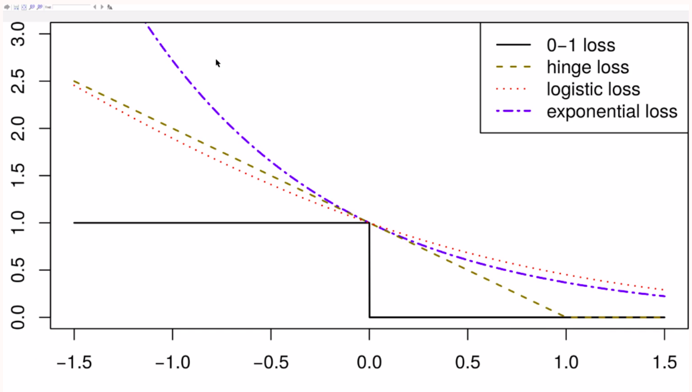
We shall consider losses of the form $\ell(h(x), y) = \phi(yh(x))$ where $\phi : \mathbb{R} \to [0, \infty)$ is convex. The associated risk is called the **ϕ-risk**.

What shall we choose for ϕ? Think about what properties are desirable. We are classifying correctly when $h(x)$ and y have the same sign, and incorrectly classifying otherwise. This sort of situation is very similar to the setup for the barrier method in IB Optimisation.

Here are a few choices of ϕ:

Example 3.10 (Convex surrogate loss functions)

All of the following loss functions are candidates for $\phi(h)$:



- Hinge loss $\max(1 - h, 0)$.
- Exponential loss $\exp\left(() - u\right)$.
- Logistic loss $\log_2(1 + e^{-u})$[a]

[a] Base 2 chosen so that all these functions, like our original indicator take the value 1 at $h = 0$

Let's prove that this is a good thing to do: that minimising ϕ-risk minimises the true misclassification we want to minimise.

Definition 3.11. The **conditional ϕ-risk** of h is

$$\mathbb{E}\left[\phi(Yh(X)) \mid X = x\right] \tag{101}$$

Working much like we did in (1.9),

$$\begin{aligned} &\mathbb{E}\left[\phi(Yh(X) \mid X = x\right] && (102)\\ =\ &\mathbb{E}\left[\phi(Yh(X)|X = x, Y = 1\right]\eta(x) + \mathbb{E}\left[\phi(Yh(X)|X = x, Y = -1\right](1 - \eta(x)) && (103)\\ =\ &\phi(h(x))\eta(x) + \phi(-h(x))(1 - \eta(x)) && (104)\end{aligned}$$

where $\eta(x) = \mathbb{P}\left[Y = 1 \mid X = x\right]$.
For a generic $\eta \in [0, 1]$ and $\alpha \in \mathbb{R}$, let

$$C_\eta(\alpha) = \phi(\alpha)\eta + \phi(-\alpha)(1 - \eta). \tag{105}$$

Definition 3.12. Say $\phi : \mathbb{R} \to [0, \infty)$ is **classification calibrated** if $\forall \eta \in [0, 1]$ with $\eta \neq 1/2$ (as before this was an annoying side case),

$$\inf_{\alpha \in \mathbb{R}} C_\eta(\alpha) < \inf_{\alpha : \alpha(2\eta - 1) \leq 0} C_\eta(\alpha). \tag{106}$$

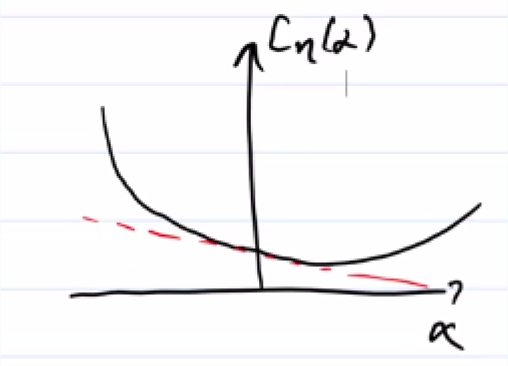
This is dense, but it can be unpacked. We're working with this 'generic' η since we want to just study functions rather than have statistics and conditioning garbage going on. We call this property 'calibrated' because what it is basically saying is that (ignoring the ambiguous '50-50' case) we incur a greater loss when we guess in contradiction to the Bayes classifier: this is the interpretation of the sign condition on the second inf.

With these formalities we can turn engineering and rules of thumb ('which surrogate loss should I pick?') into maths:

Theorem 3.13

Let $\phi : \mathbb{R} \to [0, \infty)$ be convex. Then if ϕ is differentiable at 0 and $\phi'(0) < 0$, then ϕ is classification calibrated.

Proof. Since it's a composition of ϕ things, C_η is convex and diffble at 0 with $C'_\eta(0) = (2\eta - 1)\phi'(0)$. Now separate into the cases $\eta > 1/2$, in which case $C'_\eta(0) < 0$. Then our picture is a 'skewed parabola':



and the inf inequality should now be clear: all we need check is that we must get something smaller on the RHS than all of the LHS of $\alpha = 0$. But local diffbility means we must lie below $C_\eta(0)$ somewhere on the RHS, and by convexity on the LHS we must always lie above the tangent and hence above $C_\eta(0)$ too.

The other case is similar. □

§3.4 Rademacher Complexity, Again

Recall that our ERM work led to the result (2.24) that involved the Rademacher complexity $\mathcal{R}_n(\mathcal{F})$.

In our setting now, we have $\mathcal{F} = \{(x, y) \mapsto \phi(yh(x)) \mid h \in \mathcal{H}\}$. We would like to relate $\mathcal{R}_n(\mathcal{F})$ and $\mathcal{R}_n(\mathcal{H})$.

Lemma 3.14 (The contraction lemma)

Suppose there exists some Lipschitz constant $L > 0$ with

$$|\phi(u) - \phi(u')| \leq L|u - u'|. \tag{107}$$

$\forall u, u' \in [-r, r]$ where $r = \sup h(x)$ (sup taken over all hypotheses and all inputs), since y is always ± 1 means these are the only values we'll ever evaluate ϕ at.

Then $\mathcal{R}_n(\mathcal{F}) \leq L\mathcal{R}_n(\mathcal{H})$.

Proof. Non-examinable.

The two complexities in the inequality are the complexities of what our surrogate classification function spits out, and of the actual ± 1 classifications.

This is a fairly long symbol pushing proof in the course notes. The important idea is that we need to turn $\frac{1}{n}\varepsilon_i\phi(y_i h(x_i))$ terms into $\frac{L}{n}\varepsilon_i h(x_i)$ terms. Let $i = 1$ for ease of notation going forward.

Let $A : \mathcal{H} \times \{-1, 1\}^{n-1}$ be a function that basically allows us to ignore the other $n - 1$ terms. Then we can force into the scenario to use the Lipschitz bound as follows

$$\mathbb{E}\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\varepsilon_1\phi(y_1 h(x_1)) + A(h, \varepsilon_{2:n}) \mid \varepsilon_{2:n}\right] \tag{108}$$

$$= \frac{1}{2n}\left(\sup_{h,g\in\mathcal{H}} \underbrace{\phi(y_1 h(x_1)) - \phi(y_1 g(x_1))}_{\leq L|h(x_1) - g(x_1)|} + nA(h, \varepsilon_{2:n}) + nA(g, \varepsilon_{2:n})\right) \tag{109}$$

and then considering what this absolute value means, we can accomplish the goal of turning $\frac{1}{n}\varepsilon_i\phi(y_i h(x_i))$ terms into $\frac{L}{n}\varepsilon_i h(x_i)$ terms, and we take expectation to clear the conditioning on $\varepsilon_{2:n}$ and then repeat for the other $n - 1$ terms. □

Corollary 3.15

Consider the setup of the contraction lemma and suppose $r < \infty$. Suppose that ϕ is non-increasing and let $M = \phi(-r)$. Then with probability at least $1 - \delta$, the ERM $\hat{h}$ of the ϕ-risk satisfies

$$R_\phi(\hat{(h)}) - R_\phi(h^*) \leq 2L\mathcal{R}_n(\mathcal{H}) + M\sqrt{2\log(2/\delta)/n} \tag{110}$$

But is this even useful? For $\mathcal{H} = \{x \mapsto x^T\beta\}$, r and hence M[31] AND $\mathcal{R}_n(\mathcal{H})$[32] would not be finite, so we fail spectacularly to produce any meaningful result in two ways. However, it's clear things are going wrong since we're allowing β's entries to get large. So, let's bound those entries.

