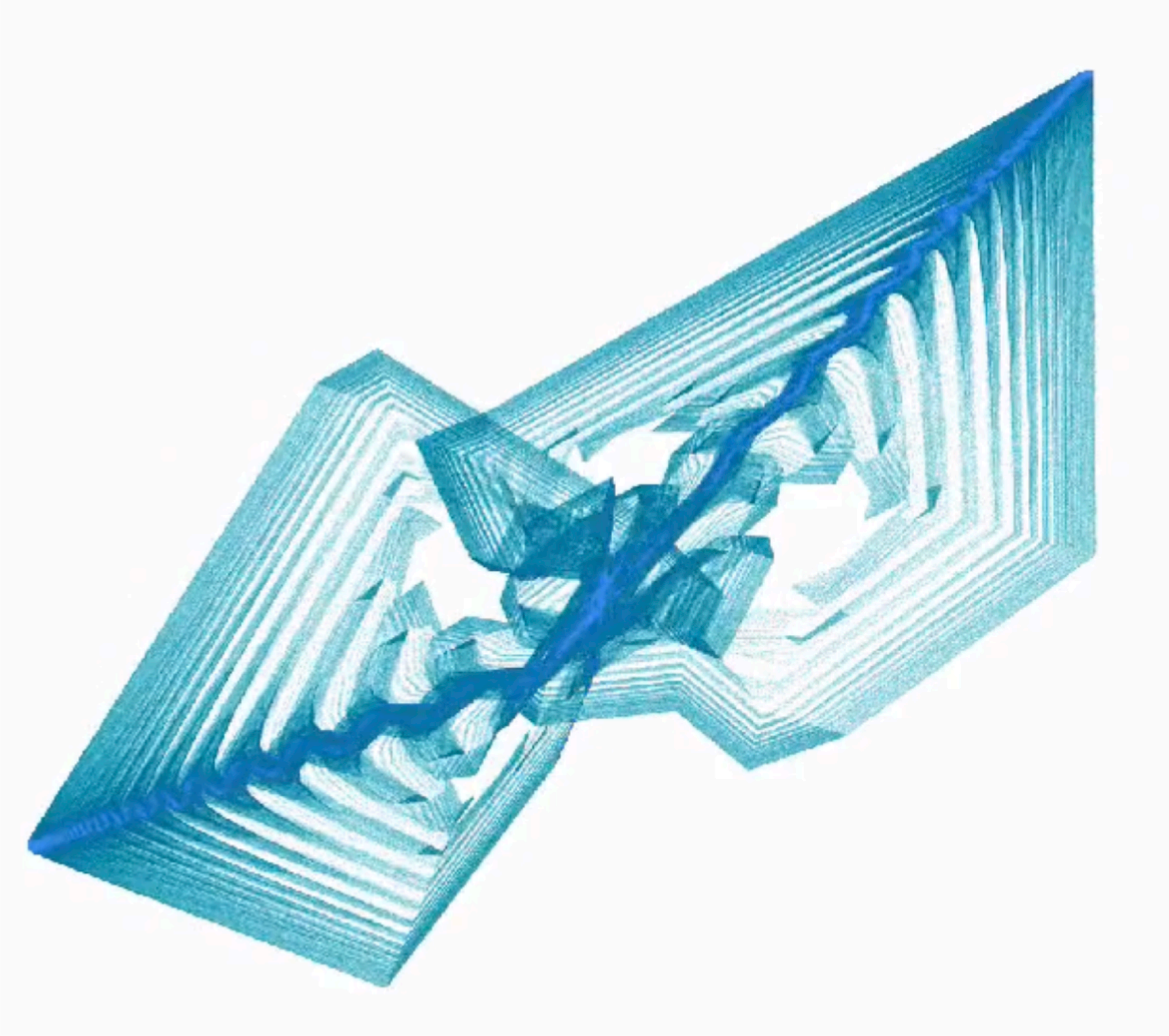
# Mathematical Foundations of Machine Learning



# Mathematical Foundations of Machine Learning

**Genesis of notes.** These notes were developed as part of a course taught by Robert Nowak at the University of Wisconsin-Madison. The reader should beware that the notes have not been carefully proofread and edited. The notes assume the reader has background knowledge of basic probability, statistics, linear algebra, and optimization.

# Contents

# List of Figures

# List of Equations

# Lecture 1: Probability in Machine Learning

## 1.1. Executive Summary

Probability and statistics are central to the design and analysis of ML algorithms. This note introduces some of the key concepts from probability useful in understanding ML. There are many great references on this topic, including [4, Chapter 2].

## 1.2. Introduction

Consider the training dataset depicted in Figure 1.1. Imagine that these data are based on $1$ to $5$ star movie reviews. The horizontal axis is the rating of *Star Wars* (SW) and the vertical axis is the rating of *Crazy Rich Asians* (CRA), a romantic comedy. Think of each point as a person's ideal ratings (maybe fractional like $3.5$), but they are forced to select $1$ to $5$ stars. Each box in the plot indicates ratings that take on a specific combination of stars. For example the box between $[2, 3] \times [3, 4]$ indicates people who gave SW a 3-star rating and CRA a 4-star rating. There are four people in this box. The color of each point indicates whether or not they "liked" *Guardians of the Galaxy* (GG); red if they liked GG, green otherwise[1]

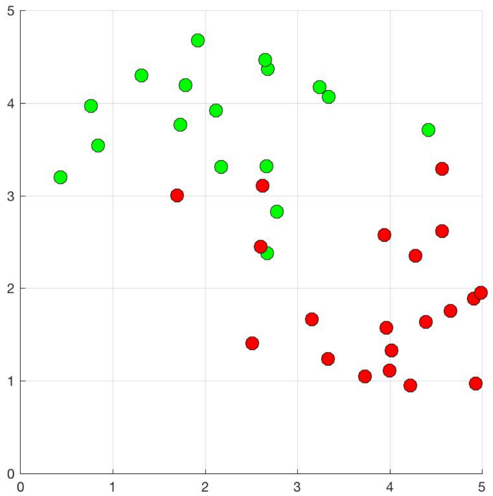


Figure 1.1: Movie ratings for $40$ students.

There are a total of 40 people in this dataset, but of course there are many more people in the world. Suppose we are really interested in understanding the movie preferences of all graduate students at UW-Madison. Suppose there are $8,094$ students. This is what the preferences of all of them might look like. This is the full "population" we would like to consider, but our available data is only the smaller sample of $40$ students. We can use ML to try to make predictions about the whole population based on the sample.

[1] Other possible scenarios are: Computer Vision- say two features correspond to distances between eyes and eyebrows, center of mouth and corners, and labels are happy or not happy face; Natural Language Processing - say features are # times words "ball" or "vote" appear and labels indicate whether document is about sports or politics.

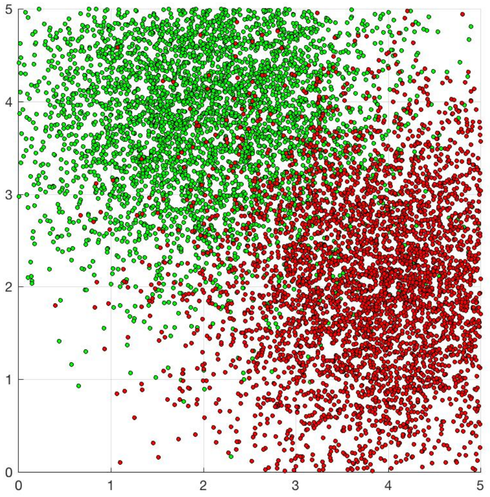


Figure 1.2: Movie ratings for all $8,094$ students.

The number of people in each box is

```
216   564   627   253    82
229   616   598   458   293
 88   273   447   631   599
 11    75   250   607   565
  1    23   100   257   231
```

Figure 1.3: Histogram of movie ratings for all $8,094$ students.

Normalizing by the total number of students gives us the probability that a randomly selected student will be in a particular box, as shown below. For example, the probability that a person is in the box $[2, 3] \times [3, 4]$, which corresponds to a 3-star rating for SW and a 4-star rating for CRA, is $598/8094 \approx 3/40$. So, if we take a random sub-sample of $40$ people, then we would expect about $3$ individuals to be in this box. In the random subsample shown in Figure 1.1, there were $4$ people in this box... very close to our expectation.

```
0.0267    0.0697    0.0775    0.0313    0.0101
0.0283    0.0761    0.0739    0.0566    0.0362
0.0109    0.0337    0.0552    0.0780    0.0740
0.0014    0.0093    0.0309    0.0750    0.0698
0.0001    0.0028    0.0124    0.0318    0.0285
```

Figure 1.4: Probabilities of movie ratings.

So let's think about things this way. There is a big population out there, and we want to learn what they think based on a random subsample. We'd like to predict things like: Will a graduate student give both SW and CRA 5-star ratings?' Will such a student like GG? If we know how a student rated SW and CRA, can we predict whether she will like GG? The complete probability structure of this problem is summarize by two sets of counts, one for those who don't like GG and one for those who do. There are 4039 students who don't like GG, and 4055 who do.

| | | | | |
|---|---|---|---|---|
| 216 | 560 | 606 | 216 | 42 |
| 228 | 604 | 527 | 247 | 35 |
| 83 | 236 | 231 | 78 | 21 |
| 7 | 39 | 35 | 18 | 3 |
| 1 | 2 | 3 | 1 | 0 |

didn't like GG

| | | | | |
|---|---|---|---|---|
| 0 | 4 | 21 | 37 | 40 |
| 1 | 12 | 71 | 211 | 258 |
| 5 | 37 | 216 | 553 | 578 |
| 4 | 36 | 215 | 589 | 562 |
| 0 | 21 | 97 | 256 | 231 |

liked GG

Figure 1.5: All relevant statistics.

## 1.3. Joint, Marginal, and Conditional Probabilities

Let $x_{1i}$ and $x_{2i}$ denote the two ratings for the $i$th person in the dataset, and let $y_i$ denote the binary $\pm 1$ label indicating whether they liked GG.

1. If we select a person at random from the population, what is the probability that they are in the $(3, 4)$ box?
2. If we select a person at random, what is the probability they gave SW a 3-star rating?
3. What is the probability that a student will like GG?
4. If a person gave SW 3-stars and CRA 4-stars, then what is the probability they will like GG?
5. What is the probability that a person who gives SW 3-stars will give CRA 1 or 2 stars?
6. What is the expected number of people in a subsample of size $n$ that will rate SW 3 stars and CRA 5 stars?

## 1.4. Histogram Classifier

Suppose we want to predict whether a student will like GG based on her ratings of SW and CRA. Consider the histogram classifier:

$$\widehat{y} = \begin{cases} +1 & \frac{p(y=+1\,|\,x_1,x_2)}{p(y=-1\,|\,x_1,x_2)} \geq 1 \\ -1 & \text{otherwise} \end{cases}$$

We can use a subsample to estimate the probabilities needed above, but they may be poor estimates if the sample size is small.

### 1.4.1. Basic Probability Calculus

Let $X$ and $Y$ be discrete random variables. The joint probability that $X$ takes the value $x$ and $Y$ takes the value $y$ is denoted by $p(x, y)$. The marginal probability that $X$ takes the value $x$ is $p(x) = \sum_y p(x, y)$, and $p(y)$ is defined analogously. The conditional probability that $Y$ takes the value $y$ given that $X$ equals $x$ is denoted by $p(y|x)$ which is the solution to the equation $p(x, y) = p(y|x)p(x)$, or in other words $p(y|x) = p(x, y)/p(x)$. If the random variables are *independent*, then $p(x, y) = p(x)p(y)$.

## 1.5. Expectation

The expected value or mean of a discrete random variable $X$ is computed by taking the expectation

$$\mathbb{E}[X] \;=\; \sum_x x\, p(x)\;,$$

where the summation is over all possible discrete values $X$ may take. In other words, the expected value is a weighted combination of all the discrete values, where each weight is the probability that $X$ will take that value. We can also take the expectation of a function of the random variable

$$\mathbb{E}[f(X)] \;=\; \sum_x f(x)\, p(x)\;.$$

Suppose we have two random variables $X$ and $Y$ and consider the sum $X+Y$. The expectation of the sum is

$$\begin{aligned}
\mathbb{E}[X+Y] &= \sum_x \sum_y (x+y) p(x,y) \\
&= \sum_x \sum_y x\, p(x,y) + \sum_x \sum_y y\, p(x,y) \\
&= \sum_x x \sum_y p(x,y) + \sum_y y \sum_x p(x,y) \\
&= \sum_x x\, p(x) + \sum_y y\, p(y) \\
&= \mathbb{E}[X] + \mathbb{E}[Y]
\end{aligned}$$

The expectation of the product is

$$\mathbb{E}[XY] \;=\; \sum_x \sum_y xy\, p(x,y)\;,$$

and, in general, is not a function of $\mathbb{E}[X]$ and $\mathbb{E}[Y]$ alone. However, if $X$ and $Y$ are independent, then

$$\begin{aligned}
\mathbb{E}[XY] &= \sum_x \sum_y xy\, p(x,y) \\
&= \sum_x \sum_y xy\, p(x) p(y) \\
&= \left(\sum_x x\, p(x)\right)\left(\sum_y y\, p(y)\right) \\
&= \mathbb{E}[X]\mathbb{E}[Y]
\end{aligned}$$

The variance of a random variable $X$ is the expectation of $(X-\mathbb{E}[X])^2$, that is the average squared deviation of $X$ from its mean value $\mu = \mathbb{E}[X]$. We write this as

$$\mathbb{V}(X) \;=\; \mathbb{E}[(X-\mu)^2]\;.$$

Another key concept is the conditional expectation defined as

$$\mathbb{E}[Y|X=x] \;=\; \sum_y y\, p(y|x)\;.$$

Technically, this is just the first moment of $p(y|x)$; in other words, the mean of the conditional distribution of $y$ given the "side information" $X=x$. The practical interpretation is that this is in some sense the best prediction of $Y$ given $X=x$. Note that if $X$ and $Y$ are independent, then $p(y|x) = p(y)$ and $\mathbb{E}[Y|X=x] = \mathbb{E}[Y]$; i.e., $X=x$ doesn't inform our prediction about $Y$.

## 1.6. Sums of Independent Random Variables

In ML applications we often encounter sums or averages of independent random variables. For example, if we select $100$ people at random from a population and ask them if they like cheese, then we can estimate the probability that an average person likes cheese by averaging the answers in our survey.

Let $X_1, X_2, \ldots$ denote independent random variables and for any $n$ let $S_n = \sum_{i=1}^n X_i$. As shown above, the mean of the sum is equal to the sum of the means:

$$\mathbb{E}[S_n] \ = \ \sum_{i=1}^n \mathbb{E}[X_i] \ .$$

This is true whether or not the terms in the sum are independent. The variance of the sum is more complicated in general, since it involves not only the variances of the individual terms, but also their covariation.

$$\begin{aligned}
\mathbb{V}(S_n) \ &= \ \mathbb{E}[(S_n - \mathbb{E}[S_n])^2] \\
&= \ \mathbb{E}[(\sum_{i=1}^n (X_i - \mathbb{E}[X_i]))^2] \\
&= \ \mathbb{E}[\sum_{i=1}^n \sum_{j=1}^n (X_i - \mathbb{E}[X_i])(X_j - \mathbb{E}[X_j])] \\
&= \ \sum_{i=1}^n \sum_{j=1}^n \mathbb{E}[(X_i - \mathbb{E}[X_i])(X_j - \mathbb{E}[X_j])]
\end{aligned}$$

However, if the terms are independent, then the variance of the sum is equal to the sum of the variances. Note that if $X_i$ and $X_j$ are independent, then

$$\mathbb{E}[(X_i - \mathbb{E}[X_i])(X_j - \mathbb{E}[X_j])] \ = \ \mathbb{E}[X_i - \mathbb{E}[X_i]] \, \mathbb{E}[X_j - \mathbb{E}[X_j]] \ = \ 0 \ ,$$

since both expectations are equal to zero. Thus, if the $X_i$ are independent, the variance simplifies to

$$\mathbb{V}(S_n) \ = \ \sum_{i=1}^n \mathbb{E}[(X_i - \mathbb{E}[X_i]))^2] \ = \ \sum_{i=1}^n \mathbb{V}(X_i) \ .$$

Here is a nice application of this property. Suppose we sample $n$ people uniformly at random[2] and ask them to rate SW and CRA. Based on this sample, we want to estimate the probability that a random person will give SW $k$ stars and CRA $\ell$ stars. Let $\mathbb{1}_{i,k,\ell}$ denote the binary "indicator" variable that is assigned the value $1$ if person $i$ gives SW $k$ stars and CRA $\ell$ stars, and the value $0$ otherwise. Then the empirical probability that a person gave SW $k$ stars and CRA $\ell$ stars is

$$\widehat{p}_{k,\ell} \ = \ \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{i,k,\ell} \ .$$

The expectation of the empirical probability is

$$\mathbb{E}[\widehat{p}_{k,\ell}] \ = \ \frac{1}{n} \sum_{i=1}^n \mathbb{E}[\mathbb{1}_{i,k,\ell}] \ = \ \frac{1}{n} \sum_{i=1}^n p_{k,\ell} \ = \ p_{k,\ell} \ ,$$

[2] We will assume that we sample people uniformly at random *with replacement*, rather than *without replacement*. This means that may select the same person more than once, but it ensures that our samples are independently and identically distributed. If the full population is large relative to the sample size, then there isn't much chance of sampling the same person more than once.

where $p_{k,\ell}$ is the "true" probability that a randomly selected person from the population at large will give SW $k$ stars and CRA $\ell$ stars. So the empirical probability is an *unbiased* estimator of the true probability. However, depending on the sample size $n$, it may have a large variance. Since the indicator random variables are independent, the variance is

$$\mathbb{V}(\widehat{p}_{k,\ell}) \;=\; \mathbb{E}[(\widehat{p}_{k,\ell} - p_{k,\ell})^2] \;=\; \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}[(\mathbb{1}_{i,k,\ell} - p_{k,\ell})^2]\,.$$

Note that $(\mathbb{1}_{i,k,\ell} - p_{k,\ell})^2$ is either $p_{k,\ell}^2$ or $(1-p_{k,\ell})^2$. In fact, it takes the value $p_{k,\ell}^2$ with probability $1-p_{k,\ell}$ and the value $(1-p_{k,\ell})^2$ with probability $p_{k,\ell}$. So its expectation is

$$\mathbb{E}[(\mathbb{1}_{i,k,\ell} - p_{k,\ell})^2] \;=\; (1-p_{k,\ell})^2 p_{k,\ell} + p_{k,\ell}^2(1-p_{k,\ell}) \;=\; p_{k,\ell}(1-p_{k,\ell})$$

Therefore, $\mathbb{V}(\widehat{p}_{k,\ell}) = \frac{p_{k,\ell}(1-p_{k,\ell})}{n}$. In other words, the variance is proportional to $1/n$ and the standard deviation is proportional to $1/\sqrt{n}$.

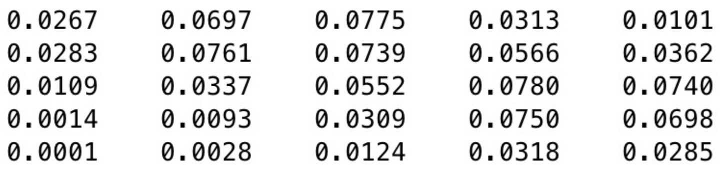
## 1.7. Excercises

1. This question is about using the `rand` or `random` function to simulation random variables. Let `rand` denote a function that generates a random number uniformly distributed on $[0, 1]$.

   **a.** Use `rand` to select one of $8,094$ students uniformly at random.

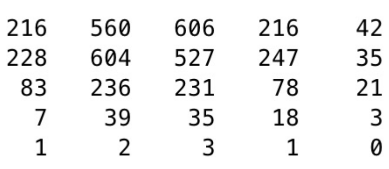
   **b.** Write a code to select $k$ students uniformly at random without replacement.

   **c.** Write a code to generate a bivariate random variable according to the distribution below

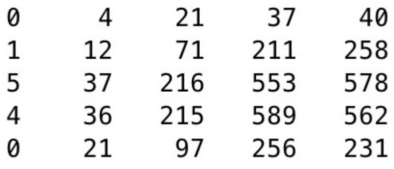


```
0.0267    0.0697    0.0775    0.0313    0.0101
0.0283    0.0761    0.0739    0.0566    0.0362
0.0109    0.0337    0.0552    0.0780    0.0740
0.0014    0.0093    0.0309    0.0750    0.0698
0.0001    0.0028    0.0124    0.0318    0.0285
```

2. Consider the movie preference prediction problem. The two arrays below denote student ratings for *Star Wars* (SW), 1-5, left to right. and *Crazy Rich Asians* (CRA), 1-5 bottom to top. The array on the left are the counts of students that did not like *Guardians of the Galaxy* (GG), and the array in the right is those that did.



```
216   560   606   216    42
228   604   527   247    35
 83   236   231    78    21
  7    39    35    18     3
  1     2     3     1     0
```



```
0     4    21    37    40
1    12    71   211   258
5    37   216   553   578
4    36   215   589   562
0    21    97   256   231
```

didn't like GG liked GG

   (a) What is the probability that a randomly selected student (RSS) will like GG and give SW a 5-star rating?

   (b) What is the probability that an RSS who likes GG, will give SW 5-stars and CRA 2-stars?

   (c) What is the probability that an RSS will give SW *at least* 3 stars and CRA *at most* 2 stars?

(d) Suppose a new student gave SW a 3-star rating and CRA a 2-star rating. Will she like GG?

(e) Suppose a new student gave SW a 3-star rating. Predict her rating for CRA.

(f) Suppose you know that a new student gave CRA a 2 star rating and also that she liked GG, what is your estimate of her (unknown) rating of SW?

(g) Suppose all you know is that a new student didn't like GG. Make predictions of her ratings for SW and CRA?

(h) Suppose a new student gave SW $X$ rating and CRA $Y$ rating, and $7 \leq X + Y \leq 9$. What is the probability she will like GG?

(i) Suppose a new student gave SW $X$ rating and CRA $Y$ rating, and $X < Y$. What is the probability she will like GG?

3. Consider the following two-player game. Each player rolls a six-sided die (and the outcomes of the rolls are independent and uniformly distributed). They cannot see their own outcome but they can see the outcome of the other player. They must submit a guess of their own roll and they do so in such a way that absolutely no information is passed between players. If they are both correct, they win. If either is wrong, they lose. What is the maximum probability of victory?

4. Suppose that scientists are studying how the brain performs a certain information-processing tasks. Three regions of the brain are involved, denoted $A$, $B$ and $C$. There is prior evidence that there are direct neural connections between regions $A$ and $B$ and regions $B$ and $C$. However, it is uncertain whether regions $A$ and $C$ are directly connected. The scientists design an experiment to test this. The activity in human subjects' brains is measured while they perform the information-processing tasks. The activity level in each region is a binary-valued variable, indicating whether the region is significantly active (1) or not (0). Let $x_A$, $x_B$, and $x_C$ denote the activity level in each region, which we will model as sequences of random variables. If there is no direct connection between regions $A$ and $C$, then we conjecture that $x_A$ and $x_C$ will be *conditionally independent* given $x_B$.

   Many measurements of these variables, for repeated trials of the task and different human subjects, are recorded. The dataset is $\{(x_A^{(i)}, x_B^{(i)}, x_C^{(i)})\}_{i=1}^n$, where $n$ is the total number of measurements. We can model each triple $(x_A^{(i)}, x_B^{(i)}, x_C^{(i)})$ as independently and identically distributed (i.e., each triple is an independent realization from the same multivariate distribution, but $(x_A^{(i)}$, $x_B^{(i)}$, and $x_C^{(i)})$ may be correlated). How would you use the data to check for whether $x_A$ and $x_C$ are conditionally independent given $x_B$?

# Lecture 2: Discrete Probability Distributions and Classification

Let $Y$ be a random variable that takes one of $m$ discrete values $\{a_1, \ldots, a_m\}$. For example, $Y$ could be a label in a binary classification problem taking values $-1$ or $+1$. The probability distribution for discrete random variables is $\mathbb{P}(Y = a_j) = p_j$, $j = 1, \ldots, m$, and $\sum_{j=1}^m p_j = 1$. Here are some common distributions.

**Bernoulli** Suppose $Y$ takes values $0$ or $1$, then

$$\mathbb{P}(Y = y) \;=\; p^y\,(1-p)^{1-y}$$

Its mean and variance are $p$ and $p(1-p)$, respectively.

**Binomial** Consider $n$ independent and identically distributed Bernoulli random variables $Y_1, Y_2, \ldots, Y_n$. Then the joint probability distribution is

$$\begin{aligned}\mathbb{P}(Y_1 = y_1, \ldots, Y_n = y_n) \;&=\; \prod_{i=1}^n \mathbb{P}(Y_i = y_i) \\ &=\; \prod_{i=1}^n p^{y_i}\,(1-p)^{1-y_i}\end{aligned}$$

The sum of i.i.d. Bernoullis follows a *Binomial* distribution

$$\mathbb{P}\Big(\sum_{i=1}^n Y_i = k\Big) \;=\; \binom{n}{k} p^k (1-p)^{n-k}$$

where $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ is the number of ways of choosing $k$ out of $n$ of the variables to have the value of $1$. The sum $K := \sum_{i=1}^n Y_i$ is said to be a binomial random variable. The mean and variance of $K$ are $np$ and $np(1-p)$, respectively.

**Multinomial** Consider $n$ independent and identically distributed random variables $Y_1, Y_2, \ldots, Y_n$ that take values in $\{a_1, \ldots, a_m\}$. Then the joint probability distribution is

$$\begin{aligned}\mathbb{P}(Y_1 = y_1, \ldots, Y_n = y_n) \;&=\; \prod_{i=1}^n \mathbb{P}(Y_i = y_i) \\ &=\; \prod_{i=1}^n \prod_{j=1}^m p_j^{\mathbb{1}\{y_i = a_j\}}\end{aligned}$$

Let $K_j$ denote the number of times the value $a_j$ appears in the sample. Then

$$\mathbb{P}(K_1 = k_1, \ldots, K_m = k_m) \;=\; \binom{n}{k_1, k_2, \ldots, k_m} \prod_{j=1}^m p_j^{k_j}$$

where $\binom{n}{k_1, k_2, \ldots, k_m} = \frac{n!}{k_1!\,k_2!\cdots k_m!}$ is the number of ways of choosing $k_1$ variables to have the value $a_1$ and $k_2$ to have value $a_2$, and so on. The mean and variance of each $K_j$ are $np_j$ and $np_j(1-p_j)$, respectively.

**Poisson** Let $Y$ be a non-negative integer-valued random variable with distribution

$$\mathbb{P}(Y = k) \;=\; \frac{e^{-\lambda}\lambda^k}{k!}$$

with parameter $\lambda > 0$. Both the mean and variance is $\lambda$.

## 2.1. Optimal Binary Classification

The goal of classification is to learn a mapping from the feature space $\mathcal{X}$ to a label space, $\mathcal{Y}$. This mapping, $f$, is called a *classifier*. For example, we might have

$$\begin{aligned} \mathcal{X} &= \mathbb{R}^d \\ \mathcal{Y} &= \{0,1\}. \end{aligned}$$

The classifier output is a prediction of the label, $\widehat{y} = f(x)$. We can measure the error of our classifier using a *loss* function; e.g., the 0-1 loss

$$\ell(\widehat{y}, y) = \mathbb{1}_{\{\widehat{y} \neq y\}} = \begin{cases} 1, & \widehat{y} \neq y \\ 0, & \widehat{y} = y \end{cases}$$

Assume that the features and labels follow a joint probability distribution. The *risk* is defined to be the expected value of the loss function, we have

$$R(f) = \mathbb{E}[\ell(f(X), Y)] = \mathbb{E}\big[\mathbb{1}_{\{f(X) \neq Y\}}\big] = \mathbb{P}(f(X) \neq Y).$$

Both the expectation and probability are with respect to a random $(X, Y)$ pair. Note that $\mathbb{1}_{\{f(X) \neq Y\}}$ is a Bernoulli random variable with probability $p = \mathbb{P}(f(X) \neq Y)$. Thus, if we have a i.i.d. dataset $\{X_i, Y_i\}_{i=1}^n$, then the total number of mistakes $f$ on these data is $\sum_{i=1}^n \mathbb{1}_{\{f(X_i) \neq Y_i\}}$ and this random variable is binomially distributed.

The performance of a given classifier can be evaluated in terms of how close its risk is to the Bayes risk.

**Definition 1** (Bayes Risk)**.** *The Bayes risk is the infimum of the risk for all classifiers:*

$$R^* = \inf_f R(f).$$

We can prove that the Bayes risk is achieved by the Bayes classifier.

**Definition 2** (Bayes Classifier)**.** *The Bayes classifier is the following mapping:*

$$f^*(x) = \begin{cases} 1, & \eta(x) \geq 1/2 \\ 0, & otherwise \end{cases}$$

*where*

$$\eta(x) \equiv \mathbb{P}_{Y|X}(Y = 1 | X = x).$$

Note that for any $x$, $f^*(x)$ is the value of $y \in \{0, 1\}$ that maximizes $P_{XY}(Y = y | X = x)$.

**Theorem 1** (Risk of the Bayes Classifier)**.**

$$R(f^*) = R^*.$$

Note that while the Bayes classifier achieves the Bayes risk, in practice this classifier is not realizable because we do not know the distribution $\mathbb{P}_{XY}$ and so cannot construct $\eta(x)$.

## 2.2. Application to Classification Error Estimation

Let $f$ be any classifier. Its probability of error is $p_f := \mathbb{P}(f(X) \neq Y) = \mathbb{E}[\mathbb{1}_{\{f(X)\neq Y\}}]$. This is generally unknown, since we typically don't know the joint distribution $\mathbb{P}_{XY}$ in practice. A common approach to estimate the error rate of a classifier $f$ is to evaluate its performance on a test set $\{X_i, Y_i\}_{i=1}^n \overset{\text{iid}}{\sim} \mathbb{P}_{XY}$. The empirical error rate is

$$\widehat{p}_f \;=\; \frac{1}{n}\sum_{i=1}^n \mathbb{1}_{\{f(X_i)\neq Y_i\}} \,.$$

Since the binary indicator variables $\mathbb{1}_{\{f(X_i)\neq Y_i\}}$ are i.i.d. $n\widehat{p}_f$ has a Binomial distribution. So the mean and variance of $\widehat{p}_f$ are $\mathbb{E}[\widehat{p}_f] = p_f$ and $\mathbb{V}(\widehat{p}_f) = \frac{p_f(1-p_f)}{n}$.

## 2.3. Application to Nearest Neighbor Classification

Suppose we are given a set of binary labeled training data $\{X_i, Y_i\}_{i=1}^n$ that are independently and identically distributed. Let $X$ be a new iid point. If we knew the data distribution, then the Bayes optimal classifier for $X$ would be to label it $1$ if $p(Y = +1|X) > P(Y = 0|X)$ and $0$ otherwise. For any $X$ let $\eta(X) = \mathbb{P}(Y = 1|X)$. The optimal classifier's probability of error is $R^* := \mathbb{E}[\min(\eta(X), 1 - \eta(X))]$. To relate this to the notation above, recall that the probability of error $\mathbb{P}(f^*(X) \neq Y) = \mathbb{E}[\mathbb{1}_{\{f^*(X)\neq Y\}}]$. The expectation is taken with respect to a random pair $(X, Y)$. We can break this into two expectations

$$\begin{aligned}\mathbb{E}[\mathbb{1}_{\{f^*(X)\neq Y\}}] &= \mathbb{E}_X\Big[\mathbb{E}_{Y|X}[\mathbb{1}_{\{f^*(X)\neq Y\}}]\Big] \\ &= \mathbb{E}_X[\min(\eta(X), 1-\eta(X))]\end{aligned}$$

The *nearest neighbor classifier* labels a new point $X$ by finding the closest point in the training set $i_X = \arg\min_i \text{dist}(X, X_i)$ and assigning the corresponding label $y_{i_X}$. The dist function could be any valid distance measure, for example the Euclidean distance $\text{dist}(X, X_i) = \|X - X_i\|_2$. Its asymptotic error rate is characterized by the following theorem.

**Theorem 2** ([8])**.** *Let $f_n^{NN}$ denote the nearest neighbor classifier based on $n$ iid training examples and for any $X$ let $\eta(X) = \mathbb{P}(Y = 1|X)$. Then*

$$\lim_{n\to\infty} \mathbb{P}(f_n^{NN}(X) \neq Y) \;=\; \mathbb{E}[2\eta(X)(1-\eta(X))]$$

The proof is a bit technical (see [12] for details), but the intuition is straightforward. If the training set is large enough, then there is a point $X' \in \{X_1, \ldots, X_n\}$ that is very close to $X$, so let's suppose that $X' = X$. Under this assumption, the labels of $X$ and $X'$, denoted $Y$ and $Y'$, are independent and identically distributed Bernoulli random variables with $p = \eta(X)$. There are two cases of error $Y' = 1$ and $Y = 0$ or $Y' = 0$ and $Y = 1$. The probability of either of these two outcomes is $\eta(X)(1 - \eta(X))$ and so the total probability of error is $2\eta(X)(1 - \eta(X))$.

Let $R_\infty^{NN} := \mathbb{E}[2\eta(X)(1 - \eta(X))]$ denote the asymptotic error of the nearest neighbor rule. Since $\eta \in [0, 1]$, we have $2\eta(1 - \eta) \geq \min(\eta, 1 - \eta)$, which implies that the asymptotic error rate of the nearest neighbor rule is never better than the optimal classifier. Also, if we let $Z = \min(\eta(X), (1 - \eta(X)))$, then

$$\begin{aligned}R_\infty^{NN} &= 2\,\mathbb{E}[Z(1-Z)] \;=\; 2\,(\mathbb{E}[Z] - \mathbb{E}[Z^2] \\ &\leq 2\,(\mathbb{E}[Z] - \mathbb{E}^2[Z])\,,\ \text{since}\ \mathbb{E}[Z^2] \geq (\mathbb{E}[Z])^2 \\ &= 2R^*(1-R^*) \;\leq\; 2R^*\end{aligned}$$

So we have shown that the asymptotic error rate of the nearest neighbor classifier is never more than twice that of the Bayes optimal classifier.

## 2.4. Application to Histogram Classifier

Let us assume that the (input) features are randomly distributed over the unit hypercube $\mathcal{X} = [0,1]^d$ (note that by scaling and shifting any set of bounded features we can satisfy this assumption), and assume that the (output) labels are binary, i.e., $\mathcal{Y} = \{0,1\}$. A histogram classifier is based on a partition the hypercube $[0,1]^d$ into $M$ smaller cubes or "bins" of equal size.

**Example 1** (Partition of hypercube in 2 dimensions)**.** *Consider the unit square* $[0,1]^2$ *and partition it into* $M$ *subsquares of equal area (assuming* $M$ *is a squared integer). Let the subsquares be denoted by* $\{B_i\}_{i=1}^M$*.*

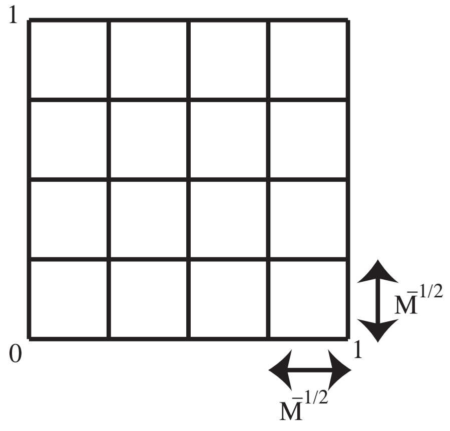


Figure 2.1: Example of hypercube $[0,1]^2$ in $M$ equally sized subsquares

A histogram classifier is any assignment of $0$ or $1$ to each bin. Given training examples $\{X_i, Y_i\}_{i=1}^n \overset{\text{iid}}{\sim} \mathbb{P}_{XY}$, a reasonable rule is to assign each bin the majority vote of the examples that fall into that bin. Specifically, for the $j$th bin define

$$\widehat{P}_j = \frac{\sum_{i=1}^n \mathbb{1}_{\{X_i \in B_j, Y_i = 1\}}}{\sum_{i=1}^n \mathbb{1}_{\{X_i \in B_j\}}} \,,$$

with the convention that $0/0 = 0$. Assign the bin the label $1$ if $\widehat{P}_j \geq 1/2$ and $0$ otherwise. Equivalently, define the following piecewise-constant estimator of $\eta(x)$:

$$\widehat{\eta}_n(x) = \sum_{j=1}^M \widehat{P}_j \mathbb{1}_{\{x \in B_j\}}$$

and classify according to

$$\widehat{f}_n^H(x) = \begin{cases} 1, & \widehat{\eta}_n(x) \geq 1/2 \\ 0, & otherwise \end{cases}$$

The histogram classifier may differ from the Bayes classifier in two ways:

**bias:** its classification rule is constant on each bin

**variance:** the majority vote may not be the optimal rule for each bin

The bias tends to 0 as $M \to \infty$, and the variance tends to 0 as $n \to \infty$. Thus, if $M, n \to \infty$ the histogram classifier may converge to the Bayes classifier. Formally, we have the following theorem which proves that histogram classifiers are *universally consistent*, meaning its error rate converges to the Bayes error rate. The histogram classifier is similar to the nearest-neighbor classifier. Both label a new example based on the examples in the training set that are close to it. The key difference is that the histogram classifier prediction is effectively the *majority vote* of a number of nearby examples. This averaging effect enables it to achieve near-optimal performance with a sufficiently large training set.

**Theorem 3** (Consistency of Histogram Classifiers). *If $M \to \infty$ and $\frac{n}{M} \to \infty$ as $n \to \infty$, then as $n \to \infty$ the error rate of the histogram classifier $\mathbb{P}(f_n^H(X) \neq Y) \to R^*$, the Bayes risk, for every distribution $\mathbb{P}_{XY}$.*

Here is a sketch of the main ideas in the proof (a more formal proof is given below). The histogram classifier assigns each bin the label of the majority of training data in the bin. Equivalently, the label for bin $B_j$ is 1 if

$$\widehat{p}_j := \frac{\sum_{i=1}^n \mathbb{1}_{\{x_i \in B_j, y_i = 1\}}}{\sum_{i=1}^n \mathbb{1}_{\{x_i \in B_j\}}}$$

is larger than $1/2$ and 0 otherwise. The bias of the histogram classifiers is due to the fact that it must assign the same label to every point in each bin, whereas the optimal classifier can be arbitrary. However, as $M \to \infty$ the bins get smaller and smaller and this piecewise constant restriction becomes less and less of a limitation. This implies that as $M$ grows, there is a histogram classifier that can approximate the optimal classifier to arbitrary accuracy. Think of it this way. Let $p_j$ be the probability of $Y = 1$ for a random $X$ in this bin. The optimal histogram decision is to assign the label 1 to the bin if $p_j > 1/2$. As the bin size shrinks to 0 around a specific point $x$, the value of $p_j$ tends to $\mathbb{P}(Y = 1|X = x)$.