[31] because we can make β have large entries, whence $h(x)$ will grow large, and all such ϕ functions considered so far grow large as inputs grow large.

[32] This is true since the contraction lemma bounds by $\mathcal{R}_n(\mathcal{F})$, and the definition of Rademacher complexity (72.) will have unbounded terms.

§3.5 ℓ_2 constraints

Consider $\mathcal{H} = \{x \mapsto x^T\beta : ||\beta||_2 \le \lambda\}$ and $\mathcal{X} = \{x \in \mathbb{R}^p : ||x||_2 \le C\}$, so by construction we resolve the $r = \infty$ issue;

$$\sup_{x\in\mathcal{X}, h\in\mathcal{H}} |h(x)| \le \lambda C \tag{111}$$

by Cauchy-Schwarz.

Theorem 3.16 (ℓ_2-constrained Rademacher bound)
For $x_{1:n} \in \mathcal{X}$ we have

$$\hat{R}(\mathcal{H}(x_{1:n})) = \frac{1}{n}\mathbb{E}\left[\sup_{\beta}\sum_{i=1}^{n}\varepsilon_i x_i^T\beta\right] \le \frac{\lambda C}{\sqrt{n}} \tag{112}$$

Proof. The summation is essentially a dot product, so actually let's use Cauchy-Schwarz once more

$$\le \frac{\lambda}{n}\mathbb{E}\left[\left\|\sum_{i=1}^{n}\varepsilon x_i\right\|_2\right] \tag{113}$$

and now use Jensen on the square root function

$$\le \frac{\lambda}{n}\left(\mathbb{E}\left[\left\|\sum_{i=1}^{n}\varepsilon_i x_i\right\|_2^2\right]\right)^{1/2} \tag{114}$$

and at this point many diagonal terms cancel (assuming we're really just squaring and ignoring the fancy norm) since $\mathbb{E}\left[\varepsilon_i x_i^T x_j \varepsilon_j\right] = 0$. Now the expectation will fall away:

$$= \frac{\lambda}{n}\left(\sum_{i=1}^{n}\|x_i\|_2^2\right)^{1/2} \le \frac{\lambda C}{\sqrt{n}} \tag{115}$$

□

Example 3.17 (Support vector machines)
Take ϕ to be hinge loss and $\mathcal{H}$ given by our ℓ_2 constrained hypotheses we've been discussing.

Then this is a so-called support vector machine. We have from (3.15) that with probability at least $1-\delta$,

$$R_\phi(\hat{h}) - R_\phi(h^*) \le \frac{2\lambda C}{\sqrt{n}} + (\lambda C + 1)\sqrt{\frac{2\log 2/\delta}{n}}. \tag{116}$$

However, we only really care about misclassification risk. But

Theorem 3.18

In fact if h^* minimises ϕ risk over $\mathcal{H}$ then

$$R_\phi(\hat{h}) - R_\phi(h^*) \geq R(\operatorname{sgn} \circ \hat{h}) - R(\operatorname{sgn} \circ h^*) \tag{117}$$

and further that $R(\operatorname{sgn} \circ h^*)$ is in fact the Bayes risk.

Proof. @todo; hint is example sheet 2, question 11 for main result, and for the part involving Bayes risk somehow is due to ϕ being classification calibrated. □

§3.6 Kernel machines (non-examinable)

Consider a very general hypothesis class $\mathcal{H} = \{\sum_{j=1}^d \phi_j(x)\beta_j \mid \beta \in \mathbb{R}^d, \|\beta\|_2 \leq \lambda\}$ where $d \in \mathbb{N} \cup \{\infty\}$ (!). Surprisingly, the optimization problem is tractable. Consider the Lagrangian form of the objective

$$\frac{1}{n}\sum_i \ell(h(X_i), Y_i) + \gamma \|\beta\|_2^2 \tag{118}$$

where γ is a Lagrange multiplier and $\Phi \in \mathbb{R}^{n \times d}$ and $\Phi_{ij} = \phi_j(X_i)$. Note that $h(X_i) = (\Phi\beta)_i$. Introduce the projection matrix $P \in \mathbb{R}^{d \times d}$ onto the *row* space (not column space!) of Φ. Then $\Phi\beta = \phi P \beta$. Also our norm decreases, since

$$\|\beta\|_2^2 = \|P\beta\|_2^2 + \|(I-P)\beta\|_2^2 \tag{119}$$

which means WLOG we can consider only the β must already lie in that row space. So $\hat{\beta} = \Phi^T\hat{\alpha}$ for some $\hat{\alpha} \in \mathbb{R}^m$. Now let $k(x, x') = \sum_j \phi_j(x)\phi_j(x')$ and $K_{ij} = k(X_i, X_j)$ so that we have a **kernel** matrix $K = \Phi\Phi^T$. Then $\hat{\alpha}$ minimises (over $\alpha \in \mathbb{R}^n$)

$$\sum_{i=1}^n \ell((K\alpha)_i, Y_i) + \gamma\alpha^T K\alpha. \tag{120}$$

This is an n-dimensional optimization problem!

The ERM $x \mapsto \sum_j \phi_j(x)\hat{\beta}_j$ is

$$\sum_j \phi_j(x)(\Phi^T\hat{\alpha})_j = \sum_j \phi_j(x)\sum_{i=1}^n \phi_j(X_i)\hat{\alpha}_i = \sum_i k(x, X_i)\hat{\alpha}_i. \tag{121}$$

In fact the only place where the dimension d even arises is in computing Φ.

It turns out that for certain families of functions, the kernel K can be computed fast (without potentially problematic sums over d terms).

Example 3.19 (An example of where the K computation is reasonable)

Suppose $\mathcal{X} = \mathbb{R}^p$ where p is large. Then if $d = p^2 + p$,

$$(\phi_1(x), ..., \phi_d(x)) = (x_1, ..., x_p, x_1x_1, x_1x_2, ..., x_1x_p, x_2x_1, ..., x_2x_p, ..., x_px_p) \quad (122)$$

i.e. the first p components are just the x guys. Then the next p^2 components are all the pairwise products in lexicographic order.

Then in a nice way,

$$k(x, x') = \sum_j x_j x'_j + \sum_j \sum_k x_j x_k x'_j x'_k = \left(\sum_j x_j x'_j + \frac{1}{2} \right)^2 - \frac{1}{4} \quad (123)$$

Recall that our $\mathcal{X} = \{x \in \mathbb{R}^p : ||x||_2 \leq C\}$. Since C appears in several of the bounds (e.g. (3.16)), we may hope that rescaling C may be useful. However, this is not the case: we will need scale up the size of β (perhaps because otherwise we're centred so close to 0 we will have very bad stability?) and hence the λC terms that appear in our inequalities won't be small.

It turns out that this situation arises in practise when our input data has a large number of features (equivalently, its dimension as a vector is large) and we expect that a low proportion of such features are useful. Then (I think!) scaling things won't fix our problems as all features will still be weighted in the same way.

§3.7 ℓ_1 constraints

Definition 3.20. The ℓ_1 norm is

$$\|u\|_1 = \sum_i |u_i| \quad (124)$$

and the ℓ_∞ norm is

$$\|u\|_\infty = \max_i |u_i|. \quad (125)$$

Suppose that $\mathcal{X} = \{x \in \mathbb{R}^p : \|x\|_\infty \leq C\}$ and let $\mathcal{H} = \{x \mapsto x^T\beta : ||\beta||_1 \leq \lambda\}$. Then $x^T\beta \leq C\lambda$.

Lemma 3.21

For any set $A \subset \mathbb{R}^n$, $\hat{\mathcal{R}}(A) = \hat{\mathcal{R}}(\text{conv}A)$,

Proof. See example sheet 2. The notation here is a generalisation of the previous (71) definition; in that case we take the Rademacher complexity of $\mathcal{F}(z_{1:n})$ which is a set, so the definition naturally generalises. □

Theorem 3.22

$$\{\beta : \|\beta\|_1 \le \lambda\} = \mathrm{conv}\,(S) \tag{126}$$

where $S = \bigcup_{j=1}^{p}\{\lambda e_j, -\lambda e_j\}$.

Proof. Note that visually this is intuitive;

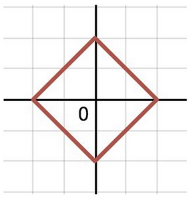


Figure 1

we're saying that the interior of this shape is the convex hull of the four vertices. Let's check the two cases: if β is such that $\|\beta\|_1 = \lambda$ then

$$\beta = \sum_{j=1}^{p} |\beta_j|\mathrm{sgn}(\beta_j)e_j = \sum_{j=1}^{p} \frac{|\beta_j|}{\lambda}\underbrace{(\lambda\mathrm{sgn}(\beta_j)e_j)}_{\in S} \tag{127}$$

and since those coefficients are at most 1, indeed we get the result.

Next, if $\|\beta\|_1 \le \lambda$, then rescale two copies of β to have norm λ, and then take a linear combination of these points and use the prior case (for suitable t_1 and t_2):

$$\beta = t_1\frac{\lambda\beta}{\|\beta\|_1} + t_2\left(\frac{-\lambda\beta}{\|\beta\|_1}\right) \in \mathrm{conv}S. \tag{128}$$

□

Given $x_1, ..., x_n$, let $L : \mathbb{R}^p \to \mathbb{R}^n$ be the linear map given by

$$L(\beta) = (x_1^T\beta, ..., x_n^T\beta)^T, \tag{129}$$

so we can write $\mathcal{H}(x_{1:n}) = L(\mathrm{conv}S) = \mathrm{conv}L(S)$ from (3.6), and then we can use (3.21):

$$\hat{\mathcal{R}}(\mathcal{H}(x_{1:n})) = \hat{\mathcal{R}}(L(S)) = \frac{\lambda}{n}\mathbb{E}\left[\max_{j=1}^{p}\left|\sum_{i=1}^{n}\varepsilon_i x_{ij}\right|\right] \tag{130}$$

where the second equality follows from the fact that we've reduced to the set S which consists of just basis vectors (scaled by $\pm\lambda$).

Now each $\pm\sum_i \varepsilon_i x_{ij}$ is sub-Gaussian with parameter

$$\sqrt{\sum_{i=1}^{n} x_{ij}^2} \le C\sqrt{n} \tag{131}$$

by (2.12) and the fact that $\|x\|_\infty \le C$. Now we can apply the max result (2.14) with the following modification:

Theorem 3.23 (Adapting the max result for absolute values)
$\mathbb{E}\left[\max_{i=1}^{n} |W_i|\right] = \mathbb{E}\left[\max(W_1, -W_1, W_2, -W_2, ..., W_n, -W_n)\right]$ i.e itself the maximum of $2n$ terms.

namely

$$\hat{R}(\mathcal{H}(x_{1:n})) \leq \frac{\lambda}{n} \times C\sqrt{n} \times \sqrt{2\log|S|} = \frac{\lambda C}{\sqrt{n}}\sqrt{2\log(2p)}. \tag{132}$$

where, alternatively, we can see the $2p$ arising as a consequence of our S being defined with ± each coordinate.

Example 3.24

Take ϕ to be hinge loss and let $\mathcal{H}_1$ be the hypothesis class $\mathcal{H} = \{x \mapsto x^T\beta : \|\beta\|_1 \leq \lambda_1\}$ and $\mathcal{X} = \{-1, 1\}^p$.

Then we can use (110), and the risk gap bound turns out to be $O\left(\lambda_1\sqrt{\frac{\log p}{n}}\right)$ (gory algebra omitted).

Due to our choice of $\mathcal{X}$, the bound for the corresponding ℓ_2 constrained hypothesis class derived in (3.17) is $O\left(\lambda_2\sqrt{\frac{p}{n}}\right)$.

To compare these, Let h_0, identified with β_0, minimise R_ϕ over the set of hypotheses with unconstrained norm.

Example 3.25

If we assume that

$$\beta_0 = \frac{1}{\sqrt{p}}(1, 1, ..., 1)^T \tag{133}$$

then for $h_0 \in \mathcal{H}_1$ to hold we need $\lambda_1 \geq \sqrt{p}$.

This corresponds to excess risk $O\left(\sqrt{\frac{p\log p}{n}}\right)$ (wait, why? Doesn't the inequality direction mean this doesn't follow? Oh I suppose we can 'bring lambda down' to the value of $\sqrt{p}$).

For $h_0 \in \mathcal{H}_2$ to hold we instead need $\lambda_2 \leq 1$, in which case the risk bound is $O\left(\sqrt{\frac{p}{n}}\right)$.

Example 3.26

If instead

$$\beta_0 = \frac{1}{\sqrt{s}}(1, ..., 1, 0, ..., 0)^T \tag{134}$$

with s non-zero entries, then the risk bounds are $O\left(\sqrt{\frac{s\log p}{n}}\right)$ and $O\left(\sqrt{\frac{p}{n}}\right)$, following through the same procedure.

Here, the $O\left(\sqrt{\frac{s\log p}{n}}\right)$ bound is the important one. This is because the form of β_0, 'throwing away' most of the features, but still having very large dimension, then we actually have a really tight bound. The fact that we don't have the s dependence in the ℓ_2 means that ℓ_1 constraints may be desirable in practise.

Still, if every feature is important, ℓ_2 constrained hypothesis classes may perform better (even though the extra $\sqrt{\log p}$ seems negligible?).

§3.8 Projection onto convex sets

Note that we've alluded to the fact that extremizing convex functions on convex sets is a computationally tractable problem. However, we haven't seen this explicitly. If we were to do gradient descent (from IB Optimization) in such a scenario what if after one step in the gradient's direction we were led to a point outside the convex set? We could move to the closest point in the convex set, doing *projective* gradient descent. This could fail though, as for example 2 is closest to 1 in the convex set $(0, 1)$ despite 1 not actually being in this set. So we may specialise to closed convex sets:

Theorem 3.27

Let $C \subset \mathbb{R}^d$ be a closed convex set. Then for each $x \in \mathbb{R}^d$, the minimiser $\pi_C(x)$ of $\|x - z\|_2$ over $z \in C$ exists and is unique. We call $\pi_C(x)$ the **projection** of x on C.

Proof. We can't immediately use tools from analysis to deduce that we achieve this value since while we are in a closed set, we're not necessarily in a bounded set so we might not have compactness. But actually, points far away will obviously not be the norm minimiser, so they don't matter. We can let $\mu = \inf_{z \in C} \|x - z\|_2$ and then consider the compact $\bar{B}(x, \mu + 1) \cap C$.

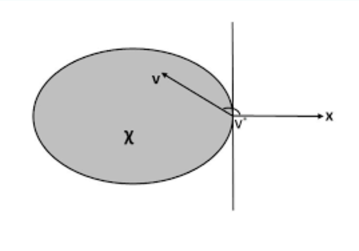
For uniqueness, we can use the *strict* convexity of the function $x \mapsto \|x\|_2^2$, and details are filled in on the example sheet. □

Theorem 3.28

In the notation of the previous result,

$$(x - \pi_C(x))^T (z - \pi_C(x)) \leq 0. \tag{135}$$

Proof. After drawing a picture, this should be intuitive (the line segments meet at an obtuse angle).



Formally, fix x, and let $\pi = \pi_C(x)$. If $z \in C$ then

$$(1 - t)\pi + tz \in C \tag{136}$$

for all $t \in [0, 1]$ by convexity definition. So

$$\|x - \pi\|_2^2 \leq \|x - \pi + t(\pi - z)\|_2^2 \tag{137}$$

$$= \|x - \pi\|_2^2 - 2t(x - \pi)^T (z - \pi) + t^2 \|\pi - z\|_2^2 . \tag{138}$$

After rearranging, we can take $t \to 0^+$ and get the result. □

Theorem 3.29

π_C is a *contraction*;

$$\|\pi_C(x) - \pi_C(z)\|_2 \leq \|x - z\|_2 \tag{139}$$

for all $x, z \in \mathbb{R}^d$.

Proof. Use (3.28) to note that

$$(x - \pi_C(x))^T(\pi_C(z) - \pi_C(x)) \leq 0 \tag{140}$$

as well as

$$(\pi_C(z) - z)^T(\pi_C(z) - \pi_C(x)) \leq 0 \tag{141}$$

(since $\pi_C(z)$ is its own projection onto the convex set). Adding these implies that

$$\|\pi_C(x) - \pi_C(z)\|_2^2 \leq |(x - z)^T(\pi_C(x) - \pi_C(z))| \tag{142}$$

now apply Cauchy-Schwarz:

$$\leq \|x - z\|_2 \, \|\pi_C(x) - \pi_C(z)\|_2 \, . \tag{143}$$

□

and divide through to deduce the result.