What we need is to show that $\widehat{p}_j$ is a good estimator of $p_j$. Let $k_j = \sum_{i=1}^n \mathbb{1}_{\{x_i \in B_j, y_i = 1\}}$ and $n_j = \sum_{i=1}^n \mathbb{1}_{\{x_i \in B_j\}}$, so that $\widehat{p}_j = k_j/n_j$. Notice that given $n_j$, the random variable $k_j$ is binomially distributed with parameter $p_j$. Therefore, $\mathbb{E}[k_j|n_j] = p_j n_j$ and $\mathbb{V}[k_j|n_j] = n_j p_j(1 - p_j)$. If $n/M \to \infty$, then the average number of examples per bin tends to infinity. So we can conclude that $n_j \to \infty$. Combining this with the fact that $k_j|n_j$ is binomially distributed, it follows that the variance of $k_j/n_j \to 0$ and therefore $k_j/n_j \to p_j$ as $n \to \infty$.

Here is the more formal proof of the theorem.

*Proof.* The proof can be found in [12, Chapter 6]. We will prove the result under some minor assumptions that simplify things. Assume that $\eta(x) = \mathbb{P}(Y = 1|X = x)$ is uniformly continuous[3] and that the marginal density $p(x) \geq c$, for some constant $c > 0$. Let $f^*$ denote the Bayes optimal classifier (defined by $\eta$). It is easily verified that

$$\mathbb{P}(\widehat{f}_n^H(X) \neq Y) - \mathbb{P}(f^*(X) \neq Y) \;\leq\; 2\mathbb{E}[|\widehat{\eta}_n(X) - \eta(X)|] \,,$$

so we will focus the convergence of this upper bound.

Let $P_j \equiv \frac{\int_{B_j} \eta(x) p_X(x) dx}{\int_{B_j} p_X(x) dx}$ (the theoretical analog of $\widehat{P}_j$) and define

$$\bar{\eta}(x) = \sum_{j=1}^M P_j \, \mathbb{1}_{\{x \in B_j\}}$$

The function $\bar{\eta}$ is constant on each bin and its value on each bin is the average of $\eta$. By the triangle inequality,

$$\mathbb{E}[|\widehat{\eta}_n(X) - \eta(X)|] \;\leq\; \underbrace{\mathbb{E}[|\bar{\eta}(X) - \eta(X)|]}_{\text{approximation error}} + \underbrace{\mathbb{E}[|\widehat{\eta}_n(X) - \bar{\eta}(X)|]}_{\text{estimation error}}$$

[3] A function $f$ is uniformly continuous if, for every $\epsilon > 0$ there exists a $\delta > 0$, such that $|x - y| < \delta$ implies $\|f(y) - f(x)\| < \epsilon$.

The expectation above is over both the training data (which define $\widehat{\eta}_n$) and the new test point $X$. We will show that $\mathbb{E}[|\widehat{\eta}_n(X) - \eta(X)|] \to 0$ as $M$ and $n$ grow, per the statement of the theorem. Since the Bayes risk is determined by $\eta(X)$, this proves that the histogram classifier's error converges to the Bayes risk.

The bias is bounded as follows.

$$\begin{aligned}
\mathbb{E}[|\bar{\eta}(X) - \eta(X)|] &= \sum_{j=1}^{M} \int_{B_j} |\bar{\eta}(x) - \eta(x)|\, p_X(x) dx \\
&\leq \sum_{j=1}^{M} \int_{B_j} \epsilon_M\, p_X(x) dx\,, \quad \text{for some small } \epsilon_M \text{ by continuity of } \eta \\
&= \sum_{j=1}^{M} \epsilon_M\, \mathbb{P}(X \in B_j), \\
&= \epsilon_M, \qquad \text{since } \sum_{j=1}^{M} \mathbb{P}(X \in B_j) = 1
\end{aligned}$$

By taking $M$ sufficiently large, $\epsilon_M$ can be made arbitrarily small. Thus $\mathbb{E}[|\eta(X) - \bar{\eta}(X)|] \to 0$.

The variance is bounded by noting that $\widehat{P}_j$ is proportional to a binomial random variable. For any $x \in [0,1]^d$, let $B(x)$ denote the histogram bin in which $x$ falls in. Define the random variables

$$N(x) = \sum_{i=1}^{n} \mathbb{1}_{\{X_i \in B(x)\}} \text{ and } K(x) = \sum_{i=1}^{n} \mathbb{1}_{\{X_i \in B(x),\, Y_i = 1\}}$$

Then

$$\widehat{\eta}_n(x) = \frac{K(x)}{N(x)}\,.$$

Note that

$$K(x) \,|\, \{N(x) = n_x\} \;\sim\; \text{Binomial}(n_x, \bar{\eta}(x))$$

since $\bar{\eta}(x)$ is the probability of a sample in $B(x)$ having the label 1 and we are conditioning on the event of observing $n_x$ samples in $B(x)$. Therefore,

$$\begin{aligned}
\mathbb{E}[\widehat{\eta}_n(x)] &= \mathbb{E}\big[\mathbb{E}[\widehat{\eta}_n(x) | N(x) = n_x]\big] \\
&= \mathbb{E}[\bar{\eta}(x)] \;=\; \bar{\eta}(x)
\end{aligned}$$

and for any $n_x > 0$

$$\mathbb{E}[|\widehat{\eta}_n(x) - \bar{\eta}(x)|^2 \,|\, N(x) = n_x] \;=\; \frac{\bar{\eta}(x)(1 - \bar{\eta}(x))}{n_x}$$

By convexity (i.e., Jensen's inequality), $\mathbb{E}[|Z|] \leq (\mathbb{E}[|Z|^2])^{\frac{1}{2}}$ for any random variable $Z$. So we have

$$\begin{aligned}
\mathbb{E}[|\widehat{\eta}_n(x) - \bar{\eta}(x)| \,|\, N(x) = n_x] &\leq \Big(\mathbb{E}[|\widehat{\eta}_n(x) - \bar{\eta}(x)|^2 \,|\, N(x) = n_x]\Big)^{1/2} \\
&= \sqrt{\frac{\bar{\eta}(x)(1 - \bar{\eta}(x))}{n_x}}
\end{aligned}$$

Since $p_X(x) \geq c > 0$ and $n/M \to \infty$, it follows that $N(x) \to \infty$. Therefore

$$\mathbb{E}[|\widehat{\eta}_n(x) - \bar{\eta}(x)| \,|\, N(x)] \overset{a.s.}{\to} 0$$

and thus $\mathbb{E}[|\widehat{\eta}_n(x) - \bar{\eta}(x)|] \to 0$. The symbol $\overset{a.s.}{\to}$ means convergence *almost surely* or equivalently *with probability* 1. Since this holds for every $x \in [0,1]^d$, it follows that $\mathbb{E}[|\widehat{\eta}_n(X) - \bar{\eta}(X)|] \to 0$, where the expectation is over the training data and a randomly chosen $X$. □

## 2.5. Summary

We defined the optimal binary classifier $f^*$ and showed that the nearest-neighbor classifier $f_n^{NN}$ and this histogram classifier $f_n^H$ can perform nearly or exactly as well in the limit as the training set size $n \to \infty$. So is this the end of the machine learning story? No. In practice, training set sizes are limited, and more sophisticated approaches such as kernel methods and neural networks tend to work much better. We will see why this may be so in later lectures.

## 2.6. Exercises

1. Consider the binary classification problem. Let $\eta(x) = \mathbb{P}(Y = 1|X = x)$ and recall that the Bayes Classifier is defined as
$$f^*(x) = \begin{cases} 1, & \eta(x) \geq 1/2 \\ 0, & \text{otherwise} \end{cases}$$
Let $g(x)$ be any other classifier. Prove that
$$\mathbb{P}(g(X) \neq Y) \;\geq\; \mathbb{P}(f^*(X) \neq Y)\,.$$

2. Consider a binary classification problem with $(X, Y) \sim \mathbb{P}_{XY}$. Recall the Bayes optimal classifier:
$$f^*(x) \;=\; \begin{cases} 1 & \text{if } \eta(x) \geq 1/2 \\ 0 & \text{otherwise} \end{cases}$$
where $\eta(x) = \mathbb{P}(Y = 1|X = x)$. Let $\widetilde{\eta}$ denote any approximation to $\eta$ and consider the "plug-in" classifier
$$f(x) \;=\; \begin{cases} 1 & \widetilde{\eta}(x) \geq 1/2 \\ 0 & \text{otherwise} \end{cases}$$
Show that
$$\mathbb{P}(f(X) \neq Y) - \mathbb{P}(f^*(X) \neq Y) \;\leq\; 2\,\mathbb{E}[|\eta(X) - \widetilde{\eta}(X)|]\,.$$

3. Let $X$ be a nonnegative random variable. Prove Markov's inequality:
$$\mathbb{P}(X \geq t) \leq \frac{\mathbb{E}[X]}{t}$$

4. A common approach to estimate the error rate of a classifier $f$ is to evaluate its performance on a test set $\{X_i, Y_i\}_{i=1}^n \overset{\text{iid}}{\sim} \mathbb{P}_{XY}$. The empirical error rate is
$$\widehat{p}_f \;=\; \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{f(X_i) \neq Y_i}\,.$$
Let $p_f := \mathbb{P}(f(x) \neq Y)$ and show that for any $\epsilon > 0$,
$$\mathbb{P}(|\widehat{p}_f - p_f| \geq \epsilon) \;\leq\; \frac{p_f(1 - p_f)}{n\epsilon^2}\,.$$

5. Let $X$ be a random variable. Prove that $\mathbb{E}[X^2] \geq (\mathbb{E}[X])^2$.

6. Let $X$ be a discrete random variable taking values in the set $\{x_1, x_2, \ldots, x_k\} \subset \mathbb{R}$. Prove *Jensen's inequality*: For any convex function $\varphi$

$$\mathbb{E}[\varphi(X)] \geq \varphi(E[X]) \ .$$

   Hint: By the definition of convexity, for any $\lambda \in [0, 1]$ and $x_1, x_2 \in \mathbb{R}$ we have $\varphi(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda\varphi(x_1) + (1 - \lambda)\varphi(x_2)$. Use this fact and induction on $k$ to prove the result. Jensen's inequality also holds for continuous real-valued random variables (essentially the limit of discrete distributions).

7. Let $X$ be a random variable. Prove that $\mathbb{E}[|X|^3] \geq (\mathbb{E}[|X|])^3$. (Hint: apply Jensen's inequality)

8. Show that the mean and variance of a Poisson random variable with distribution $\mathbb{P}(X = k) = e^{-\lambda}\frac{\lambda^k}{k!}$ is $\lambda$.

9. Suppose $X_1$ and $X_2$ are independent Poisson random variables with parameters $\lambda_1$ and $\lambda_2$. 1) Show that the conditional distribution of $X_1$ given $X_1 + X_2 = n$ is binomial with parameters $n$ and $\frac{\lambda_1}{\lambda_1+\lambda_2}$. 2) Show that $X_1 + X_2$ is also a Poisson random variable with parameter $\lambda_1 + \lambda_2$.

10. What is the minimum probability of error classifier in the multiclass setting? Assume $m > 2$ classes and knowledge of the joint distribution $\mathbb{P}_{XY}$.

11. Derive an expression for the minimum probability of error in the multiclass setting in terms of appropriate functionals of the joint distribution (similar to the expression we derived in the binary classification setting).

12. Consider estimating the error rate of a classifier $f$ in a multiclass setting. Given an expression for the estimator based on a training set $\{(X_i, Y_i)\}_{i=1}^n \overset{\text{iid}}{\sim} \mathbb{P}_{XY}$.

13. Following from the previous problem, suppose that $n = 1000$ and the estimated error rate is $0.05$. Are you confident that the true error is probably less than $0.10$?

# Lecture 3: Multivariate Gaussian Models and Classification

A common framework in ML is to suppose that the training data $\{\boldsymbol{x}_i, y_i\}_{i=1}^n \overset{\text{iid}}{\sim} P_{xy}$, where $P_{xy}$ is the joint distribution over pairs $(\boldsymbol{x}, y)$. The *generative* approach to designing a classifier is to fit a model to the training data, and then use this model to derive a classifier; e.g., based on $p(y|\boldsymbol{x})$. One of the most common probability models in ML is the multivariate Gaussian (or Normal) model, abbreviated MVN. Let $\boldsymbol{x} \in \mathbb{R}^d$. The MVN density function is given by

$$p(\boldsymbol{x}) \;=\; \frac{1}{\sqrt{(2\pi)^d|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right),$$

where $\boldsymbol{\mu} \in \mathbb{R}^d$ is the mean, $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$ is the covariance matrix, and $|\boldsymbol{\Sigma}|$ is the determinant of the covariance matrix. If $\boldsymbol{x}$ is a random vector distributed according to this density function, then we write $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ for shorthand notation. Some bivariate Gaussians are depicted in the figure below. In this note we will examine how the MVN model can be used to derive linear (and nonlinear) classifiers.

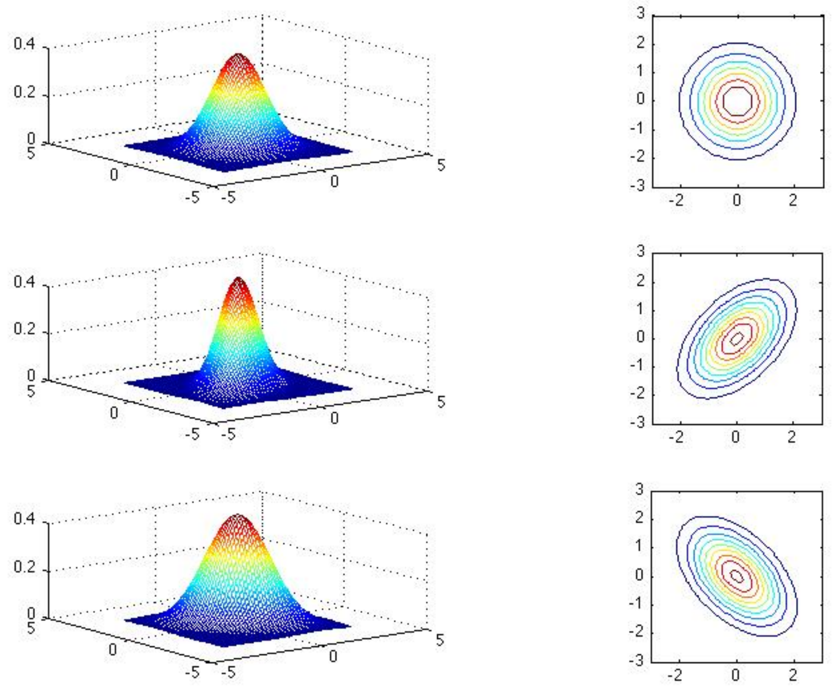


Figure 3.1: Multivariate Gaussian Distributions. The top row show the Gaussian density (left) and its contour plot (right) with mean $[0\ 0]^T$ and covariance $[1\ 0; 0\ 1]$. The second row is the same except that the covariance is $[1\ 0.5; 0.5\ 1]$; positive correlation. The third row is the same except that the covariance is $[1\ -0.5; -0.5\ 1]$; negative correlation.

One of the most important and special properties of the Gaussian distribution is that linear transformations of Gaussian random variables are Gaussian distributed. For example, the sum of two Gaussian random variables is Gaussian. The general rule for affine transformations is: if $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then for any compatible matrix $\boldsymbol{A}$ and vector $\boldsymbol{b}$ the transformed variable $\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} \sim \mathcal{N}(\boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}, \boldsymbol{A}\Sigma\boldsymbol{A}^T)$. This is a very special property that does not hold for random variables in general. This linear transformation property can be proved using Fourier transforms.

## 3.1. MVN Models and Classification

Consider a multiclass prediction problem and let $p(\boldsymbol{x}|y=j)$, $j=0,1,\ldots,k$ denote the *class conditional* distributions of the feature $\boldsymbol{x}$; i.e., the features of examples belonging to class $j$ are distributed according to the density function $p(\boldsymbol{x}|y=j)$. In this note we focus on the special case where the class-conditional densitities are Gaussian:

$$\boldsymbol{x}|y=j \;\sim\; \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j).$$

Note that each class-conditional density is Gaussian with its own mean and covariance.

Recall that the optimal classification rule is

$$\widehat{y}(\boldsymbol{x}) \;=\; \arg\max_j p(y=j|\boldsymbol{x}) \;,$$

where $p(y=j|\boldsymbol{x})$ is the probability of that $y=j$ given the feature $\boldsymbol{x}$. We say it is optimal because it minimizes the probability of error. This can be related to the class-conditional densities via Bayes Rule:

$$p(y=j|\boldsymbol{x}) \;=\; \frac{p(\boldsymbol{x}|y=j)p(y=j)}{p(\boldsymbol{x})} \;.$$

$p(y=j)$ is the marginal probability that $y=j$, i.e., the probability that an random example has label $j$. This is also sometimes called the *prior* probability that $y=j$, since it is the probability prior to having seen the feature associated with $y$. $p(\boldsymbol{x})$ is the marginal density of $\boldsymbol{x}$. When predicting the label given $\boldsymbol{x}$, the denominator $p(\boldsymbol{x})$ is a constant, so the optimal classification rule can be expressed as

$$\widehat{y}(\boldsymbol{x}) \;=\; \arg\max_j p(\boldsymbol{x}|y=j)p(y=j) \;.$$

Consider the special case of binary classification. In this case,

$$\begin{aligned}
\widehat{y}(\boldsymbol{x}) \;&=\; \arg\max_{j\in\{0,1\}} p(\boldsymbol{x}|y=j) \\
&=\; \begin{cases} 1 & \text{if } \frac{p(\boldsymbol{x}|y=1)}{p(\boldsymbol{x}|y=0)} > \frac{p(y=0)}{p(y=1)} \\[2ex] 0 & \text{if } \frac{p(\boldsymbol{x}|y=1)}{p(\boldsymbol{x}|y=0)} < \frac{p(y=0)}{p(y=1)} \end{cases} \\[2ex]
&=\; \begin{cases} 1 & \text{if } \log\frac{p(\boldsymbol{x}|y=1)}{p(\boldsymbol{x}|y=0)} > \log\frac{p(y=0)}{p(y=1)} \\[2ex] 0 & \text{if } \log\frac{p(\boldsymbol{x}|y=1)}{p(\boldsymbol{x}|y=0)} < \log\frac{p(y=0)}{p(y=1)} \end{cases}
\end{aligned}$$

This is called the log-likelihood ratio test. If we have Gaussian class-conditional densities, then the log-likelihood ratio is a quadratic function in $\boldsymbol{x}$. So the decision boundary (the set of $\boldsymbol{x}$ where the log-likelihood ratio is 0) is a quadratic curve/surface in the feature space. For the special case of Gaussian class-conditional densities with equal covariances (i.e., $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}$) and equal prior probabilities (i.e., $p(y=0) = p(y=1) = 1/2$), the log-likelihood ratio simplifies to

$$\widehat{y}(\boldsymbol{x}) \;=\; \begin{cases} 1 & \text{if } 2(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} \;\geq\; \boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 \\[2ex] 0 & \text{otherwise} \end{cases}$$

If we let $\boldsymbol{w} = 2\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$ and $b = \boldsymbol{\mu}_0^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1$, then we see that this is just a linear classifier: $\boldsymbol{w}^T\boldsymbol{x} + b > 0$.

## 3.2. Optimality of Likelihood Ratio

If the class-conditional densities are *known* and the prior probabilities of each class are *equal*, then the maximum likelihood classification rule $\widehat{y}(\boldsymbol{x}) = \arg\max_j p(\boldsymbol{x}|y = j)$ is optimal, in the sense that it minimizes the probability of making a mistake[4]. Here we provide another interpretation of this optimality for the special case of two classes (binary classification). The generalization to multiple classes is straightforward.

To keep the notation simple, let $p_j$, $j = 0, 1$, denote the two class-conditional distributions. Assume that we observe a random variable distributed according to one of two distributions.

$$\begin{aligned} H_0 : \boldsymbol{x} &\sim p_0 \\ H_1 : \boldsymbol{x} &\sim p_1 \end{aligned}$$

Deciding which of the two best fits an observation of $\boldsymbol{x}$ is called a *simple* binary hypothesis test, simple because the two distributions are known precisely (i.e., without unknown parameters or other uncertainties). A decision is made by partitioning the range of $\boldsymbol{x}$ into two disjoint regions. Let us denote the regions by $R_0$ and $R_1$. If $\boldsymbol{x} \in R_i$, then we decide that $H_i$ is the best match to the data; i.e., we decide that the data were distributed according to $p_i$. The key question is how to design the *decision regions* $R_0$ and $R_1$. Note that since $R_0 \bigcup R_1$ is assumed to be the entire range of $\boldsymbol{x}$, $R_1$ is simply the complement of $R_0$, and so the choice of either region determines the other.

There are four possible outcomes in a test of this form, depending on the decision we make ($H_0$ or $H_1$), and the true distribution of the data (also $H_0$ or $H_1$). Let us denote these as $(0, 0)$, $(0, 1)$, $(1, 0)$, and $(1, 1)$, where the first argument denotes the decision based on the regions $R_0$ and $R_1$ and the second denotes the true distribution that generated $\boldsymbol{x}$. Note that the outcomes $(0, 1)$ and $(1, 0)$ are mistakes or *errors*. The test made the wrong decision about which distribution generated $\boldsymbol{x}$.

In order to optimize the choice of decision regions, we can specify a *cost* for incorrect (and correct, if we wish) decisions. Without loss of generality, let's assume the costs are non-negative. Let $c_{i,j}$ be the cost associated with outcome $(i, j)$, $i, j \in \{0, 1\}$. The costs reflect the relative importance of correct and incorrect decisions. Since our aim is to design a test that makes few mistakes, it is reasonable to assume that $c_{1,0}$ and $c_{0,1}$ are larger than $c_{0,0}$ and $c_{1,1}$; in fact often it is reasonable to assign a zero cost to correct decisions. The costs $c_{0,1}$ and $c_{1,0}$ may be different. For example, it may be that one type of error is more problematic than the other.

The overall cost associated with a test (i.e., with decision regions $R_0$ and $R_1$) is usually called the *Bayes Cost*, and it is defined as follows.

$$C = \sum_{i,j=0}^{1} c_{i,j} \pi_j \, \mathbb{P}(\text{decide } H_i \mid H_j \text{ is true})$$

where $\pi_j := p(y = j)$, $j = 0, 1$, is called the prior probability of $H_j$, and $\mathbb{P}(\text{decide } H_i \mid H_j \text{ is true})$ denotes the probability of deciding $H_i$ when $H_j$ generated $\boldsymbol{x}$. $\pi_j$ is the probability that an observation will be generated according to $p_j$. The prior probabilities sum to 1, since we assume the data are generated either according to $p_0$ or $p_1$, but they need not be equal. One distribution may be more probable than the other (e.g., more people are healthy than have a disease). Our goal is to design the decision regions in order to minimize the *Bayes Cost*.

The Bayes Cost can be expressed directly in terms of the decision regions as follows. We will assume that $p_0$ and $p_1$ are continuous densities, but an analogous representation exists when they are discrete probability mass

[4] If the prior probabilities are not equal, then the optimal classification rule is simply $\widehat{y}(\boldsymbol{x}) = \arg\max_j p(\boldsymbol{x}|y = j)\, p(y = j)$, where $p(y = j)$ is the prior probability of class $j$.

functions (i.e., replace integrals with sums in expressions below).

$$\begin{aligned} C &= \sum_{i,j=0}^{1} c_{i,j}\pi_j \, \mathbb{P}(\text{decide } H_i \,|\, H_j \text{ is true}) \\ &= \sum_{i,j=0}^{1} c_{i,j}\pi_j \, \mathbb{P}(\boldsymbol{x} \in R_i \,|\, H_j \text{ is true}) \\ &= \sum_{i,j=0}^{1} c_{i,j}\pi_j \int_{R_i} p_j(\boldsymbol{x})\, d\boldsymbol{x} \end{aligned}$$

The choice of $R_0$ and $R_1$ that minimizes the cost $C$ becomes obvious if we expand the sum above.

$$\begin{aligned} C &= \sum_{i,j=0}^{1} c_{i,j}\pi_j \int_{R_i} p_j(\boldsymbol{x})\, d\boldsymbol{x} \\ C &= \int_{R_0} (c_{0,0}\pi_0\, p_0(\boldsymbol{x}) + c_{0,1}\pi_1\, p_1(\boldsymbol{x}))\, d\boldsymbol{x} \; + \; \int_{R_1} (c_{1,0}\pi_0\, p_0(\boldsymbol{x}) + c_{1,1}\pi_1\, p_1(\boldsymbol{x}))\, d\boldsymbol{x} \end{aligned}$$

The integrands are non-negative, so it follows that we should let $R_0$ be the set of $\boldsymbol{x}$ for which the first integrand is smaller than the second. That is,

$$\begin{aligned} R_0 &:= \{\boldsymbol{x} \,:\, c_{0,0}\pi_0\, p_0(\boldsymbol{x}) + c_{0,1}\pi_1\, p_1(\boldsymbol{x}) < c_{1,0}\pi_0\, p_0(\boldsymbol{x}) + c_{1,1}\pi_1\, p_1(\boldsymbol{x})\} \\ R_1 &:= \{\boldsymbol{x} \,:\, c_{0,0}\pi_0\, p_0(\boldsymbol{x}) + c_{0,1}\pi_1\, p_1(\boldsymbol{x}) > c_{1,0}\pi_0\, p_0(\boldsymbol{x}) + c_{1,1}\pi_1\, p_1(\boldsymbol{x})\} \end{aligned}$$

Therefore, the optimal test (relative to the assigned costs) takes the following simple form:

$$\frac{p_1(\boldsymbol{x})}{p_0(\boldsymbol{x})} \underset{H_0}{\overset{H_1}{\gtrless}} \frac{\pi_0(c_{1,0} - c_{0,0})}{\pi_1(c_{0,1} - c_{1,1})} \, .$$

Note that the term on the right hand side is a constant that depends on the prior probabilities and the costs (i.e., it does not depend on $\boldsymbol{x}$). The term of the left hand side is a ratio of probability densities evaluated at $\boldsymbol{x}$. The value of a probability density at the observed $\boldsymbol{x}$ is called the *likelihood* of $\boldsymbol{x}$ under that model. Thus, $\frac{p_1(\boldsymbol{x})}{p_0(\boldsymbol{x})}$ is called the *likelihood ratio* and the test is called the *likelihood ratio test* (LRT). No matter what the prior probabilities are or how costs are assigned, the optimal test *always* has takes the form

$$\frac{p_1(\boldsymbol{x})}{p_0(\boldsymbol{x})} \underset{H_0}{\overset{H_1}{\gtrless}} \gamma \, ,$$

where $\gamma > 0$ is the *threshold* of the test. We have shown is that the LRT, with an appropriate threshold, is optimal.

## 3.3. Exercises

1. Consider a binary classification problem with class-conditional densities $p(\boldsymbol{x}|y = j)$ given by MVNs $\mathcal{N}(\boldsymbol{\mu}_j, \Sigma)$, $j = 0, 1$.

   (a) Show that the log likelihood ratio test reduces to the form

   $$(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x} \underset{H_0}{\overset{H_1}{\gtrless}} \gamma$$

   where $\gamma$ is a threshold.

(b) Let $t(\boldsymbol{x}) = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x}$ denote the scalar *test statistic*. What is the distribution of $t(\boldsymbol{x})$ when $\boldsymbol{x}$ is drawn from $\mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma})$ (i.e., when it is from class $j$)?

(c) Let $Q(z) = \int_z^\infty \frac{1}{\sqrt{2\pi}} e^{-t^2/2}\, dt$, the probability that a $\mathcal{N}(0,1)$ random variable exceeds the value $z$. Show that the probability of mistakenly predicting a label of 1 when the true label is 0 is given by

$$Q\left( \frac{\gamma - (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_0}{\Big( (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) \Big)^{1/2}} \right)$$

(d) Let $\boldsymbol{A} \in R^{k \times d}$, with $k < d$. Then $\widetilde{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} \in \mathbb{R}^k$. Let us view $\widetilde{\boldsymbol{x}}$ as a means of reducing the dimensionality of $\boldsymbol{x}$. What are the class-conditional distributions of $\widetilde{\boldsymbol{x}}$? If we base our classification on $\widetilde{\boldsymbol{x}}$ rather than $\boldsymbol{x}$, will the optimal classifier perform better or worse?

(e) Prove that the covariance matrix $\boldsymbol{\Sigma}$ is positive semidefinite; i.e., $\boldsymbol{v}^T \boldsymbol{\Sigma} \boldsymbol{v} \geq 0$ for every $\boldsymbol{v} \in \mathbb{R}^d$.

(f) Let $\boldsymbol{\Sigma} = \sum_{i=1}^d \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^T$ be the eigendecomposition. Since $\boldsymbol{\Sigma}$ is symmetric and positive semidefinite (by construction) the eigenvalues $\lambda_i \geq 0$. $\boldsymbol{u}_i$ is the eigenvector associated with eigenvalue $\lambda_i$; i.e., $\boldsymbol{\Sigma}\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i$. Express the inverse $\boldsymbol{\Sigma}^{-1}$ in terms of the eigenvalues and eigenvectors.

(g) Use the results above to design a linear dimensionality reduction from $d$ to $k$ that introduces the least distortion to the Mahanalobis distance, the distance appearing in the MVN model.

(h) Suppose that

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.9 & 0.9 \\ 0.9 & 1 & 0.9 \\ 0.9 & 0.9 & 1 \end{bmatrix}.$$

Find the eigendecomposition and the best reduction to $k = 1$ dimension.

2. The Bayes optimal classifier minimizes the total probability of error at the point $x$. Recall there are two types of errors, false positives and false negatives. The $f^*$ treats both as equally costly losses (loss of 1 for both types of error).

(a) Suppose that we assign different costs to the two types of error. Let $c_{01}$ be the cost/loss of predicting $\widehat{y} = 0$ when the true label is $y = 1$ and let $c_{10}$ be the cost/loss of the other type of error. Suppose that we do not know the prior probabilities, so we simply guess that they are equal. Derive an expression for the optimal classifier in this case.

(b) Now suppose that we additionally know the probabilities of the two classes; i.e., we know that $\mathbb{P}(Y = 1) = p$ and $\mathbb{P}(Y = 0) = 1 - p$. Show that the $f^*$ above does not minimize the average probability of error (expectation over $X$) if $p \neq 1/2$.

(c) Derive a new classifier that is optimal for given costs $c_{01}$ and $c_{10}$ and prior probability $p$.

3. Consider a binary classification problem where the features $\boldsymbol{x} \in \{-1, +1\}^d$ and the label $y \in \{-1, 1\}$.

(a) Let $\boldsymbol{x} = [x_1, \cdots, x_d]^T$ and suppose that $y = x_1$ and $x_2, \ldots, x_d$ are independent of $x_1$ and $y$ and are i.i.d. binary random variables with $\mathbb{P}(x_i = +1) = \mathbb{P}(x_i = -1) = 1/2$ for $i = 2, \ldots, d$. What is the Bayes optimal classifier and Bayes Risk?

(b) Now consider the nearest-neighbor classifier based on a labeled training set of size $n = 2$ consisting of one randomly chosen example form each class. Find a lower bound on the probability of error of the nearest-neighbor classifier for $d = 2, 3$. (If both training points are equally close to the test point, then randomly pick either label with equal probability). HINT: Consider the test point a fixed binary vector and consider the randomness in the two training points.

(c) Use the binomial distribution to derive an expression for the error bound for general values of $d$. How does the computational complexity of this calculation grow as a function of $d$?

(d) Conjecture about the limiting error rate as $d \to \infty$.

# Lecture 4: Learning MVN Classifiers

Suppose we are given training data $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$. We can fit MVN models to these data (separating the data according to their labels), and then derive a classifier based on the fitted models.

A key question is how to fit models to the data. There are many ways to do this, but one natural approach is to match the empirical (data-based) moments to the moments of the MVN model. The MVN model is defined by its first two moments, the mean and covariance. So we can just set the mean and covariance of the model to the empirical mean and covariance.

Consider the subset of the training data with the label $j$, that is $\{\boldsymbol{x}_i\}_{i:y_i=j}$, where $j$ is one of the possible labels. Let $n_j$ denote the number of examples in this set. The empirical mean and covariance of these data are computed as follows:

$$\begin{aligned}\widehat{\boldsymbol{\mu}}_j &= \frac{1}{n_j}\sum_{i:y_i=j}\boldsymbol{x}_i\\ \widehat{\boldsymbol{\Sigma}}_j &= \frac{1}{n_j}\sum_{i:y_i=j}(\boldsymbol{x}_i-\widehat{\boldsymbol{\mu}}_j)(\boldsymbol{x}_i-\widehat{\boldsymbol{\mu}}_j)^T\end{aligned}$$

Then we "plug-in" these estimates to obtain a MVN model for data in the class $j$; i.e., $p(\boldsymbol{x}|y=j)$ is the MVN density $\mathcal{N}(\widehat{\boldsymbol{\mu}}_j, \widehat{\boldsymbol{\Sigma}}_j)$.

If we assume the two class-conditional distributions have the same covariance, then we can estimate the covariance as follows.

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^n(\boldsymbol{x}_i-\widehat{\boldsymbol{\mu}}_{y_i})(\boldsymbol{x}_i-\widehat{\boldsymbol{\mu}}_{y_i})^T$$

As discussed above, the optimal classifier is linear in this case. This type of linear classifier is referred to as *Fisher's linear discriminant* [16].

## 4.1. Analysis of the "Plug-in" MVN Classifier

Consider the following simplified case. Suppose we are given $n$ i.i.d. training examples

$$\boldsymbol{x}|y=+1 \sim \mathcal{N}(\boldsymbol{\theta},\boldsymbol{I}) \text{ and } \boldsymbol{x}|y=-1 \sim \mathcal{N}(-\boldsymbol{\theta},\boldsymbol{I})\ .$$

Assume that $\boldsymbol{\theta}\in\mathbb{R}^d$ and that $\mathbb{P}(y=+1)=\mathbb{P}(y=-1)$. In other words, the two classes are equally probable and have means $\boldsymbol{\theta}$ and $-\boldsymbol{\theta}$ and the covariance matrices are both *known* to be $\boldsymbol{I}$, the identity matrix. The likelihood ratio classifier (Bayes classifier) minimizes the probability of error. The log likelihood ratio in this case is

$$\log\left(\frac{p(\boldsymbol{x}|y=+1)}{p(\boldsymbol{x}|y=-1)}\right) = \frac{1}{2}(\boldsymbol{x}+\boldsymbol{\theta})^T(\boldsymbol{x}+\boldsymbol{\theta})-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\theta})^T(\boldsymbol{x}-\boldsymbol{\theta}) = 2\boldsymbol{x}^T\boldsymbol{\theta}$$

So the optimal classification rule is

$$f^*(\boldsymbol{x}) = \begin{cases}+1 & \text{if } \boldsymbol{x}^T\boldsymbol{\theta}>0\\ -1 & \text{if } \boldsymbol{x}^T\boldsymbol{\theta}<0\end{cases}.$$

This achieves the minimum probability of error, which is given by

$$\begin{aligned}\mathbb{P}(f^*(\boldsymbol{x}) \neq y) &= \mathbb{P}(\boldsymbol{x}^T\boldsymbol{\theta} > 0|y=-1)\mathbb{P}(y=-1) + \mathbb{P}(\boldsymbol{x}^T\boldsymbol{\theta} < 0|y=+1)\mathbb{P}(y=+1) \\ &= \mathbb{P}(\boldsymbol{x}^T\boldsymbol{\theta} > 0|y=-1)\,,\end{aligned}$$

where the last equality follows since the two types of error are equal because of the symmetry of the problem. Note that $\boldsymbol{x}^T\boldsymbol{\theta}|y=-1 \sim \mathcal{N}(-\|\boldsymbol{\theta}\|^2, \|\boldsymbol{\theta}\|^2)$. So the probability of error is equal to the probability that a $\mathcal{N}(0, \|\boldsymbol{\theta}\|^2)$ random variable exceeds $\|\boldsymbol{\theta}\|^2$. This can be bounded by Markov's inequality: if $z \sim \mathcal{N}(0, \|\boldsymbol{\theta}\|^2)$, then $\mathbb{P}(z > \|\boldsymbol{\theta}\|^2) \leq \frac{\mathbb{E}[z^2]}{\|\boldsymbol{\theta}\|^4} = \frac{1}{\|\boldsymbol{\theta}\|^2}$. So we have shown that $\mathbb{P}(f^*(\boldsymbol{x}) \neq y) \leq \frac{1}{\|\boldsymbol{\theta}\|^2}$, which makes sense since the error should decrease as the distance between the means increases (note that the distance between the means is $2\|\boldsymbol{\theta}\|$).