§3.9 Subgradients

What if the functions we want to apply gradient descent to functions that aren't differentiable? For example, hinge loss is not differentiable at $x = 1$ (3.10).

Let's now introduce some terminology closely related to the supporting hyperplane theorem:

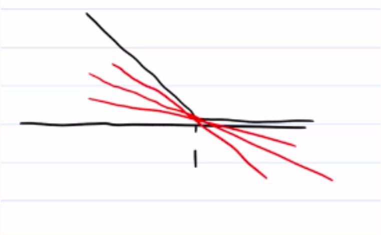
Definition 3.30. $g \in \mathbb{R}^d$ is a **subgradient** of a convex function $f : \mathbb{R}^d \to \mathbb{R}$ at $x \in \mathbb{R}^d$ if

$$f(z) - f(x) \geq g^T(z - x) \tag{144}$$

for all $z \in \mathbb{R}^d$. We call the set of all such g the **subgradient**, denoted $\partial f(x)$.

Example 3.31

The hinge loss $\phi(u) = \max(0, 1 - u)$ has $\partial\phi(1) = [-1, 0]$:



Theorem 3.32

If $f : \mathbb{R}^d \to \mathbb{R}$ is convex, then $\partial f(x)$ is non-empty for all $x \in \mathbb{R}^d$.

Proof. Non-examinable.

There is intuition that, actually, if our convex function is not differentiable then it will be easier rather than harder to find a supporting hyperplane; example (3.31) showed that the sub-differential set was a whole interval rather than just a point.

We shall generalise the idea used in proving Jensen's inequality from IA Probability, where we took the sup of the gradients of the chords joining $(x, f(x))$ to a point on its left, and the inf of the gradients of the chords joining it to a point on its right.

Let C be f's epigraph (3.9). Then take a sequence

$$w_1, w_2, \dots \in \mathbb{R}^{d+1} \setminus C \tag{145}$$

such that $w_k \to (x, f(x))$. Then we can use (3.28) to find a v_k with $\|v_k\|_2 = 1$ such that

$$v_k^T w \leq v_k^T w_k \tag{146}$$

for all $w \in C$. For example, a rescaled $w_k - \pi_C(w_k)$ works. Then the v_k lie in a compact unit ball, so by Bolzano-Weierstrass find a convergent subsequence that converges to some $v = (v_1, v_2)$ where $v_1 \in \mathbb{R}^d$ and $v_2 \in \mathbb{R}$. So

$$v_1^T z + v_2 y \leq v_1^T x + v_2 f(x) \tag{147}$$

for all $(z, y) \in C$. Setting $z = x$ and growing y to be larger than $f(x)$, we get that $v_2 \leq 0$. z being unconstrained (in $\mathbb{R}^d$) means that $v_2 \neq 0$ either.

So after dividing through by v_2 and rearranging, v_2/v_1 is in $\partial f(x)$. □

Theorem 3.33

f differentiable implies that $\partial f(x) = \{\nabla f(x)\}$.

Proof. Like most of this section, this is intuitively obvious in two dimensions, and we need to symbol push to generalise to higher dimensions.

Let $g \in \mathbb{R}^d$ be the subgradient. Then we have, for any $z \in \mathbb{R}^d$

$$\nabla f(x)^T z = \lim_{t \downarrow 0} \frac{f(x + tz) - f(x)}{t} \geq g^T z \tag{148}$$

where the inequality follows from subgradient definition.

But taking $z = g - \nabla f(x)$ we get that $\|\nabla f(x) - g\|_2^2 \leq 0$ and hence the result. □

Theorem 3.34 (Subgradient calculus)
Let $f, f_1, f_2 : \mathbb{R}^d \to \mathbb{R}$ be convex. Then

- $\partial(\alpha f)(x) = \{\alpha g : g \in \partial f(x)\}$ for $\alpha > 0$.
- $\partial(f_1 + f_2)(x) = \{g_1 + g_2 : g_1 \in \partial f_1(x), g_2 \in \partial f_2(x)\}$
- Suppose that $h : \mathbb{R}^m \to \mathbb{R}$ is a composition $h = f \circ g$ where g is an affine function possibly from one dimension to another). Then $\partial h(x) = A^T \partial f(Ax+b)$

Proof. The first and third properties are immediate from writing out definitions.
Why does the second property hold? It holds since □

Example 3.35 (SVMs continued (see also (3.17)))
Consider

$$f(\beta) = \frac{1}{n}\sum_{i=1}^{n} \max(1 - y_i x_i^T \beta, 0). \tag{149}$$

Let $\phi(u) = \max(1-u, 0)$. Then combining (3.31) with $u > 1$ and $u < 1$ (where ϕ is differentiable, we can use (3.34) to decompose $f(\beta)$:
Write

$$h_i(\beta) = \max(1 - y_i x_i^T \beta) \tag{150}$$

so $\partial h_i(\beta) = \{-y_i x_i t : t \in [0,1]\}$ when $y_i x_i^T \beta = 1$. From the second two properties now,

$$\partial f(\beta) = -\frac{1}{n}\sum_{i=1}^{n} y_i x_i t_i \tag{151}$$

where the range of each of the t_is is either the singleton sets $\{-1\}$ or $\{0\}$ if $y_i x_i^T \beta \neq 1$, or else the interval $[0,1]$.

§3.10 Gradient Descent

Suppose we're solving the optimization problem

$$\min_{\beta \in C} f(\beta) \tag{152}$$

where C is closed and convex.

Let $\beta_1 \in C$ be an initial guess and $k \in \mathbb{N}$ be the number of steps, and the sequence of positive step sizes $(\eta_s)_{s=1}^{k-1}$. Then the gradient descent algorithm is the following:

```
for $s = 1, ..., k-1$ do
    $g_s \leftarrow \partial f(\beta_s)$                    ▷ Set to *any* subgradient
    $z_{s+1} \leftarrow \beta_s - \eta_s g_s$
    $\beta_{s+1} \leftarrow \pi_C(z_{s+1})$
end for
    return $\bar{\beta} = \frac{1}{k}\sum_{s=1}^{k} \beta_s$     ▷ An average over all steps, NOT $\beta_s$
```

Remark 3.36. $\bar{\beta} \in C$ by convexity.

Remark 3.37. The choice to return this average of β values rather than β_s may not be too unlike that choice; it is likely the gradients will decrease in size as we optimize, and hence the β_i will cluster around β_s.

[a] Please send any corrections and/or feedback to asc70@cam.ac.uk

Theorem 3.38

Suppose $\hat{\beta}$ is a minimiser of a convex function $f : \mathbb{R}^p \to \mathbb{R}$ over a closed convex set $C \subset \mathbb{R}^p$, playing a very similar role to the ERMs we've worked so much with.

Suppose we have two boundedness assumptions $\sup_{\beta \in C} \|\beta\| \leq R < \infty$ and $\sup_{\beta \in C} \sup_{g \in \partial f(\beta)} \|g\|_2 \leq L < \infty$. Then if $\eta_s \equiv \eta = 2R/L\sqrt{k}$ the output $\bar{\beta}$ of the gradient descent algorithm above satisfies

$$f(\bar{\beta}) - f(\hat{\beta}) \leq \frac{2LR}{\sqrt{k}}. \tag{153}$$

Proof. $f(\beta) \geq f(\beta_s) + g_s^T(\beta - \beta_s)$ for all β, so

$$f(\beta_s) - f(\hat{\beta}) \leq g_s^T(\beta_s - \hat{\beta}) = \frac{1}{\eta}(\beta_s - z_{s+1})^T(\beta_s - \hat{\beta}) \tag{154}$$

$$= \frac{1}{2\eta}\left(\left\|\hat{\beta} - \beta_s\right\|_2^2 + \|z_{s+1} - \beta_s\|_2^2 - \left\|\hat{\beta} - z_{s+1}\right\|_2^2\right) \tag{155}$$

$$\tag{156}$$

(see (5) for definitions) for the first equality, and the second inequality is a sort of polarisation identity. Now use projection-is-contraction result (3.29) to note that $\left\|z_{s+1} - \hat{\beta}\right\|_2^2 \geq \left\|\pi_C(z_{s+1}) - \pi_C(\hat{\beta})\right\|_2^2 = \left\|\beta_{s+1} - \hat{\beta}\right\|_2^2$, so now we can apply our Lipschitz-like subgradient L bound:

$$f(\beta_s) - f(\hat{\beta}) \leq \frac{1}{2\eta}\left(\eta^2 \underbrace{\|g_s\|_2^2}_{\leq L^2} + \left\|\hat{\beta} - \beta_s\right\|_2^2 - \left\|\hat{\beta} - \beta_{s+1}\right\|_2^2\right) \tag{157}$$

and thus when we sum this quantity, we magically get telescoping:

$$\frac{1}{k}\sum_{s=1}^{k} f(\beta_s) - f(\hat{\beta}) \leq \frac{\eta L^2}{2} + \frac{\left\|\hat{\beta} - \beta_1\right\|_2^2 - \left\|\hat{\beta} - \beta_{k+1}\right\|_2^2}{2\eta k} \tag{158}$$

$$\leq \frac{\eta L^2}{2} + \frac{2R^2}{\eta k} \tag{159}$$

since $\max_{x,y \in C} \|x - y\|_2^2 = 4R^2$, and we just ignore that minus term.

Now the $\eta = 2R/L\sqrt{k}$ choice in fact is the minimiser, and by Jensen's,

$$\frac{1}{k}\sum_s f(\beta_s) \leq f(\bar{\beta}) \tag{160}$$

and hence the result follows. □

Remark 3.39. In general, $t^* = \operatorname{argmin}_t \frac{A}{t} + \frac{t}{B}$ satisfies $\frac{A}{t^*} = \frac{t^*}{B}$.

Example 3.40 (SVMs again)

In the notation continued from (3.35) we say the subgradients at some β were of the form

$$g = \frac{1}{n}\sum_{i=1}^{n} y_i x_i t_i \tag{161}$$

where we had $|t_i| \leq 1$. So by the triangle inequality we have the subgradient bound

$$\|g\|_2^2 \leq C \tag{162}$$

where C is the ℓ_2 norm bound for the $x \in \mathcal{X}$.

So application of the last result gives the bound

$$f(\bar{\beta}) - f(\hat{\beta}) \leq 2C\lambda/\sqrt{k} \tag{163}$$

where λ is the bound on the β in the hypothesis class.

Example 3.41 (Mandatory COVID Example (!))

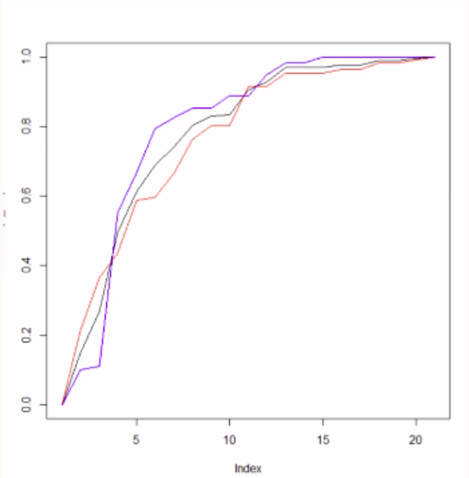
Gradient descent is a more general method than a tool for ERM.

We have data on initial cases from Wuhan, and specifically the time B when they entered Wuhan, the time E when they left Wuhan, and the time when symptoms began S. Suppose each person was infected with COVID at time T. Then of course the incubation period $S-T$ is a very important thing we need to calculate! In this case, we may care to what extent $S-T$ depends on individuals' factors.

The negative log likelihood (of each probability that the incubation period is n days) is a convex function, and so we have formally setup applying gradient descent to a linear model.

Applying this to men and women who contracted COVID and plotting a CDF, we see that the incubation period for women (in blue) and the men (red) are different; in fact it is statistically significant that system onset is more likely to be very early or very late for men; this variance is large. However on average the incubation period is similar between the genders.

Of course, this is the 'real world' so cultural differences could account for this, as men may be more likely to report symptoms earlier etc.



§3.11 Stochastic Gradient Descent

In general a significant computational bottleneck on applying gradient descent is computing the (sub)gradient, as for example even in ERM it will be the sum of n terms where n is large.

Stochastic gradient descent (SGD) can circumvent this issue in the case of minimising convex functions of the form

$$f(\beta)=\mathbb{E}\left[\tilde{f}(\beta;U)\right] \tag{164}$$

where $\tilde{f}:\mathbb{R}^p\times\mathcal{U}\to\mathbb{R}$ is such that $\beta\mapsto\tilde{f}(\beta;U)$ is convex for all $u\in\mathcal{U}$, and U is a random variable taking values in $\mathcal{U}$.

Example 3.42

This setup encompasses ERM: let U be distributed uniformly on $\{1, ..., n\}$. Then the ERM objective function with $\mathcal{H} = \{h_\beta : \beta \in C\}$ can be written as

$$\frac{1}{n}\sum_{i=1}^{n}\ell(h_\beta(x_i), y_i) = \mathbb{E}\left[\ell(h_\beta(x_U), y_U)\right] = \mathbb{E}\left[\tilde{f}(\beta; U)\right] \tag{165}$$

so long as our training data is fixed. The only randomness is in U.

Written in pseudocode, using the same input data as (5) in addition to several i.i.d. copies $U_1, ..., U_{k-1}$ of U.

```
for $s = 1, ..., k-1$ do
    $\tilde{g}_s \leftarrow \partial f(\beta_s; U_s)$                ▷ See remark (3.43)
    $z_{s+1} \leftarrow \beta_s - \eta_s \tilde{g}_s$
    $\beta_{s+1} \leftarrow \pi_C(z_{s+1})$
end for
    return $\bar{\beta} = \frac{1}{k}\sum_{s=1}^{k}\beta_s$        ▷ An average over all steps, NOT $\beta_s$
```

Remark 3.43. The subgradient $\partial\tilde{f}$ is taken with respect to β_s *only*: we 'fix' the U_s things in $\tilde{f}(\beta; U_s)$ first.

Remark 3.44. This is pretty much the same as (5) except we now have tildes, and of course f is different.

Theorem 3.45

Suppose $\hat{\beta}$ is a minimiser of f as above over a closed convex set $C \subset \mathbb{R}^p$.

Suppose that we have the same bound $\sup_{\beta \in C} \|\beta\|_2 \leq R < \infty$ and same $\eta_s \equiv \eta = 2R/L\sqrt{k}$ condition as we had before for the not-stochastic case (3.38) and that we have a slightly different subgradient bound:

$$\sup_{\beta \in C} \mathbb{E}\left[\sup_{\tilde{g} \in \partial \tilde{f}(\beta_i; U)} \|\tilde{g}\|_2^2\right] \leq L^2 < \infty. \tag{166}$$

that involves squaring, since this is actually how we used the bound last time, and we can't avoid this since $\mathbb{E}\left[X^2\right] \neq \mathbb{E}\left[X\right]^2$.

Then the output $\bar{\beta}$ of the gradient descent algorithm above satisfies an analogous result to (153) but with expectations:

$$\mathbb{E}\left[f(\bar{\beta}) - f(\hat{\beta})\right] \leq \frac{2LR}{\sqrt{k}}. \tag{167}$$

Proof. This is not the same as (3.38) since we won't get telescoping due to the changing U_s guys. So as ever, we sneakily use conditional expectation by defining

$$g_s = \mathbb{E}\left[\tilde{g}_s \mid \beta_s\right]. \tag{168}$$

U_s is independent of β_s (because β_s is computed from U_{s-1}) so the following manipulation works

$$\mathbb{E}\left[\tilde{f}(\beta; U_s) \mid \beta_s\right] = \mathbb{E}\left[\tilde{f}(\beta; U_s)\right] = f(\beta) \geq f(\beta_s) + g_s^T(\beta - \beta_s) \tag{169}$$

for all β, so $g \in \partial f(\beta_s)$. So we can insert $\tilde{g}$ as follows

$$f(\beta_s) - f(\hat{\beta}) \leq g_s^T(\beta_s - \hat{\beta}) = \mathbb{E}\left[\tilde{g}_s^T(\beta - \hat{\beta}) \mid \beta_s\right] \tag{170}$$

and now we can use the same (polarisation-like) manipulations from (3.38), where everything will be surrounded with $\mathbb{E}\left[... \mid \beta_s\right]$. We get that the above is

$$\leq \frac{1}{2\eta}\mathbb{E}\left[\eta^2 \|\tilde{g}_s\|_2^2 + \left\|\hat{\beta} - \beta_s\right\|_2^2 - \left\|\hat{\beta} - \beta_{s+1}\right\|_2^2 \mid \beta_s\right] \tag{171}$$

now use the tower property to bring that expectation down, and then we will get telescoping:

$$\mathbb{E}\left[\frac{1}{n}\sum_{s=1}^{k} f(\beta_s)\right] - f(\hat{\beta}) \leq \frac{\eta L^2}{2} + \frac{\left\|\beta_1 - \hat{\beta}\right\|_2^2}{2\eta k} \leq \frac{2LR}{\sqrt{k}}. \tag{172}$$

and once plugging in our η choice and using Jensen's finishes this. □

§4 Popular Machine Learning Methods

In this section

How do we choose the λ constraining our e.g. ℓ_2-constrained hypothesis class? This is a case where we want to consider multiple different machine learning methods, i.e. each λ gives rise to a different method.