Now let's consider a learning set-up in which we don't know the value of $\boldsymbol{\theta}$, but we do have a training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$. Note that $\boldsymbol{x}_i|y_i \sim \mathcal{N}(y_i\boldsymbol{\theta}, \boldsymbol{I})$ and so $y_i\boldsymbol{x}_i|y_i \sim \mathcal{N}(y_i^2\boldsymbol{\theta}, \boldsymbol{I}) = \mathcal{N}(\boldsymbol{\theta}, \boldsymbol{I})$, and therefore $y_i\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\theta}, \boldsymbol{I})$. Thus, a natural estimator of $\boldsymbol{\theta}$ is

$$\widehat{\boldsymbol{\theta}} = \frac{1}{n}\sum_{i=1}^n y_i\boldsymbol{x}_i\,.$$

Now we will plug this estimate into the form of the Bayes classifier to obtain the classification rule

$$\widehat{f}(\boldsymbol{x}) = \begin{cases} +1 & \text{if } \boldsymbol{x}^T\widehat{\boldsymbol{\theta}} > 0 \\ -1 & \text{if } \boldsymbol{x}^T\widehat{\boldsymbol{\theta}} < 0 \end{cases}\,.$$

Due to the symmetry of the problem, the probability of error of this classifier is

$$\mathbb{P}(\widehat{f}(\boldsymbol{x}) \neq y) = \mathbb{P}(\boldsymbol{x}^T\widehat{\boldsymbol{\theta}} > 0|y=-1)\,,$$

The test statistic $\boldsymbol{x}^T\widehat{\boldsymbol{\theta}}$ doesn't have as simple a distribution as the optimal statistic $\boldsymbol{x}^T\boldsymbol{\theta}$ since both $\boldsymbol{x}$ and $\widehat{\boldsymbol{\theta}}$ are random. It is important, however, that these are independent of each other, since we assume that the new “test" point $\boldsymbol{x}$ is independent of the training data. Specifically, $\boldsymbol{x} \sim \mathcal{N}(-\boldsymbol{\theta}, \boldsymbol{I})$ and $\widehat{\boldsymbol{\theta}} \sim \mathcal{N}(\boldsymbol{\theta}, \boldsymbol{I}/n)$. Equivalently, $\boldsymbol{x} = -\boldsymbol{\theta} + \boldsymbol{e}_1$ and $\widehat{\boldsymbol{\theta}} = \boldsymbol{\theta} + \boldsymbol{e}_2$, where $\boldsymbol{e}_1 \sim \mathcal{N}(0, \boldsymbol{I})$ and $\boldsymbol{e}_2 \sim \mathcal{N}(0, \boldsymbol{I}/n)$ . So we can expand the statistic as follows

$$\begin{aligned}\boldsymbol{x}^T\widehat{\boldsymbol{\theta}} &= (-\boldsymbol{\theta} + \boldsymbol{e}_1)^T(\boldsymbol{\theta} + \boldsymbol{e}_2) \\ &= -\|\boldsymbol{\theta}\|^2 + (\boldsymbol{e}_1 - \boldsymbol{e}_2)^T\boldsymbol{\theta} + \boldsymbol{e}_1^T\boldsymbol{e}_2\end{aligned}$$

With this we can write the probability of error as follows.

$$\begin{aligned}\mathbb{P}(\widehat{f}(\boldsymbol{x}) \neq y) &= \mathbb{P}(-\|\boldsymbol{\theta}\|^2 + (\boldsymbol{e}_1 - \boldsymbol{e}_2)^T\boldsymbol{\theta} + \boldsymbol{e}_1^T\boldsymbol{e}_2 > 0) \\ &= \mathbb{P}\big((\boldsymbol{e}_1 - \boldsymbol{e}_2)^T\boldsymbol{\theta} + \boldsymbol{e}_1^T\boldsymbol{e}_2 > \|\boldsymbol{\theta}\|^2\big)\,.\end{aligned}$$

Now we can apply Markov's inequality to get the bound

$$\mathbb{P}(\widehat{f}(\boldsymbol{x}) \neq y) \leq \frac{\mathbb{E}\big[\big((\boldsymbol{e}_1 - \boldsymbol{e}_2)^T\boldsymbol{\theta} + \boldsymbol{e}_1^T\boldsymbol{e}_2\big)^2\big]}{\|\boldsymbol{\theta}\|^4}\,.$$

The expectation can be easily computed as follows.

$$\begin{aligned}\mathbb{E}\big[\big((\boldsymbol{e}_1 - \boldsymbol{e}_2)^T\boldsymbol{\theta} + \boldsymbol{e}_1^T\boldsymbol{e}_2\big)^2\big] &= \mathbb{E}[((\boldsymbol{e}_1 - \boldsymbol{e}_2)^T\boldsymbol{\theta} + \boldsymbol{e}_1^T\boldsymbol{e}_2)^T((\boldsymbol{e}_1 - \boldsymbol{e}_2)^T\boldsymbol{\theta} + \boldsymbol{e}_1^T\boldsymbol{e}_2)] \\ &= \mathbb{E}[\boldsymbol{\theta}^T(\boldsymbol{e}_1 - \boldsymbol{e}_2)(\boldsymbol{e}_1 - \boldsymbol{e}_2)^T\boldsymbol{\theta} + \boldsymbol{e}_2^T\boldsymbol{e}_1(\boldsymbol{e}_1 - \boldsymbol{e}_2)^T\boldsymbol{\theta} + \boldsymbol{\theta}^T(\boldsymbol{e}_1 - \boldsymbol{e}_2)\boldsymbol{e}_1^T\boldsymbol{e}_2 + \boldsymbol{e}_2^T\boldsymbol{e}_1\boldsymbol{e}_1^T\boldsymbol{e}_2]\,.\end{aligned}$$

Now if we first take the expectation with respect to $\boldsymbol{e}_1$ we get

$$\mathbb{E}[((\boldsymbol{e}_1 - \boldsymbol{e}_2)^T\boldsymbol{\theta} + \boldsymbol{e}_1^T\boldsymbol{e}_2)^2|\boldsymbol{e}_2] = \|\boldsymbol{\theta}\|^2 + \boldsymbol{\theta}^T\boldsymbol{e}_2\boldsymbol{e}_2^T\boldsymbol{\theta} + \boldsymbol{e}_2^T\boldsymbol{\theta} + \boldsymbol{\theta}^T\boldsymbol{e}_2 + \boldsymbol{e}_2^T\boldsymbol{e}_2\,,$$

since all cross terms vanish, i.e., $\mathbb{E}[\boldsymbol{e}_1^T\boldsymbol{e}_2|\boldsymbol{e}_2] = 0$ because $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ are independent and $\mathbb{E}[\boldsymbol{e}_1] = 0$. Taking the expectation with respect to $\boldsymbol{e}_2$ yields

$$\mathbb{E}[(\boldsymbol{e}_1 - \boldsymbol{e}_2)^T\boldsymbol{\theta} + \boldsymbol{e}_1^T\boldsymbol{e}_2)^2] \;\; = \;\; \|\boldsymbol{\theta}\|^2 + \frac{1}{n}\|\boldsymbol{\theta}\|^2 + d/n \,.$$

Thus, we have the bound

$$\mathbb{P}(\widehat{f}(\boldsymbol{x}) \neq y) \;\; \leq \;\; \frac{(1 + 1/n)\|\boldsymbol{\theta}\|^2 + d/n}{\|\boldsymbol{\theta}\|^4} \,.$$

Notice that if $n \gg d$, then the bound is essentially equal to the one we obtained for the Bayes classifier.

## 4.2. Comparison of MVN Plug-in Classifier and Histogram Classifier

Recall that Theorem 3 shows that the histogram classifier is able to achieve performance of the optimal classifier in any situation (arbitrary distributions) given enough training data. The MVN plug-in classifier is also able to achieve performance of the optimal classifier, but with the additional *strong* assumptions on the training data. So should why don't we prefer the histogram classifier, since it requires no assumptions on the data? If the class-conditional densities are MVN with (known) covariances, then the MVN plug-in classifier's error bound comes close to matching that of the optimal classifier when $n > d$ (i.e., the number of training data exceeds the dimension of the feature space). The histogram classifier would require far more data in the same situation. Minimally, the histogram partition would split each coordinate dimension in two. This means the histogram would have at least $2^d$ bins. The histogram classifier also requires at least one example in each bin, leading to the requirement that $n > 2^d$. In other words, the training set size must grow *exponentially* with dimension. This is the *curse of dimensionality*. Because of the curse, histogram classifiers are usually only used in low-dimensional problems. For high-dimensional problems, approaches based on strong modeling assumptions (even though they may not reflect the true distributions) can yield much better results.

## 4.3. Exercises

1. Are the empirical mean $\widehat{\boldsymbol{\mu}}$ and covariance $\widehat{\boldsymbol{\Sigma}}$ unbiased estimators? If not, are they *asymptotically* unbiased? Derive explicit formulas for the expectations.

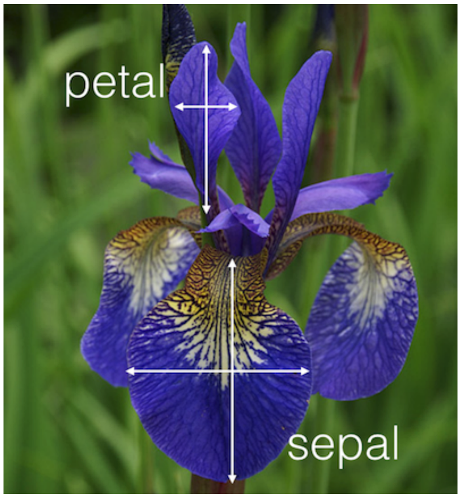
2. The analysis above assumes MVN class-conditional distributions. Show that the error bound for $\widehat{f}$ applies for any class-conditional distributions with the same means and covariance.

3. Derive another error bound for the case where the two means are different, say $\boldsymbol{\theta}_{+1} \neq \boldsymbol{\theta}_{-1}$.

4. Generalize things further by assuming the class-conditional distributions have the same covariance, but that it is a general covariance matrix $\boldsymbol{\Sigma}$ instead of the identity matrix $\boldsymbol{I}$.

5. In 1936 Ronald Fisher published a famous paper on classification titled "The use of multiple measurements in taxonomic problems." In the paper, Fisher study the problem of classifying iris flowers based on measurements of the sepal and petal widths and lengths, depicted in the image below.

   Fisher's dataset is available in Matlab (`fisheriris.mat`) and is widely available on the web (e.g., Wikipedia). The dataset consists of $50$ examples of three types of iris flowers. The sepal and petal measurements can be used to classify the examples into the three types of flowers. Approach the iris classification problem using a *generative modeling* approach. Assume that each class-conditional density is MVN.



(a) Fit the MVN models to the training data using estimates of means and covariances. Design a classifier based on these fitted models.

(b) Constrain the MVN models so that the decision boundaries between pairs of classes are linear.

## 5.1. Introduction

Consider a binary classification problem. For the special case of Gaussian class-conditional densities with equal covariances (i.e., $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}$) and equal prior probabilities (i.e., $\mathbb{P}(y = 1) = \mathbb{P}(y = 0)$), the log-likelihood ratio simplifies to

$$\widehat{y}(\boldsymbol{x}) \;=\; \begin{cases} 1 & \text{if } 2(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} \;\geq\; \boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 \\ \\ 0 & \text{otherwise} \end{cases}$$

If we let $\boldsymbol{w} = 2\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$ and $b = \boldsymbol{\mu}_0^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1$, then we see that this is just a linear classifier: if $\boldsymbol{w}^T\boldsymbol{x} + b \geq 0$, then the predicted label is $\widehat{y} = 1$. To get a sense of the difficulty this classification problem, let us consider the expected value of the test statistic $\boldsymbol{w}^T\boldsymbol{x}+b$. Consider a random example from the class 1 distribution: $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$. Then we have

$$\begin{aligned} \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|y=1)}[\boldsymbol{w}^T\boldsymbol{x} + b] &= 2(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 \\ &= \boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 - 2\boldsymbol{\mu}_0^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 \\ &= (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)\ , \end{aligned}$$

the Mahalanobis distance between the means. Similarly,

$$\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|y=0)}[\boldsymbol{w}^T\boldsymbol{x} + b] \;=\; -(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)\ .$$

So for any feature $x$ we can write the test statistic as

$$\boldsymbol{w}^T\boldsymbol{x} + b \;=\; \pm\,(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) \;+\; \zeta\ ,$$

where the sign on the first term depends on whether $\boldsymbol{x}$ comes from class 1 or 0 and $\zeta$ is a zero-mean random variable[5]. This suggests that larger values of $(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$ make it easier to classify new examples. In other words, the Mahalanobis distance between the means is a natural measure of the classification difficulty (i..e, larger distance implies easier classification). In this lecture, we generalize this idea to any class-conditional distributions.

## 5.2. Optimal Classifiers and Likelihood Functions

Suppose we have feature/label pairs $(\boldsymbol{x}, y) \sim P$. In this note, let us assume a binary classification setting with $y \in \{0, 1\}$. If $P$ is known, then the optimal classifier (i.e., the classifier that minimizes the probability of error) is easily derived. Denote the joint probability function by $p(\boldsymbol{x}, y)$. Recall that conditional probability of $y$ given $\boldsymbol{x}$ is given by

$$p(y|\boldsymbol{x}) = \frac{p(\boldsymbol{x}, y)}{p(\boldsymbol{x})}\ .$$

[5] In fact, since the test statistic is linear and $\boldsymbol{x}$ is MVN, the test statistic is also Gaussian and it is easy to check that $\zeta \sim \mathcal{N}(0, 4(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0))$.

In particular, the probability that $y = 1$ given $\boldsymbol{x}$ is $p(1|\boldsymbol{x})$. To clarify what this refers to, we sometimes write it more explicitly as $p(y = 1|\boldsymbol{x})$. This is a probability (i.e., a number in $[0, 1]$) that depends on the value of $\boldsymbol{x}$. If $p(y = 1|\boldsymbol{x}) \geq 1/2$, then the optimal classifier predicts the label to be $1$, and otherwise it predicts $0$. Since this conditional probability plays a crucial role in the optimal classification we give it a special notation:

$$\eta(\boldsymbol{x}) \ := \ p(y = 1|\boldsymbol{x}) \, .$$

The optimal classifier can be expressed as

$$f^*(\boldsymbol{x}) \ = \ \begin{cases} 1 & \eta(\boldsymbol{x}) \geq 1/2 \\ 0 & \eta(\boldsymbol{x}) < 1/2 \end{cases} .$$

Equivalently, since $p(y = 0|\boldsymbol{x}) = 1 - p(y = 1|\boldsymbol{x})$ we have

$$f^*(\boldsymbol{x}) \ = \ \begin{cases} 1 & \frac{\eta(\boldsymbol{x})}{1-\eta(\boldsymbol{x})} \geq 1 \\ 0 & \text{otherwise} \end{cases} .$$

We can express the optimal classifier in a different way, in terms of the class-conditional probability functions (or the conditional probabilities of $\boldsymbol{x}$ given $y$). Note that

$$p(y|\boldsymbol{x}) = \frac{p(\boldsymbol{x}, y)}{p(\boldsymbol{x})} \ = \ \frac{p(\boldsymbol{x}|y)p(y)}{p(\boldsymbol{x})} \, .$$

So $\eta(\boldsymbol{x}) = \frac{p(\boldsymbol{x}|y=1)p(y=1)}{p(\boldsymbol{x})}$ and $p(y = 0|\boldsymbol{x} = \boldsymbol{x}) = \frac{p(\boldsymbol{x}|y=0)p(y=0)}{p(\boldsymbol{x})}$ and thus the optimal classifier is

$$f^*(\boldsymbol{x}) \ = \ \begin{cases} 1 & \frac{p(\boldsymbol{x}|y=1)p(y=1)}{p(\boldsymbol{x}|y=0)p(y=0)} \geq 1 \\ & \\ 0 & \text{otherwise} \end{cases} ,$$

which we recognize as the likelihood ratio test (notice that the common denominator $p(\boldsymbol{x})$ cancels in the ratio). The value of $p(\boldsymbol{x}|y = j)$ is called the *likelihood* that $y = j$ given $\boldsymbol{x}$, and $\frac{p(\boldsymbol{x}|y=1)}{p(\boldsymbol{x}|y=0)}$ is called the *likelihood ratio*.

## 5.3. Kullback-Leibler Divergence: Intrinsic Difficulty of Classification

Let $p_j(\boldsymbol{x}) = p(\boldsymbol{x}|y = j)$ for $j = 0, 1$ and denote the log likelihood ratio by

$$\Lambda(\boldsymbol{x}) = \log \frac{p_1(\boldsymbol{x})}{p_0(\boldsymbol{x})} \, .$$

Consider a random test point $\boldsymbol{x} \sim q$, where $q$ may be $p_0$, $p_1$, or some other distribution. The log likelihood ratio classifier (assuming equal priors) is

$$\Lambda(\boldsymbol{x}) \underset{H_0}{\overset{H_1}{\gtrless}} 0 \, .$$

The resulting log likelihood ratio $\Lambda(\boldsymbol{x})$ is a random variable, and we can decompose it into its deterministic and stochastic components:

$$\Lambda(\boldsymbol{x}) \ = \ \mathbb{E}[\Lambda(\boldsymbol{x})] \ + \ (\Lambda(\boldsymbol{x}) - \mathbb{E}[\Lambda(\boldsymbol{x})]) \, .$$

The first term is a deterministic number and the second term is a zero-mean random variable. In other words, $\Lambda(\boldsymbol{x})$ has a distribution with mean $\mathbb{E}[\Lambda(\boldsymbol{x})]$. If the mean is far above or below the threshold 0, then we expect the classifier to perform well, and conversely if the mean is close to zero it will perform poorly. Note that

$$\begin{aligned}\mathbb{E}[\Lambda(\boldsymbol{x})] &= \int q(\boldsymbol{x}) \log \frac{p_1(\boldsymbol{x})}{p_0(\boldsymbol{x})}\, d\boldsymbol{x} \\ &= \int q(\boldsymbol{x}) \log \frac{q(\boldsymbol{x})}{p_0(\boldsymbol{x})}\, d\boldsymbol{x} - \int q(\boldsymbol{x}) \log \frac{q(\boldsymbol{x})}{p_1(\boldsymbol{x})}\, d\boldsymbol{x}\end{aligned}$$

These integrals are the Kullback-Leibler (KL) divergences from of $p_0$ and $p_1$ from $q$, denoted by $D(q\|p_0)$ and $D(q\|p_1)$. If $q = p_1$, then the mean is $\mathbb{E}[\Lambda(\boldsymbol{x})] = D(p_1 \,\|\, p_0) \geq 0$, since the KL divergence is non-negative. If $q = p_0$, then $\mathbb{E}[\Lambda(\boldsymbol{x})] = -D(p_0 \,\|\, p_1) \leq 0$. The larger the divergences, the easier the classification problem.

To illustrate further, consider an example $\boldsymbol{x}$ chosen randomly from class 1, that is $\boldsymbol{x} \sim p_1(\boldsymbol{x}) = p(\boldsymbol{x}|y = 1)$. The log likelihood ratio is then

$$\Lambda(\boldsymbol{x}) \;=\; D(p_1 \,\|\, p_0) \;+\; \zeta(\boldsymbol{x}) \;,$$

where the KL divergence $D(p_1 \,\|\, p_0) \geq 0$ and $\zeta(\boldsymbol{x}) := \Lambda(\boldsymbol{x}) - D(p_1 \,\|\, p_0)$ is a zero-mean random variable.

### 5.3.1. Gaussian Class-Conditional Distributions

If $\boldsymbol{x}|y = j \sim \mathcal{N}(\mu_j, \Sigma)$ for $j = 0, 1$ (note common covariance), then

$$D(p_0 \,\|\, p_1) = D(p_1 \,\|\, p_0) \;=\; \frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)\;,$$

whichi is proportional to the squared *Mahalanobis distance* between the means. To gain some insight, consider the scalar case: $x|y = j \sim \mathcal{N}(\mu_j, \sigma^2)$ for $j = 0, 1$, with $\mu_0 = -\mu_1 = \mu > 0$. In this case, $D(p_0 \,\|\, p_1) = D(p_1 \,\|\, p_0) = 2\mu^2/\sigma^2$. The log likelihood ratio is proportional to $\Lambda(x) = 2\mu x$. Consider a test point $x \sim p_1$. Then $\Lambda(x) \sim \mathcal{N}(2\mu^2, 4\mu^2\sigma^2)$. Note that the KL divergence is equal to *signal-to-noise ratio*, i.e., the ratio of the squared mean of $x$ over its variance.

### 5.3.2. Separable Classes

The classes are separable if and only if the class-conditional densities do not overlap (i.e., the supports are disjoint subsets of the feature space). Consider the KL divergences:

$$\begin{aligned}D(p_1 \,\|\, p_0) &= \int p_1(\boldsymbol{x}) \log \frac{p_1(\boldsymbol{x})}{p_0(\boldsymbol{x})}\, d\boldsymbol{x} \\ D(p_0 \,\|\, p_1) &= \int p_0(\boldsymbol{x}) \log \frac{p_0(\boldsymbol{x})}{p_1(\boldsymbol{x})}\, d\boldsymbol{x}\end{aligned}$$

Consider a point $\boldsymbol{x}$ in the feature space where $p_1(\boldsymbol{x}) > 0$ and $p_0(\boldsymbol{x}) = 0$. The value of the integrand in $D(p_1 \,\|\, p_0)$ is infinite at that point, and therefore so is $D(p_1 \,\|\, p_0)$. This agrees with common sense: if the class-conditional distributions are non-overlapping, then perfect (error-free) classification is possible. However, the optimal classification rule may be nonlinear and can be very difficult to learn from training data, so KL divergence does not tell the whole story. The margin between separable classes (mininum distance between the supports) is a more meaningful indicator of the difficulty of *learning* a classifier. Also notice that the KL divergence may be infinite even if the supports of the two class-conditional densities have a large overlap (i.e., even if the two classes are difficult to distinguish); if one has non-zero probability mass outside the support of the other, then the KL divergence will be infinite. All this shows that the KL divergence may not be an appropriate measure of classification difficulty if the two class-conditional distributions do not have the same support.

### 5.3.3. Statistical Hypothesis Testing

The KL divergence between two distributions $p_1$ and $p_0$ may be infinite even if they partially overlap. This happens whenever the integrand $p_1(\boldsymbol{x}) \log \frac{p_1(\boldsymbol{x})}{p_0(\boldsymbol{x})}$ is infinite on a non-trivial subset of the feature space. Clearly in such cases the classes are non-separable and perfect classification from a single sample is impossible. However, the infinite value of the KL divergence still tells us something important. Suppose we are given a set of i.i.d. examples $\{\boldsymbol{x}_i\}_{i=1}^n$, drawn from either $p_1$ or $p_0$. The statistical testing problem is to decide which distribution *all* the examples come from. In this case, the log-likelihood ratio is

$$\Lambda(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \sum_{i=1}^{n} \log \frac{p_1(\boldsymbol{x}_i)}{p_0(\boldsymbol{x}_i)} \ ,$$

which is proportional to the average $\frac{1}{n} \sum_{i=1}^{n} \log \frac{p_1(\boldsymbol{x}_i)}{p_0(\boldsymbol{x}_i)}$. The strong law of large numbers implies that if the examples come from $p_1$, then

$$\frac{1}{n} \sum_{i=1}^{n} \log \frac{p_1(\boldsymbol{x}_i)}{p_0(\boldsymbol{x}_i)} \to D(p_1 \,\|\, p_0) \ , \text{ as } n \to \infty \ .$$

So as $n$ grows it is possible to perfectly decide which distribution generated the "batch" of examples. More generally, the size of $D(p_1 \,\|\, p_0)$ and $D(p_0 \,\|\, p_1)$ determines how many examples are sufficient to confidently decide which distribution they came from.

### 5.3.4. Nonnegativity of KL Divergence

The key property in question is that $D(q||p) \geq 0$, with equality if and only if $q = p$. To prove this, we will use Jensen's Inequality. Recall, a function is *convex* if $\forall \ \lambda \in [0, 1]$

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda) f(y)$$

The left hand side of this inequality is the function value at some point between $x$ and $y$, and the right hand side is the value of a straight line connecting the points $(x, f(x))$ and $(y, f(y))$. In other words, for a convex function the function value between two points is always lower than the straight line between those points.

**Jensen's Inequality:** If a function $f(x)$ is convex, then

$$\mathbb{E}[f(x)] \geq f(\mathbb{E}[x])$$

Now if we rearrange the KL divergence formula,

$$\begin{aligned} D(q||p) &= \int q(x) \log \frac{q(x)}{p(x)} dx \\ &= \mathbb{E}_q \left[ \log \frac{q(x)}{p(x)} \right] \\ &= -\mathbb{E}_q \left[ \log \frac{p(x)}{q(x)} \right] \end{aligned}$$

we can use Jensen's inequality, since $-\log z$ is a convex function.

$$\begin{aligned}
&\geq -\log\left(\mathbb{E}_q\left[\frac{p(x)}{q(x)}\right]\right)\\
&= -\log\left(\int q(x)\frac{p(x)}{q(x)}dx\right)\\
&= -\log\left(\int p(x)dx\right)\\
&= -\log(1)\\
&= 0
\end{aligned}$$

Therefore $D(q||p) \geq 0$.

### 5.3.5. Mutual Information

KL divergence also plays a key role in quantifying the information that one random variable conveys about another. In the context of machine learning, one may be interested in quantifying how informative a feature $\boldsymbol{x}$ is for predicting a label $y$. If the feature and label are statistically independent, then the feature is useless. With this in mind, consider $D(p(x,y)\,\|\,p(x)p(y))$ , the KL divergence between the joint distribution $p(x,y)$ and the distribution $p(x)p(y)$. If $\boldsymbol{x}$ and $y$ are independent, then $p(x,y) = p(x)p(y)$ and the KL divergence is $0$. If the feature $\boldsymbol{x}$ is highly predictive of the label, then the joint distribution is very different than the factorized form and the KL divergence is large. This particular KL divergence has a special name: the *mutual information* between $\boldsymbol{x}$ and $y$. This quantify is often used for feature engineering and selection. Consult [10] and [4, Chapter 1] for more on information-theoretic concepts.

## 5.4. Exercises

1. Consider the following two-dimensional class-conditional densities:

$$\begin{aligned}
p(\boldsymbol{x}|y=+1) &= a\mathbb{1}_{\{\boldsymbol{x}\in[0,1]\times[0,1]\}} + 4(1-a)\mathbb{1}_{\{\boldsymbol{x}\in[0,0.5]\times[0,0.5]\}}\\
p(\boldsymbol{x}|y=-1) &= a\mathbb{1}_{\{\boldsymbol{x}\in[0,1]\times[0,1]\}} + 4(1-a)\mathbb{1}_{\{\boldsymbol{x}\in[0.5,1]\times[0.5,1]\}}\ ,
\end{aligned}$$

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

   (a) Assuming equal prior probabilities for the two classes, express the optimal classification rule and error rate in terms of $a$.

   (b) Express the KL divergences

   $$D(p(\boldsymbol{x}|y=+1)\,\|\,p(\boldsymbol{x}|y=-1)) \text{ and } D(p(\boldsymbol{x}|y=-1)\,\|\,p(\boldsymbol{x}|y=+1))$$

   as functions of $a$.

   (c) Now suppose that do not know the class conditional densities, but you are given a set of training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \overset{iid}{\sim} p(\boldsymbol{x}, y)$ and the features are known to lie within the unit square. How would you design a classifier using these data? Can you bound its probability of error and compare it to the optimal error rate?

2. Consider a binary classification problem, and assume the class conditional densities have the form $x|y = 0 \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{I})$ and $x|y = 1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{I})$ and equal prior probabilities. Suppose you estimate the means from a training dataset, denoted $\widehat{\boldsymbol{\mu}}_0$ and $\widehat{\boldsymbol{\mu}}_1$. Moreover, assume that you know that $\|\boldsymbol{\mu}_j - \widehat{\boldsymbol{\mu}}_j\|_2 \leq \epsilon$ for $j = 0, 1$ and some $\epsilon > 0$. Plug these estimates into the log likelihood ratio. The resulting classifier may be suboptimal due to the errors in the estimates. Quantify the suboptimality by computing the KL divergences of this classifier and comparing them to the KL divergences of the optimal classifier (based on the true means).

3. Consider a binary classification problem with equal prior probabilities and class-conditional densities $x|y = 0 \sim$ uniform$[0, 1]$ and $x|y = 1 \sim$ uniform$[t, 1 + t]$ for some $t \in \mathbb{R}$.

   (a) Compute the mutual information between the feature $x$ and the label $y$ as a function of $t$. Interpret the result for $t = 0$, $|t| \geq 1$, and $t \in (0, 1)$.

   (b) Compute the minimum probability of misclassification as a function of $t$.

# Lecture 6: Maximum Likelihood Estimation

Maximum likelihood estimation is one of the most popular approaches in statistical inference and machine learning. It is usually viewed as a methodology for estimating the parameters of a probabilistic model, but in fact the core principle of maximum likelihood is density estimation.

## 6.1. The Maximum Likelihood Estimator

Consider a family of probability distributions indexed by a parameter $\theta$. The parameter may be a scalar or multidimensional. For example, the family of multivariate Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is indexed by the mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, so in this case $\theta = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$. In general, we will denote a family of density functions by $p(x|\theta)$, $\theta \in \Theta$, where $\Theta$ denotes the set of all possible values the parameter can take. Given data $\boldsymbol{x}$, the Maximum Likelihood Estimate (MLE) is

$$\widehat{\theta} = \arg\max_{\theta \in \Theta} p(\boldsymbol{x}|\theta)$$

where $p(\boldsymbol{x}|\theta)$ as a function of $\boldsymbol{x}$ with the parameter $\theta$ fixed is the probability density function or mass function. Viewing $p(\boldsymbol{x}|\theta)$ as a function of $\theta$ with the data $x$ fixed is called the "likelihood function." Sometimes will denote $p(\boldsymbol{x}|\theta)$ by $p_\theta(\boldsymbol{x})$.

This is a generalization of the hypothesis testing problem considered in the previous lecture. Suppose that $\theta \in \{0, 1\}$, i.e., $\theta$ is a binary-valued parameter which could indicate which of to two classes $y = 0$ or $y = 1$ generated the data $\boldsymbol{x}$. In this case, maximum likelihood estimation of $\theta$ is equivalent to the log-likelihood ratio test. Denote the log-likelihood ratio by $\Lambda(\boldsymbol{x}) = \log \frac{p_1(\boldsymbol{x})}{p_0(\boldsymbol{x})}$. Then the Maximum Likelihood Estimator (MLE) of $\theta$ is

$$\widehat{\theta} \ = \left\{ \begin{array}{ll} 1 & \text{if } \Lambda(\boldsymbol{x}) > 0 \\ 0 & \text{if } \Lambda(\boldsymbol{x}) < 0 \end{array} \right.$$

## 6.2. ML Estimation and Density Estimation

ML Estimation is equivalent to density estimation. Assume

$$\boldsymbol{x}_i \overset{\text{iid}}{\sim} q, \quad i = 1, \cdots, n, \quad \text{where } q \text{ is an unknown probability density}$$

and suppose we wish to model these data using with a parametrized family of distributions $\{p_\theta\}_{\theta \in \Theta}$. The ML Estimation is equivalent to finding the density in $\{p_\theta\}_{\theta \in \Theta}$ that best fits the data. i.e., the generative model with the highest density/probability value on the points $\boldsymbol{x} = \{\boldsymbol{x}_i\}$. The *true* distribution of the data points is $\Pi_{i=1}^n q(\boldsymbol{x}_i)$, because the $\boldsymbol{x}_i$ are i.i.d. The likelihood function for $\theta$ is $\Pi_{i=1}^n p_\theta(\boldsymbol{x}_i)$. The true generating density $q$ may not be a member of the parametric family under consideration. The MLE is the solution to

$$\max_{\theta \in \Theta} \Pi_{i=1}^n p_\theta(\boldsymbol{x}_i) \ \text{ or equivalently } \ \max_{\theta \in \Theta} \sum_{i=1}^n \log p_\theta(\boldsymbol{x}_i) \ .$$

The argument of the latter optimization is called the log-likelihood function of $\theta$. The logarithm is often applied because it simplifies the optimization in cases where the probability models involve exponential functions (e.g., the Gaussian). We can also express the MLE as the solution of the minimization

$$\min_{\theta\in\Theta} - \sum_{i=1}^{n} \log p_\theta(\boldsymbol{x}_i)\ .$$

## 6.3. Examples of MLE

**Example 2. Structured Mean.** *Let $\boldsymbol{B} \in \mathbb{R}^{n\times k}$ be a known matrix and assume the $n \times n$ covariance matrix $\boldsymbol{\Sigma}$ is known. Consider the model*

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{B\theta})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{B\theta})\}\ ,\ \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{\theta} \in \mathbb{R}^k$$

The MLE $\widehat{\boldsymbol{\theta}}$ is given by

$$\begin{aligned} \widehat{\boldsymbol{\theta}} &= \arg\min_{\boldsymbol{\theta}} - \log p(\boldsymbol{x}|\boldsymbol{\theta}) \\ &= \arg\min_{\boldsymbol{\theta}} (\boldsymbol{x} - \boldsymbol{B\theta})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{B\theta}) \\ &= (\boldsymbol{B}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} \end{aligned}$$

**Example 3. Poisson mean estimation.** *Let $x_i \overset{\text{iid}}{\sim}$ Poisson$(\lambda)$, $i = 1, \ldots, n$. Then the negative log-likelihood is*

$$\sum_{i=1}^{n} - \log\left(e^{-\lambda}\frac{\lambda^{x_i}}{x_i!}\right) \ = \ \sum_{i=1}^{n}\Big(\lambda - x_i\log(\lambda) + \log(x_i!)\Big)$$

*The derivative of each term with respect to $\lambda$ is $1 - x_i/\lambda$, so if we set the derivative equal to zero we find the minimizer is $\widehat{\lambda} = \frac{1}{n}\sum_{i=1}^{n} x_i$.*

**Example 4. Linear Regression.** *Suppose $y_i = \boldsymbol{w}^T\boldsymbol{x}_i + \epsilon_i$, for $i = 1, \ldots, n$. Here $\{\boldsymbol{x}_i, y_i\}_{i=1}^{n}$ are observed, with $\boldsymbol{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, and the $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ are unobserved noises.*

$$p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}) \propto \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \boldsymbol{w}^T\boldsymbol{x}_i)^2\right) \ = \ \exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{y} - \boldsymbol{Xw}\|^2\right)$$

*where $\boldsymbol{y}$ is a column vector of the $\{y_i\}$ and $\boldsymbol{X}$ is an $n \times d$ matrix whose $i$th row is $\boldsymbol{x}_i$.*

The maximum likelihood estimate $\widehat{\boldsymbol{w}}$ is given by,

$$\begin{aligned} \widehat{\boldsymbol{w}} &= \arg\min_{\boldsymbol{w}} - \log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}) \\ &= \arg\min_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{Xw}\|^2 \\ &= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}\ ,\ \text{assuming } \boldsymbol{X} \text{ has full rank.} \end{aligned}$$

## 6.4. MLE and KL Divergence

We can view the negative log-likelihood function as sum of "loss" functions of the form

$$\ell_i(q, p_\theta) \ := \ -\log p_\theta(\boldsymbol{x}_i)$$

which measure the loss incurred when using $p_\theta$ to model the distribution of $\boldsymbol{x}_i$, which is actually distributed according to $q$. So we may view this as an empirical measure of how well $p_\theta$ fits $q$ at the point $\boldsymbol{x}_i$. The notation here makes the dependence of the loss on both $q$ and the choice of $p_\theta$ explicit, but it is commmon to simply write $\ell_i(\theta)$, since we view the loss as a function of the parameter $\theta$. The expected value of a loss is called the "risk". The risk associated with negative log-likelihood loss function is

$$\begin{aligned} R(q, p_\theta) &= \mathbb{E}[\ell_i(q, p_\theta)] \\ &= \mathbb{E}_{\boldsymbol{x}_i \sim q}[-\log p_\theta(\boldsymbol{x}_i)] \\ &= -\int q(\boldsymbol{x}) \log p_\theta(\boldsymbol{x})\, d\boldsymbol{x} \end{aligned}$$

It is also common just to write $R(\theta)$ to denote the risk, but we will be more explicit here. We can compare the value of the risk of $p_\theta$ with the that of the true distribution $q$. The *excess risk*

$$R(q, p_\theta) - R(q, q)$$

quantifies how much larger the risk is when we use $p_\theta$ instead of $q$. Note that

$$\begin{aligned} R(q, p_\theta) - R(q, q) &= \mathbb{E}[\log q(\boldsymbol{x}) - \log p_\theta(\boldsymbol{x})] \\ &= \mathbb{E}\left[\log \frac{q(\boldsymbol{x})}{p_\theta(\boldsymbol{x})}\right] \\ &= \int q(\boldsymbol{x}) \log \frac{q(\boldsymbol{x})}{p_\theta(\boldsymbol{x})} dx \\ &=: D(q \,\|\, p_\theta) \\ &= \geq 0 \end{aligned}$$

with equality if $p_\theta = q$. Recall that $D(q \,\|\, p_\theta)$ is the *Kullback-Leibler* divergence of $p_\theta$ from $q$. This shows that $q$ minimizes the risk. We can consider

$$\theta^* = \arg\min_\theta D(q \| p_\theta)$$

to be the *optimal* value of $\theta$. The density $p_{\theta^*}$ is the member of the parametric class that is closest in KL divergence to the data-generating distribution $q$. If we have multiple iid observations $\boldsymbol{x}_i \overset{\text{iid}}{\sim} q,\ i = 1, \cdots, n$, then the MLE is

$$\widehat{\theta}_n = \arg\min_\theta -\sum_{i=1}^n \log p_\theta(\boldsymbol{x}_i)$$

By strong law of large numbers, for any $\theta \in \Theta$

$$\frac{1}{n} \sum_{i=1}^n \log \frac{q(\boldsymbol{x}_i)}{p_\theta(\boldsymbol{x}_i)} \xrightarrow{\text{a.s.}} D(q \| p_\theta)\ .$$

So intuitively $\widehat{\theta}_n \to \theta^*$ as $n \to \infty$. To conclude, the analysis above shows that ML Estimation is essentially trying to find a parametric probability model $p_\theta$ that best fits the true data distribution in the sense of KL divergence.

## 6.5. Exercises

1. Let $x_1, \ldots, x_n \overset{iid}{\sim}$ Uniform$(0, \theta)$.

**a.** Find the MLE $\widehat{\theta}_n$.

**b.** Give a mathematical expression for the exact distribution of $\widehat{\theta}_n$.

**c.** Show that the MLE is consistent in MSE; i.e., $\mathbb{E}[|\widehat{\theta}_n - \theta|^2] \to 0$ as $n \to \infty$. Hint: Use the result in part (b) to compute the mean and variance of $\widehat{\theta}_n$.

2. Suppose we are monitoring credit card payments for a population of $N$ people, and model the number of days ($\tau$) that will pass before each person defaults on their payments as independent, identically exponentially distributed random variables with parameter $\theta > 0$. That is, for $i = 1, \ldots, N$, $\tau_i \sim \exp(\theta)$, so that $p(\tau_i|\theta) = \theta e^{-\tau_i \theta}$. Find the MLE of $\theta$.

3. *Robust estimation.* There are different approaches to finding the "center" of a dataset. Suppose $x_1, \ldots, x_n$ are scalars. The sample mean is one statistic that summarizes the central location of the data. Of course, if one assumes the data are i.i.d. realizations of a $\mathcal{N}(\theta, \sigma^2)$ random variable, then the sample mean is the MLE of $\theta$. The sample mean is the minimizer of

$$e_2(\theta) = \sum_{i=1}^{n} (x_i - \theta)^2$$

Another alternative is to minimize the sum of absolute errors

$$e_1(\theta) = \sum_{i=1}^{n} |x_i - \theta|$$

The minimizer of $e_1$ has some nice robustness properties, which you will investigate in this problem. It can also be viewed as the MLE of $\theta$ if the data are assumed to be i.i.d. realizations from a Laplacian distribution of the form $p(x; \theta) = \frac{1}{2b} e^{-|x-\theta|/b}$, where $b > 0$.

**a.** Find closed-form expressions for $\widehat{\theta}_i = \arg\min_\theta e_i(\theta)$, $i = 1, 2$.

**b.** Suppose that $n = 3$ and $(x_1, x_2, x_3) = (1.1, 0.9, 1.0)$. What are $\widehat{\theta}_i$, $i = 1, 2$ in this case?

**c.** Suppose that the value of $x_3$ was misrecorded as $x_3 = 100.0$ instead of $1.0$. What are $\widehat{\theta}_i$, $i = 1, 2$ in this case? Which estimator is more robust to this sort of error?

4. Suppose that Twitter assigns each new user an integer $k + 1$, where $k$ is the number of users who registered before. At the moment there are $N$ Twitter accounts, but Twitter keeps this number secret. However, the integer assigned to each user can be found in their profile and you are going to exploit this in order to estimate $N$. Sample $n$ users uniformly at random from the set of all users. Let $x_1, \ldots, x_n$ be the integers assigned to these users. These integers are distributed uniformly at random from the set $\{1, \ldots, N\}$.

**a.** Consider the estimator $\widehat{N}_1 := \frac{2}{n}(\sum_{i=1}^{n} x_i) - 1$. Compute the mean, variance, and MSE $\mathbb{E}[|N - \widehat{N}_1|^2]$ of $\widehat{N}_1$? Is this estimator unbiased?

**b.** Let $\widehat{N}_2$ denote the MLE of $N$. Derive an expression for $\widehat{N}_2$. Compute the mean, variance, and MSE $\mathbb{E}[|N - \widehat{N}_2|^2]$ of $\widehat{N}_2$? Hint: Recall that if $Y$ is a non-negative integer-valued random variable, then $\mathbb{E}[Y] = \sum_{i=1}^{\infty} \mathbb{P}(Y \geq i)$.

**c.** Which estimator would you use and why? Can you propose an even better estimator?

# Lecture 7: Sufficient Statistics

Consider a random variable $x$ whose distribution $p$ is parametrized by $\theta \in \Theta$ where $\theta$ is a scalar or a vector. Denote this distribution as $p(x|\theta)$. In many machine learning applications we need to make some decision about $\theta$ from observations of $x$, where the density can be one of many in a family of distributions, $\{p(x|\theta)\}_{\theta\in\Theta}$, indexed by different choices of the parameter $\theta$. More generally, suppose we make $n$ independent observations $x_1, x_2, \dots, x_n$ where $p(x_1 \dots x_n|\theta) = \prod_{i=1}^n p(x_i|\theta)$. These observations can be used to infer or estimate the correct value for $\theta$. This problem can be posed as follows. Let $\boldsymbol{x} = [x_1, x_2, \dots, x_n]$ be a vector containing the $n$ observations.

**Question:** Is there a lower dimensional function of $\boldsymbol{x}$, say $t(\boldsymbol{x})$, that alone carries all the relevant information about $\theta$? For example, if $\theta$ is a scalar parameter, then one might suppose that all relevant information in the observations can be summarized in a scalar statistic.

**Goal:** Given a family of distributions $\{p(x|\theta)\}_{\theta\in\Theta}$ and one or more observations from a particular distribution $p(x|\theta^*)$ in this family, find a data compression strategy that preserves all information pertaining to $\theta^*$. The function identified by such strategy is called a *sufficient statistic*.

## 7.1. Sufficient Statistics

Maximum likelihood estimation is based exclusively on the *shape* likelihood function $p(x|\theta)$; the goal is to find a maximum point. Any processing operations that preserve the shape will not affect the ML estimation process. This is key to the idea of *sufficient statistics*.

**Example 5** (Bernoulli Random Variables)**.** *Suppose $x$ is a $0/1$ - valued variable with $\mathbb{P}(x = 1) = \theta$ and $\mathbb{P}(x = 0) = 1 - \theta$. That is $x \sim p(x|\theta) = \theta^x(1-\theta)^{1-x}, (x \in [0, 1])$. We observe $n$ independent realizations $x_1, \dots, x_n$ with $p(x_1, \dots, x_n|\theta) = \prod_{i=1}^n \theta^{x_i}(1-\theta)^{1-x_i} = \theta^k(1-\theta)^{n-k}$; $k = \sum_{i=1}^n x_i$ (number of 1's).*
*Note that $x = \sum_{i=1}^n x_i$ is a random variable with values in $\{0, 1 \dots, n\}$*

$$p(k|\theta) = \binom{n}{k}\theta^k(1-\theta)^{n-k}, \text{ a binomial distribution with } \binom{n}{k} = \frac{n!}{(n-k)!k!}$$

*The joint probability mass function of $(x_1, \dots, x_n)$ and $k$ is*

$$p(x_1, \dots, x_n, k|\theta) = \begin{cases} p(x_1, \dots, x_n|\theta); & \text{if } k = \sum x_i \\ 0; & \text{otherwise} \end{cases}$$

*Therefore*

$$p(x_1, \dots, x_n|k, \theta) \; = \; \frac{\theta^k(1-\theta)^{n-k}}{\binom{n}{k}\theta^k(1-\theta)^{n-k}} = \frac{1}{\binom{n}{k}}$$

*This shows that the conditional probability of $x_1, \dots, x_n$ given $k = \sum x_i$ is uniformly distributed over the $\binom{n}{k}$ sequences that have exactly $k$ 1's. In other words, the condition distribution of $x_1, \dots, x_n$ given $k$ is independent of $\theta$. So $k$ carries all relevant infomation about $\theta$!*

**Note:** $k = \sum x_i$ compresses $\{0, 1\}^n$ (n bits) to $\{0, \dots, n\}$ ($\log n$ bits).

**Definition 3.** *Let $x$ denote a random variable whose distribution is parameterized by $\theta \in \Theta$. Let $p(x|\theta)$ denote the density of mass function. A statistic $t(x)$ is* sufficient *for $\theta$ if the distribution of $x$ given $t(x)$ is independent of $\theta$; i.e., $p(x|t, \theta) = p(x|t)$*

**Theorem 4** (Fisher-Neyman Factorization). *Let $x$ be a random variable with density $p(x|\theta)$ for some $\theta \in \Theta$. The statistic $t(x)$ is sufficient for $\theta$ if and only if the density can be factorized into a function $a(x)$ and a function $b(t,\theta)$, a function of $\theta$ but only depending on $x$ through the $t(x)$; i.e.,*

$$p(x|\theta) = a(x)b(t,\theta)$$

*Proof.* (if/sufficiency) Assume $p(x|\theta) = a(x)b(t|\theta)$

$$p(t|\theta) = \int_{x:t(x)=t} p(x|\theta)dx = \left(\int_{x:t(x)=t} a(x)dx\right)b(t,\theta)$$

This shows that

$$p(x|t,\theta) = \frac{p(x,t|\theta)}{p(t|\theta)} = \frac{p(x|\theta)}{p(t|\theta)} = \frac{a(x)}{\int_{x:t(x)=t} a(x)dx} \text{ independent of } \theta$$

$$\Rightarrow t(x) \text{ is a sufficient statistic}$$

(only if/necessity) If $p(x|t,\theta) = p(x|t)$ independent of $\theta$ then $p(x|\theta) = p(x,t|\theta) = p(x|t,\theta)p(t|\theta) = \underbrace{p(x|t)}_{a(x)}\underbrace{p(t|\theta)}_{b(t,\theta)}$ □

**Example 6** (Bernoulli Random Variables). $x_1, \ldots, x_n \overset{\text{iid}}{\sim}$ *Bernoulli*$(\theta)$ *and let* $x = (x_1, \ldots, x_n)$*. Then* $p(x|\theta) = \theta^k(1-\theta)^{n-k} = \underbrace{\frac{1}{\binom{n}{k}}}_{a(x)}\underbrace{\binom{n}{k}\theta^k(1-\theta)^{n-k}}_{b(k,\theta)} \Rightarrow k$ *is sufficient for* $\theta$.

**Example 7** (Poisson). *Let $\lambda$ be an average number of packets/sec sent over a network. Let $x$ be a random variable representing number of packets seen in $1$ second. Assume $p(x|\lambda) = e^{-\lambda}\frac{\lambda^x}{x!}$.*
*Given $x_1, \ldots, x_n \overset{\text{iid}}{\sim} p(x|\lambda)$ , we have*

$$p(x_1, \ldots, x_n|\lambda) = \prod_{i=1}^n e^{-\lambda}\frac{\lambda^{x_i}}{x_i!} = \underbrace{\prod_{i=1}^n \frac{1}{x_i!}}_{a(x)} \underbrace{e^{-n\lambda}\lambda^{\sum x_i}}_{b(\sum x_i,\lambda)} .$$

*So $\sum_{i=1}^n x_i$ is a sufficient statistic for $\lambda$.*

**Example 8** (Gaussian). $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ *is $d$-dimensional.*
$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \overset{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ *and let $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denote all the parameters of the model.*

$$\begin{aligned} p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n|\boldsymbol{\theta}) = \prod_{i=1}^n p(\boldsymbol{x}_i;\boldsymbol{\theta}) &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi^d|\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\boldsymbol{x}_i-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu})} \\ &= 2\pi^{-nd/2}|\boldsymbol{\Sigma}|^{-n/2}e^{-\frac{1}{2}\sum_{i=1}^n(\boldsymbol{x}_i-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu})} \end{aligned}$$

*Define sample mean*

$$\widehat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i$$

*and sample covariance*

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})^T$$

$$\begin{aligned}
\exp(-\frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{x}_i - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu})) &= \exp(-\frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}} + \widehat{\boldsymbol{\mu}} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}} + \widehat{\boldsymbol{\mu}} - \boldsymbol{\mu})) \\
&= \exp(-\frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}) - \sum_{i=1}^{n}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})^T\boldsymbol{\Sigma}^{-1}(\widehat{\boldsymbol{\mu}} - \boldsymbol{\mu}) - \frac{1}{2}\sum_{i=1}^{n}(\widehat{\boldsymbol{\mu}} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\widehat{\boldsymbol{\mu}} - \boldsymbol{\mu})) \\
&= \exp(-\frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}))\exp(-\frac{1}{2}\sum_{i=1}^{n}(\widehat{\boldsymbol{\mu}} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\widehat{\boldsymbol{\mu}} - \boldsymbol{\mu})) \\
&= \exp(-\frac{1}{2}tr(\boldsymbol{\Sigma}^{-1}\sum_{i=1}^{n}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})^T))\exp(-\frac{1}{2}\sum_{i=1}^{n}(\widehat{\boldsymbol{\mu}} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\widehat{\boldsymbol{\mu}} - \boldsymbol{\mu})) \\
&= \exp(-\frac{1}{2}tr(\boldsymbol{\Sigma}^{-1}(n\widehat{\boldsymbol{\Sigma}})))\exp(-\frac{1}{2}\sum_{i=1}^{n}(\widehat{\boldsymbol{\mu}} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\widehat{\boldsymbol{\mu}} - \boldsymbol{\mu}))
\end{aligned}$$

*Note that the second term on the second line is zero because $\frac{1}{n}\sum_i \boldsymbol{x}_i = \widehat{\boldsymbol{\mu}}$. For any matrix $\boldsymbol{B}$, $tr(\boldsymbol{B})$ is the sum of the diagonal elements. On the fourth line above we use the* trace *property, $tr(\boldsymbol{AB}) = tr(\boldsymbol{BA})$.*

$$p(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n|\theta) = \underbrace{2\pi^{-nd/2}|\boldsymbol{\Sigma}|^{-n/2}\exp(-\frac{1}{2}\sum_{i=1}^{n}(\widehat{\boldsymbol{\mu}} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\widehat{\boldsymbol{\mu}} - \boldsymbol{\mu}))\exp(-\frac{1}{2}tr(\boldsymbol{\Sigma}^{-1}n\widehat{\boldsymbol{\Sigma}}))}_{b(\widehat{\boldsymbol{\mu}},\widehat{\boldsymbol{\Sigma}},\theta)}\ \underbrace{1}_{a(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n)}$$

## 7.2. Minimal Sufficient Statistic

**Definition 4.** *A sufficient statistic is* minimal *if the dimension of $t(x)$ cannot be further reduced and still be sufficient.*

**Example 9.** *Let $x_1,\ldots,x_n \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$*

$$\begin{aligned}
\boldsymbol{u}(x_1,\ldots,x_n) = [x_1 + x_2,\ldots,x_{n-1} + x_n]^T \ ,\ &\ \boldsymbol{u} \textit{ is a } \tfrac{n}{2}\textit{-dimensional statistic} \\
t(x_1,\ldots,x_n) = \sum_{i=1}^{n} x_i \ &\ \textit{is a 1-dimensional statistic}
\end{aligned}$$

*$t$ is sufficient, and since $t = \sum_{i=1}^{n/2} u_i$ it follows that $\boldsymbol{u}$ is also a sufficient statistic (but not minimal).*

## 7.3. Rao-Blackwell Thereom

Sufficient statistics arise naturally in ML estimation, but there are many other criteria for estimation. The Rao-Blackwell theorem provides additional support for the use of sufficient statistics.

**Theorem 5.** *[5] Assume $x \sim p(x|\theta)$, $\theta \in \mathbb{R}$, and $t(x)$ is a sufficient statistic for $\theta$. Let $f(x)$ be an estimator of $\theta$ and consider the mean square error $\mathbb{E}[(f(x)-\theta)^2]$. Define $g(t(x)) = \mathbb{E}[f(x)|t(x)]$.*

*Then $\mathbb{E}[(g(t(x))-\theta)^2] \leq \mathbb{E}[(f(x)-\theta)^2]$, with equality if and only if $f(x) = g(t(x))$ with probability 1; i.e., if the function $f$ is equal to $g$ composed with $t$.*

*Proof.* First note that because $t(x)$ is a sufficient statistic for $\theta$, it follows that $g(t(x)) = \mathbb{E}[f(x)|t(x)]$ does not depend on $\theta$, and so it too is a valid estimator (i.e., if $t(x)$ were not sufficient, then $g(t(x))$ might be a function of $t(x)$ and $\theta$ and therefore not computable from the data alone).

Next recall the following basic facts about conditional expectation. Suppose $x$ and $y$ are random variables. Then

$$\mathbb{E}[f(x)|y] = \int f(x)p(x|y)dx$$

where $p(x|t)$ is conditional density of $x$ given $t(x)$. Furthermore, for any random variables $x$ and $y$

$$\begin{aligned}\mathbb{E}[\mathbb{E}[f(x)|y)]] &= \int \mathbb{E}[f(x)|y]p(y)dy \\ &= \int \left( \int f(x)p(x|y)dx \right) p(y)dy \\ &= \int f(x) \left( \int p(x|y)p(y)dy \right) dx \\ &= \int f(x)p(x)dx = \mathbb{E}[x]\end{aligned}$$

This is sometimes called the *smoothing* property.

Now consider the conditional expectation

$$\mathbb{E}[f(x)-\theta|t(x)] = g(t(x)) - \theta$$

By Jensen's Inequality

$$(\mathbb{E}[f(x)-\theta|t(x)])^2 \leq \mathbb{E}[(f(x)-\theta)^2|t(x)]\ .$$

Therefore

$$(g(t(x))-\theta)^2 \leq \mathbb{E}[(f(x)-\theta)^2|t(x)]$$

Take expectation of both sides (recall the smoothing property above) yields

$$\mathbb{E}[(g(t(x))-\theta)^2] \leq \mathbb{E}[(f(x)-\theta)^2]$$

□

## 7.4. Exercises

1. Let $x_1, x_2, \ldots, x_n$ be independent and identically distributed random variables. Each $x_i \sim \text{Uniform}(a,b)$. That is, each random variable is uniformly distributed on the interval $(a,b)$. Find scalar sufficient statistics for $a$ and $b$.

2. Suppose that $x_1, \ldots, x_n \overset{\text{iid}}{\sim} \text{Exp}(\lambda)$ for some $\lambda > 0$, that is $p(x|\lambda) = \lambda e^{-\lambda x}$ for $x \geq 0$ and $0$ otherwise. Furthermore, suppose that $y_1, \ldots, y_m \overset{\text{iid}}{\sim} \text{Exp}(\lambda/2)$ and independent of $\{x_i\}$. Find a one-dimensional sufficient statistic $\lambda$.

3. Let $\boldsymbol{x} = [x_1\ x_2]^T$ denote a realization of a bivariate Gaussian random vector. Assume

$$\mathbb{E}[\boldsymbol{x}] = \begin{bmatrix} \mu \\ \mu \end{bmatrix} := \boldsymbol{\mu} \text{ and } \mathbb{E}[(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T] = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$

where $\rho$ is *known* and $\mu$ is an unknown mean parameter of interest. Note that $x_1$ and $x_2$ are correlated.

**(a)** Find a 1-dimensional sufficient statistic $t$ for $\mu$.

**(b)** Show that $x_1$ is an unbiased estimator of $\mu$.

**(c)** Verify the Rao-Blackwell Theorem by deriving the conditional expectation $E[x_1|t]$ and showing that it is also an unbiased estimator of $\mu$ and that its variance is lower than that of $x_1$. Hint: First determine the conditional density of $x_1|t$.

# Lecture 8: Asymptotic Analysis of the MLE

Finding the MLE is essentially density estimation based on a set of i.i.d. samples drawn from a distribution $q$. As the number of samples increases, the MLE tends to the parameter value of the density that is closest to the generating distribution $q$. The MLE is asymptotically Gaussian distributed with covariance given by the inverse of the Fisher Information Matrix.

## 8.1. Convergence of log likelihood to KL

Suppose we make $n$ i.i.d. samples $x_1, \dots, x_n$ drawn from distribution $q$, and consider a parametric family of densities $\{p(x|\theta)\}$. We will use the notation $p(x|\theta)$ and $p_\theta$ interchangeably. The samples could be scalars or vector-valued; this does not affect the analysis in this note. The MLE of $\theta$ is

$$\begin{aligned}\widehat{\theta}_n &= \arg\max_\theta \prod_{i=1}^n p(x_i|\theta) \;=\; \arg\min_\theta \; -\sum_{i=1}^n \log p(x_i|\theta) \\ &= \arg\min_\theta \sum_{i=1}^n \log q(x_i) - \log p(x_i|\theta) \;=\; \arg\min_\theta \sum_{i=1}^n \log \frac{q(x_i)}{p(x_i|\theta)}\end{aligned}$$

By strong law of large numbers (SLLN) for any $\theta \in \Theta$

$$\frac{1}{n}\sum_{i=1}^n \log \frac{q(x_i)}{p(x_i|\theta)} \xrightarrow{\text{a.s.}} D(q\|p_\theta)$$

where $p_\theta$ denotes the parametric density and the quantity $D(q\|p_\theta)$ is the so-called *Kullback-Leibler* divergence of $p_\theta$ from $q$. Let $\theta^* = \arg\min_\theta D(q\|p_\theta)$, the parameter associated with the parametric density that is closest to $q$. We would like to show that the MLE converges to $\theta^*$ in the following sense:

$$D\big(q\|p_{\widehat{\theta}_n}\big) \longrightarrow D(q\|p_{\theta^*})$$

Note that since $\widehat{\theta}_n$ maximizes $\sum_{i=1}^n \log p(x_i|\theta)$ we have

$$\begin{aligned} 0 &\geq \frac{1}{n}\sum_{i=1}^n \log \frac{p(x_i|\theta^*)}{p(x_i|\widehat{\theta}_n)} \\ &= \frac{1}{n}\sum_{i=1}^n \log \frac{p(x_i|\theta^*)}{p(x_i|\widehat{\theta}_n)}\frac{q(x_i)}{q(x_i)} \\ &= \frac{1}{n}\sum_{i=1}^n \log \frac{q(x_i)}{p(x_i|\widehat{\theta}_n)} - \frac{1}{n}\sum_{i=1}^n \log \frac{q(x_i)}{p(x_i|\theta^*)} \\ &\approx D\big(q\|p_{\widehat{\theta}_n}\big) \;-\; D(q\|p_{\theta^*})\;. \end{aligned}$$

This shows that $D(q\|p_{\theta^*}) \gtrsim D\big(q\|p_{\widehat{\theta}_n}\big)$ (i.e., this is an approximate inequality), but by definition $D(q\|p_{\theta^*}) \leq D(q\|p_\theta)$ for all $\theta$. This suggests that $\widehat{\theta}_n \to \theta^*$ in the sense that $D\big(q\|p_{\widehat{\theta}_n}\big) \to D(q\|p_{\theta^*})$. The subtle issue here is that $\widehat{\theta}_n$ is a random variable depending on all the $x_i$ (not a fixed $\theta \in \Theta$), so technically one needs to be a bit more careful when considering the convergence of $\frac{1}{n}\sum_{i=1}^n \log \frac{q(x_i)}{p(x_i|\widehat{\theta}_n)}$, which is not a sum of independent random variables. However, the claimed convergence to KL holds under mild regularity assumptions on the likelihood function; this argument can be made precise [24, Chpt. 3.4]

## 8.2. Asymptotic Distribution of MLE

Above, we argued that under mild assumptions $\widehat{\theta}_n$ converges to $\theta^*$. Next we will characterize the asymptotic distribution of $\widehat{\theta}_n$ under the assumption that the data are generated by $q = p_{\theta^*}$. The notation $\widehat{\theta}_n \overset{\text{asymp}}{\sim} p$ means that as $n \to \infty$ the distribution of $\widehat{\theta}_n$ (a random variable because it is a function of a random set of data) tends to the distribution $p$.

**Theorem 6.** *(Asymptotic Distribution of MLE) Let $x_1, \ldots, x_n$ be iid observations from $p(x|\theta^*)$, where $\theta^* \in \mathbb{R}^d$. Let $\widehat{\theta}_n = \arg\max_\theta \prod_{i=1}^n p(x_i|\theta) = \arg\max_\theta \sum_{i=1}^n \log p(x_i|\theta)$, define $L(\theta) := \sum_{i=1}^n \log p(x_i|\theta)$, and assume $\frac{\partial L(\theta)}{\partial \theta_j}$ and $\frac{\partial^2 L(\theta)}{\partial \theta_j \partial \theta_k}$ exist for all j,k. Furthermore, assume that $p(x|\theta)$ satisfies the regularity*[6] *condition $\mathbb{E}[\frac{\partial \log p(x|\theta)}{\partial \theta}] = 0$. Then*

$$\widehat{\theta}_n \overset{asymp}{\sim} \mathcal{N}(\theta^*, n^{-1} I^{-1}(\theta^*))$$

*where $I(\theta^*)$ is the Fisher-Information Matrix (FIM), whose elements are given by*

$$[I(\theta^*)]_{j,k} = -\mathbb{E}_{x \sim p_\theta^*}\left[\frac{\partial^2 \log p(x|\theta)}{\partial \theta_j \partial \theta_k}|_{\theta=\theta^*}\right]$$

The theorem tells that the distribution tends to a Gaussian. It also tells us that $\widehat{\theta}_n$ is asymptotically unbiased, since the mean of the limiting Gaussian distribution if $\theta^*$. It also characterizes the asymptotic covariance of the estimator; the covariance decays to zero like $1/n$, but the structure of the covariance is determined by the Fisher information matrix. The Fisher Information matrix is the expected value of the negative of the Hessian matrix of the log-likelihood function at the point $\theta^*$. It measures the curvature of the log-likelihood surface. For example, in the case where $\theta$ is scalar, the Fisher Information matrix is simply the negative of the second derivative of the log-likelihood function. Since we are maximizing the log-likelihood, the curvature should be negative. The more negative the curvature, the more sharply defined is the location of the maximum. Therefore, more negative curvatures lead to less variable estimates, which is precisely what is revealed by the limiting distribution above. Figure 8.1 depicts this in the case of a scalar parameter.

**Remark:** Consider the scalar case with $d = 1$. The theorem above tells us that for large $n$ the variance of the MLE is $\mathbb{E}[(\widehat{\theta} - \theta^*)^2] \approx I^{-1}(\theta^*)/n$. This shows that the larger the Fisher Information $I(\theta^*)$, the fewer samples we need to obtain a good estimate of $\theta^*$. Thus, knowing or bounding the Fisher information can help us decide how many training examples are sufficient.

*Proof.* We will prove the theorem for the special case when $\theta$ is scalar. The proof for multidimensional vectors follows the same steps using multivariate calculus. By the mean value theorem,

$$\frac{\partial L(\theta)}{\partial \theta}|_{\theta=\widehat{\theta}_n} \;=\; \frac{\partial L(\theta)}{\partial \theta}|_{\theta=\theta^*} + \frac{\partial^2 L(\theta)}{\partial \theta^2}|_{\theta=\widetilde{\theta}}\,(\widehat{\theta}_n - \theta^*)\,,$$

where $\widetilde{\theta}$ is some value between $\theta^*$ and $\widehat{\theta}_n$. By definition, $\frac{\partial L(\theta)}{\partial \theta}|_{\theta=\widehat{\theta}_n} = 0$, so

$$0 \;=\; \frac{\partial L(\theta)}{\partial \theta}|_{\theta=\theta^*} + \frac{\partial^2 L(\theta)}{\partial \theta^2}|_{\theta=\widetilde{\theta}}\,(\widehat{\theta}_n - \theta^*)$$

[6] The regularity condition amounts to assuming that we can interchange the order of differentiation and integration/expectation (as we will do in the proof). Formally, this is an application of the dominated convergence theorem and requires certain conditions on $p(x|\theta)$ to be met. The conditions hold for most of the distributions we encounter in practice, but it's not true when the support of the distribution depends on $\theta$ (e.g., if $p(x|\theta)$ is the uniform density on $[0, \theta]$).

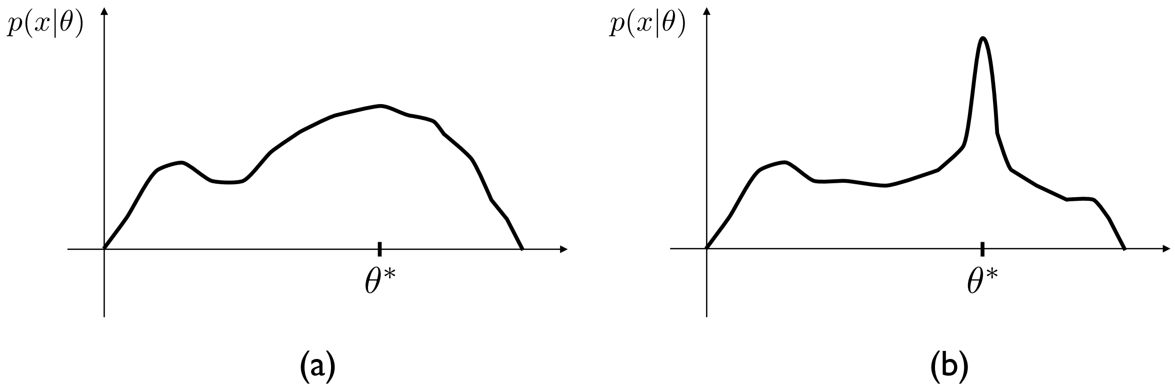


Figure 8.1: Likelihood functions for scalar parameter $\theta$. (a) Low curvature cases will lead to greater estimator variances compared to (b) high curvature cases.

From equation above we have

$$\widehat{\theta}_n - \theta^* \;=\; -\frac{\frac{\partial L(\theta)}{\partial \theta}|_{\theta=\theta^*}}{\frac{\partial^2 L(\theta)}{\partial \theta^2}|_{\theta=\widetilde{\theta}}}$$

Let us consider $\sqrt{n}(\widehat{\theta}_n - \theta^*)$. The reason scaling the difference by $\sqrt{n}$ is that this is the normalization needed to stabilize the limiting distribution. For example, if $x_1, \ldots, x_n$ were iid observations from the distribution $N(\theta^*, 1)$, then it is easy to see that $\sqrt{n}(\widehat{\theta}_n - \theta^*) \sim N(0, 1)$. So, from above we have

$$\sqrt{n}(\widehat{\theta}_n - \theta^*) \;=\; -\frac{\frac{1}{\sqrt{n}}\frac{\partial L(\theta)}{\partial \theta}|_{\theta=\theta^*}}{\frac{1}{n}\frac{\partial^2 L(\theta)}{\partial \theta^2}|_{\theta=\widetilde{\theta}}}\;. \tag{8.1}$$

Note that the denominator is the average curvature of the individual log-likelihood terms, and this average will converge to the mean/expected curvature (a constant larger than zero unless the log-likelihood functions are exactly "flat"). So think of the denominator as behaving like a non-zero constant. We will identify that constant later.

First, let's study the numerator.

$$\frac{1}{\sqrt{n}}\frac{\partial L(\theta)}{\partial \theta}|_{\theta=\theta^*} \;=\; \frac{1}{\sqrt{n}}\sum_{i=1}^{n}\frac{\partial \log p(x_i|\theta)}{\partial \theta}|_{\theta=\theta^*}$$

The expectation (mean) of this quantity is $0$, as shown below.

$$\begin{aligned}
\mathbb{E}\left[\frac{\partial \log p(x|\theta)}{\partial \theta}|_{\theta=\theta^*}\right] &= \int \frac{\partial \log p(x|\theta)}{\partial \theta}|_{\theta=\theta^*} p(x|\theta^*)dx \\
&= \int \frac{1}{p(x|\theta^*)}\frac{\partial p(x|\theta)}{\partial \theta}|_{\theta=\theta^*}\, p(x|\theta^*)dx \\
&= \int \frac{\partial p(x|\theta)}{\partial \theta}|_{\theta=\theta^*} dx \\
&= \frac{\partial}{\partial \theta}\left[\int p(x|\theta)dx\right]|_{\theta=\theta^*} \;= 0\;,
\end{aligned}$$

since $\int p(x|\theta)dx = 1$ for all $\theta$ and the derivative of a constant is $0$. Recall the Central Limit Theorem. If $z_1, \ldots, z_n$ are iid random variables with mean $\mathbb{E}[z_1] = 0$ and variance $\mathbb{E}[z_1^2] = \sigma^2$, then $\frac{1}{\sqrt{n}}\sum_i z_i \xrightarrow{D} \mathcal{N}(0, \sigma^2)$, meaning the

the random variable defined by summation has a distribution that tends to the Gaussian as $n \to \infty$. Therefore, by the CLT we have

$$\frac{1}{\sqrt{n}}\frac{\partial L(\theta)}{\partial\theta}|_{\theta=\theta^*} \;=\; \frac{1}{\sqrt{n}}\sum_{i=1}^{n}\frac{\partial \log p(x_i|\theta)}{\partial\theta}|_{\theta=\theta^*} \;\stackrel{D}{\to}\; \mathcal{N}\Big(0, \mathbb{V}\Big[\frac{\partial\log p(x|\theta)}{\partial\theta}|_{\theta=\theta^*}\Big]\Big) \,.$$

The notation $\stackrel{D}{\to}$ means convergence in distribution. Since the mean is $0$, its variance is just the second moment

$$\mathbb{V}\Big[\frac{\partial\log p(x|\theta)}{\partial\theta}|_{\theta=\theta^*}\Big] \;=\; \mathbb{E}\Big[\Big(\frac{\partial\log p(x|\theta)}{\partial\theta}|_{\theta=\theta^*}\Big)^2\Big] \,.$$

The variance can be related to the curvature of the log-likelihood function as follows. First observe that

$$\begin{aligned}\frac{\partial^2\log p(x|\theta)}{\partial\theta^2} &= \frac{\partial}{\partial\theta}\Big(\frac{1}{p(x|\theta)}\frac{\partial p(x|\theta)}{\partial\theta}\Big)\\ &= -\frac{1}{p^2(x|\theta)}\Big(\frac{\partial p(x|\theta)}{\partial\theta}\Big)^2 + \frac{1}{p(x|\theta)}\frac{\partial^2 p(x|\theta)}{\partial\theta^2}\end{aligned}$$

Now let's take the expectation

$$\begin{aligned}\mathbb{E}\Big[\frac{\partial^2\log p(x|\theta)}{\partial\theta^2}|_{\theta=\theta^*}\Big] &= -\int\Big(\frac{1}{p(x|\theta^*)}\frac{\partial p(x|\theta)}{\partial\theta}|_{\theta=\theta^*}\Big)^2 p(x|\theta^*)\,dx \;+\; \int\frac{1}{p(x|\theta^*)}\frac{\partial^2 p(x|\theta)}{\partial\theta^2}|_{\theta=\theta^*}\,p(x|\theta^*)\,dx\\ &= -\int\Big(\frac{\partial\log p(x|\theta)}{\partial\theta}|_{\theta=\theta^*}\Big)^2 p(x|\theta^*)\,dx \;+\; \int\frac{\partial^2 p(x|\theta)}{\partial\theta^2}|_{\theta=\theta^*}\,dx\\ &= -\mathbb{E}\Big[\Big(\frac{\partial\log p(x|\theta)}{\partial\theta}|_{\theta=\theta^*}\Big)^2\Big] + \frac{\partial^2}{\partial\theta^2}\Big(\int p(x|\theta)\,dx\Big)|_{\theta=\theta^*}\,.\end{aligned}$$

Since $\int p(x|\theta)\,dx = 1$ the second term is $0$. Therefore, the variance is equal to the negative expected curvature:

$$\mathbb{E}\Big[\Big(\frac{\partial\log p(x|\theta)}{\partial\theta}|_{\theta=\theta^*}\Big)^2\Big] \;=\; -\mathbb{E}\Big[\frac{\partial^2\log p(x|\theta)}{\partial\theta^2}|_{\theta=\theta^*}\Big] \;=\; I(\theta^*)\,.$$

Now consider the denominator of (8.1). By the Strong Law of Large Numbers (SLLN), this average converges to its mean value, the negative Fisher Information, almost surely:

$$\frac{1}{n}\frac{\partial^2 L(\theta)}{\partial\theta^2}|_{\theta=\widetilde{\theta}} \;=\; \frac{1}{n}\sum_{i=1}^{n}\frac{\partial^2\log p(x_i|\theta)}{\partial\theta^2}|_{\theta=\widetilde{\theta}} \;\stackrel{a.s.}{\to}\; \mathbb{E}\Big[\frac{\partial^2\log p(x_i|\theta)}{\partial\theta^2}|_{\theta=\theta^*}\Big] \;=\; -I(\theta^*)\,.$$

Note that in the equation above we substituted $\theta^*$ for $\widetilde{\theta}$ since we assume that $\widehat{\theta}_n \stackrel{a.s.}{\to} \theta^*$ (and therefore so does $\widetilde{\theta}$). To summarize, the numerator of (8.1) converges in distribution to a Gaussian

$$\frac{1}{\sqrt{n}}\frac{\partial L(\theta)}{\partial\theta}|_{\theta=\theta^*} \stackrel{D}{\to} \mathcal{N}(0, I(\theta^*))\,,$$

and the denominator $\frac{1}{n}\frac{\partial^2 L(\theta)}{\partial\theta^2}|_{\theta=\widetilde{\theta}} \stackrel{a.s.}{\to} -I(\theta^*)$. So for large $n$, the numerator behaves like a Gaussian random variable and the denominator is almost constant. The ratio therefore converges in distribution to a Gaussian rescaled by the limiting constant of the denominator

$$\sqrt{n}(\widehat{\theta}_n - \theta^*) \;\stackrel{D}{\to}\; \frac{1}{I(\theta^*)}\mathcal{N}(0, I(\theta^*)) \;\equiv\; \mathcal{N}(0, I^{-1}(\theta^*))\,.$$

This type of convergence is rigorously proved by *Slutsky's Theorem* [23]. □

## 8.3. Exercises

1. Congratulations! You have just been hired to by Google to work on their online ad auction team. Website advertising spaces are sold by an auction. For simplicity, let's assume the following auction model. There is one auction for each ad space. Each bidder places a single bid per ad space, and the highest bidder wins that auction. Since Google runs the ad auction service, they can observe all the bids. The website selling the ad space only observes the final winning bid.

   Your first assignment at Google is to determine how much the sellers can learn about the distribution of bids from observations of only the highest bids in each auction. Here is a mathematical model of the observation process. Suppose there are $n$ bidders in an auction and there (non-negative) bids are $x_1, x_2, \ldots, x_n \overset{iid}{\sim} \theta e^{-\theta x}$ (i.e., exponentially distributed with parameter $\theta$). The seller observes $y := \max_{i=1,\ldots,n} x_i$.

   **a.** Derive an expression for the probability density of $y$. Hint: Start by finding the cumulative distribution function of $y$.

   **b.** How would you find the MLE of $\theta$ given $y$?

   **c.** Let $\widehat{\theta}$ denote the MLE. Show that for $n$ sufficiently large $\widehat{\theta} \approx \frac{\log n}{y}$.

2. Suppose that $n_1$ people are given treatment 1 and $n_2$ people are given treatment 2. let $x_1$ bet the number of people on treatment 1 who respond favorably to the treatment and let $x_2$ be the number of people on treatment 2 who respond favorably. Assume $x_1 \sim$ Binomial$(n_1, p_1)$, $x_2 \sim$ Binomial$(n_2, p_2)$, and that $x_1$ and $x_2$ are statistically independent. Let $\phi = p_1 - p_2$.

   **a.** Find the MLE $\widehat{\phi}$ for $\phi$.

   **b.** Find the Fisher information matrix $I(p_1, p_2)$.

   **c.** Use the result of (b) to characterize the large-sample distribution of $\widehat{\phi}$.

   **d.** Assume that $n_1 = n_2$. Based on the result of (c), roughly how many people are required in the study so that $\mathbb{P}(|\widehat{\phi} - \phi| > 0.01) < 0.05$?

   **e.** Verify your conclusions in (d) by simulating this problem (in Python, Matlab, R, etc); i.e., generate many iid realizations of $x_1$ and $x_2$ with $n_1 = n_2$ from part (d) and generate a Monte Carlo estimate of $\mathbb{P}(|\widehat{\phi} - \phi| > 0.01)$. You can chose $p_1$ and $p_2$ as you like (e.g., $p_1 = p_2$ will suffice).

3. Suppose we observe measurements of the form

$$y_i \;=\; f_{\boldsymbol{w}}(\boldsymbol{x}_i) \;+\; \epsilon_i \;,\; i = 1, \ldots, n \;,$$

   where $\boldsymbol{x}_i \in \mathbb{R}^d$ are *known* (deterministic), the $\epsilon_i$ are i.i.d. zero-mean, unit-variance random variables, and $f_{\boldsymbol{w}}$ is a function parametermized by an *unknown* weight vector $\boldsymbol{w}$.