This leads to **cross-validation**, which is the biggest take-away from the course.

Definition 4.1. A **machine learning method** is a function

$$H : D \to \mathbb{R}^{\mathcal{X}} \tag{173}$$

where D is the training data, i.e. a machine learning method takes as input the training data $D = (X_i, Y_i)_{i=1}^n$ and outputs a hypothesis $H_D : \mathcal{X} \to \mathbb{R}$.

Let $H^1, ..., H^m$ be a collection of machine learning methods. Ideally, we would choose the 'best' method H^j, where

$$\mathbb{E}\left[\ell(H_D^j(X), Y) \mid D\right] \tag{174}$$

where this expectation is taken over essentially a 'new' independent sample from $\mathcal{X} \times \mathcal{Y}$. This is intractable generally since we only 'see' D (i.e. see everything we've done on ERM!).

An easier approach is to try to minimise the expectation of (174):

$$\mathbb{E}\left[\mathbb{E}\left[\ell(H_D^j(X), Y) \mid D\right]\right]. \tag{175}$$

Definition 4.2. **Cross-validation** is the process of splitting the dataset D into v **folds** $A_1, ..., A_v$ that partition D.

Define $D_{-k} = D \setminus A_k$, and $H_{-k}^j = H_{D-k}^j$.

Definition 4.3. The cross-validation error CV is defined as

$$\mathrm{CV}(j) = \frac{1}{n} \sum_{k=1}^{v} \sum_{i \in A_k} \ell(H_{-k}^j(X_i), Y_i) \tag{176}$$

> **Remark 4.4.** (4.3) is a (usually upwards) biased estimate of (175), because we have a factor of $\frac{1}{n}$ and $n - |A_k| < n$.

> **Remark 4.5.** $v = n$ corresponds to **leave-one-out-cross-validation** which gives the least bias, but can have high variance as the summand in (4.3) will tend to be positively correlated (Why is this exclusive to $v = n$, and why does the implication follow?).

§4.1 Adaboost

Given a base set $\mathcal{B}$ of 'base' classifiers $h : \mathcal{X} \to \{-1, +1\}$ with the property

$$h \in \mathcal{B} \implies -h \in \mathcal{B}, \tag{177}$$

consider the class

$$\mathcal{H} = \left\{ \sum_{m=1}^{M} \beta_m h_m : \beta_m \in \mathbb{R}, h_m \in \mathcal{B}, 1 \le m \le M \right\} \tag{178}$$

M is called the **tuning parameter** which corresponds to .

Adaboost can be motivated as a greedy ERM over $\mathcal{H}$ using exponential loss.

Setting $\hat{f}_0$ to be the zero function $x \mapsto 0$ as our 'initial guess' at a hypothesis, adaboost performs the following update for each step $1 \leq m \leq M$:

$$(\hat{\beta}_m, \hat{h}_m) = \operatorname*{argmin}_{\beta \geq 0, h \in \mathcal{B}} \frac{1}{n} \sum_{i=1}^{n} \exp\left(-Y_i\{\hat{f}_{m-1}(X_i) + \beta h(X_i)\}\right) \tag{179}$$

$$\hat{f}_m = \hat{f}_{m-1} + \hat{\beta}_m \hat{h}_m \tag{180}$$

where property (177) allows us to only need consider $\beta \geq 0$.

The final classification is performed according to $\operatorname{sgn} \circ \hat{f}_M$.

> **Remark 4.6.** As mentioned, this is a greedy *algorithm*, it is quite unlike the previous ERM procedures that generally guarentee some sort of minimization.

To make this process implementable, we can separate the process of the h and β updates as follows:

$$w_i^{(m)} = \frac{1}{n} \exp\left(-Y_i \hat{f}_{m-1}(X_i)\right). \tag{181}$$

Then the expression we take an argmin over in (180) is, decomposing,

$$e^{\beta} \sum_{i=1}^{n} w_i^{(m)} \mathbb{1}\left[h(X_i) = Y_i\right] + e^{-\beta} \sum_{i=1}^{n} w_i^{(m)} \mathbb{1}\left[h(X_i) \neq Y_i\right] \tag{182}$$

$$(e^{\beta} - e^{-\beta}) \sum_{i=1}^{n} w_i^{(m)} \mathbb{1}\left[h(X_i) \neq Y_i\right] + e^{-\beta} \sum_{i=1}^{n} w_i^{(m)}. \tag{183}$$

Now this motivates us to define a 'weighted error'

$$\operatorname{err}_m(h) = \frac{\sum_{i=1}^{n} w_i^{(m)} \mathbb{1}\left[h(X_i) \neq Y_i\right]}{\sum_{i=1}^{n} w_i^{(m)}}. \tag{184}$$

which is defined assuming that no $h \in \beta$ perfectly classifies the data. This setup allows us to rewrite the h update in (180) as

$$\hat{h}_m = \operatorname*{argmin}_{h \in \mathcal{B}} \operatorname{err}_m(h) \tag{185}$$

Note that this reduction of the minimisation problem relies centrally on the $e^{\beta} - e^{-\beta}$ term in (183) being positive, which is fine since we have property (177).

Now we can just consider this new $\hat{h}_m$ as fixed, and differentiating (183) with respect to β in order to get the expression for the updated β as follows:

$$\hat{\beta}_m = \frac{1}{2} \log\left(\frac{1 - \operatorname{err}_m(\hat{h}_m)}{\operatorname{err}_m(\hat{h}_m)}\right). \tag{186}$$

Example 4.7 (Adaboost applied to decision stumps)
Let $\mathcal{X} = \mathbb{R}^p$ and consider the class of **decision stumps**

$$\mathcal{B} = \{h_{a,j,1}(x) = \operatorname{sgn}(x_j - a), h_{a,j,2}(x) = \operatorname{sgn}(a - x_j) : a \in \mathbb{R}, 1 \le j \le p\}. \tag{187}$$

i.e. the set of basic classifiers that pick a coordinate, and a parameter, and classify based on what side of that parameter we fall.

To perform adaboost, finding the optimal weights $w_1, ..., w_n > 0$ (omitting superscript m), for each $1 \le j \le p$, first sort $\{X_{ij}\}_{i=1}^n$ assuming distinctness: $X_{(1)j} < ... < X_{(n)j}$.

Now fixing j, WLOG assume that $X_{(i)j} = X_{ij} = x_i$. Now

$$\operatorname{err}(h_{x_{k+1},j,1}) - \operatorname{err}(h_{x_k,j,1}) = \frac{Y_{k+1}w_{k+1}}{\sum_l w_l}. \tag{188}$$

by referring to (184), and noting that having sorted anything, moving our parameter just slightly to the right will only change one classification (and increase or decrease the error, depending on the sign of Y_i).

So to pick the optimal $h_{a,j,1}$ (over variable a), we need consider a bunch of cumulative sums, and something similar happens with $h_{a,j,2}$. We can sort the X before performing adaboost, and hence after preprocessing the complexity of the algorithm is $O(np)$.

> **Remark 4.8.** The original $\mathcal{H}$ is uncountable, but it turns out the only a values that we care about (as the resulting classifier will behave the same on the training data) are the values inbetween coordinates X_{ij} in the finite input D which consists of only np values total.

This same strategy of taking a simple set of hypotheses and then 'boosting' them into a better hypothesis is more general than adaboost.