   (a) What are the conditional mean and variance of $y_i$ given $\boldsymbol{x}_i$?

   (b) Let $p$ denote the probability density of the $\epsilon_i$. What optimization would you solve to find the maximum likelihood estimate (MLE) of $\boldsymbol{w}$?

   For the rest of the questions, assume that $f_{\boldsymbol{w}}(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x}$, a linear function and that $\epsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$. Also, remember that we are treating the $\boldsymbol{x}_i$ as known, deterministic vectors. Assume that the linear span of $\{\boldsymbol{x}_i\}_{i=1}^n$ is $\mathbb{R}^d$.

   (a) Give an explicit (linear-algebraic) expression in terms of $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ for the MLE of $\boldsymbol{w}$ and its distribution.

(b) What is the Fisher information matrix for the MLE?

(c) Suppose that you have a training sample budget. Assume $n \gg d$, but may only use $d$ training examples. If our aim is to minimize the mean-square error $\mathbb{E}[\|\boldsymbol{w} - \widehat{\boldsymbol{w}}\|^2]$, propose a method for selecting the $d$ examples. Hint: Express $\mathbb{E}[\|\boldsymbol{w} - \widehat{\boldsymbol{w}}\|^2]$ in terms of the Fisher information matrix.

# Lecture 9: Maximum Likelihood Estimation and Empirical Risk Minimization

Suppose we have a "training" dataset of independent and identically distributed examples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$. The goal of prediction is to predict the unknown label $y$ from an observation of a new $\boldsymbol{x}$, under the assumption that $(\boldsymbol{x}, y)$ is independently and identically distributed to the training data. Specifically, the goal is to "learn" a function $f$ using the training data so that $f(\boldsymbol{x})$ agrees with $y$ in some sense. We look at two approaches that lead to similar (sometimes identical) solutions to this learning problem: Maximum Likelihood Estimation and Empirical Risk Minimization.

## 9.0.1. Maximum Likelihood Estimation Approach

The MLE approach starts by assuming some form for the conditional distribution of $y$ given $\boldsymbol{x}$. For example, we might assume that the conditional distribution is Gaussian or Bernoulli, which lead to least squares or logistic regression, respectively (more on this later). One of the main ideas in this approach is to use models from the *exponential family* of distributions, which includes many of the common distributions including the Gaussian and Bernoulli. Using *any* distribution from the exponential family produces a MLE optimization problem that is convex in the prediction function $f$. This is because any such distribution can be written in terms a parameter $\theta$, called the *natural parameter* of the distribution, as

$$p(y|\theta) \propto \exp\big(-\ell(y,\theta)\big)$$

for some function $\ell$ that depends on the specific distribution under consideration. A key fact (which we prove below) is that if $p(y|\boldsymbol{x})$ is in the exponential family, then $\ell$ is convex in $\theta$. The idea is to model the natural parameter as a function of $\boldsymbol{x}$, so that the distribution of $y$ is determined by $\boldsymbol{x}$. Substituting $\theta = f(\boldsymbol{x})$, we have

$$p(y|\boldsymbol{x}) \propto \exp\big(-\ell(y, f(\boldsymbol{x}))\big)$$

The MLE is the solution to

$$\max_{f\in\mathcal{F}} \sum_{i=1}^n \log p(y_i|\boldsymbol{x}_i) \;\equiv\; \min_{f\in\mathcal{F}} \sum_{i=1}^n \ell\big(y_i, f(\boldsymbol{x}_i)\big)$$

where $\mathcal{F}$ is a certain set of allowable functions. The fact that $f$ is convex in $f$, ensures that all local minima are global.

## 9.0.2. Empirical Risk Minimization

One of the most common approaches to machine learning is called *Empirical Risk Minimization* (ERM). ERM is based on a choice of a loss function $\ell$ that measures the disagreement between a label $y$ and a predictor $f(\boldsymbol{x})$. The value of $\ell(y, f(\boldsymbol{x}))$ is called the the loss of $f$ on the example $(\boldsymbol{x}, y)$. Common losses included the squared error loss $\ell(y, f(\boldsymbol{x})) = (y - f(\boldsymbol{x}))^2$ and the hinge loss $\ell(y, f(\boldsymbol{x})) = \max(0, 1 - yf(\boldsymbol{x}))$, which is common in binary classification problems with the label set $y \in \{-1, +1\}$. The expected value of the loss, with respect to the joint distribution of $(\boldsymbol{x}, y)$ is called the *risk*. Given a class of functions $\mathcal{F}$, one would like to find the $f \in \mathcal{F}$ that minimizes the risk, i.e., to find a solution to

$$\min_{f\in\mathcal{F}} \mathbb{E}\big[\ell(y, f(\boldsymbol{x}))\big]\ .$$

ERM approximates the expectation with an i.i.d. set of training examples $\{(\boldsymbol{x}_i, y_i)\}$ and finds the solution to

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{n} \ell(y_i, f(\boldsymbol{x}_i)) \ .$$

This is reasonable since $\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(\boldsymbol{x}_i)) \to \mathbb{E}\big[\ell(y, f(\boldsymbol{x}))\big]$ as $n \to \infty$. Notice that the ERM optimization has the same form as the MLE. So, given a choice of loss function, we can view ERM as MLE with a conditional distribution model $p(y|\boldsymbol{x}) \propto \exp\big(-\ell(y, f(\boldsymbol{x}))\big)$.

### 9.0.3. Example: Gaussian Models and Least Squares

Consider the following model for a labeled dataset $\{\boldsymbol{x}_i, y_i\}_{i=1}^{n}$. Suppose that the labels $y_i$ are conditionally independent given their correponding features $\boldsymbol{x}_i$ and that $y_i \sim \mathcal{N}(f_{\boldsymbol{w}}(\boldsymbol{x}_i), 1)$. This means that $\mathbb{E}[y_i|\boldsymbol{x}_i] = f_{\boldsymbol{w}}(\boldsymbol{x}_i)$, where $f_{\boldsymbol{w}}$ is a function parameterized by $\boldsymbol{w}$. The variance of $y_i|\boldsymbol{x}_i$ is 1 (more generally, we could assume any other value for the variance). In this case, the log-likelihood of $\boldsymbol{w}$ given $\{\boldsymbol{x}_i, y_i\}$ is

$$L(\boldsymbol{w}) \ = \ -\frac{1}{2}\sum_{i=1}^{n}\Big(y_i - f_{\boldsymbol{w}}(\boldsymbol{x}_i)\Big)^2 \ + \ C$$

where $C$ is a constant that does not depend on $\boldsymbol{w}$. Thus, the MLE of $\boldsymbol{w}$ is given by the *least squares* optimization

$$\widehat{\boldsymbol{w}} \ = \ \arg\min_{\boldsymbol{w}} \sum_{i=1}^{n}\Big(y_i - f_{\boldsymbol{w}}(\boldsymbol{x}_i)\Big)^2$$

If we assume a linear model, that is $f_{\boldsymbol{w}}(\boldsymbol{x}_i) = \boldsymbol{w}^T\boldsymbol{x}_i$, then we have the classical least squares problem

$$\arg\min_{\boldsymbol{w}} \sum_{i=1}^{n}(y_i - \boldsymbol{x}_i^T\boldsymbol{w})^2$$

and the MLE is a solution to the linear system

$$\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} = \ \boldsymbol{X}^T\boldsymbol{y}$$

where $\boldsymbol{X}$ is a matrix with rows $\boldsymbol{x}_i^T$ and $\boldsymbol{y}$ is a vector with rows equal to $y_i$, $i = 1, \dots, n$.

*Generalized linear models* extend this sort of linear modeling approach to other distributions, including cases where the conditional distribution of $y_i$ given $\boldsymbol{x}_i$ is binomial (binary classification), multinomial (multi-class), Poisson, exponential, and other probability models in the *exponential family*. The common theme is that the conditional probability model takes the form $p(y|\boldsymbol{x}) \propto \exp(-\ell(y, \boldsymbol{w}^T\boldsymbol{x}))$, where $\ell(y, \boldsymbol{w}^T\boldsymbol{x})$ is a convex function of $\boldsymbol{w}$.

## 9.1. The Exponential Family

The Exponential Family is a class of distributions with the following form:

$$p(y|\boldsymbol{\theta}) \ = \ b(y)\exp(\boldsymbol{\theta}^T\,\boldsymbol{t}(y) - a(\boldsymbol{\theta})) \ .$$

The parameter $\boldsymbol{\theta}$ is called the *natural parameter* of the distribution and $\boldsymbol{t}(y)$ is the *sufficient statistic*. The quantity $e^{-a(\boldsymbol{\theta})}$ is a normalization constant, ensuring that $p(y|\theta)$ sums or integrates to 1. If $y$ is continuous

$$1 \;=\; \int p(y|\boldsymbol{\theta})\,dy \;=\; e^{-a(\boldsymbol{\theta})} \int b(y)\exp(\boldsymbol{\theta}^T\,\boldsymbol{t}(y))\,dy$$

which shows that $a(\boldsymbol{\theta}) = \log\big(\int b(y)\exp(\boldsymbol{\theta}^T\,\boldsymbol{t}(y))\,dy\big)$. If $y$ is discrete, then the integral is replaced by a summation. $a(\boldsymbol{\theta})$ is called the *log partition function*. The factor $b(y)$ is the non-negative *base measure*, and in many cases it is equal to 1. Many familiar distributions belong to the exponential family (e.g., Gaussian, exponential, log-normal, gamma, chi-squared, beta, Dirichlet, Bernoulli, Poisson, geometric). To use exponential family distributions to model the conditional distribution of $y$ given $\boldsymbol{x}$, we take $\boldsymbol{\theta}$ to be a parametric function of $\boldsymbol{x}$, e.g., a linear function $\theta = \boldsymbol{w}^T\boldsymbol{x}$.

Note that the negative log-likelihood function of $\boldsymbol{\theta}$ is

$$-\log p(y|\boldsymbol{\theta}) \;=\; -\boldsymbol{\theta}^T\,\boldsymbol{t}(y) + a(\boldsymbol{\theta}) - \log b(y)\;.$$

Remarkably, this is a convex function of $\boldsymbol{\theta}$, which means that a maximum likelihood estimator can be easily computed using convex optimization. To show it is convex, note that the first term is linear and hence convex in $\boldsymbol{\theta}$. We will verify $a(\boldsymbol{\theta})$ is convex by showing that for any $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$, and $\lambda \in [0,1]$

$$a\big(\lambda\boldsymbol{\theta}_1 + (1-\lambda)\boldsymbol{\theta}_2\big) \;\le\; \lambda a(\boldsymbol{\theta}_1) + (1-\lambda)a(\boldsymbol{\theta}_2)\;.$$

We will assume $y$ is continuous, but the following argument holds in the discrete case by replacing integrals with summations. We will use Hölder's inequality which states that $p, q \ge 1$ such that $1/p + 1/q \le 1$ and any two functions $f(y)$, $g(y)$ and measure $b(y)$

$$\int |f(y)g(y)|\,b(y)dy \le \Big(\int |f(y)|^p\,b(y)dy\Big)^{1/p}\Big(\int |g(y)|^q\,b(y)dy\Big)^{1/q}\;.$$

Consider

$$\begin{aligned}
\exp\Big(a\big(\lambda\boldsymbol{\theta}_1 + (1-\lambda)\boldsymbol{\theta}_2\big)\Big) &= \int \exp\Big(\big(\lambda\boldsymbol{\theta}_1 + (1-\lambda)\boldsymbol{\theta}_2\big)^T\,\boldsymbol{t}(y)\Big)\,b(y)dy \\
&= \int \exp\big(\boldsymbol{\theta}_1^T\boldsymbol{t}(y)\big)^\lambda\,\exp\big(\boldsymbol{\theta}_2^T\,\boldsymbol{t}(y)\big)^{(1-\lambda)}\,b(y)dy \\
&\le \Big(\int \exp\big(\boldsymbol{\theta}_1^T\boldsymbol{t}(y)\big)b(y)dy\Big)^\lambda\Big(\int \exp\big(\boldsymbol{\theta}_2^T\boldsymbol{t}(y)\big)b(y)dy\Big)^{1-\lambda} \\
&= \exp(a(\boldsymbol{\theta}_1))^\lambda\,\exp(a(\boldsymbol{\theta}_2))^{1-\lambda}
\end{aligned}$$

where we applied Hölder's inequality with $p = 1/\lambda$ and $q = 1/(1-\lambda)$. Taking the logarithm of both sides yields the desired result.

Next we will consider several applications of the GLM framework.

**Gaussian Distribution: Classical Least Squares**

$$p(y|\mu) \;=\; \frac{1}{\sqrt{2\pi}}e^{-(y-\mu)^2/2} \;=\; \frac{1}{\sqrt{2\pi}}e^{-y^2/2}\exp(\mu\,y - \mu^2/2)$$

and we identify $b(y) = \frac{1}{\sqrt{2\pi}}e^{-y^2/2}$, $\theta = \mu$, $t(y) = y$, and $a(\theta) = \theta^2/2$. Note that the natural parameter in this case is the mean $\mu$. If we use this as a model for the conditional distribution $y_i|\boldsymbol{x}_i$ and let $\theta = \boldsymbol{w}^T\boldsymbol{x}_i$, then we have the least squares linear model above. In other words, in least squares we model $y_i$ as Gaussian with unit variance and model its mean as a linear function of the corresponding feature $\boldsymbol{x}_i$.

**Binomial Distribution: Logistic Regression**

$$p(y|\mu) \;=\; \mu^y(1-\mu)^{1-y} \;=\; \exp\Big(y\log(\mu)+(1-y)\log(1-\mu\Big) \;=\; \exp\Big(y\log\Big(\frac{\mu}{1-\mu}\Big)+\log(1-\mu)\Big)$$

and we identify $b(y)=1$, $\theta=\log(\mu/(1-\mu))$, $t(y)=y$, and $a(\theta)=\log(1+e^\theta)$. The natural parameter in this case is *not* the mean. Reparameterizing the Binomial distribution in terms of its natural parameter, we have

$$p(y|\theta) \;=\; \exp\Big(\theta\, y - \log(1+e^\theta)\Big)$$

Consider the log-likelihood function $\log p(y|\theta)$.

$$\begin{aligned}\log p(y=1|\theta) &= \theta-\log(1+e^\theta)=\log(e^\theta)-\log(1+e^\theta)=\log\Big(\frac{1}{1+e^{-\theta}}\Big)\\ \log p(y=0|\theta) &= \log\Big(\frac{1}{1+e^\theta}\Big)\end{aligned}$$

For convenience, let us use the binary labels ±1 instead of $0$ and $1$; i.e., reassign $y\rightarrow 2y-1$. Then we have

$$\log p(y|\theta) \;=\; \log\Big(\frac{1}{1+e^{-y\theta}}\Big)$$

To use this as a model for the conditional distribution $y|\boldsymbol{x}$, let $\theta=\boldsymbol{w}^T\boldsymbol{x}$. Consider iid examples $\{(\boldsymbol{x}_i,y_i)\}_{i=1}^n$. In this case, we model $y_i$ as Bernoulli (with labels $\pm1$) and model its natural parameter as a linear function of the corresponding feature $\boldsymbol{x}_i$. This is a (generalized) linear model for a binary classification setting. The maximum likelihood estimator of $\boldsymbol{w}$ is the solution to

$$\min_{\boldsymbol{w}}\sum_{i=1}^n\log\Big(1+e^{-y_i\boldsymbol{x}_i^T\boldsymbol{w}}\Big)$$

If $\widehat{\boldsymbol{w}}$ is the solution, then thepredicted label for a new $\boldsymbol{x}$ is given by

$$\widehat{y} \;=\; \text{sign}(\widehat{\boldsymbol{w}}^T\boldsymbol{x})$$

This optimization is called *logistic regression* because the function $f(\theta):=\log(1/(1+e^{-\theta}))$ is known as the *logistic function*. The function $\ell(\theta):=\log(1+e^{-\theta})$ is called the *logistic loss function*. The logistic loss function is convex in $\theta$. To see this, note that the second derivative of $\ell(\theta)$ is

$$\frac{e^{-\theta}}{1+e^{-\theta}}\Big(1-\frac{e^{-\theta}}{1+e^{-\theta}}\Big)\geq 0$$

**Multinomial Distribution: Multinomial Logistic Regression**

Let $y$ be a random variable that takes value $k$ with probability $\mathbb{P}(y = k) = q_k$, $k = 1, \ldots, K$. The likelihood of $q_1, \ldots, q_K$ is

$$p(y|q_1, \ldots, q_K) \ = \ \sum_{k=1}^{K} \mathbb{1}_{\{y=k\}}\, q_k \ = \ \exp\left(\sum_{k=1}^{K} \mathbb{1}_{\{y=k\}} \log q_k\right) \ = \ \exp\Big(\boldsymbol{\theta}^T \boldsymbol{t}(y) - a(\boldsymbol{\theta})\Big)$$

where $\boldsymbol{\theta} \in \mathbb{R}^k$, $q_k = e^{\theta_k} / \sum_{m=1}^{K} e^{\theta_m}$, $a(\boldsymbol{\theta}) = \log\left(\sum_{k=1}^{K} e^{\theta_k}\right)$, and $\boldsymbol{t}(y)$ is the “one-hot" vector

$$\boldsymbol{t}(y) \ = \ \begin{bmatrix} \mathbb{1}_{\{y=1\}} \\ \vdots \\ \mathbb{1}_{\{y=K\}} \end{bmatrix}$$

The function $e^{\theta_k} / \sum_{m=1}^{K} e^{\theta_m}$ is called the *logit* or *soft-max* function. Notice that this parameterization ensures that the probabilities sum to $1$. Reparameterizing the likelihood function in terms of $\boldsymbol{\theta}$ yields

$$L(\boldsymbol{\theta}) \ := \ \log p(y|\theta_1, \ldots, \theta_K) \ = \ \log\left(\sum_{k=1}^{K} \mathbb{1}_{\{y=k\}} \frac{e^{\theta_k}}{\sum_{m=1}^{K} e^{\theta_m}}\right) \ = \ \sum_{k=1}^{K} \mathbb{1}_{\{y=k\}} \log\left(\frac{e^{\theta_k}}{\sum_{m=1}^{K} e^{\theta_m}}\right)$$

To use this as a model for the conditional distribution $y|\boldsymbol{x}$, we introduce weight vectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K$ and let $\theta_k = \boldsymbol{w}_k^T \boldsymbol{x}$. This is a linear model for a multiclass classification setting. Consider iid examples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$. The maximum likelihood estimator of $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K$ is the solution to

$$\min_{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K} -\sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{1}_{\{y_i=k\}} \log\left(\frac{e^{\boldsymbol{w}_k^T \boldsymbol{x}_i}}{\sum_{m=1}^{K} e^{\boldsymbol{w}_m^T \boldsymbol{x}_i}}\right)$$

The objective is a convex function in the weight vectors, because of the convexity with respect to $\boldsymbol{\theta}$. Each term in the sum is a *cross-entropy* loss. This name comes from the following interpretation. Let $\{p_k\}$ and $\{q_k\}$ be probability mass functions. The cross-entropy $\{q_k\}$ relative to $\{p_k\}$ is $-\sum_k p_k \log q_k$. Each term in the objective above is the cross-entropy between $\{\mathbb{1}_{\{y=k\}}\}$ and $\{q_k\}$, where $q_k = \frac{e^{\boldsymbol{w}_k^T \boldsymbol{x}_i}}{\sum_{m=1}^{K} e^{\boldsymbol{w}_m^T \boldsymbol{x}_i}}$.

Finally, let $\widehat{\boldsymbol{w}}_1, \ldots, \widehat{\boldsymbol{w}}_K$ denote the MLEs. The predicted label for a new $\boldsymbol{x}$ is given by

$$\widehat{y} \ = \ \arg \max_{k \in \{1, \ldots, K\}} \widehat{\boldsymbol{w}}_k^T \boldsymbol{x}$$

## 9.2. Exercises

1. Suppose that $y \geq 0$ and $p(y|\mu) = \frac{1}{\mu} e^{-y/\mu}$, the exponential density function.

   (a) Show that $\mathbb{E}[y] = \mu$.

   (b) Express this density in the exponential family form $p(y|\theta) \ = \ b(y) \exp(\theta\, t(y) - a(\theta))$.

   (c) Suppose that $y_1, \ldots, y_n$ are iid according to $p(y|\theta)$. Show that the negative log-likelihood is a convex function of $\theta$.

(d) What is the MLE for $\theta$?

2. Consider the Binomial and Multinomial GLMs:

$$\text{Binomial} : p(y|\boldsymbol{x}, \boldsymbol{w}) \;=\; \exp\left( y \log\left( \frac{1}{1+e^{-\boldsymbol{x}^T\boldsymbol{w}}} \right) + (1-y)\log\left( \frac{1}{1+e^{\boldsymbol{x}^T\boldsymbol{w}}} \right) \right)$$

$$\text{Multinomial} : p(y=\ell|\boldsymbol{x}, \boldsymbol{w}_1, \ldots, \boldsymbol{w}_k) \;=\; \frac{\exp(\boldsymbol{x}^T\boldsymbol{w}_\ell)}{\sum_{j=1}^{k}\exp(\boldsymbol{x}^T\boldsymbol{w}_j)} \;,\; \text{for } \ell = 1, \ldots, k$$

(a) Show that the two models are equivalent for the binary classification, $k = 2$, case. HINT: Relate $\boldsymbol{w}$ in the Binomial model to $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ in the Multinomial model.

(b) What is the limiting behavior of the $p(y|\boldsymbol{x}, \boldsymbol{w})$ as $\|\boldsymbol{w}\| \to \infty$?

(c) Consider the class-conditional densities $\boldsymbol{x}|y = 1 \sim \mathcal{N}(\boldsymbol{w}, \boldsymbol{I})$ and $\boldsymbol{x}|y = 0 \sim \mathcal{N}(-\boldsymbol{w}, \boldsymbol{I})$. Assuming equal prior probabilities, find an expression for $p(y = 1|\boldsymbol{x}, \boldsymbol{w})$.

(d) Suppose we have $n$ training examples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ that are independently sampled with equal probability from the two Gaussian class-conditional densities above and consider the estimate

$$\widehat{\boldsymbol{w}} \;=\; \frac{1}{n}\sum_{i=1}^{n}(2y_i - 1)\boldsymbol{x}_i \;.$$

i. What is the distribution of $\widehat{\boldsymbol{w}}$?

ii. What is the asymptotic distribution of $\frac{\widehat{\boldsymbol{w}}}{\|\widehat{\boldsymbol{w}}\|}$? Hint: Use that fact that by the SLLN, $\|\widehat{\boldsymbol{w}}\| \overset{a.s.}{\to} \|\boldsymbol{w}\|$.

3. Recall the negative log-likelihood function for the multinomial GLM

$$-L(\boldsymbol{\theta}) \;=\; -\log\left( \sum_{k=1}^{K} \mathbb{1}_{\{y=k\}} \frac{e^{\theta_k}}{\sum_{m=1}^{K} e^{\theta_m}} \right)$$

Prove directly that this is convex in $\boldsymbol{\theta}$ by showing that the Hessian matrix is positive semidefinite via the following steps.

(a) Denote the Hessian matrix by $-\nabla^2 L(\boldsymbol{\theta})$. The $j, k$-th element of this matrix is $-\frac{\partial^2 L(\boldsymbol{\theta})}{\partial\theta_j\partial\theta_k}$. Let $\boldsymbol{q}$ be the $K \times 1$ vector of the probabilities $q_j = e^{\theta_j}/\sum_{k=1}^{K} e^{\theta_k}$, $j = 1, \ldots, K$. Show that

$$-\nabla^2 L(\boldsymbol{\theta}) \;=\; \text{diag}(\boldsymbol{q}) - \boldsymbol{q}\boldsymbol{q}^T$$

(b) The Hessian is positive semidefinite by shown that for any $\boldsymbol{v} \in \mathbb{R}^K$

$$\boldsymbol{v}^T\nabla^2 L(\boldsymbol{\theta})\boldsymbol{v} \;=\; \sum_{k=1}^{K} q_k v_k^2 - \Big( \sum_{k=1}^{K} q_k v_k \Big)^2 \;\geq\; 0 \;.$$

Hint: You can interpret the two sums in the previous expression as expectations and apply Jensen's inequality.

# Lecture 10: Linear Models

The main idea of the Generalized Linear Model (GLM) is to model $p(y|\boldsymbol{x})$ in terms of a linear function (i.e., weighted combination) of the features. Specifically, given a chosen probability distribution for $y$, we set the natural parameter to be $\theta = \boldsymbol{w}^T\boldsymbol{x}$, and so we will write this model as $p(y|\boldsymbol{w}^T\boldsymbol{x})$. In other words, we consider a parametric family of models for the conditional distribution of $y$ given $\boldsymbol{x}$ indexed by $\boldsymbol{w} \in \mathbb{R}^d$, which we denote as $\left\{p(y|\boldsymbol{w}^T\boldsymbol{x})\right\}_{\boldsymbol{w}\in\mathbb{R}^d}$. The MLE of $\boldsymbol{w}$ is the solution to

$$\min_{\boldsymbol{w}} \sum_{i=1}^{n} -\log p(y_i|\boldsymbol{w}^T\boldsymbol{x}_i) \ .$$

This will be relatively easy to solve if $-\log p(y|\boldsymbol{w}^T\boldsymbol{x})$ is differentiable and convex in $\boldsymbol{w}$, which is the case for GLMs. In other words, $p(y|\boldsymbol{w}^T\boldsymbol{x}) \propto \exp(-\ell(y, \boldsymbol{w}^T\boldsymbol{x}))$ for a function $\ell$ satisfying the following properties:

1. $\ell$ is convex in $\boldsymbol{w}$.
2. $\ell : \mathbb{R} \to [0, \infty)$ so that $p(y|\boldsymbol{w}^T\boldsymbol{x}) = \exp(\log p(y|\boldsymbol{w}^T\boldsymbol{x})) \propto \exp(-\ell(y, \boldsymbol{w}^T\boldsymbol{x})) \in [0, 1]$.

The $\ell$ is a called a *loss function* that measures the error/distortion between $y_i$ and the value predicted by $\boldsymbol{w}^T\boldsymbol{x}_i$. The general form of the optimization is

$$\min_{\boldsymbol{w}} \sum_{i=1}^{n} \ell(y_i, \boldsymbol{w}^T\boldsymbol{x}_i)$$

If $\ell$ is convex in $\boldsymbol{w}$, then the overall optimization is also convex.

## 10.1. Common Loss Functions

The GLM is a systematic framework for obtaining convex loss functions that are matched to the (assumed) data distribution. Here is a list of common loss functions used in machine learning for regression and/or binary classification. Some are derived via the GLM framework (e.g., quadratic, logistic), while others are not (e.g., hinge loss).

**Quadratic/Gaussian:** $\ell(y_i, \boldsymbol{w}^T\boldsymbol{x}_i) \ = \ (y_i - \boldsymbol{w}^T\boldsymbol{x}_i)^2$

**Absolute/Laplacian:** $\ell(y_i, \boldsymbol{w}^T\boldsymbol{x}_i) \ = \ |y_i - \boldsymbol{w}^T\boldsymbol{x}_i|$

**Logistic:** $\ell(y_i, \boldsymbol{w}^T\boldsymbol{x}_i) \ = \ \log(1 + \exp(-y_i\boldsymbol{w}^T\boldsymbol{x}_i))$

**Hinge:** $\ell(y_i, \boldsymbol{w}^T\boldsymbol{x}_i) \ = \ \max(0, 1 - y_i\boldsymbol{w}^T\boldsymbol{x}_i)$

**0/1:** $\ell(y_i, \boldsymbol{w}^T\boldsymbol{x}_i) \ = \ \mathbb{1}_{\{y_i\boldsymbol{w}^T\boldsymbol{x}_i<0\}}$

where we use the convention $y_i \in \{-1, +1\}$ for the final three "classification" loss functions. All of these are convex functions, and thus easy to optimize, except for the $0/1$ loss. Notice that in the context of classification, we can express the quadratic/Gaussian loss and the absolute/Laplacian loss in terms of $y_i \boldsymbol{x}_i^T \boldsymbol{w}$ as well, since $y_i \in \{-1, +1\}$ implies that $|y_i| = 1$ and therefore

$$|y_i - \boldsymbol{x}_i^T \boldsymbol{w}| = |y_i||1 - y_i \boldsymbol{x}_i^T \boldsymbol{w}| = |1 - y_i \boldsymbol{x}_i^T \boldsymbol{w}| \ .$$

Figure 10.1 compares the various loss functions in the binary classification setting as a function of $y_i \boldsymbol{w}^T \boldsymbol{x}_i$. Recall that the classification decision will be $\text{sign}(\boldsymbol{x}_i^T \boldsymbol{w})$. Thus, if $y_i \boldsymbol{w}^T \boldsymbol{x}_i > 0$, then the decision is correct. The $0/1$ loss is ideal, since its expected value is exactly the probability of error. The other loss functions can be viewed as convex approximations to the $0/1$ loss. The absolute value and quadratic losses have the undesirable feature of assigning large losses to some points that are correctly classified (these losses are more suitable for regression problems). The logistic and hinge losses reduce this problem, and thus they are preferred for classification tasks.

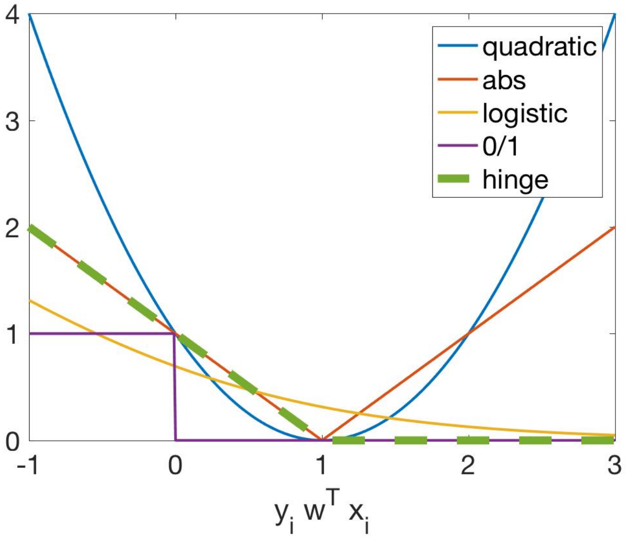


Figure 10.1: Comparison of loss functions for binary classification, with $y_i \in \{-1, +1\}$.

## 10.2. Optimization

In general, the optimization problem

$$\min_{\boldsymbol{w}} \sum_{i=1}^{n} \ell(y_i, \boldsymbol{w}^T \boldsymbol{x}_i)$$

does not have a closed-form solution, and we need to solve it by gradient descent or other iterative algorithms. Denote the gradient of the loss function with respect to $\boldsymbol{w}$ by $\nabla \ell$. Then gradient descent starts from an initial $\boldsymbol{w}_0$ and proceeds according to the iteration

$$\boldsymbol{w}_t \ = \ \boldsymbol{w}_{t-1} - \mu \sum_{i=1}^{n} \nabla \ell(y_i, \boldsymbol{w}_{t-1}^T \boldsymbol{x}_i) \ , \ t = 1, 2, \ldots$$

where $\mu > 0$ is a step size. If the loss is convex in $\boldsymbol{w}$, then gradient descent will converge to a global minimum (if $\mu$ is sufficiently small). If the loss is continuous but non-convex, then it may converge to a (suboptimal) local minimum. If the loss is discontinuous, as in the case of $0/1$ loss, then gradient descent cannot be used to solve the optimization.

In the special case of the quadratic loss (Gaussian negative log likelihood), we have a closed-form solution to the optimization. The optimization can be expressed in matrix-vector form by defining $\boldsymbol{X}$ to be an $n \times d$ matrix with $i$th row equal to $\boldsymbol{x}_i^T$ and $\boldsymbol{y}$ to be an $n \times 1$ vector with $i$th row equal to $y_i$. Then the optimization is

$$\min_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|^2 .$$

Setting the derivative with respect to $\boldsymbol{w}$ to $\boldsymbol{0}$ results in the system of equations

$$-\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) \;=\; \boldsymbol{0}$$

and so (assuming $\boldsymbol{X}$ is full rank) the solution is

$$\widehat{\boldsymbol{w}} \;=\; \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y} .$$

## 10.3. Exercises

1. Consider a binary classification problem with training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, with $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$.

   (a) Suppose that training data are linearly separable. That is, there exists a vector $\boldsymbol{w} \in \mathbb{R}^d$ such that $y_i = \text{sign}(\boldsymbol{w}^T\boldsymbol{x}_i)$, $i = 1, \ldots, n$. Construct an example of this situation in $d = 2$ with $n = 3$ points that are not collinear and with at least one from each class. Sketch your example in the two-dimensional plane. In this simple setting, you can compare the behaviors of learning with various loss functions without the need for numerical optimizations.

   (b) Formulate the learning problem using $0/1$-loss. What is the minimum value of the objective (loss)? Find a solution to the learning problem. Is it unique?

   (c) Formulate the learning problem as a logistic regression problem. What is the minimum value of the logistic regression objective (loss)? Find a solution to the logistic regression. Is it unique?

   (d) Derive an expression for the gradient descent steps for logistic regression (i.e., what is the gradient of the loss function?).

   (e) Formulate the learning problem using hinge loss instead. Prove that the hinge loss is convex.

   (f) What is the minimum value of the hinge loss objective? Find a solution in this case. Is it unique?

   (g) Finally, consider squared error loss. Find a solution in this case. Is it unique?

2. Suppose that $y_i \sim \text{Poisson}(\lambda_i)$, $i = 1, \ldots, n$. Derive a GLM for this model, i.e., model the natural parameter $\boldsymbol{\theta} = \boldsymbol{X}\boldsymbol{w}$, where $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ is known and $\boldsymbol{w} \in \mathbb{R}^p$ is an unknown parameter. We can interpret this sort of model as follows. The columns of $\boldsymbol{X}$ can denote a basis for representing the natural parameter $\boldsymbol{\theta} \in \mathbb{R}^n$ of the Poisson process. If $p < n$, then the basis spans a subspace of $\mathbb{R}^n$ of dimension at most $p$.

   (a) Express the distribution of $\boldsymbol{y}$ in the standard exponential family form and identify the natural parameter $\boldsymbol{\theta} \in \mathbb{R}^n$. How is $\boldsymbol{\theta}$ related to $\boldsymbol{\lambda} = \{\lambda_i\}_{i=1}^n$.

   (b) For a $p < n$ of your choice, pick a basis $\boldsymbol{X}$ and generate data according to the Poisson model with $\boldsymbol{\theta} = \boldsymbol{X}\boldsymbol{w}$. You can use any standard software for your experiment, and you can pick $\boldsymbol{w}$ as you like. For example, you could take $\boldsymbol{X}$ to be a basis for linear functions of the form $\theta_i = w_1\, i + w_2$, $i = 1, \ldots, n$.

   (c) If we use a basis for linear functions to represent $\boldsymbol{\theta}$, then what type of function is the canonical parameter $\boldsymbol{\lambda}$?

   (d) Now use the GLM to find the MLE of $\boldsymbol{\theta}$. Transform this MLE to obtain an estimate of $\boldsymbol{\lambda}$. There are built-in routines from finding the MLE in most standard software packages, usually under a name like `glmfit`.

# Lecture 11: Gradient Descent

The least squares optimization problem

$$\min_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2$$

has the solution $\widehat{\boldsymbol{w}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$, when $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ is full-rank. In general, inverting the $d \times d$ matrix $(\boldsymbol{X}^T\boldsymbol{X})^{-1}$ requires $O(d^3)$ floating point operations, which can be demanding both in terms of time and space if $d$ is large. Iterative solvers can be used to obtain good approximations to the solution at a lower cost. The Landweber iteration is given by

$$\boldsymbol{w}_{t+1} \;=\; \boldsymbol{w}_t \;+\; \tau\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}_t)\ , \ \text{ for } t = 0, 1, \ldots \text{ for some } \tau > 0.$$

This is equivalent to a gradient descent method, which we will examine in these notes. This iteration requires matrix multiplications $\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}_t$ at each step. If $d$ and $n$, the number of training examples is large, then even this iterative method can be prohibitive. In recent years, dataset sizes have grown faster than computer speeds, and consequently large-scale machine learning is limited by computing time rather than the sample size. Incremental versions of gradient descent, that process just one sample at each step rather than the whole dataset, are increasingly useful because they are scalable to extremely large datasets and problem sizes. Incremental gradient descent algorithms are also readily extended to handle regularized methods such as ridge regression and lasso, and a variety of loss functions including the hinge loss. Such algorithms are the focus of this note.

## 11.1. Gradient Descent

Suppose we are given a training set $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ and consider the least squares problem

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}\in\mathbb{R}^d} \sum_{i=1}^n (y_i - \boldsymbol{w}^T\boldsymbol{X}_i)^2. \tag{11.1}$$

If $\boldsymbol{X}$ is full-rank, then a closed-form solution exists

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}\in\mathbb{R}^d} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}\ . \tag{11.2}$$

An alternative to the matrix-inverse approach is to minimize the squared error using gradient descent. This requires computing the gradient of the squared error, which is $-2\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})$. Note that the gradient is zero at the optimal solution, so the optimal $\boldsymbol{w}^*$ is the solution to the normal equations $\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} = \boldsymbol{X}^T\boldsymbol{y}$.

To gain a further insight, consider the gradient descent algorithm. Starting with an initial weight vector $\boldsymbol{w}_0$ (e.g., $\boldsymbol{w}_0 = 0$), gradient descent iterates the following step for $t = 1, 2, \ldots$

$$\begin{aligned} \boldsymbol{w}_t &= \boldsymbol{w}_{t-1} - \frac{1}{2}\gamma\left(\nabla\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2\right)\big|_{\boldsymbol{w}=\boldsymbol{w}_{t-1}} \\ &= \boldsymbol{w}_{t-1} + \gamma\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}_{t-1}) \end{aligned}$$

where $\gamma > 0$ is a step-size. Note that the algorithm takes a step in the negative gradient direction (i.e., ‘downhill’). The choice of the step size is crucial. If the steps are too large, then the algorithm may diverge. If they are too small, then convergence may take a long time. We can understand the effect of the step size as follows. Note that we can write the iterates as

$$\begin{aligned} \boldsymbol{w}_t &= \boldsymbol{w}_{t-1} + \gamma(\boldsymbol{X}^T y - \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}_{t-1}) \\ &= \boldsymbol{w}_{t-1} + \gamma\boldsymbol{X}^T\boldsymbol{X}((\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T y - \boldsymbol{w}_{t-1}) \\ &= \boldsymbol{w}_{t-1} - \gamma\boldsymbol{X}^T\boldsymbol{X}(\boldsymbol{w}_{t-1} - \boldsymbol{w}^*) \end{aligned}$$

Subtracting $\widehat{\boldsymbol{w}}$ from both sides gives us

$$\boldsymbol{v}_t = \boldsymbol{v}_{t-1} - \gamma \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{v}_{t-1}$$

where $\boldsymbol{v}_t = \boldsymbol{w}_t - \widehat{\boldsymbol{w}}$, $t = 1, 2, \ldots$ So we have

$$\begin{aligned}\boldsymbol{v}_t &= (\mathbf{I} - \gamma \boldsymbol{X}^T\boldsymbol{X})\boldsymbol{v}_{t-1} \\ &= (\mathbf{I} - \gamma \boldsymbol{X}^T\boldsymbol{X})^t\boldsymbol{v}_0\end{aligned}$$

Thus the sequence $\boldsymbol{v}_t \to 0$ if all the eigenvalues of $(\mathbf{I} - \gamma \boldsymbol{X}^T\boldsymbol{X})$ are less than 1. This holds[7] if $\gamma < 2\lambda^{-1}_{\mathtt{max}}(\boldsymbol{X}^T\boldsymbol{X})$. We see that the eigenvalues of $\boldsymbol{X}^T\boldsymbol{X}$ play a key role in gradient descent algorithms. Let $\alpha < 1$ denote the largest eigenvalue of $(\mathbf{I} - \gamma \boldsymbol{X}^T\boldsymbol{X})$. Then for any initialization $\boldsymbol{w}_0$, we have

$$\|\boldsymbol{w}_t - \widehat{\boldsymbol{w}}\| \leq \alpha^t \|\boldsymbol{w}_t - \widehat{\boldsymbol{w}}\| = O(\alpha^t)$$

which shows that the error of gradient descent converges exponentially in $t$. This is a consequence of the fact that the quadratic loss with a full rank $\boldsymbol{X}$ is a *strongly* Figure 11.1 depicts convex functions and discusses the notion of strong convexity.

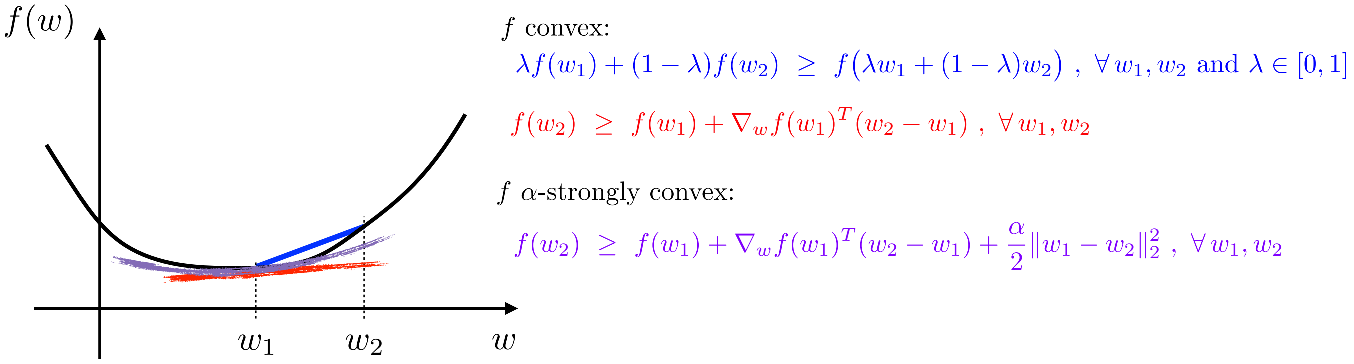

$f(w)$
$w_1$
$w_2$
$w$
$f$ convex:
$\lambda f(w_1) + (1-\lambda) f(w_2) \geq f(\lambda w_1 + (1-\lambda) w_2)$ , $\forall\, w_1, w_2$ and $\lambda \in [0,1]$
$f(w_2) \geq f(w_1) + \nabla_w f(w_1)^T (w_2 - w_1)$ , $\forall\, w_1, w_2$
$f$ $\alpha$-strongly convex:
$f(w_2) \geq f(w_1) + \nabla_w f(w_1)^T (w_2 - w_1) + \frac{\alpha}{2}\|w_1 - w_2\|_2^2$ , $\forall\, w_1, w_2$

Figure 11.1: A function $f$ is said to be convex if for all $\lambda \in [0,1]$ and $\boldsymbol{w}_1, \boldsymbol{w}_2$ we have $\lambda f(\boldsymbol{w}_1) + (1-\lambda) f(\boldsymbol{w}_2) \geq f(\lambda \boldsymbol{w}_1 + (1-\lambda)\boldsymbol{w}_2)$. If $f$ is differentiable, then an equivalent definition is that $f(\boldsymbol{w}_2) \geq f(\boldsymbol{w}_1) + \nabla f(\boldsymbol{w}_1)^T(\boldsymbol{w}_2 - \boldsymbol{w}_1)$. A function is *strictly* convex if both inequalities are strict (i.e., hold with $\geq$ replace by $>$). A function $f$ is said to be $\alpha$-strongly convex if $f(\boldsymbol{w}_2) \geq f(\boldsymbol{w}_1) + \nabla f(\boldsymbol{w}_1)^T(\boldsymbol{w}_2 - \boldsymbol{w}_1) + \frac{\alpha}{2}\|\boldsymbol{w}_2 - \boldsymbol{w}_1\|_2^2$ for all $\boldsymbol{w}_1, \boldsymbol{w}_2$. If $f$ is twice differentiable, then an equivalent definition is that $\nabla^2 f(\boldsymbol{w}) \succ \alpha \boldsymbol{I}$ for all $\boldsymbol{w}$.

## 11.2. Stochastic Gradient Descent for Least Squares

Stochastic gradient descent (SGD) is an incremental version of gradient descent, where the gradient is replaced by the gradient with respect to just one training example, rather than the entire training set. The SGD iterates are

$$\begin{aligned}\boldsymbol{w}_t &= \boldsymbol{w}_{t-1} - \frac{1}{2}\gamma\, \nabla (y_{i_t} - \boldsymbol{x}_{i_t}^T\boldsymbol{w})^2\big|_{\boldsymbol{w}=\boldsymbol{w}_{t-1}} \\ &= \boldsymbol{w}_{t-1} + \gamma(y_{i_t} - \boldsymbol{w}_{t-1}^T\boldsymbol{x}_{i_t})\boldsymbol{x}_{i_t} \ ,\end{aligned}$$

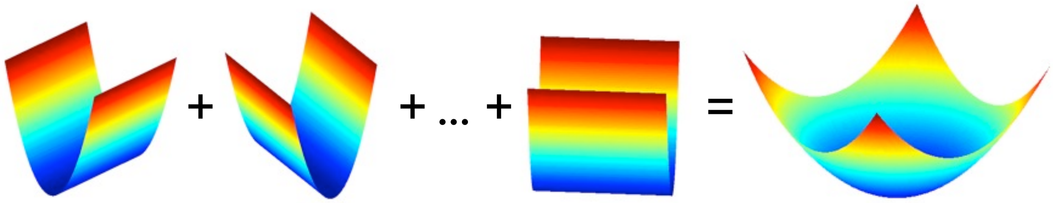
where $(y_{i_t}, \boldsymbol{x}_{i_t})$ is one of the training examples. There are several choices for the training example used at each step. For example, the algorithm can simply cycle through the training examples as it iterates $i_t = [t \bmod m] + 1$.

[7] Let $\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T = \boldsymbol{I} - \gamma\boldsymbol{X}^T\boldsymbol{X}$, the eigendecomposition, where $\boldsymbol{D}$ is diagonal and $\boldsymbol{U}$ is orthogonal (i.e., $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$). Note that $\boldsymbol{U}^T(\boldsymbol{I} - \gamma\boldsymbol{X}^T\boldsymbol{X})\boldsymbol{U} = \boldsymbol{I} - \gamma\boldsymbol{U}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{U}$, so $\boldsymbol{U}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{U}$ is diagonal and its diagonal elements are the non-negative eigenvalues of $\boldsymbol{X}^T\boldsymbol{X}$. Let $\lambda_i(\boldsymbol{X}^T\boldsymbol{X})$ denote the $i$th eigenvalue of $\boldsymbol{X}^T\boldsymbol{X}$ and note that the convergence condition becomes $|1 - \gamma\lambda_i(\boldsymbol{X}^T\boldsymbol{X})| < 1$.

This form of algorithm is usually referred to as incremental gradient descent. The term *stochastic* gradient descent selects each example uniformly at random from the full dataset. In this case, the average or *expected* value of the gradient is equal to the full gradient; i.e., $i_t \sim \text{uniform}(1, 2, \ldots, m)$ and

$$\mathbb{E}\big[\nabla(y_{i_t} - \boldsymbol{x}_{i_t}^T \boldsymbol{w})^2\big] \;=\; \frac{1}{n}\sum_{i=1}^{n} \nabla(y_i - \boldsymbol{x}_i^T \boldsymbol{w})^2 \,.$$

One can think of the SGD algorithm as considering each term in the sum of $(11.1)$ individually. Geometrically, for $t > d$ the complete sum of $(11.1)$ tends to look like a convex, quadratic bowl while each individual term is described by a degenerate quadratic in the sense that in all but 1 of the $d$ orthogonal directions, the function is flat. This concept is illustrated in Figure 11.2 with $f_t$ equal to $(y_{i_t} - \boldsymbol{x}_{i_t}^T \boldsymbol{w})^2$. Intuitively, each individual function $f_t$ only tells us about at most one dimension of the total $d$ so we should anticipate the algorithm will require $t \gg d$ iterations.



$$f_1(\mathbf{w}) \;+\; f_2(\mathbf{w}) + \;\ldots + \; f_T(\mathbf{w}) \;=\; \sum_{t=1}^{T} f_t(\mathbf{w})$$

Figure 11.2: The SGD algorithm can be thought of as considering each of the loss terms of $(11.1)$ individually. Because each term is "flat" in all but $1$ of the total $d$ directions, this implies that each term is convex but not *strongly* convex (see Figure 11.1). However, if $T > d$ we typically have that the complete sum *is* strongly convex which can be exploited to achieve faster rates of convergence.

## 11.3. Gradients and Subgradients

In general, the objective function $f$ might not be differentiable. For example, we might replace a squared $\ell_2$ regularizer with the $\ell_1$ norm, or change the squared error loss function to the hinge loss. The idea of a gradient can be extended to non-differentiable functions by introducing the notion of *subgradients*. Recall that for a convex function $f$ that is differentiable at $\boldsymbol{w}$, for all $\boldsymbol{u}$ we have

$$f(\boldsymbol{u}) \;\geq\; f(\boldsymbol{w}) \,+\, (\boldsymbol{u} - \boldsymbol{w})^T \nabla f(\boldsymbol{w}), \tag{11.3}$$

i.e., the gradient at $\boldsymbol{w}$ defines the slope of a tangent that lies below $f$, as depicted in Figure 11.1 . If $f$ is not differentiable at $\boldsymbol{w}$, we can write a similar inequality:

$$f(\boldsymbol{u}) \geq f(\boldsymbol{w}) + (\boldsymbol{u} - \boldsymbol{w})^T \boldsymbol{v} \tag{11.4}$$

where we call $\boldsymbol{v}$ a *subgradient*. The formal definition is below.

**Definition 11.3.1.** Any vector $\boldsymbol{v}$ that satisfies (11.4) is called a subgradient of $f$ at $\boldsymbol{w}$. The set of subgradients of $f$ at $\boldsymbol{w}$ is called the differential set and denoted $\partial f(\boldsymbol{w})$. So we write $\boldsymbol{v} \in \partial f(\boldsymbol{w})$.

If $f$ is differentiable at $\boldsymbol{w}$, then there is only one subgradient at $\boldsymbol{w}$, and it is equal to the gradient at $\boldsymbol{w}$. Subgradients exist even if $f$ is not differentiable at a certain point (e.g., the blue $\boldsymbol{x}$ on the left of Figure 11.1). For such cases, we can replace the gradient in SGD with any of the subgradients. We will use $\nabla f_t(\boldsymbol{w}_t)$ in the SGD iterations, with the understanding that if the gradient does not exist, then a subgradient (any one if there are multiple) are used instead.

## 11.4. Exercises

1. (a) Suppose $f : \mathbb{R} \to \mathbb{R}$ and $g : \mathbb{R}^d \to \mathbb{R}$ are both convex functions. In addition, suppose $f$ is non-decreasing. Prove $h(\boldsymbol{x}) := f \circ g(\boldsymbol{x}) = f(g(\boldsymbol{x}))$ is a convex function in $\boldsymbol{x}$.

   (b) Suppose $f : \mathbb{R} \to \mathbb{R}$ is a convex function and non-increasing, and $g : \mathbb{R}^d \to \mathbb{R}$ is a concave function. Prove $h(\boldsymbol{x}) := f \circ g(\boldsymbol{x}) = f(g(\boldsymbol{x}))$ is a convex function in $\boldsymbol{x}$.

   (c) Suppose $f : \mathbb{R} \to \mathbb{R}$ is a convex function, and $g : \mathbb{R}^d \to \mathbb{R}$ is an affine function, namely $g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$ for some known $\boldsymbol{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$. Prove $h(\boldsymbol{x}) := f \circ g(\boldsymbol{x}) = f(g(\boldsymbol{x}))$ is a convex function in $\boldsymbol{x}$.

2. (a) Derive an expression for the gradient descent iteration to minimize the ridge regression objective:

$$\min_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 \ + \ \lambda\|\boldsymbol{w}\|_2^2 \ , \ \lambda > 0 \ .$$

   (b) Derive subgradient for the absolute value function $f(x) = |x|$.

   (c) Derive an expression for a (sub)gradient descent iteration to minimize the Lasso criterion:

$$\min_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 \ + \ \lambda\|\boldsymbol{w}\|_1 \ , \ \lambda > 0 \ .$$

# Lecture 12: Analysis of Stochastic Gradient Descent

To study the convergence behavior of stochastic gradient descent, let us consider the more general problem

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}\in\mathbb{R}^d} \frac{1}{T}\sum_{t=1}^{T} \ell_t(\boldsymbol{w}) \tag{12.1}$$

where each $\ell_t : \mathbb{R}^d \to \mathbb{R}$ is a convex function (see Figure 11.1). In the context of LS, $\ell_t(\boldsymbol{w}) := (y_{i_t} - \boldsymbol{x}_{i_t}^T\boldsymbol{w})^2$, which is quadratic and hence convex in $\boldsymbol{w}$. In logistic regression with $y_i \in \{-1,+1\}$, $\ell_t(\boldsymbol{w}) := \log(1+\exp(-y_{i_t}\boldsymbol{x}_{i_t}^T\boldsymbol{w}))$, which is also convex in $\boldsymbol{w}$. The problem of (12.1) is known as an unconstrained online convex optimization program [27]. The general SGD iteration is given by

$$\boldsymbol{w}_{t+1} \;=\; \boldsymbol{w}_t - \gamma_t \nabla\ell_t(\boldsymbol{w}_t) \tag{12.2}$$

where $\gamma_t$ is a positive, non-increasing sequence of step sizes and the algorithm is initialized with some arbitrary $\boldsymbol{w}_1 \in \mathbb{R}^d$. Each term in the sum of (12.1) typically corresponds to the loss on a different training example. If the training set is finite, then the iteration process could make passes over the training set, say in a cyclical or randomized fashion. The following theorem characterizes the performance of the SGD iteration (12.2) assuming the gradients are bounded.

**Theorem 12.0.1** (see [27], constant stepsize)**.** *Let $\ell_t$ be convex and $\|\nabla\ell_t(\boldsymbol{w})\|_2 \le G$ for all $t$ and all $\boldsymbol{w}$. Further define the optimal value $\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}\in\mathbb{R}^d}\sum_{t=1}^T \ell_t(\boldsymbol{w})$. Using algorithm* (12.2) *with $\gamma_t = \gamma$ (constant stepsize) and arbitrary $\boldsymbol{w}_1 \in \mathbb{R}^d$ we have*

$$\frac{1}{T}\sum_{t=1}^{T}(\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{w}^*)) \;\le\; \frac{\|\boldsymbol{w}_1 - \boldsymbol{w}^*\|_2^2}{2\gamma T} + \frac{\gamma}{2}G^2 \qquad \textit{for all } T$$

Before proving the theorem, note that this is a strong result. It only uses the fact that the $\ell_t$ functions are convex and that the gradients of $\ell_t$ are bounded. In particular, it assumes *nothing* about how the $\ell_t$ functions relate to each other from time to time.

*Proof.* We begin by observing that

$$\begin{aligned}\|\boldsymbol{w}_{t+1} - \boldsymbol{w}^*\|_2^2 &= \|\boldsymbol{w}_t - \gamma\nabla\ell_t(\boldsymbol{w}_t) - \boldsymbol{w}^*\|_2^2 \\ &= \|\boldsymbol{w}_t - \boldsymbol{w}^*\|_2^2 - 2\gamma\nabla\ell_t(\boldsymbol{w}_t)^T(\boldsymbol{w}_t - \boldsymbol{w}^*) + \gamma^2\|\nabla\ell_t(\boldsymbol{w}_t)\|_2^2\end{aligned}$$

and after rearranging we have that

$$\nabla\ell_t(\boldsymbol{w}_t)^T(\boldsymbol{w}_t - \boldsymbol{w}^*) \le \frac{\|\boldsymbol{w}_t - \boldsymbol{w}^*\|_2^2 - \|\boldsymbol{w}_{t+1} - \boldsymbol{w}^*\|_2^2}{2\gamma} + \frac{\gamma}{2}G^2.$$

By the convexity of $\ell_t$ for all $t$, $\boldsymbol{w}_t$ we have $\ell_t(\boldsymbol{w}^*) - \ell_t(\boldsymbol{w}_t) \ge \nabla\ell_t(\boldsymbol{w}_t)^T(\boldsymbol{w}^* - \boldsymbol{w}_t)$. Thus, summing both sides of this equation from $t = 1$ to $T$ and dividing by $T$, we have

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{w}^*) \;&\le\; \frac{1}{T}\sum_{t=1}^{T}\left(\frac{\|\boldsymbol{w}_t - \boldsymbol{w}^*\|_2^2 - \|\boldsymbol{w}_{t+1} - \boldsymbol{w}^*\|_2^2}{2\gamma} + \frac{\gamma}{2}G^2\right) \\ &=\; \frac{\|\boldsymbol{w}_1 - \boldsymbol{w}^*\|_2^2}{2\gamma T} - \frac{\|\boldsymbol{w}_{T+1} - \boldsymbol{w}^*\|_2^2}{2\gamma T} + \frac{\gamma}{2}G^2 \\ &\le\; \frac{\|\boldsymbol{w}_1 - \boldsymbol{w}^*\|_2^2}{2\gamma T} + \frac{\gamma}{2}G^2\,.\end{aligned}$$

Notice that we use the fact that the sum above is a telescoping series. □

Note that in Theorem 12.0.1, as we increase the number of iterations ($T \to \infty$), the first term in the bound tends to zero but the second term does not. One way to ensure that we have a diminishing average error is to choose a constant stepsize that shrinks with the number of iterations we plan on doing. The following corollary gives this specialized result.

**Corollary 12.0.2.** *Let $\ell_t$ be convex and $\|\nabla \ell_t(\boldsymbol{w})\|_2 \leq G$ for all $t$ and all $\boldsymbol{w}$. Further define the optimal value $\boldsymbol{w}^* = \arg\min_{\boldsymbol{w} \in \mathbb{R}^d} \sum_{t=1}^{T} \ell_t(\boldsymbol{w})$. Using algorithm* (12.2) *with $\gamma_t = 1/\sqrt{T}$ (constant stepsize) and arbitrary $\boldsymbol{w}_1 \in \mathbb{R}^d$ we have*

$$\frac{1}{T}\sum_{t=1}^{T}(\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{w}^*)) \;\leq\; \frac{\|\boldsymbol{w}_1 - \boldsymbol{w}^*\|_2^2 + G^2}{2\sqrt{T}} \qquad \textit{for all } T$$

The proof of Corollary 12.0.2 is straightforward; simply substitute $\gamma = 1/\sqrt{T}$ into Theorem 12.0.1.

To illustrate how the results above apply to learning problems, consider the case where our goal is to solve $\min_{\boldsymbol{w}} \sum_{i=1}^{n} \ell_i(\boldsymbol{w})$, where $\ell_i(\boldsymbol{w})$ is the loss incurred by $\boldsymbol{w}$ on training example $(\boldsymbol{x}_i, y_i)$. For example, $\ell_i(\boldsymbol{w}) = \log(1 + \exp(-y_i \boldsymbol{w}^T \boldsymbol{x}_i))$ in the case of logistic regression. Let $\boldsymbol{w}^*$ denote a solution. This solution may not be unique if the overall sum of losses is not strictly convex. This is one reason that we measure the difference between objective values rather than the weight vectors themselves. To apply the theory above, let $T = kn$ for some positive integer $k \geq 1$ and let $\ell_t(\boldsymbol{w}) = \ell_{i_t}$, with $i_t = 1, 2, \ldots, n, 1, 2, \ldots, n, \ldots$. That is, let $i_t$ simply make cyclic passes over the training set. In this case, we have

$$\frac{1}{T}\sum_{t=1}^{T}(\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{w}^*)) \;=\; \frac{1}{T}\sum_{t=1}^{T}\ell_{i_t}(\boldsymbol{w}_t) - \frac{1}{n}\sum_{i=1}^{n}\ell_i(\boldsymbol{w}^*)$$

Now let us apply Corollary 12.0.2, which tells us for large $T$ (i.e., large $k$ in this setting), the difference above tends to zero. Since $\boldsymbol{w}^*$ minimizes the sum of losses, this shows that $\frac{1}{T}\sum_{t=1}^{T}\ell_{i_t}(\boldsymbol{w}_t)$ tends to the global minimum. This implies that $\frac{1}{n}\sum_{i=1}^{n}\ell_i(\boldsymbol{w}_t) \to \frac{1}{n}\sum_{i=1}^{n}\ell_i(\boldsymbol{w}^*)$ as $t$ grows.

Using a very small but constant stepsize, Corollary 12.0.2, as in may lead to slow initial convergence. One way around this is to use a *diminishing* stepsize. For technical reasons that will be come clear in our analysis of the dimishing stepsize case, we will modify the algorithm as follows. Assume that the solution $\boldsymbol{w}^*$ satisfies $\|\boldsymbol{w}^*\| \leq B$ for some constant $B$. This is a reasonable assumption in practice; it just says that the solution vector isn't arbitrarily large. Now at each iteration $t$ we do two things:

$$\begin{aligned} &1. \quad \boldsymbol{v}_{t+1} = \boldsymbol{w}_t - \gamma_t \nabla \ell_t(\boldsymbol{w}_t) \\ &2. \quad \boldsymbol{w}_{t+1} = \begin{cases} \boldsymbol{v}_{t+1} & \text{if } \|\boldsymbol{v}_{t+1}\| \leq B \\ \frac{B\,\boldsymbol{v}_{t+1}}{\|\boldsymbol{v}_{t+1}\|} & \text{otherwise} \end{cases} \end{aligned} \qquad (12.3)$$

The first step is the gradient descent, the second is a projection step that ensures the weight vectors always satisfy $\|\boldsymbol{w}_t\| \leq B$, which we assume the solution $\boldsymbol{w}^*$ must also satisfy. We will show that this algorithm, with $\gamma_t = \frac{1}{\sqrt{t}}$, satisfies a bound similar to the one we derived for $\gamma_t = \frac{1}{\sqrt{T}}$. Before giving the result, we will require an auxiliary result that will help us find bounds.

**Lemma 12.0.3.** *For any $t = 1, 2, \ldots,$ the following inequality holds:*

$$\sum_{t=1}^{T}\frac{1}{\sqrt{t}} \;\leq\; 2\sqrt{T}$$

*Proof.* This can be shown by thinking of the sum as the sums of areas of several rectangles of width 1 and height $1/\sqrt{t}$. The sum is bounded above by an integral:

$$\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \;\leq\; 1 + \int_{1}^{T} \frac{\mathrm{d}t}{\sqrt{t}} \;=\; 1 + 2(\sqrt{T} - 1) \;=\; 2\sqrt{T} - 1 \;\leq\; 2\sqrt{T}$$

□

We are now ready to prove the bound result with diminishing stepsize.

**Theorem 12.0.4** (diminishing stepsize)**.** *Let $\ell_t$ be convex and assume that $\|\nabla \ell_t(\boldsymbol{w})\|_2 \leq G$ for all $t$ and all $\boldsymbol{w}$ and that $\|\boldsymbol{w}^*\| \leq B$. Further define the optimal value $\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}\in\mathbb{R}^d} \sum_{t=1}^{T} \ell_t(\boldsymbol{w})$. Using algorithm* (12.3) *with $\gamma_t = 1/\sqrt{t}$ (diminishing stepsize) and arbitrary $\boldsymbol{w}_1 \in \mathbb{R}^d$ we have*

$$\frac{1}{T}\sum_{t=1}^{T} (\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{w}^*)) \;\leq\; \frac{2B^2 + G^2}{\sqrt{T}} \qquad \textit{for all } T$$

*Proof.* The first part of the proof is very similar to that of Theorem 12.0.1. First observe that $\|\boldsymbol{w}_t - \boldsymbol{w}^*\| \leq \|\boldsymbol{v}_t - \boldsymbol{w}^*\|$ for all $t$, since the projection step never increases the error. Therefore,

$$\begin{aligned}
\|\boldsymbol{w}_{t+1} - \boldsymbol{w}^*\|_2^2 &\leq \|\boldsymbol{v}_{t+1} - \boldsymbol{w}^*\|_2^2 \\
&= \|\boldsymbol{w}_t - \gamma \nabla \ell_t(\boldsymbol{w}_t) - \boldsymbol{w}^*\|_2^2 \\
&= \|\boldsymbol{w}_t - \boldsymbol{w}^*\|_2^2 - 2\gamma_t \nabla \ell_t(\boldsymbol{w}_t)^T(\boldsymbol{w}_t - \boldsymbol{w}^*) + \gamma^2 \|\nabla \ell_t(\boldsymbol{w}_t)\|_2^2 \\
&\leq \|\boldsymbol{w}_t - \boldsymbol{w}^*\|_2^2 - 2\gamma_t \nabla \ell_t(\boldsymbol{w}_t)^T(\boldsymbol{w}_t - \boldsymbol{w}^*) + \gamma_t^2 G^2 \,.
\end{aligned}$$

Proceed as in Theorem 12.0.1 but don't divide by $T$ to obtain:

$$\sum_{t=1}^{T} \ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{w}^*) \quad\leq\quad \sum_{t=1}^{T} \left( \frac{\|\boldsymbol{w}_t - \boldsymbol{w}^*\|_2^2 - \|\boldsymbol{w}_{t+1} - \boldsymbol{w}^*\|_2^2}{2\gamma_t} + \frac{\gamma_t}{2} G^2 \right)$$

Rearranging, substituting $\gamma_t = 1/\sqrt{t}$, and applying Lemma 12.0.3, we obtain:

$$\begin{aligned}
&\sum_{t=1}^{T} \ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{w}^*) \\
&\qquad \leq \left( \frac{\|\boldsymbol{w}_1 - \boldsymbol{w}^*\|_2^2}{2\gamma_1} - \frac{\|\boldsymbol{w}_{T+1} - \boldsymbol{w}^*\|_2^2}{2\gamma_{T+1}} \right) + \sum_{t=2}^{T} \frac{\|\boldsymbol{w}_t - \boldsymbol{w}^*\|_2^2}{2} \left( \frac{1}{\gamma_t} - \frac{1}{\gamma_{t-1}} \right) + G^2 \sum_{t=1}^{T} \frac{\gamma_t}{2} \\
&\qquad \leq \frac{\|\boldsymbol{w}_1 - \boldsymbol{w}^*\|_2^2}{2} + \sum_{t=2}^{T} \frac{\|\boldsymbol{w}_t - \boldsymbol{w}^*\|_2^2}{2} \left( \sqrt{t} - \sqrt{t-1} \right) + G^2 \sum_{t=1}^{T} \frac{1}{2\sqrt{t}} \\
&\qquad \leq 2B^2 \sum_{t=1}^{T} \left( \sqrt{t} - \sqrt{t-1} \right) + G^2 \sqrt{T} \\
&\qquad \leq \sqrt{T} \big( 2B^2 + G^2 \big) \,,
\end{aligned}$$

where we used the fact that $\|\boldsymbol{w}_t - \boldsymbol{w}^*\| \leq \|\boldsymbol{w}_t\| + \|\boldsymbol{w}^*\| \leq 2B$ and that $\sum_{t=1}^{T} \big( \sqrt{t} - \sqrt{t-1} \big)$ is a telescoping series. Divide both sides by $T$ completes the proof. □

This agnostic approach to the functions $\ell_t$ allow us to apply the theorem to analyzing SGD for least squares. In this case we have $\ell_t(\boldsymbol{w}) = (y_{i_t} - \boldsymbol{w}^T\boldsymbol{x}_{i_t})^2$ so that $\nabla\ell_t(\boldsymbol{w}) = -2(y_{i_t} - \boldsymbol{w}^T\boldsymbol{x}_{i_t})\boldsymbol{x}_{i_t}$ we see that

$$\|\nabla\ell_t(\boldsymbol{w})\| \;\le\; 2\left(\|y_{i_t}\boldsymbol{x}_{i_t}\| \;+\; \|\boldsymbol{w}_t\boldsymbol{x}_{i_t}^T\boldsymbol{x}_{i_t}\|\right) \;\le\; 2\left(\max_i |y_i|\|\boldsymbol{x}_i\| \;+\; \|\boldsymbol{w}_t\|\max_i \|\boldsymbol{x}_i\|^2\right).$$

To further simplify this expression, suppose we are in the classification setting where $|y_i| = 1$ and assume that the features are bounded $\|\boldsymbol{x}_i\| \le C$. Then we have

$$\|\nabla\ell_t(\boldsymbol{w})\| \;\le\; 2C\left(1 + BC^2\right) \;=:\; G.$$

Plugging this into the theorem, we have

$$\frac{1}{T}\sum_{t=1}^{T}(\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{w}^*)) \le \frac{2B^2 + 2C^2(1+BC^2)^2}{\sqrt{T}}$$

for all $T$. The takeaway message is that if we assume *nothing* about the features, apart from $\|\boldsymbol{x}_i\| \le B$, then our residual sum of squared errors converges to the residual using $\boldsymbol{w}^*$ at a rate proportional to $1/\sqrt{T}$.

## 12.1. Exercises

1. Derive an expression for an SGD iteration to minimize the sum of logistic losses:

$$\min_{\boldsymbol{w}} \sum_{i=1}^{n} \log(1 + \exp(-y_i\boldsymbol{x}_i^T\boldsymbol{w})).$$

2. Prove that the logistic loss function $\log(1 + \exp(-z))$ is strictly convex in $z \in \mathbb{R}$.

3. Derive an expression for an SGD iteration to minimize the sum of hinge losses:

$$\min_{\boldsymbol{w}} \sum_{i=1}^{n} (1 - y_i\boldsymbol{x}_i^T\boldsymbol{w})_+ .$$

4. Analyze stochastic GD (SGD) for Perceptron Algorithm. Let $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ be a training set with $y_i \in \{-1, +1\}$ and $\|\boldsymbol{x}_i\| \le B$ for all $i$. Assume that there exists a $\boldsymbol{w}$ satisfying the "margin" condition $y_i\boldsymbol{w}^T\boldsymbol{x}_i \ge 1$ for $i = 1, \ldots, n$. Let $\boldsymbol{w}^*$ be a vector that has the minimum norm among vectors that satisfy the margin condition. Define the function

$$f(\boldsymbol{w}) \;=\; \max_i(1 - y_i\boldsymbol{w}^T\boldsymbol{x}_i),$$

which is margin (distance) of the classification boundary to the nearest training example (assuming $\boldsymbol{w}$ correctly classifies all the points (i.e., $y_i\boldsymbol{w}^T\boldsymbol{x}_i > 0$ for all $i$). This can be viewed as a loss function (recall the hinge loss is $\max(0, 1 - y_i\boldsymbol{w}^T\boldsymbol{x}_i)$, essentially the same thing).

   (a) Show that $\min_{\boldsymbol{w}:\|\boldsymbol{w}\|\le\|\boldsymbol{w}^*\|} f(\boldsymbol{w}) = 0$ and show that any $\boldsymbol{w}$ that satisifies $f(\boldsymbol{w}) < 1$ yields a linear classifier that correctly classifies the training data.

   (b) Show how to calculate a subgradient of $f$.

   (c) Analyze the performance of SGD in this setting, using the theoretical bounds and analysis discussed in class.

# Lecture 13: Bayesian Inference

Consider a family of probability distributions indexed by a parameter $\boldsymbol{\theta}$. The parameter may be a scalar or multidimensional. In general, we will denote a family of distributions by $p(x|\boldsymbol{\theta})$, $\boldsymbol{\theta} \in \boldsymbol{\theta}$, where $\boldsymbol{\theta}$ denotes the set of all possible values the parameter can take. Viewed as a function of $\boldsymbol{\theta}$ with $\boldsymbol{x}$ fixed, $\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{\theta})$ is called the *likelihood* (function) of $\boldsymbol{\theta}$. Let $p(\boldsymbol{\theta})$ be a *prior probability* distribution on $\boldsymbol{\theta}$. This distribution is a models how specific values of $\boldsymbol{\theta}$ are more or less probable *a priori*, that is before observing the data $\boldsymbol{x}$. Bayes' Rule allows us to compute the *posterior* distribution of $\boldsymbol{\theta}$:

$$p(\boldsymbol{\theta}|\boldsymbol{x}) \;=\; \frac{p(\boldsymbol{x}|\boldsymbol{\theta})\, p(\boldsymbol{\theta})}{p(\boldsymbol{x})}$$

The posterior distribution reflects the probability of different values of $\boldsymbol{\theta}$ in light of the observed data $\boldsymbol{x}$.

## 13.1. Back to the Basics

Recall the simple setting where $\theta$ takes just one of two values, say $0$ and $1$. This corresponds to two probability models for $\boldsymbol{x}$, $p_1(\boldsymbol{x}) = p(\boldsymbol{x}|\theta = 1)$ and $p_0(\boldsymbol{x}) = p(\boldsymbol{x}|\theta = 0)$. Deciding which model is better for observed data $\boldsymbol{x}$ is a simple binary hypothesis test. The decision rule that minimizes the probability of error is to decide $\widehat{\theta} = 1$ if $p(\theta = 1|\boldsymbol{x}) > p(\theta = 0|\boldsymbol{x})$ and decide $\widehat{\theta} = 0$ otherwise (and flip a fair coin if $p(\theta = 1|\boldsymbol{x}) = p(\theta = 0|\boldsymbol{x})$). From Bayes' Rule, we know that this is equivalent to the likelihood ratio test

$$\frac{p_1(\boldsymbol{x})}{p_0(\boldsymbol{x})} \overset{H_1}{\underset{H_0}{\gtrless}} \frac{p(\theta = 0)}{p(\theta = 1)}$$

where $p(\theta = 1) = 1 - p(\theta = 0)$ is the prior probability that $\theta = 1$. We see here how the optimal decision about which model is best for the data hinges on our prior knowledge about $\theta$. Bayesian inference is the natural extension of this idea to more general settings, e.g., where $\theta$ may be a continuous parameter. This is analogous to the idea of maximum likelihood, but now incorporating prior knowledge about $\theta$ that shapes how we decide which model $p(\boldsymbol{x}|\theta)$ is best for a given observation $\boldsymbol{x}$.

To view this in the general Bayesian framework, consider the following example. Suppose that $x_i|\theta \overset{\text{iid}}{\sim} \mathcal{N}(\theta, 1)$, $i = 1, \ldots, n$; i.e., a Gaussian likelihood function. Let the prior probability model be $p(\theta) = \frac{1}{2}\delta(\theta) + \frac{1}{2}\delta(\theta - \mu)$, where $\delta(\theta - \mu)$ is the Dirac delta function at $\mu$. This means the prior is zero at all points except $0$ or $\mu$. The MLE of $\theta$ is $\widehat{\theta} = \frac{1}{n}\sum_{i=1}^{n} x_i$. However, the prior allows for only two possible probability models, Gaussian with mean $\theta = 0$ or $\theta = \mu$ (since the prior is zero at all other values of $\theta$). This implies that the posterior $p(\theta|\boldsymbol{x})$ is also nonzero only at these two points. The best model is the one with larger posterior probability. This is equivalent to the simple hypothesis testing problem above.

## 13.2. Posterior Distributions and Decisions

Bayesian inference methods have three core elements.

**Prior Distribution:** $p(\boldsymbol{\theta})$ encodes prior knowledge about which values of $\boldsymbol{\theta}$ are more plausible models for the inference problem at hand. In essence, $p(\boldsymbol{\theta})$ is a non-negative weighting function over the set $\Theta$ of all values that $\boldsymbol{\theta}$ may take. Prior knowledge can come in form of physical reasoning, desired constraints or regularity properties, or beliefs about $\boldsymbol{\theta}$.

**Likelihood Function:** The likelihood function is $p(\boldsymbol{x}|\boldsymbol{\theta})$ viewed as a function of $\boldsymbol{\theta}$. Maximizing this function itself is maximum likelihood estimation.

**Posterior Distribution:** The posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{x}) \propto p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})$ combines prior knowledge with information derived from the data $\boldsymbol{x}$. Maximizing the posterior with respect to $\boldsymbol{\theta}$ produces the *Maximum a Posteriori* (MAP) estimator. Note that $\log p(\boldsymbol{\theta}|\boldsymbol{x}) = \log p(\boldsymbol{x}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) +$ constant, so the log-posterior is just a linear combination of the log-likelihood and the log-prior. From this point of view, $-\log p(\boldsymbol{\theta})$ can be view as a regularization term in the estimation process.