§4.2 Gradient Boosting

This technique is used widely e.g. in most winning entries to kaggle competitions.

It can be motivated by the following thought experiment: consider applying gradient descent directly in order to minimise the risk $R(h) = \mathbb{E}\left[\ell(h(X), Y)\right]$ (over all function h). This would involve the following steps:

- Have an initial guess $f_0 : \mathcal{X} \to \mathbb{R}$.
- For each $m = 1, ..., M$ compute

$$\begin{aligned} g_m(x) &= \left.\frac{\partial \mathbb{E}\left[\ell(\theta, Y) | X = x\right]}{\partial \theta}\right|_{f_{m-1}(x)} && (189) \\ &= \mathbb{E}\left[\left.\frac{\partial \ell(\theta, Y)}{\partial \theta}\right|_{f_{m-1}(x)} \middle| X = x\right] && (190) \end{aligned}$$

  where we make suitable regularity conditions to exchange ∂ and $\mathbb{E}[...]$.
- Update $f_m = f_{m-1} - \eta g_m$ where $\eta > 0$ is a small step length.

But this is, as ever in the course, idealized since the expectation is over the distribution of Y.

Now recall from (1.7) that (190) (thought of as a function of x) is the minimiser of

$$\mathbb{E}\left[\left(\left.\frac{\partial \ell(\theta, Y)}{\partial \theta}\right|_{f_{m-1}(X)} - f(X)\right)^2\right] \tag{191}$$

over all functions $f : \mathcal{X} \to \mathbb{R}$. This is the motivation for **gradient boosting** where we empirically minimise (191) using a regression approach.

Definition 4.9. Let H be a base regression method that takes as its argument training data D and outputs a hypothesis $H_D : \mathcal{X} \to \mathbb{R}$. *ell* the loss may be any type of loss (including a convex surrogate).

Then **gradient boosting** takes as input the data $X_{1:n}$ and $Y_{1:n}$ as well as $\eta > 0$, H and M and

$$\hat{\mu} = \operatorname*{argmin}_{\mu \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^{n} \ell(\mu, Y_i). \tag{192}$$

We set $f_0(x) = \hat{\mu}$, i.e we initially guess the best constant function (with respect to the data, and the loss function).

The idea of the algorithm is to use our base regression procedure itself to generate a 'gradient hypothesis' $\hat{g}(m)$ at each stage of the algorithm that is intended to be a close approximation of f in (191). This will therefore be an empirical approximation of the 'true' gradient (190).

```
for $m = 1, ..., M$ do
    $W_i \leftarrow \frac{\partial}{\partial \theta} \ell(\theta, Y_i) \mid_{\theta = \hat{f}_{m-1}(X_i)}$        ▷ One gradient per item of training data.
    $\hat{g}_m \leftarrow H_{(X_{1:n}, W_{1:n})}$        ▷ i.e. apply our method $H$.
    $\hat{f}_m \leftarrow \hat{f}_{m-1} - \eta \hat{g}_m$
end for
    return $\hat{f}_M$        ▷ Possibly composed with sgn in the classification setting.
```

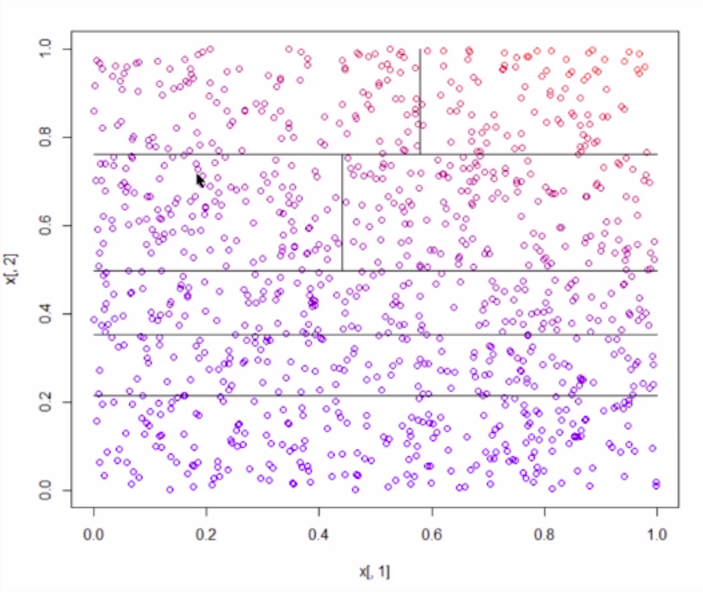
§4.3 Decision Trees

Gradient boosting is particularly effective at optimising 'decision tree methods' that are generalisations of the decision tree stumps (4.7).

Firstly let's get an example of what a decision tree is:

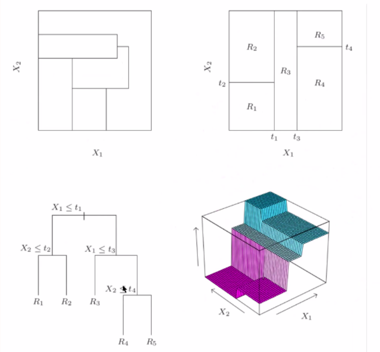
Example 4.10 (Decision tree by example, not formality)

Suppose we are in a $0-1$ classification setting and wish to recursively draw decision boundaries based on the decision boundaries that maximise the reduction in least squares error. Restricting to horizontal and vertical lines only, we get a plot that has a bunch of nested regions:



It is computationally efficient to compute such boundaries due to the reduction in size of each region with each iteration.

The method is called a decision tree procedure since there are several ways to visualize our resultant classifier, and one of them is a decision tree (think 'if-else' statement in code), at the bottom left:



Formally,

Definition 4.11. Given data $(X_1, Y_1), ..., (X_n, Y_n)$, a **decision tree** performs the following:

1. Take as input the maximum number of regions J. Initialize $\hat{\mathcal{R}} = \{\mathbb{R}^p\}$.
2. For each region $R \in \hat{\mathcal{R}}$ such that $I = \{i : X_i \in R\}$ has $|I| > 1$ perform the following:
   - For each $j = 1, ..., p$ let S_j be the set of midpoints between adjacent $\{X_{ij}\}_{i \in I}$ (i.e specialise to one coordinate then sort points in R by this coordinate and extract the midpoints).

- Find the predictor $\hat{j}_R$ and the split point $\hat{s}_R$ to minimise the residual sum of squares drop

$$\underbrace{\min_{c_1\in\mathbb{R}} \sum_{i\in I:X_{ij}\leq s} (Y_i - c_1)^2 + \min_{c_2\in\mathbb{R}} \sum_{i\in I:X_{ij}>s} (Y_i - c_2)^2}_{\text{The residual sum of squares when we split at } s} - \underbrace{\min_{c\in\mathbb{R}} \sum_{i\in I} (Y_i - c)^2}_{\text{RSS before the split}} . \tag{193}$$

where the last, pre-split term does not affect the optimization but is included for our analysis to come.

3. Let $\hat{R}$ be the region yielding the lowest value of the drop in RSS, and update our partition $\hat{\mathcal{R}}$ accordingly (unenlightening symbols omitted).
4. Repeat steps 2 and 3 until $|\hat{\mathcal{R}}| = J$.
5. Writing $\hat{\mathcal{R}} = \{\hat{R}_1, ..., \hat{R}_J\}$, let $\hat{I}_j = \{i : X_i \in \hat{R}_j\}$ and

$$\hat{\gamma}_j = \frac{1}{|\hat{I}_j|} \sum_{i\in\hat{I}_j} Y_i. \tag{194}$$

i.e. a weighted average of the results of things in region.

6. Return the classifier $\hat{T} : \mathbb{R}^p \to \mathbb{R}$ that classifies based on the plurality of the region that data points fall into:

$$\hat{T}(x) = \sum_{j=1}^{J} \hat{\gamma}_j \mathbb{1}\left[x \in \hat{R}_j\right]. \tag{195}$$

§4.4 Random Forests

Consider the regression setting (i.e where $Y_i \in \mathbb{R}$) with squared error loss. Let $\hat{T}_D$ be a decision tree trained on iid data $D = (X_i, Y_i)_{i=1}^n$. Let $\bar{T}(x) = \mathbb{E}\left[\hat{T}_D(x)\right]$.

> **Remark 4.12.** Here, we consider the training data to be random, hence the expectation with the argument x.