Bayesian inference methods consider the full posterior distribution, since it tells the probability of any value of $\boldsymbol{\theta}$. Often, we are interested in a specific estimate of $\boldsymbol{\theta}$. The two most common estimators are

**Maximum a Posteriori Estimator:** $\widehat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\boldsymbol{x})$, the mode of $p(\boldsymbol{\theta}|\boldsymbol{x})$

**Posterior Mean Estimator:** $\widehat{\boldsymbol{\theta}}_{\text{PM}} = \int \boldsymbol{\theta}\, p(\boldsymbol{\theta}|\boldsymbol{x})\, d\boldsymbol{\theta}$, the mean of $p(\boldsymbol{\theta}|\boldsymbol{x})$

## 13.3. Example: Temperature Estimation

Consider a family of probability distributions indexed by a parameter $\theta$. The parameter may be a scalar or multidimensional. In general, we will denote a family of density functions by $p(x|\theta)$, $\theta \in \Theta$, where $\Theta$ denotes the set of all possible values the parameter can take. Recall that the Maximum Likelihood Estimator (MLE) is

$$\widehat{\theta}_{\text{MLE}} = \arg\max_{\theta\in\Theta} p(x|\theta)$$

where $p(x|\theta)$ as a function of $x$ with the parameter $\theta$ fixed is the probability density function or mass function. Viewing $p(x|\theta)$ as a function of $\theta$ with $x$ fixed is called the "likelihood function."

The MLE procedure implicitly treats all $\theta \in \Theta$ are equally plausible, but this might not be reasonable in all situations. For example, suppose that $x$ is a temperature measurement somewhere on the surface of the earth and $\theta$ is the mean temperature at that location. We know that temperatures outside the range $-100\,^\circ C \le x \le 100\,^\circ C$ have never been observed anywhere on earth, so we can safely restrict $\theta$ to the interval $[-100, 100]$; i.e., $\Theta = [-100, 100]$. However, we also know that it is much more probably that temperatures will be in the range $[-30, 30]$, than below $-30$ or above $+30$. So perhaps it makes sense to "weight" different temperature ranges more than others, since they are simply more probable *a priori*; i.e., before making any measurements.

One way to weight the set $\Theta$ to reflect prior knowledge of the plausibility of different $\theta$ (and hence different $p(x|\theta)$ models) is to place a *prior probability distribution* on $\Theta$. Let $p(\theta)$ denote such a distribution. We can view $p(\theta)$ as a non-negative weighting function over the set $\Theta$, and use this weighting to modify our optimization as follows

$$\widehat{\theta}_{\text{MAP}} = \arg\max_{\theta\in\Theta} p(x|\theta)p(\theta)$$

This optimization tends to prefer solutions where $p(\theta)$ is large. This is called the *Maximum a Posteriori* (MAP) estimator. The name MAP derives from the fact that

$$\begin{aligned}\max_{\theta\in\Theta} p(x|\theta)p(\theta) &= \max_{\theta\in\Theta} \frac{p(x|\theta)p(\theta)}{p(x)} \\ &= \max_{\theta\in\Theta} p(\theta|x)\end{aligned}$$

and $p(\theta|x)$ is called the *posterior distribution* of $\theta$ give $x$.

Continuing with the temperature example above, Suppose the prior is

$$p(\theta) \;=\; \begin{cases} \frac{1}{130} & -30 \le \theta \le 30 \\[2ex] \frac{1}{260} & 30 < |\theta| \le 100 \end{cases}$$

In other words, the prior places twice the probability on values in the range $[-30, 30]$. Then $|\widehat{\theta}_{\text{MAP}}| \le |\widehat{\theta}_{\text{MLE}}|$. This is easy to verify. Suppose that the likelihood achieves its maximum on the interval $[-30, 30]$. Then so does the posterior density function. If the likelihood is maximized outside this region, then the maximum of the posterior may be at the same point or at some other point in the range $[-30, 30]$, due to the scaling factor of 2 in this range. So in this case the prior *biases* the estimator towards lower temperatures. Although bias is usually undesirable, the MAP estimator may also have a lower variance (since it may reduce the magnitude of the estimate), and thus the overall mean-squared error of the MAP estimator may be smaller than that of the MLE. We will look at a more concrete example of this *bias-variance* tradeoff later in this note.

## 13.4. Example: Twitter Monitoring

Suppose that we are monitoring Twitter for mentions of a particular topic or hashtag. Each hour for $n$ hours we count how many tweets were posted on the topic. Denote these counts by $\boldsymbol{x} = (x_1, \ldots, x_n)$. We will assume the counts are i.i.d. The Poisson probability distribution is a reasonable model for these data:

$$p(\boldsymbol{x}|\theta) \;=\; \prod_{i=1}^{n} e^{-\theta} \frac{\theta^{x_i}}{x_i!} \;,\; \theta > 0 \;.$$

The parameter $\theta$ is the mean of the Poisson distribution, and the MLE is given by

$$\widehat{\theta}_{\text{MLE}} \;=\; \frac{1}{n} \sum_{i=1}^{n} x_i \;.$$

Now imagine that we have some prior knowledge about whether the topic is hot and trending (e.g., probably a large $\theta$) or rare (i,e., small $\theta$). We can represent this knowledge in terms of an exponential prior distribution:

$$p(\theta) \;=\; \alpha e^{-\alpha\theta} \;,\; \alpha > 0 \;.$$

The larger the value of $\alpha$, the more quickly the prior density function decays away from 0. As $\alpha \to 0$, the prior tends to a uniform (flat) distribution. The posterior distribution is

$$p(\theta|\boldsymbol{x}) \;\propto\; \alpha e^{-\alpha\theta} \prod_{i=1}^{n} e^{-\theta} \frac{\theta^{x_i}}{x_i!} \;=\; \alpha \prod_{i=1}^{n} e^{-\theta(1+\alpha/n)} \frac{\theta^{x_i}}{x_i!}$$

and

$$-\log p(\theta|\boldsymbol{x}) \;=\; \sum_{i=1}^{n} \theta(1+\alpha/n) \;-\; x_i \log(\theta) \;+\; \text{constant} \;.$$

Minimizing this expression with respect to $\theta$ yield the MAP estimator

$$\widehat{\theta}_{\text{MAP}} \;=\; \frac{1}{(n+\alpha)} \sum_{i=1}^{n} x_i \;=\; \frac{n\,\widehat{\theta}_{\text{MLE}}}{n+\alpha} \;.$$

So we see that the MAP estimator is a "shrunken" version of the MLE (i.e., it is scaled down towards $0$ by a factor of $n/(n+\alpha)$). Notice that as the sample size $n$ grows, the MAP estimator converges to the MLE. This illustrates a general fact: the prior only plays a significant role if the sample size is relatively small.

This shrinkage may be desirable if we are only going to count for a few hours and we believe that the number of tweets on the topic will probably be relatively small. To understand this further, consider the bias-variance decomposition of the mean-squared error (MSE). If $\widehat{\theta}$ be denotes any estimator of the true value $\theta$, then

$$\begin{aligned} \text{MSE}(\widehat{\theta}) \;&:=\; \mathbb{E}[(\theta-\widehat{\theta})^2] \;=\; \mathbb{E}[(\theta-\mathbb{E}[\widehat{\theta}]+\mathbb{E}[\widehat{\theta}]-\widehat{\theta})^2] \\ &=\; \underset{\text{bias}}{(\theta-\mathbb{E}[\widehat{\theta}])^2} \;+\; \underset{\text{variance}}{\mathbb{E}[(\mathbb{E}[\widehat{\theta}]-\widehat{\theta})^2]} \end{aligned}$$

Note that cross-term $\mathbb{E}[(\theta-\mathbb{E}[\widehat{\theta}])(\mathbb{E}[\widehat{\theta}]-\widehat{\theta})] = (\theta-\mathbb{E}[\widehat{\theta}])\mathbb{E}[(\mathbb{E}[\widehat{\theta}]-\widehat{\theta})] = 0$. Now let us compute the bias and variance of each estimator above. For the MLE we have

$$\mathbb{E}[\widehat{\theta}_{\text{MLE}}] \;=\; \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[x_i] \;=\; \theta$$

and so it is unbiased. Since $\sum_{i=1}^{n} x_i$ is a sum of independent random variables, its variance is equal to the some of the variance of each $x_i$. The variance of $x_i \sim \text{Poisson}(\theta)$ is $\theta$. So the variance of the MLE is

$$\mathbb{V}[\widehat{\theta}_{\text{MLE}}] \;=\; \sum_{i=1}^{n}\mathbb{V}[x_i/n] \;=\; \frac{\theta}{n}$$

Since the MAP estimator is just a scaled version of the MLE, its mean and variance are

$$\begin{aligned} \mathbb{E}[\widehat{\theta}_{\text{MAP}}] \;&=\; \frac{n}{n+\alpha}\,\theta \\ \mathbb{V}[\widehat{\theta}_{\text{MAP}}] \;&=\; \left(\frac{n}{n+\alpha}\right)^2\frac{\theta}{n} \end{aligned}$$

So its variance is smaller than that of the MLE, by a factor of $\left(\frac{1}{1+\alpha/n}\right)^2$, but it is biased. The squared bias of the MAP estimator is

$$\begin{aligned} \left(\theta-\mathbb{E}[\widehat{\theta}_{\text{MAP}}]\right)^2 \;&=\; \left(\theta-\frac{n}{n+\alpha}\,\theta\right)^2 \\ &=\; \left(\frac{\alpha}{n+\alpha}\right)^2\theta^2 \end{aligned}$$

So the two estimators have the following MSEs

$$\begin{aligned} \text{MSE}(\widehat{\theta}_{\text{MLE}}) \;&=\; \frac{\theta}{n} \\ \text{MSE}(\widehat{\theta}_{\text{MAP}}) \;&=\; \left(\frac{\alpha}{n+\alpha}\right)^2\theta^2 \;+\; \left(\frac{n}{n+\alpha}\right)^2\frac{\theta}{n} \end{aligned}$$

Since we don't know $\theta$, it is impossible to determine which estimator has a lower MSE. However, we can gain some insight by minimizing this quantity. Setting the derivative with respect to $\alpha$ to zero yields the optimal value

$$\alpha^* \;=\; \frac{1}{\theta}\,,$$

which tends to $0$ as $\theta$ grows (i.e., if $\theta$ is very large, then we should use the MLE). So if our a priori expectation is that we will get about $\theta_{\text{guess}}$ tweets on the topic per hour, then we could set $\alpha$ using this value.

# 13.5. Multivariate Normal Distributions in Bayesian Inference

The multivariate Gaussian/normal (MVN) distribution holds a special place in Bayesian inference. The Gaussian distribution is, of course, one of the most important models of probabilistic uncertainty. The Central Limit Theorem tells us that the sum of many independent random effects tends to a Gaussian distribution. Its prominence in Bayesian inference owes largely to the fact that if both the likelihood and prior distribution are multivariate Gaussian, then so is the posterior distribution. In fact, the mean and covariance of the posterior can be computed by simple linear-algebraic operations applied to the means and covariances of the likelihood and prior. This led to important Bayesian methods like *Wiener* [25] and *Kalman* [17] filters, which are widely used in signal and image processing and control systems.

The following theorem demonstrates a special property of jointly MVN randon variables.

**Theorem 13.5.1.** *Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be jointly Gaussian random vectors, whose joint distribution can be expressed as*

$$\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} \sim \mathcal{N}\left( \begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy} \end{bmatrix} \right)$$

*Then the conditional distribution of $\boldsymbol{y}$ given $\boldsymbol{x}$ is*

$$\boldsymbol{y}|\boldsymbol{x} \sim \mathcal{N}\big(\boldsymbol{\mu}_y + \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_x), \boldsymbol{\Sigma}_{yy} - \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}\big).$$

*Proof.* Without loss of generality assume that $\boldsymbol{x}$ and $\boldsymbol{y}$ are zero-mean random vectors. Therefore

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}, \boldsymbol{y})}{p(\boldsymbol{x})} \ \propto \ \frac{|\boldsymbol{\Sigma}|^{-1}\text{exp}\Big\{ -\frac{1}{2}\big[\boldsymbol{x}^T \quad \boldsymbol{y}^T\big]\boldsymbol{\Sigma}^{-1}\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix}\Big\}}{|\boldsymbol{\Sigma}_{xx}|^{-1}\text{exp}\big\{ -\frac{1}{2}\boldsymbol{x}^T\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{x}\big\}}$$

where

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy} \end{bmatrix}.$$

To simplify the formula we need to determine $\boldsymbol{\Sigma}^{-1}$. The inverse can be written as:

$$\begin{bmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{\Sigma}_{xx}^{-1} & 0 \\ \boldsymbol{0} & 0 \end{bmatrix} + \begin{bmatrix} -\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy} \\ \boldsymbol{I} \end{bmatrix} \boldsymbol{Q}^{-1}\big[-\boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1} \quad \boldsymbol{I}\big]$$

where

$$\boldsymbol{Q} := \boldsymbol{\Sigma}_{yy} - \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}.$$

This formula for the inverse is easily verified by multiplying it by $\boldsymbol{\Sigma}$ to get the identity matrix. Substituting the inverse into $p(\boldsymbol{y}|\boldsymbol{x})$ yields

$$p(\boldsymbol{y}|\boldsymbol{x}) \ \propto \ |\boldsymbol{Q}|^{-1}\text{exp}\Big\{ -\frac{1}{2}(\boldsymbol{y} - \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{x})^T\boldsymbol{Q}^{-1}(\boldsymbol{y} - \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{x})\Big\}$$

which shows that $\boldsymbol{y}|\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{x}, \boldsymbol{Q})$. For the general case when $\mathbb{E}[\boldsymbol{x}] = \boldsymbol{\mu}_x$ and $\mathbb{E}[\boldsymbol{y}] = \boldsymbol{\mu}_y$ then

$$(\boldsymbol{y} - \boldsymbol{\mu}_y)|(\boldsymbol{x} - \boldsymbol{\mu}_x) \sim \mathcal{N}(\boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_x), \boldsymbol{Q})$$

$$\boldsymbol{y}|\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_y + \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_x), \boldsymbol{Q})$$

□

To apply this result to Bayesian inference, consider the likelihood $\boldsymbol{x}|\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\Sigma})$, for some covariance $\boldsymbol{\Sigma}$. Assume a MVN prior for $\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\theta,\theta})$. Since $\boldsymbol{x} = \boldsymbol{\theta} + \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, the marginal distribution is $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma} + \boldsymbol{\Sigma}_{\theta,\theta})$ and the cross-covariance between $\boldsymbol{x}$ and $\boldsymbol{\theta}$ is just the variance common to both, i.e., $\boldsymbol{\Sigma}_{x,\theta} = \boldsymbol{\Sigma}_{\theta,\theta}$. The result above tells us that

$$\boldsymbol{\theta}|\boldsymbol{x} \ \sim \ \mathcal{N}\Big(\boldsymbol{\Sigma}_{\theta,\theta}(\boldsymbol{\Sigma} + \boldsymbol{\Sigma}_{\theta,\theta})^{-1}\boldsymbol{x}\, ,\, \boldsymbol{\Sigma}_{\theta,\theta} - \boldsymbol{\Sigma}_{\theta,\theta}(\boldsymbol{\Sigma} + \boldsymbol{\Sigma}_{\theta,\theta})^{-1}\boldsymbol{\Sigma}_{\theta,\theta}\Big).$$

The posterior mean and MAP estimator are the same

$$\widehat{\boldsymbol{\theta}} \ = \ \boldsymbol{\Sigma}_{\theta,\theta}(\boldsymbol{\Sigma} + \boldsymbol{\Sigma}_{\theta,\theta})^{-1}\boldsymbol{x}\ ,$$

which is referred to as the *Wiener filter* in the signal processing literature.

## 13.6. Bayesian Linear Modeling

Recall the idea of a Generalized Linear Model (GLM) is to model $p(y|\boldsymbol{x})$ in terms of a linear function (i.e., weighted combination) of the features. Specifically, given a chosen probability distribution for $y$, we set the natural parameter to be $\theta = \boldsymbol{w}^T\boldsymbol{x}$, and so we will write this model as $p(y|\boldsymbol{w}^T\boldsymbol{x})$. GLMs have the special property that $p(y|\boldsymbol{w}^T\boldsymbol{x}) \propto \exp(-\ell(y, \boldsymbol{w}^T\boldsymbol{x}))$ for a convex loss function $\ell$. The Bayesian approach here is to place a prior probability model on the weights $\boldsymbol{w}$. Let $p(\boldsymbol{w})$ denote the prior. The posterior

$$p(\boldsymbol{w}|\boldsymbol{x}, y) \propto p(\boldsymbol{w})\exp(-\ell(y, \boldsymbol{w}^T\boldsymbol{x}))\ .$$

Given data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, the MAP estimator of $\boldsymbol{w}$ is the solution to

$$\min_{\boldsymbol{w}} \sum_{i=1}^n \ell(y_i, \boldsymbol{w}^T\boldsymbol{x}_i) \ - \ \log p(\boldsymbol{\theta})\ .$$

For example, priors of the form $p(\boldsymbol{w}) \propto \exp\big(-\frac{\lambda}{2}\|\boldsymbol{w}\|_2^2\big)$ or $p(\boldsymbol{w}) \propto \exp(-\lambda\|\boldsymbol{w}\|_1)$ lead to “ridge" or “lasso" regularization, respectively.

## 13.7. Exercises

1. Consider a sequence of independent coin tosses (i.i.d. Bernoulli random variables)

$$\boldsymbol{x} = [x_1, \ldots, x_n]^T$$

Let $x_i = 1$ denote “heads” and $x_i = 0$ “tails,” and $\mathbb{P}(x_i = 1) = \theta$, $\mathbb{P}(x_i = 0) = 1 - \theta$. If we observe $\boldsymbol{x}$, then the likelihood function is

$$p(\boldsymbol{x}|\theta) \ = \ \theta^{s(\boldsymbol{x})}(1-\theta)^{N-s(\boldsymbol{x})}$$

where $s(\boldsymbol{x}) = \sum_{n=1}^n x_i$. Suppose we are interested in estimating $\theta$ given $\boldsymbol{x}$. Let’s take a Bayesian approach. *A priori* we *believe* that $\theta \approx 1/2$ (*i.e.,* we’re tossing a reasonably fair coin), and we know $0 \leq \theta \leq 1$.

**a.** Show that a beta density prior for $\theta$ reflects this prior information. The beta density is given by

$$p(\theta; \alpha) = \frac{\Gamma(2\alpha)}{\Gamma(\alpha)^2}\theta^{\alpha-1}(1-\theta)^{\alpha-1}$$

where $\alpha \geq 1$ is a shape parameter to be specified by the user and $\Gamma$ is the Euler gamma function $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt$.

**b.** Show that the beta density is a *conjugate* prior for the Bernoulli distribution. That is, show that the posterior density also has a beta form.

**c.** Find the posterior mean estimator $\mathbb{E}[\theta|\boldsymbol{x}]$, the mean of the posterior distribution which is a function of $\alpha$ and the data. How does $\alpha$ effect estimator performance? What is the asymptotic (large $n$) behavior of the estimator?

2. Suppose $x \sim p(x|\theta)$ and let $R$ be a risk function (e.g., MSE). An estimator $\widehat{\theta}$ is said to be minimax optimal if it minimizes the maximum risk $\sup_\theta R(\widehat{\theta}, \theta)$. The minimax estimator is related to Bayesian estimators as follows. Let $p(\theta)$ denote a prior probability distribution and let

$$\widehat{\theta}_p \;:=\; \arg\min_{\widehat{\theta}(x)} \int R(\widehat{\theta}(x), \theta)\, p(\theta) d\theta \;,$$

where the maximization is over all possible estimators. The estimator $\widehat{\theta}_p$ the minimizes the average or *Bayes* risk under prior $p$. For example, if the risk is MSE, then $\widehat{\theta}_p = \mathbb{E}[\theta|x]$, the mean of the posterior distribution. If the estimator $\widehat{\theta}_p$ satisfies

$$\int R(\widehat{\theta}_p, \theta)\, p(\theta)\, d\theta \;=\; \sup_\theta R(\widehat{\theta}_p, \theta) \;,$$

then it is minimax optimal. To see this let $\widehat{\theta}$ be any estimator and note that

$$\begin{aligned} \sup_\theta R(\widehat{\theta}, \theta) \;&\geq\; \int R(\widehat{\theta}, \theta)\, p(\theta)\, d\theta \\ &\geq\; \int R(\widehat{\theta}_p, \theta)\, p(\theta)\, d\theta \;=\; \sup_\theta R(\widehat{\theta}_p, \theta) \;. \end{aligned}$$

A corollary of this is that if $\widehat{\theta}_p$ has constant risk, then it is minimax (since the average risk will be equal to the worst case). Use this to find a minimax optimal estimator under MSE risk for $x \sim \text{Binomial}(\theta, n)$, the distribution consider in the problem 2 above. Hint: Use a Beta prior with parameter $\alpha$ chosen so that the posterior mean has constant risk.

3. Recall that the solution to the optimization

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^n} \|\boldsymbol{y} - \boldsymbol{\theta}\|_2^2 \;+\; \lambda \|\boldsymbol{\theta}\|_1$$

is the soft-thresholding operator applied to each element of $\boldsymbol{y}$. Reasoning in a similar manner, determine the solution to

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^n} \|\boldsymbol{y} - \boldsymbol{\theta}\|_2^2 \;+\; \lambda \|\boldsymbol{\theta}\|_0$$

where $\|\boldsymbol{\theta}\|_0$ is equal to the number of non-zero elements in $\boldsymbol{\theta}$.

4. The idea of a gradient can be extended to non-differentiable functions by introducing the notion of *subgradients*. Recall that for a convex function $f$ that is differentiable at $\boldsymbol{w}$, for all $\boldsymbol{u}$ we have

$$f(\boldsymbol{u}) \;\;\geq\;\; f(\boldsymbol{w}) \;+\; (\boldsymbol{u} - \boldsymbol{w})^T \nabla f(\boldsymbol{w}),$$

i.e., the gradient at $\boldsymbol{w}$ defines the slope of a tangent that lies below $f$. If $f$ is not differentiable at $\boldsymbol{w}$, we can write a similar inequality:

$$f(\boldsymbol{u}) \geq f(\boldsymbol{w}) + (\boldsymbol{u} - \boldsymbol{w})^T \boldsymbol{v} \tag{13.1}$$

where we call $\boldsymbol{v}$ a *subgradient*. The formal definition is below.

**Definition 13.7.1.** Any vector $\boldsymbol{v}$ that satisfies (13.1) is called a subgradient of $f$ at $\boldsymbol{w}$. The set of subgradients of $f$ at $\boldsymbol{w}$ is called the differential set and denoted $\partial f(\boldsymbol{w})$. So we write $\boldsymbol{v} \in \partial f(\boldsymbol{w})$.

If $f$ is differentiable at $\boldsymbol{w}$, then there is only one subgradient at $\boldsymbol{w}$, and it is equal to the gradient at $\boldsymbol{w}$.

Let $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ be a set of training data and arrange these data into matrix $\boldsymbol{X}$ and vector $\boldsymbol{y}$. Derive the SGD steps for the following optimization problems. If the gradient does exist at a point, then you may use any subgradient instead (in general there may be many, so you can pick any one).

(a) $\min_w \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|^2$

(b) $\min_w \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|^2 + \lambda\|\boldsymbol{w}\|_2^2$

(c) $\min_w \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|^2 + \lambda\|\boldsymbol{w}\|_1$

(d) $\min_w \sum_{i=1}^n \log(1 + \exp(-y_i \boldsymbol{w}^T \boldsymbol{x}_i))$

(e) $\min_w \sum_{i=1}^n \max(0, 1 - y_i \boldsymbol{w}^T \boldsymbol{x}_i)$

(f) $\min_w \sum_{i=1}^n \max(0, 1 - y_i \boldsymbol{w}^T \boldsymbol{x}_i) + \lambda\|\boldsymbol{w}\|_2^2$

# Lecture 14: Proximal Gradient Algorithms

Consider optimization problems of the following form

$$\min_{\boldsymbol{w}\in\mathbb{R}^p} f(\boldsymbol{w}) \,+\, g(\boldsymbol{w}) \ ,$$

where the functions $f$ and $g$ are convex, and $f$ is also differentiable. Special cases include “ridge" and “lasso" regression, where $f(\boldsymbol{w}) = \sum_{i=1}^n \ell(y_i, \boldsymbol{w}^T\boldsymbol{x}_i)$ with some convex loss function $\ell$ and $g(\boldsymbol{w}) = \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$ or $\lambda\|\boldsymbol{w}\|_1$, respectively. $\lambda \geq 0$ is a regularization parameter that adjusts the tradeoff between the loss and the regularization. A general class of iterative algorithms known as proximal gradient methods can be used to solve these optimizations. Proximal gradient algorithms are easy to implement when the function $g$ has a computationally efficient proximal operator, and have state-of-the-art performance.

The *proximal operator* for $g$ is defined as follows. For any $t > 0$ and $\boldsymbol{v} \in \mathbb{R}^d$

$$\text{prox}_{g,t}(\boldsymbol{v}) \ := \ \arg\min_{\boldsymbol{u}} \left(\frac{1}{2}\|\boldsymbol{u}-\boldsymbol{v}\|^2 + t\, g(\boldsymbol{u})\right) .$$

The solution to the optimization is a point close (proximal) to the input $\boldsymbol{v}$ and with a relatively small $g$ value. The parameter $t$ controls the tradeoff between staying close to $\boldsymbol{v}$ and minimizing $g$. For example, suppose $g(\boldsymbol{w}) = \|\boldsymbol{w}\|_1$. Then the proximal operator is

$$\text{prox}_{g,t}(\boldsymbol{v}) \ = \ \arg\min_{\boldsymbol{u}} \left(\frac{1}{2}\|\boldsymbol{u}-\boldsymbol{v}\|^2 + t\,\|\boldsymbol{u}\|_1\right) \ = \ \arg\min_{\boldsymbol{u}} \sum_{i=1}^d \left(\frac{1}{2}(u_i - v_i)^2 + t\,|u_i|\right) .$$

We see that the optimization objective is *separable* in the coordinates. Each coordinate $u_i$ can be optimized separately. In fact, there is a closed-form solution known as the *soft-threshold* operation

$$\arg\min_{u_i} \left(\frac{1}{2}(u_i - v_i)^2 + t\,|u_i|\right) \ = \ \text{sign}(v_i)\max(0, |v_i| - t) \ .$$

The parameter $t$ plays the role of a threshold. Note that if $|v_i| \leq t$, then the solution is $0$.

**Proposition 14.0.2.** *The solution* $\widehat{u}$ *to the optimization*

$$\min_{u\in\mathbb{R}} \left(\frac{1}{2}(u-v)^2 + t\,|u|\right)$$

*is the soft-threshold operation* $\widehat{u} = \textit{sign}(v)\max(0, |v| - t)$*.*

*Proof.* The subgradient of the objective function with respect to $u$ is

$$\partial_u \left(\frac{1}{2}(u-v)^2 + t\,|u|\right) \ = \ u - v + t\,\text{sign}(u) \ .$$

The subgradient must be zero at the solution, yielding the equation $v = u + t\,\text{sign}(u)$. Consider $u$ as a function of $v$. If $|v| < t$, then $u = -t\,\text{sign}(u) + v$ implies $\widehat{u} = 0$, since otherwise the $\text{sign}(\widehat{u}) \neq \text{sign}(-t\,\text{sign}(\widehat{u}) + v)$. If $v \geq t$, then $\text{sign}(\widehat{u}) = +1$ and $\widehat{u} = v - t$. To see this, observe that if we assume $\text{sign}(\widehat{u}) = -1$, then the solution would be $\widehat{u} = v + \lambda > 0$, which contradicts the assumed sign. Similarly, if $v \leq -t$, then the solution is $\widehat{u} = v + t$. □

This particular case arises in the lasso regression problem, which was a major motivation for the development of proximal gradient algorithms. This type of algorithm was first introduced for the lasso regression (i.e., squared error loss and $\ell_1$ regularization) in [15]. Proximal gradient algorithms were further developed in many subsequent papers, including [26, 3], and the analysis in this lecture follows ideas developed in these papers.

## 14.1. Proximal Gradient Algorithm with Squared Error Loss

To introduce the idea of proximal gradient algorithms, let us first consider the special case of squared error loss, $f(\boldsymbol{w}) = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2$. We can write the objective function as

$$\begin{aligned} L(\boldsymbol{w}) &:= \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 \ + \ g(\boldsymbol{w}) \\ &= \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}^{(k)} + \boldsymbol{X}\boldsymbol{w}^{(k)} - \boldsymbol{X}\boldsymbol{w}\|_2^2 \ + \ g(\boldsymbol{w}) \\ &= \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}^{(k)}\|_2^2 \ + \ 2(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}^{(k)})^T\boldsymbol{X}(\boldsymbol{w}^{(k)} - \boldsymbol{w}) \ + \ \|\boldsymbol{X}(\boldsymbol{w}^{(k)} - \boldsymbol{w})\|_2^2 \ + \ g(\boldsymbol{w}) \end{aligned}$$

The goal of an iterative algorithm is to choose $\boldsymbol{w}$ to reduce our objective. Notice that the first term $C := \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}^{(k)}\|_2$ is a constant that doesn't depend on the variable $\boldsymbol{w}$. The remaining terms are still somewhat complicated, but we can simplify it using the following bound

$$\begin{aligned} L(\boldsymbol{w}) &= C \ + \ 2(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}^{(k)})^T\boldsymbol{X}(\boldsymbol{w}^{(k)} - \boldsymbol{w}) \ + \ \|\boldsymbol{X}(\boldsymbol{w}^{(k)} - \boldsymbol{w})\|_2^2 \ + \ g(\boldsymbol{w}) \\ &\leq C \ + \ 2(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}^{(k)})^T\boldsymbol{X}(\boldsymbol{w}^{(k)} - \boldsymbol{w}) \ + \ \|\boldsymbol{X}\|_2^2\,\|\boldsymbol{w}^{(k)} - \boldsymbol{w}\|_2^2 \ + \ g(\boldsymbol{w}) \\ &\leq C \ + \ 2(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}^{(k)})^T\boldsymbol{X}(\boldsymbol{w}^{(k)} - \boldsymbol{w}) \ + \ t^{-1}\,\|\boldsymbol{w}^{(k)} - \boldsymbol{w}\|_2^2 \ + \ g(\boldsymbol{w})\ , \end{aligned} \tag{14.1}$$

where $\|\boldsymbol{X}|_2$ is the spectral norm of $\boldsymbol{X}$ (the largest singular value) and $0 < t < 1/\|\boldsymbol{X}\|_2^2$. Observe that the upper bound on the right hand side "touches" the original objective $L$ at the point $\boldsymbol{w} = \boldsymbol{w}^{(k)}$. The parameter $t$ will play the same role that it did in the gradient descent iteration. Now we will choose $\boldsymbol{w}$ to minimize this bound, that is

$$\begin{aligned} \boldsymbol{w}^{(k+1)} &:= \arg\min_{\boldsymbol{w}}\bigl\{2(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}^{(k)})^T\boldsymbol{X}(\boldsymbol{w}^{(k)} - \boldsymbol{w}) \ + \ t^{-1}\,\|\boldsymbol{w}^{(k)} - \boldsymbol{w}\|_2^2 \ + \ g(\boldsymbol{w})\bigr\} \\ &= \arg\min_{\boldsymbol{w}}\bigl\{2t(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}^{(k)})^T\boldsymbol{X}(\boldsymbol{w}^{(k)} - \boldsymbol{w}) \ + \ \|\boldsymbol{w}^{(k)} - \boldsymbol{w}\|_2^2 \ + \ tg(\boldsymbol{w})\bigr\}\ . \end{aligned}$$

Note that if we take $\boldsymbol{w} = \boldsymbol{w}^{(k)}$, then the value of the objective above is $g(\boldsymbol{w}^{(k)})$. The minimization will produce a value at least this small (making no progress), but it may be possible that $\boldsymbol{w}^{(k+1)}$ has a value less than $g(\boldsymbol{w}^{(k)})$. It is easy to see (sketch a picture of $L$ and the upper bound) that finding $\boldsymbol{w}^{(k+1)}$ to reduce the upper bound must also reduce the original objective, i.e., $\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}^{(k+1)}\|_2^2 + g(\boldsymbol{w}^{(k+1)}) < \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}^{(k)}\|_2^2 + g(\boldsymbol{w}_k)$, and progress is made toward reducing the original objective function. This inequality is strict for the following reason. Let $L_k^u$ denote the upper bounding function in (14.1). Then $L(\boldsymbol{w}^{(k+1)}) \leq L_k^u(\boldsymbol{w}^{(k+1)}) < L_k^u(\boldsymbol{w}^{(k)}) = L(\boldsymbol{w}^{(k)})$, where the first inequality is because $L_k^u$ upper bounds $L$ at all points and the second is strict because $L_k^u$ is strictly convex due to the $\|\boldsymbol{w}^{(k)} - \boldsymbol{w}\|^2$ term (unless $\boldsymbol{w}^{(k)}$ is already the minimum point, in which case we have equality).

Next define $\boldsymbol{v} := t\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}^{(k)}) = -t\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{w}^{(k)} - \boldsymbol{y})$. Then we can write the optimization above as

$$\boldsymbol{w}^{(k+1)} \ = \ \arg\min_{\boldsymbol{w}}\bigl\{2\,\boldsymbol{v}^T(\boldsymbol{w}^{(k)} - \boldsymbol{w}) \ + \ \|\boldsymbol{w}^{(k)} - \boldsymbol{w}\|_2^2 \ + \ tg(\boldsymbol{w})\bigr\}\ .$$

So we can "complete the square" to obtain

$$\begin{aligned} \boldsymbol{w}^{(k+1)} &= \arg\min_{\boldsymbol{w}}\bigl\{\|\boldsymbol{v} + \boldsymbol{w}^{(k)} - \boldsymbol{w}\|_2^2 \ - \ \|\boldsymbol{v}\|_2^2 \ + \ tg(\boldsymbol{w})\bigr\} \\ &= \arg\min_{\boldsymbol{w}}\bigl\{\|\boldsymbol{v} + \boldsymbol{w}^{(k)} - \boldsymbol{w}\|_2^2 \ + \ tg(\boldsymbol{w})\bigr\}\ , \end{aligned}$$

since $\boldsymbol{v}$ is a constant that does not depend on the optimization variable $\boldsymbol{w}$. Finally, define

$$\begin{aligned} \boldsymbol{z}_k &:= \boldsymbol{v} + \boldsymbol{w}^{(k)} \\ &= \boldsymbol{w}^{(k)} - t\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{w}^{(k)} - \boldsymbol{y})\ , \end{aligned}$$

which you will recognize as the gradient descent iterate. Thus, the next iterate is given by

$$\boldsymbol{w}^{(k+1)} \ = \ \arg\min_{\boldsymbol{w}}\bigl\{\|\boldsymbol{z}_k - \boldsymbol{w}\|_2^2 \ + \ tg(\boldsymbol{w})\bigr\}\ .$$

Note that if $g$ is separable (i.e., it is a sum of terms, each involving just one coordinate of $\boldsymbol{w}$), then the optimization above is separable. This is a separable approximation (and upper bound) of the original optimization. The solution to the optimization the *proximal operator* of the function $g$. So this sort of iterative optimization is often referred to as a *proximal point algorithm*. Note that if $g = 0$, then the solution is $\boldsymbol{w}^{(k+1)} = \boldsymbol{z}_k$, the gradient descent iterate. The optimization is also easy to solve for many choices of $g$.

## 14.2. Proximal Gradient Algorithm

Now let $f(\boldsymbol{w})$ be any convex loss function. The proximal gradient algorithm is follows a simple iteration. Initialize with $\boldsymbol{w}^{(0)}$ arbitrarily chosen and for $k \geq 1$

$$\boldsymbol{w}^{(k)} \;=\; \operatorname{prox}_{g,t}(\boldsymbol{w}^{(k-1)} - t\nabla f(\boldsymbol{w}^{(k-1)}))\;,$$

where for any $\boldsymbol{v} \in \mathbb{R}^p$

$$\operatorname{prox}_{g,t}(\boldsymbol{v}) \;:=\; \arg\min_{\boldsymbol{u}} \left(\frac{1}{2}\|\boldsymbol{u} - \boldsymbol{v}\|^2 + tg(\boldsymbol{u})\right)\;,$$

is called a *proximal operator* for the function $g$. The parameter value $\boldsymbol{w}^+ = \operatorname{prox}_{g,t}(\boldsymbol{w} - t\nabla f(\boldsymbol{w}))$ minimizes the sum of $g(\boldsymbol{u})$ and a *separable* quadratic approximation of $f(\boldsymbol{u})$ around the point $\boldsymbol{w}$. The separability of this approximation is the key to efficient algorithms. Note that if we define $\boldsymbol{v} = \boldsymbol{w} - t\nabla f(\boldsymbol{w})$ and if the regularization function is also separable (e.g., $g(\boldsymbol{u}) = \lambda\|\boldsymbol{u}\|_1 = \sum_{i=1}^p |u_i|$), then we can write the proximal operator optimization as

$$\min_{\boldsymbol{u}} \sum_{i=1}^{p} \left(\frac{1}{2}(u_i - v_i)^2 \;+\; tg(u_i)\right)\;,$$

and we can solve for each scalar element separately. In the case $g(\boldsymbol{u}) = \lambda\|\boldsymbol{u}\|_1$, the proximal gradient algorithm iterates a gradient descent step followed by a soft-thresholding operation, and so is sometimes called an *iterative soft-thresholding algorithm* (ISTA). The solutions in this case tend to be sparse vectors.

## 14.3. Analysis of Proximal Gradient Algorithm

We will employ the following assumptions in our analysis. Unless otherwise noted, norms denote the Euclidean norm.

1. $f$ is convex on $\mathbb{R}^p$ and the gradient $\nabla f$ is $L$-Lipschitz, i.e.,

$$\|\nabla f(\boldsymbol{w}) - \nabla f(\boldsymbol{v})\| \leq L\|\boldsymbol{w} - \boldsymbol{v}\| \text{ for all } \boldsymbol{w}, \boldsymbol{v} \in \mathbb{R}^p\;.$$

2. $g$ is convex

An immediate consequence of the first assumption is the following Taylor series bound.