Let (X, Y) be independent of D and distributed like the training data. Recall the expectation decomposition (1.7), then we can compute

$$\mathbb{E}\left[R(\hat{T}_D)\right] = \mathbb{E}\left[(Y - \hat{T}_D(x))^2\right] \tag{196}$$

$$= \mathbb{E}\left[(Y - \underbrace{\mathbb{E}\left[Y \mid X, D\right]}_{\mathbb{E}[Y|X]})^2\right] + \mathbb{E}\left[(\mathbb{E}\left[Y|X\right] - \hat{T}_D(X))^2\right] \tag{197}$$

Use the tower property, and fudge in $\bar{T}(X)$ noting that $\bar{T} = \mathbb{E}\left[\hat{T}_D\right]$ (so cross-terms disappear) to write

$$\mathbb{E}\left[\mathbb{E}\left[(Y-\mathbb{E}\left[Y \mid X\right])^2\right]|X\right]+\mathbb{E}\left[\left(\hat{T}_D-\mathbb{E}\left[\hat{T}_D\right]\right)^2\right]+\mathbb{E}\left[(\bar{T}-\mathbb{E}\left[Y|X\right])^2\right] \tag{198}$$

both these first two terms are now conditional variances:

$$= \underbrace{\mathbb{E}\left[\operatorname{Var}(Y|X)\right]}_{\text{'Irreducible error', tree independent}}+\underbrace{\mathbb{E}\left[\operatorname{Var}(\hat{T}_D|X)\right]}_{\text{Variance of the tree}}+\underbrace{\mathbb{E}\left[(\bar{T}-\mathbb{E}\left[Y|X\right])^2\right]}_{\text{Squared bias}}. \tag{199}$$

As we increase the number of regions, while the squared bias will reduce, However, the variance of the tree will tend to increase due the γ coefficients in the decision tree construction being far more variable.

Definition 4.13. A **random forest** procedure samples from the data D with replacement in order to form datasets $D_1^*, ..., D_B^*$. It then fits trees $\hat{T}^{(b)}$ to the data D_b^*, but when searching for the best variable to split upon, restrict ourselves to a random sample of m_{try} of the p predictors.

Then we average out the trees: we output $f_{\text{rf}} = \frac{1}{B}\sum_{b=1}^{B}\hat{T}^{(b)}$.

What is the reason for this sampling procedure? For one thing, it may reduce the computational complexity. It also makes the $\hat{T}^{(b)}$ more independent.

If for $b_1 \neq b_2$ and some $x \in \mathbb{R}^p$ we have that $\operatorname{Corr}(\hat{T}^{(b_1)}(x), \hat{T}^{(b_2)}(x)) = \rho \geq 0$ then when can directly compute the forest variance via the formula,

$$\operatorname{Var}(f_{\text{rf}}) = \left(\rho + \frac{1-\rho}{B}\right)\operatorname{Var}(\hat{T}^{(1)}(x)) \tag{200}$$

so we can't expect increasing B alone decreasing the forest variance; we also need decrease ρ. This can be done by choosing a small m_{try} since this will increase likelihood that the individual forests come from different classifiers. But then squared bias would (probably...) increase.

§4.5 Neural Networks

This will be the last machine learning technique in the course, and we focus only on feed-forward neural networks, the state-of-the-art for many problems.

Neural networks for classification problems are based around a class of hypotheses $\mathcal{H}$, that is rich in that it is consists of many function compositions.

Definition 4.14. A **neural network** is based around a set of hypotheses $h(x)$ defined by

$$h(x) = A^{(d)} \circ g \circ A^{(d-1)} \circ g \circ ... \circ A^{(1)}(x) \tag{201}$$

where d is the depth of the neural network, $A^{(k)} : \mathbb{R}^{m_k} \to \mathbb{R}^{m_{k+1}}$ is an affine function $A(v) = \beta^{(k)}v + \mu^{(k)}$, and $g : \mathbb{R}^m \to \mathbb{R}^m$ is a non-linear activation function that acts component-wise by a function $\psi(u)$. Some popular choices include $\operatorname{ReLU}(x) = \max(x, 0)$ and $\sigma(x) = 1/(1 + e^{-u})$.

> **Remark 4.15.** If g was not non-linear, the whole function would collapse into a single affine function!

Let's set up some definitions in order to show how we can apply SGD to such hypotheses:

Definition 4.16. $h^{(0)} = x$ is the input layer to the neural network, $x^{(k)} := A^{(k)}(h^{(k-1)})$, the **hidden layers** of the network are given by $h^{(k)} = g(x^{(k)})$ for $k < d$ and the output (layer) is $x^{(d)} = h(x)$.

Further terminology is that the vector input to h is called the 'input layer', the output $g \circ A^{(i)} \circ g \circ \ldots \circ A^{(1)}(x)$ at each stage is called the 'hidden layer' output, and the final output is called the output layer. The terminology is used as each component can be thought of as a nodes in a graph, with edges all edges between nodes in adjacent layers. The process of training a neural network is the process of applying SGD to the parameters (β, μ) of all the affine functions, with surrogate loss ϕ. This turns out to be computationally tractable:

§4.5.1 Back-Propagation

We let $z = \phi(yh(x)) = \phi(yx^{(d)})$ be our loss. Then initially compute

$$\frac{\partial z}{\partial x^{(d)}} = y\phi'(yx^{(d)}). \tag{202}$$

Recall that our output $x^{(d)}$ is a function of the last hidden layer which is a simple affine function, so

$$\frac{\partial z}{\partial \beta_{1k}^{(d)}} = \frac{\partial z}{\partial x^{(d)}} h_k^{(d-1)}, \tag{203}$$

$$\frac{\partial z}{\partial \mu^{(d)}} = \frac{\partial z}{\partial x^{(d)}}. \tag{204}$$

by differentiating the affine function expression $A = \beta x + \mu$.
Now we can propagate further backwards by writing

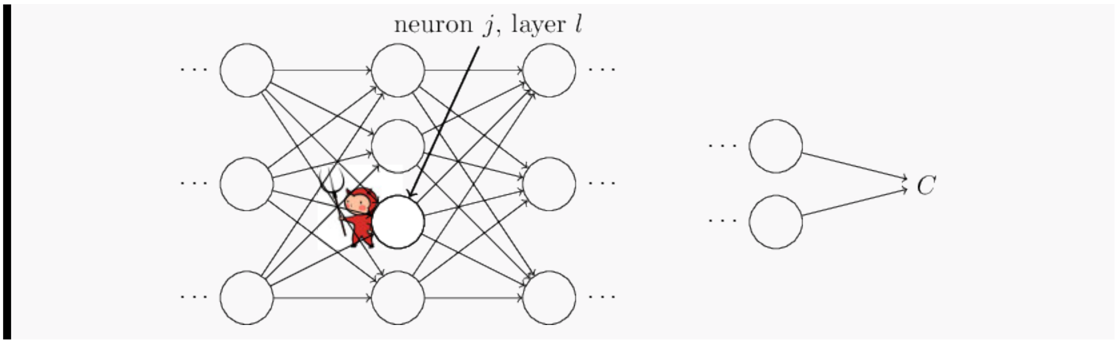
$$\frac{\partial z}{\partial h_j^{(d-1)}} = \frac{\partial z}{\partial x^{(d)}} \beta_{1j}^{(d)} \tag{205}$$

and then passing through the affine function

$$\frac{\partial z}{\partial x_j^{(d-1)}} = \frac{\partial z}{\partial j_j^{d-1}} \psi'(x_j^{(d-1)}). \tag{206}$$

And now we're back to a $\frac{\partial z}{\partial x}$ expression as in (202), so we can do this all again to get the gradients with respect to all the parameters further back in the network.

> **Remark 4.17.** A much lighter introduction (which is slower paced!) can be found at `http://neuralnetworksanddeeplearning.com`. Thie author implements a neural network in basic python, while (importantly for mathmos!) explaining the choices made in both the design of the neural net architecture, and implementation choices.



References

[1] Rajen D. Shah (2021), *Mathematics of Machine Learning*, http://www.statslab.cam.ac.uk/~rds37/teaching/machine_learning/notes.pdf.

[2] Philippe Rigollet, *18.657: Mathematics of Machine Learning*, https://ocw.mit.edu/courses/mathematics/18-657-mathematics-of-machine-learning-fall-2015/lecture-notes/MIT18_657F15_LecNote.pdf.