**Lemma 14.3.1.**

$$f(\boldsymbol{v}) \;\;\leq\;\; f(\boldsymbol{w}) + \nabla f(\boldsymbol{w})^T(\boldsymbol{v} - \boldsymbol{w}) + \frac{L}{2}\|\boldsymbol{v} - \boldsymbol{w}\|^2\;, \qquad (14.2)$$

*Proof.* Since $f$ is differentiable, by the Fundamental Theorem of Calculus

$$f(\boldsymbol{v}) = f(\boldsymbol{w}) + \int_0^1 \nabla f\big(\boldsymbol{w} + \gamma(\boldsymbol{v} - \boldsymbol{w})\big)^T (\boldsymbol{v} - \boldsymbol{w})\, d\gamma\ .$$

Therefore, we have

$$f(\boldsymbol{v}) - f(\boldsymbol{w}) - \nabla f(\boldsymbol{w})^T(\boldsymbol{v} - \boldsymbol{w}) \ = \ \int_0^1 \Big(\nabla f\big(\boldsymbol{w} + \gamma(\boldsymbol{v} - \boldsymbol{w})\big) - \nabla f(\boldsymbol{w})\Big)^T (\boldsymbol{v} - \boldsymbol{w})\, d\gamma\ .$$

By the Cauchy-Schwarz inequality and the Lipschitz assumption above, we have

$$\Big(\nabla f\big(\boldsymbol{w} + \gamma(\boldsymbol{v} - \boldsymbol{w})\big) - \nabla f(\boldsymbol{w})\Big)^T (\boldsymbol{v} - \boldsymbol{w}) \ \leq\ \|\nabla f\big(\boldsymbol{w} + \gamma(\boldsymbol{v} - \boldsymbol{w})\big) - \nabla f(\boldsymbol{w})\|\, \|\boldsymbol{v} - \boldsymbol{w}\| \ \leq\ L\gamma\|\boldsymbol{v} - \boldsymbol{w}\|^2\ .$$

and since $\int_0^1 \gamma\, d\gamma = \frac{1}{2}$, the result follows. □

Recall that because $f$ is convex, $f(\boldsymbol{v}) \geq f(\boldsymbol{w}) + \nabla f(\boldsymbol{w})^T(\boldsymbol{v} - \boldsymbol{w})$. So the Lipschitz assumption on the gradient is guaranteeing that $f(\boldsymbol{v})$ cannot be too much larger than this.

First note that we can express the proximal gradient update rule as

$$\boldsymbol{w}^{(k)} \ = \ \boldsymbol{w}^{(k-1)} - tF(\boldsymbol{w}^{(k-1)})\ ,$$

where

$$F(\boldsymbol{w}) \ := \ \frac{1}{t}\Big(\boldsymbol{w} - \operatorname{prox}_{g,t}(\boldsymbol{w} - t\nabla f(\boldsymbol{w}))\Big)\ .$$

We can interpret $tF(\boldsymbol{w}^{(k-1)})$ as the step we take to adjust the weights at each iteration. Next, take $\boldsymbol{v} \ = \ \boldsymbol{w} - tF(\boldsymbol{w})$ in Equation (14.2) to obtain

$$f(\boldsymbol{w} - tF(\boldsymbol{w})) \ \leq\ f(\boldsymbol{w}) - t\nabla f(\boldsymbol{w})^T F(\boldsymbol{w}) + \frac{t^2 L}{2}\|F(\boldsymbol{w})\|^2\ .$$

If $0 < t \leq \frac{1}{L}$, then

$$f(\boldsymbol{w} - tF(\boldsymbol{w})) \ \leq\ f(\boldsymbol{w}) - t\nabla f(\boldsymbol{w})^T F(\boldsymbol{w}) + \frac{t}{2}\|F(\boldsymbol{w})\|^2\ .$$

We will assume this condition on $t$ for the rest of the analysis. Also, we will make use of the following fact in two places in the proof: If $g$ is any convex function and $\boldsymbol{x}$ is in the subdifferential of $g$ at the point $\boldsymbol{w}$ (i.e., if $g$ is differentiable at $\boldsymbol{u}$, then $\boldsymbol{x}$ is tangent to $g$ at $\boldsymbol{w}$, otherwise it is any subdifferential at $\boldsymbol{w}$), then for any $\boldsymbol{u} \in \mathbb{R}^p$

$$g(\boldsymbol{u}) \ \geq\ f(\boldsymbol{w}) + \boldsymbol{x}^T(\boldsymbol{u} - \boldsymbol{w})\ .$$

In words, this says that "tangent" lines lie below the convex function.

**Lemma 14.3.2.** *For all* $\boldsymbol{v} \in \mathbb{R}^p$

$$f(\boldsymbol{w} - tF(\boldsymbol{w})) + g(\boldsymbol{w} - tF(\boldsymbol{w})) \ \leq\ f(\boldsymbol{v}) + g(\boldsymbol{v}) + F(\boldsymbol{w})^T(\boldsymbol{w} - \boldsymbol{v}) - \frac{t}{2}\|F(\boldsymbol{w})\|^2\ . \tag{14.3}$$

*Proof.* Using the Taylor's Series bound 14.2

$$f(\boldsymbol{w} - tF(\boldsymbol{w})) + g(\boldsymbol{w} - tF(\boldsymbol{w})) \ \ \leq\ \ f(\boldsymbol{w}) - t\nabla f(\boldsymbol{w})^T F(\boldsymbol{w}) + \frac{t}{2}\|F(\boldsymbol{w})\|^2 + g(\boldsymbol{w} - tF(\boldsymbol{w}))\ .$$

By convexity of $f$, for any $\boldsymbol{v}$ we have $f(\boldsymbol{v}) \geq f(\boldsymbol{w}) + \nabla f(\boldsymbol{w})^T(\boldsymbol{v} - \boldsymbol{w})$, and so $f(\boldsymbol{w}) \leq f(\boldsymbol{v}) + \nabla f(\boldsymbol{w})^T(\boldsymbol{w} - \boldsymbol{v})$. This yields

$$\begin{aligned} f(\boldsymbol{w} - tF(\boldsymbol{w})) + g(\boldsymbol{w} - tF(\boldsymbol{w})) &\leq f(\boldsymbol{v}) + \nabla f(\boldsymbol{w})^T(\boldsymbol{w} - \boldsymbol{v}) - tf(\boldsymbol{w})^T F(\boldsymbol{w}) \\ &\quad + \frac{t}{2}\|F(\boldsymbol{w})\|^2 + g(\boldsymbol{w} - tF(\boldsymbol{w}))\,. \end{aligned} \tag{14.4}$$

Also, since $g$ is convex we can upper bound it in a similar fashion. Because $g$ may not be differentiable, we consider the subdifferentials of $g$. Recall the definition of the $\text{prox}_{g,t}$ operator

$$\text{prox}_{g,t}(\boldsymbol{v}) \;:=\; \arg\min_{\boldsymbol{u}} \underbrace{\left(\frac{1}{2}\|\boldsymbol{u} - \boldsymbol{v}\|^2 + tg(\boldsymbol{u})\right)}_{=:p(\boldsymbol{u})}\,.$$

Now because $\text{prox}_{g,t}(\boldsymbol{v})$ is a minimizer of $p(\boldsymbol{u})$, the zero vector $\mathbf{0}$ is in the subdifferential of $p$ at the point $\text{prox}_{g,t}(\boldsymbol{v})$. Note that the subdifferential is

$$\partial p(\text{prox}_{g,t}(\boldsymbol{v})) \;=\; \text{prox}_{g,t}(\boldsymbol{v}) - \boldsymbol{v} + t\partial g(\text{prox}_{g,t}(\boldsymbol{v}))\,.$$

Setting this expression to $\mathbf{0}$ shows that

$$\boldsymbol{v} - \text{prox}_{g,t}(\boldsymbol{v}) \;\in\; \partial g(\text{prox}_{g,t}(\boldsymbol{v}))\,.$$

Now take $\boldsymbol{v} = \boldsymbol{w} - t\nabla f(\boldsymbol{w})$ to obtain

$$\boldsymbol{w} - t\nabla f(\boldsymbol{w}) - \text{prox}_{g,t}(\boldsymbol{w} - t\nabla f(\boldsymbol{w})) \;\in\; \partial g(\text{prox}_{g,t}(\boldsymbol{w} - t\nabla f(\boldsymbol{w})))\,.$$

Recall that $\text{prox}_{g,t}(\boldsymbol{w} - t\nabla f(\boldsymbol{w})) = \boldsymbol{w} - tF(\boldsymbol{w})$, so we have

$$F(\boldsymbol{w}) - \nabla f(\boldsymbol{w}) \;\in\; \partial g(\text{prox}_{g,t}(\boldsymbol{w} - t\nabla f(\boldsymbol{w})))\,.$$

By convexity of $g$, we have the bound $g(\boldsymbol{w} - tF(\boldsymbol{w})) \leq g(\boldsymbol{v}) + \big(F(\boldsymbol{w}) - \nabla f(\boldsymbol{w})\big)^T(\boldsymbol{w} - tF(\boldsymbol{w}) - \boldsymbol{v})$ for all $\boldsymbol{v}$. Plugging this into Equation (14.4) yields

$$\begin{aligned} f(\boldsymbol{w} - tF(\boldsymbol{w})) + g(\boldsymbol{w} - tF(\boldsymbol{w})) &\leq f(\boldsymbol{v}) + \nabla f(\boldsymbol{w})^T(\boldsymbol{w} - \boldsymbol{v}) - tf(\boldsymbol{w})^T F(\boldsymbol{w}) + \frac{t}{2}\|F(\boldsymbol{w})\|^2 \\ &\quad + g(\boldsymbol{v}) + \big(F(\boldsymbol{w}) - \nabla f(\boldsymbol{w})\big)^T(\boldsymbol{w} - tF(\boldsymbol{w}) - \boldsymbol{v}) \\ &= f(\boldsymbol{v}) + g(\boldsymbol{v}) + F(\boldsymbol{w})^T(\boldsymbol{w} - \boldsymbol{v}) - \frac{t}{2}\|F(\boldsymbol{w})\|^2 \end{aligned}$$

□

Lemma 1 shows that each iteration of the algorithm makes progress toward the global minimum. To simplify notation, let us denote the overall objective function by $\phi := f + g$. Take $\boldsymbol{v} = \boldsymbol{w}$ in Equation 14.3 to obtain

$$\begin{aligned} \phi(\boldsymbol{w} - tF(\boldsymbol{w})) &\leq \phi(\boldsymbol{w}) - \frac{t}{2}\|F(\boldsymbol{w})\|^2 \\ &\leq \phi(\boldsymbol{w})\,. \end{aligned}$$

Now let $\boldsymbol{w}^\star$ denote a global minimizer $\phi$, and use Equation (14.3) with $\boldsymbol{v} = \boldsymbol{w}^\star$ to obtain

$$\begin{aligned} \phi(\boldsymbol{w} - tF(\boldsymbol{w})) - \phi(\boldsymbol{w}^\star) &\leq F(\boldsymbol{w})^T(\boldsymbol{w} - \boldsymbol{w}^\star) - \frac{t}{2}\|F(\boldsymbol{w})\|^2 \\ &= \frac{1}{2t}\Big(\|\boldsymbol{w} - \boldsymbol{w}^\star\|^2 - \|\boldsymbol{w} - \boldsymbol{w}^\star - tF(\boldsymbol{w})\|^2\Big) \\ &= \frac{1}{2t}\Big(\|\boldsymbol{w} - \boldsymbol{w}^\star\|^2 - \|\boldsymbol{w} - tF(\boldsymbol{w}) - \boldsymbol{w}^\star\|^2\Big)\,. \end{aligned}$$

Since $\phi(\boldsymbol{w} - tF(\boldsymbol{w})) - \phi(\boldsymbol{w}^\star) \geq 0$, we have

$$\|\boldsymbol{w} - tF(\boldsymbol{w}) - \boldsymbol{w}^\star\|^2 \ \leq\ \|\boldsymbol{w} - \boldsymbol{w}^\star\|^2 \ , \tag{14.5}$$

which shows that each iteration decreases the distance to a global minimizer.

Now we can analyze the rate of convergence of the algorithm. Add inequalities in Equation (14.5) for $\boldsymbol{w}^{(i-1)} = \boldsymbol{w}$ and $\boldsymbol{w}^{(i)} = \boldsymbol{w} - tF(\boldsymbol{w})$:

$$\begin{aligned}\sum_{i=1}^{k} \left(\phi(\boldsymbol{w}^{(i)}) - \phi(\boldsymbol{w}^\star)\right) &\leq \frac{1}{2t}\sum_{i=1}^{k}\left(\|\boldsymbol{w}^{(i-1)} - \boldsymbol{w}^\star\|^2 - \|\boldsymbol{w}^{(i)} - \boldsymbol{w}^\star\|^2\right) \\ &= \frac{1}{2t}\left(\|\boldsymbol{w}^{(0)} - \boldsymbol{w}^\star\|^2 - \|\boldsymbol{w}^{(k)} - \boldsymbol{w}^\star\|^2\right) \\ &\leq \frac{1}{2t}\|\boldsymbol{w}^{(0)} - \boldsymbol{w}^\star\|^2\end{aligned}$$

Now since we showed above that $\phi(\boldsymbol{w}^{(i)})$ is nonincreasing,

$$\phi(\boldsymbol{w}^{(k)}) - \phi(\boldsymbol{w}^\star) \ \leq\ \frac{1}{k}\sum_{i=1}^{k}\left(\phi(\boldsymbol{w}^{(i)}) - \phi(\boldsymbol{w}^\star)\right) \ \leq\ \frac{1}{2kt}\|\boldsymbol{w}^{(0)} - \boldsymbol{w}^\star\|^2 \ .$$

Finally, recall that we can take $t = \frac{1}{L}$ yielding

$$\phi(\boldsymbol{w}^{(k)}) - \phi(\boldsymbol{w}^\star) \ \leq\ \frac{L}{2k}\|\boldsymbol{w}^{(0)} - \boldsymbol{w}^\star\|^2 \ .$$

This shows that $\phi(\boldsymbol{w}^{(k)}) - \phi(\boldsymbol{w}^\star) \leq \epsilon$ after $O(\epsilon^{-1})$ iterations.

## 14.4. Exercises

Suppose that we observe

$$\boldsymbol{y} \ = \ \boldsymbol{X}\boldsymbol{w}^* + \boldsymbol{\epsilon} \tag{14.6}$$

where $\boldsymbol{X} \in \mathbb{R}^{n\times n}$ has orthonormal columns and $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. Consider the regularized optimization for estimating $\boldsymbol{w}^*$

$$\min_{\boldsymbol{w}} \ \frac{1}{2}\,\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 \ + \ \frac{\lambda}{p}\,\|\boldsymbol{w}\|_p^p$$

for $p = 1$ or 2 and $\lambda > 0$.

1. Show/explain how the optimization problem above can be transformed into an equivalent optimization of the form

$$\min_{\boldsymbol{w}} \ \frac{1}{2}\,\|\widetilde{\boldsymbol{y}} - \boldsymbol{w}\|_2^2 \ + \ \frac{\lambda}{p}\,\|\boldsymbol{w}\|_p^p$$

   and explain how $\widetilde{\boldsymbol{y}}$ is related to $\boldsymbol{y}$.

2. Let $\lambda = 0$.

(a) Give an expression for $\widehat{\boldsymbol{w}}$, the solution to optimization above, and

(b) Consider the prediction error for a new observation of the form $y = \boldsymbol{x}^T\boldsymbol{w}^* + \epsilon$, for arbitrary, fixed $\boldsymbol{x} \in \mathbb{R}^n$ and independent noise $\epsilon \sim \mathcal{N}(0, 1)$. Show that the expected squared prediction error

$$\mathbb{E}[(y - \boldsymbol{x}^T\widehat{\boldsymbol{w}})^2] \;=\; \|\boldsymbol{x}\|_2^2 + 1 \;.$$

3. Let $\lambda > 0$ and $p = 2$.

   (a) Give an expression for $\widehat{\boldsymbol{w}}$, the solution to optimization above in this case, and

   (b) Consider the prediction error for a new observation of the form $y = \boldsymbol{x}^T\boldsymbol{w}^* + \epsilon$, as above. Derive an expression for the expected squared prediction error $\mathbb{E}[(y - \boldsymbol{x}^T\widehat{\boldsymbol{w}})^2]$, and show that it reduces to the expression in the MLE case when $\lambda = 0$.

4. The analysis of proximal gradient assumes that $\nabla f$ is $L$-Lipschitz, and the step size is then assumed to satisfy $t \leq 1/L$. The analysis also assumes that $t$ is also the parameter of the proximal operator $\text{prox}_{g,t}$. What is the effect of changing the proximal operator parameter to some other values $\tau > t$?

# Lecture 15: The Lasso and Soft-Thresholding

The squared error “lasso" regression problem solves the optimization

$$\min_{w} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{w}\|_1 \ .$$

Recall that the $\ell_1$ regularization encourages sparse solutions. Suppose that $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{X}\boldsymbol{w}, \sigma^2\boldsymbol{I})$ for some $\boldsymbol{w} \in \mathbb{R}^d$ and that $\boldsymbol{w}$ is sparse (many of elements in $\boldsymbol{w}$ are exactly zero). Ideally, the solution to the lasso optimization produce a $\widehat{\boldsymbol{w}}$ also sparse in the same locations. Under certain assumptions on $\boldsymbol{X}$, this sort of result can be proved [18, 7, 13]. The simplest setting where such results can be established is when $\boldsymbol{X} = \boldsymbol{I}$, the identity matrix, which we will study here.

Consider the “direct” observation model where $\boldsymbol{y} \in \mathbb{R}^n$ is given by

$$\boldsymbol{y} = \boldsymbol{w} + \boldsymbol{\epsilon} \ , \ \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2\boldsymbol{I}) \ .$$

Suppose that many of the weights/coefficients in $\boldsymbol{w}$ are equal to zero. The MLE of $\boldsymbol{w}$ is simply $\boldsymbol{y}$, and its MSE is $n\sigma^2$. Instead, consider the regularized problem

$$\min_{w} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{w}\|_1 \ .$$

Its solution is soft-thresholding estimator

$$\widehat{w}_i = \text{sign}(y_i) \ \max(|y_i| - \lambda, 0) \ , \ \lambda > 0$$

which can perform much better, especially if $\boldsymbol{w}$ is sparse.

Before we analyze the soft-thresholding estimator, let us consider an ideal thresholding estimator. Suppose that an oracale tells us the magnitude of each $w_i$. The *oracle* estimator is

$$\widehat{w}_i^O \ = \ \begin{cases} y_i & \text{if } |w_i|^2 \geq \sigma^2 \\ 0 & \text{if } |w_i|^2 < \sigma^2 \end{cases}$$

In other words, we estimate a coefficient if and only if the signal power is at least as large as the noise power. The MSE of this estimator is

$$\mathbb{E}\sum_{i=1}^{n}(\widehat{w}_i^O - w_i)^2 = \sum_{i=1}^{n}\min(|w_i|^2, \sigma^2)$$

Notice that the MSE of the oracle estimator is always less than or equal to the MSE of the MLE. If $w$ is sparse, then the MSE of the oracle estimator can be much smaller. If all but $k < n$ coefficients are zero, then the MSE of the oracle estimator is at most $k\sigma^2$. Remarkably, the soft-thresholding estimator comes very close to achieving the performance of the oracle, and shown by the following theorem from [14].

The theorem uses the threshold $\lambda = \sqrt{2\sigma^2 \log n}$. This choice of threshold is motivated by the following observation. Assume, for the moment, that we observe no signal at all, just noise (i.e., $w_i = 0$ for $i = 1, \ldots, n$). In this case, we should set the threshold so that it is larger than the magnitude of any of the $y_i$ (so they are all set to zero). If we take $\lambda = \sqrt{2\sigma^2 \log \frac{n}{\delta}}$, then using the Gaussian tail bound and the union bound we have $\mathbb{P}(\bigcup_{i=1}^n\{|y_i| \geq \lambda\}) \leq \delta$.

**Theorem 15.0.1.** *Assume the direct observation model above and let*

$$\widehat{w}_i = \text{sign}(y_i) \ \max(|y_i| - \lambda, 0)$$

*with $\lambda = \sqrt{2\sigma^2 \log n}$. Then*

$$\mathbb{E}\|\widehat{w} - w\|_2^2 \;\leq\; (2\log n + 1)\left\{\sigma^2 + \sum_{i=1}^{n} \min(|w_i|^2, \sigma^2)\right\}$$

The theorem shows that the soft-thresholding estimator mimics the MSE performance of the oracle estimator to within a factor of roughly $2\log n$. For example, if $\boldsymbol{w}$ is $k$-sparse (with non-zero coefficients larger than $\sigma$ in magnitude), then the MSE of the oracle is $k\sigma^2$ and the MSE of the soft-thresholding estimator is at most $(2\log n+1)(k+1)\sigma^2 \approx 2k\log n\,\sigma^2$ when $n$ is large. This also corresponds to a huge improvement over the MLE if $2k\log n \ll n$.

**Intution:** Consider the case with $\sigma^2 = 1$ (the general case follows by simple rescaling). First recall that if $y \sim \mathcal{N}(0,1)$, then $\mathbb{P}(|y| \geq \lambda) \leq e^{-\lambda^2/2}$. This inequality is easily derived as follows. Since $\mathbb{P}(y \geq \lambda) = \mathbb{P}(y \leq -\lambda)$, we only need to show that $\mathbb{P}(y \geq \lambda) = \frac{1}{2\pi}\int_\lambda^\infty e^{-y^2/2}dy \leq \frac{1}{2}e^{-\lambda^2/2}$. Note that

$$\frac{\frac{1}{2\pi}\int_\lambda^\infty e^{-y^2/2}dy}{\frac{1}{2}e^{-\lambda^2/2}} \;=\; \frac{\frac{1}{2\pi}\int_\lambda^\infty e^{-(y^2-\lambda^2)/2}dy}{\frac{1}{2}} \;=\; \frac{\frac{1}{2\pi}\int_\lambda^\infty e^{-(y-\lambda)(y+\lambda)/2}dy}{\frac{1}{2}} \; .$$

The desired inequality results by making change of variable $t = y + \lambda$ to yield

$$\frac{\frac{1}{2\pi}\int_\lambda^\infty e^{-y^2/2}dy}{\frac{1}{2}e^{-\lambda^2/2}} \;=\; \frac{\frac{1}{2\pi}\int_0^\infty e^{-t(t+2\lambda)/2}dt}{\frac{1}{2}} \;\leq\; \frac{\frac{1}{2\pi}\int_0^\infty e^{-t^2/2}dt}{\frac{1}{2}} \;=\; 1 \; .$$

Now observe that if $\lambda = \sqrt{2\log n}$, then $\mathbb{P}(|y_i| \geq \lambda | w_i = 0) \leq e^{-\log n} = \frac{1}{n}$. Using this we have

$$\mathbb{E}\left[\sum_{i:w_i=0} \mathbb{1}_{\{\widehat{w}_i \neq 0\}}\right] \;\leq\; \sum_{i:w_i=0} \frac{1}{n} \;\leq\; 1 \; .$$

In other words, using this threshold we expect that at most one of the $w_i = 0$ will not be estimated as $\widehat{w}_i = 0$. Next consider cases when $w_i \neq 0$. Let us suppose that $|w_i| \gg \lambda$, so that $\widehat{w}_i = y_i - \lambda \text{sign}(y_i)$. In other words, if $|w_i| \gg \lambda$, then with high probability (tending to 1 as $|w_i|$ increases, we have $|y_i| \geq \lambda$. In this case,

$$(w_i - \widehat{w}_i)^2 \;=\; (-\epsilon_i + \lambda \text{sign}(y_i))^2 \;\leq\; \epsilon_i^2 + 2|\epsilon_i|\lambda + \lambda^2 \; .$$

Taking the expecation of this upper bound yields

$$\mathbb{E}[(w_i - \widehat{w}_i)^2] \;\leq\; 1 + 2\lambda + \lambda^2 \;\leq\; 3\lambda^2 + 1 \; , \text{ assuming } \lambda > 1 \; .$$

Thus, if $\boldsymbol{w}$ has only $k$ nonzero weights, then this intuition suggests that

$$\sum_{i=1}^{n} \mathbb{E}[(w_i - \widehat{w}_i)^2] \;=\; O(k\log n) \; .$$

This is formalized in the following proof of Theorem 1.

*Proof:* To simplify the analysis, assume that $\sigma^2 = 1$. The general result follows directly. It suffice to show that

$$\mathbb{E}[(\widehat{w}_i - w_i)^2] \;\leq\; (2\log n + 1)\left\{\frac{1}{n} + \min(w_i^2, 1)\;\right\}$$

for each $i$. So let $y \sim \mathcal{N}(w,1)$ and let $f_\lambda(y) = \text{sign}(y)\max(|y|-\lambda, 0)$. We will show that with $\lambda = \sqrt{2\log n}$

$$\mathbb{E}[(f_\lambda(y)-w)^2] \;\le\; (2\log n + 1)\left\{\frac{1}{n} + \min(w^2,1)\right\} .$$

First note that $f_\lambda(y) = y - \text{sign}(y)(|y|\wedge\lambda)$, where $a \wedge b$ is shorthand notation for $\min(a,b)$. It follows that

$$\begin{aligned}\mathbb{E}[(f_\lambda(y)-w)^2] &= \mathbb{E}[(y-w)^2] - 2\mathbb{E}[\text{sign}(y)(|y|\wedge\lambda)(y-w)] + \mathbb{E}[y^2\wedge\lambda^2] \\ &= 1 - 2\mathbb{E}[\text{sign}(y)(|y|\wedge\lambda)(y-w)] + \mathbb{E}[y^2\wedge\lambda^2]\end{aligned}$$

The expected value in the second term is equal to $\mathbb{P}(|y|<\lambda)$, which is verified as follows.

The expectation can be split into integrals over four intervals, $(\infty,-t]$, $(-t,0]$, $(0,t]$, and $(t,\infty)$. Each integrand is a linear or quadratic function of $y$ times the Gaussian density function. Let $\phi(x) := \frac{1}{\sqrt{2\pi}}e^{-x^2/2}$ and $\Phi(x)$ be the cumulative distribution function of $\phi(x)$, and consider the following indefinite Gaussian integral forms:

$$\begin{aligned}\int \phi(x)\,dx &= \Phi(x)\ ,\ \text{by definition of } \Phi, \\ \int x\phi(x)\,dx &= \frac{1}{\sqrt{2\pi}}\int xe^{-x^2/2}\,dx \;=\; \underbrace{-\frac{1}{\sqrt{2\pi}}\int e^u\,du}_{u=-x^2/2} \;=\; -\frac{1}{\sqrt{2\pi}}e^u \;=\; -\phi(x)\ , \\ \int x^2\phi(x)\,dx &= \Phi(x) - x\phi(x)\ .\end{aligned}$$

The last form is verified as follows. Let $u = x$ and $dv = x\phi(x)dx$. Then integration by parts $\int udv = uv - \int vdu$ and $\int x\phi(x)dx = -\phi(x)$ show that

$$\int x^2\phi(x)\,dx = x\int x\phi(x)dx - \int\int x\phi(x)dx = -x\phi(x) + \int\phi(x) = \Phi(x) - x\phi(x)\ .$$

The Gaussian distribution we are considering has mean $w$ so the shifted integral forms below, which follow immediately from the derviations above by variable substitution, will be used in our analysis:

$$\begin{aligned}&(i) & \textstyle\int \phi(x-w)dx &= \Phi(x-w) \\ &(ii) & \textstyle\int x\phi(x-w)dx &= w\Phi(x-w) - \phi(x-w) \\ &(iii) & \textstyle\int x^2\phi(x-w)dx &= (1+w^2)\Phi(x-w) - (x+w)\phi(x-w)\end{aligned}$$

Using these forms we compute

$$\begin{aligned}\mathbb{E}[\text{sign}(x)(|x|\wedge\lambda)(x-w)] &= \int_{-\infty}^{\infty}\text{sign}(x)(|x|\wedge\lambda)(x-w)\,\phi(x-w)\,dx \\ &= \underbrace{\int_{-\infty}^{-\lambda} -\lambda(x-w)\phi(x-w)\,dx}_{\lambda\phi(-\lambda-w)} + \underbrace{\int_{-\lambda}^{0} x(x-w)\phi(x-w)dx}_{\Phi(-w)-\Phi(-\lambda-w)-\lambda\phi(-\lambda-w)} \\ &\quad + \underbrace{\int_{0}^{\lambda} x(x-w)\phi(x-w)dx}_{\Phi(\lambda-w)-\Phi(-w)-\lambda\phi(\lambda-w)} + \underbrace{\int_{\lambda}^{\infty} \lambda(x-w)\phi(x-w)dx}_{\lambda\phi(\lambda-w)} \\ &= \Phi(\lambda-w) - \Phi(-\lambda-w) = \mathbb{P}(|x|<\lambda)\end{aligned}$$

So we have shown that

$$\mathbb{E}[(f_\lambda(y)-w)^2] \;=\; 1 - 2\mathbb{P}(|y|<\lambda) + \mathbb{E}[y^2\wedge\lambda^2]$$

Note first that since $y^2 \wedge \lambda^2 \leq \lambda^2$ we have

$$\mathbb{E}[(f_\lambda(y) - w)^2] \leq 1 + \lambda^2 = 1 + 2\log n < (2\log n + 1)(1/n + 1) .$$

On the other hand, since $y^2 \wedge \lambda^2 \leq y^2$ we also have

$$\mathbb{E}[(f_\lambda(y) - w)^2] \ \leq \ 1 - 2\mathbb{P}(|y| < \lambda) + w^2 + 1 \ = \ 2(1 - \mathbb{P}(|y| < \lambda)) + w^2 \ = \ 2\mathbb{P}(|y| \geq \lambda) + w^2 .$$

The proof will be finished if we show that

$$2\mathbb{P}(|y| \geq \lambda) \ \leq \ (2\log n + 1)/n + (2\log n)w^2 .$$

Define $g(w) := 2\mathbb{P}(|y| \geq \lambda)$ and note that $g$ is symmetric about $0$. Using a Taylor's series with remainder we have

$$g(w) \leq g(0) + \frac{1}{2}\sup|g''|w^2 ,$$

where $g''$ is the second derivative of $g$. Note that $g(w) = 2[1 - \mathbb{P}(z \leq \lambda - w) + \mathbb{P}(z \leq -\lambda - w)]$, where $z \sim \mathcal{N}(0,1)$. Using the Gaussian tail bound $\mathbb{P}(z > \lambda) \leq \frac{1}{2}e^{-\lambda^2/2}$ and plugging in $\lambda = \sqrt{2\log n}$ we obtain $g(0) \leq 2/n$. Note that $g'(w) = 2[\phi(\lambda - w) - \phi(-\lambda - w)]$ and $g'(0) = 0$. The integral $(ii)$ above shows that the derivative of $\phi(\lambda - w)$ with respect to $w$ is equal to $(\lambda - w)\phi(\lambda - w)$. So we have $g''(w) = 2[(\lambda - w)\phi(\lambda - w) - (-\lambda - w)\phi(-\lambda - w)]$. It is easy to verify that $|g''(w)| < 1$, since $\sup_x |x\phi(x)| < 0.25$. To simplify the final bound, note that $4\log n > 1$ if $n \geq 2$, so it follows that $\sup_w g''(w) < 4\log n$ for all $n \geq 2$.

## 15.1. Exercises

1. Suppose that we observe

$$\boldsymbol{y} \ = \ \boldsymbol{X}\boldsymbol{w}^* + \boldsymbol{\epsilon} \tag{15.1}$$

where $\boldsymbol{X} \in \mathbb{R}^{n\times n}$ has orthonormal columns and $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. Consider the regularized optimization for estimating $\boldsymbol{w}^*$

$$\min_{\boldsymbol{w}} \ \frac{1}{2}\,\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 \ + \ \lambda\,\|\boldsymbol{w}\|_1$$

   (a) What value of $\lambda$ would you suggest for this case and why?

   (b) Suppose that $\boldsymbol{w}^*$ has only $k < n$ nonzero elements. Consider the prediction error for a new observation of the form $y = \boldsymbol{x}^T\boldsymbol{w}^* + \epsilon$, as above. Show that the expected squared prediction error can be bounded as follows

$$\mathbb{E}[(y - \boldsymbol{x}^T\widehat{\boldsymbol{w}})^2] \ \leq \ (2\log n + 1)(k+1)\|\boldsymbol{x}\|^2 + 1 .$$

   Hint: You may use the soft-thresholding result from class which states that the solution to the optimization in (14.6), for appropriately chosen $\lambda$, produces an estimator $\widehat{\boldsymbol{w}}$ satisfying the bound

$$\mathbb{E}[\|\boldsymbol{w}^* - \widehat{\boldsymbol{w}}\|_2^2] \ \leq \ (2\log n + 1)\Big(1 + \sum_{i=1}^n \min(|w_i^*|^2, 1)\Big) .$$

2. Consider the data model

$$y_i \ = \ w_i \ + \ \epsilon_i \ , \ i = 1, 2, \ldots, n$$

where $\epsilon_i \overset{iid}{\sim} \mathcal{N}(0,1)$. Let's model the weight sparsity using the Gaussian mixture model $w_i \overset{iid}{\sim} (1-p)\mathcal{N}(0, 0.1) + p\mathcal{N}(0, 10)$. Let $\widehat{\boldsymbol{w}}$ denote the estimate of $\boldsymbol{w}$ using a soft-thresholding operation with threshold $\lambda > 0$. Find a bound on the MSE of this estimator in terms of $p$ and $\lambda$. How would you optimize the choice of $\lambda$ in terms of $p$?

3. Let $\boldsymbol{X}$ be an $n \times n$ diagonal matrix and denote the $i$th diagonal entry as $x_i$. Assume $\boldsymbol{X}$ is full rank (i.e., $x_i \neq 0$ for all $i$). Suppose we observe $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{\epsilon}$, with $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. Equivalently, $y_i = x_i w_i + \epsilon_i$, $i = 1, \dots, n$. Consider the optimization

$$\frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{w}\|_1$$

and recall that our theory for the case $\boldsymbol{X} = \boldsymbol{I}$ suggests that $\lambda$ should be proportional to the standard deviation of the noise. Is this reasonable for general diagonal matrix $\boldsymbol{X}$? Suppose that a certain $|x_i| \gg 1$ or $|x_i| \ll 1$ to get some intuition. Can you suggest a modified regularization that might make more sense in this setting? The goal is to find an estimator $\widehat{w}_i$ so the $\sum_i \mathbb{E}[(\widehat{w}_i - w_i)^2]$ is small.

# Lecture 16: Concentration Inequalities

The most important form of statistic considered in this course is a sum of independent random variables.

**Example 10.** *A biologist is studying the new artificial lifeform called* synthia. *She is interested to see if the synthia cells can survive in cold conditions. To test synthia's hardiness, the biologist will conduct $n$ independent experiments. She has grown $n$ cell cultures under ideal conditions and then exposed each to cold conditions. The number of cells in each culture is measured before and after spending one day in cold conditions. The fraction of cells surviving the cold is recorded. Let $x_1, \ldots, x_n$ denote the recorded fractions. The average $\widehat{p} := \frac{1}{n}\sum_{i=1}^n x_i$ is an estimator of the survival probability.*

Understanding behavior of sums of independent random variables is extremely important. For instance, the biologist in the example above would like to know that the estimator is reasonably accurate. Let $X_1, \ldots, X_n$ be independent and identically distributed random variables with variance $\sigma^2 < \infty$ and consider the average $\widehat{\mu} := \frac{1}{n}\sum_{i=1}^n X_i$. First note that $\mathbb{E}[\widehat{\mu}] = \mathbb{E}[X]$. An easy calculation shows that the variance of $\widehat{\mu}$ is $\sigma^2/n$. So the average has the same mean value as the random variables and the variance is reduced by a factor of $n$. Lower variance means less uncertainty. So it is possible to reduce uncertainty by averaging. The more we average, the less the uncertainty (assuming, as we are, that the random variables are independent, which implies they are uncorrelated).

The argument above quantifies the effect of averaging on the variance, but often we would like to say more about the distribution of the average. The *Central Limit Theorem* is a classic result showing that the probability distribution of the average of $n$ independent and identically distributed random variables with mean $\mu$ and variance $\sigma^2 < \infty$ tends to a Gaussian distribution with mean $\mu$ and variance $\sigma^2/n$, regardless of the form of the distribution of the variables. By 'tends to' we mean in the limit as $n$ tends to infinity.

In many applications we would like to say something more about the distributional characteristics for finite values of $n$. One approach is to calculate the distribution of the average explicitly. Recall that if the random variables have a density $p_X$, then the density of the sum $\sum_{i=1}^n X_i$ is the $n$-fold convolution of the density $p_X$ with itself (again this hinges on the assumption that the random variables are independent; it is easy to see by considering the characteristic function of the sum and recalling that multiplication of Fourier transforms is equivalent to convolution in the inverse domain). However, this exact calculation can be sometimes difficult or impossible, if for instance we don't know the density $p_X$, and so sometimes probability bounds are more useful.

Let $Z$ be a non-negative random variable and take $t > 0$. Then

$$\begin{aligned} \mathbb{E}[Z] &\geq \mathbb{E}[Z\,\mathbb{1}_{\{Z\geq t\}}] \\ &\geq \mathbb{E}[t\,\mathbb{1}_{\{Z\geq t\}}] \;=\; t\,\mathbb{P}(Z \geq t) \end{aligned}$$

The result $\mathbb{P}(Z \geq t) \leq \mathbb{E}[Z]/t$ is called *Markov's Inequality*. We can generalize this inequality as follows. Let $\phi$ be any non-decreasing, non-negative function. Then

$$\mathbb{P}(Z \geq t) \;=\; \mathbb{P}(\phi(Z) \geq \phi(t)) \;\leq\; \frac{\mathbb{E}[\phi(Z)]}{\phi(t)}\;.$$

We can use this to get a bound on the probability 'tails' of any random variable $Z$. Let $t > 0$

$$\begin{aligned} \mathbb{P}(|Z - \mathbb{E}[Z]| \geq t) &= P((Z-\mathbb{E}[Z])^2 \geq t^2) \\ &\leq \frac{\mathbb{E}[(Z-\mathbb{E}[Z])^2]}{t^2} \\ &= \frac{\text{Var(Z)}}{t^2}\;, \end{aligned}$$

where Var(Z) denotes the variance of $Z$. This inequality is known as *Chebyshev's Inequality*. If we apply this to the average $\widehat{\mu} = \frac{1}{n}\sum_{i=1}^{n} X_i$, then we have

$$\mathbb{P}(|\widehat{\mu} - \mu| \geq t) \;\leq\; \frac{\sigma^2}{nt^2}$$

where $\mu$ and $\sigma^2$ are the mean and variance of the random variables $\{X_i\}$. This shows that not only is the variance reduced by averaging, but the tails of the distribution (probability of observing values a distance of more than $t$ from the mean) are smaller.

The tail bound given by Chebyshev's Inequality is loose, and much tighter bounds are possible under slightly stronger assumptions. For example, if $X_i \overset{iid}{\sim} \mathcal{N}(\mu, 1)$, then $\widehat{\mu} \sim \mathcal{N}(\mu, 1/n)$. The following tail-bound for the Gaussian density shows that in this case $\mathbb{P}(|\widehat{\mu} - \mu| \geq t) \leq e^{-nt^2/2}$.

**Theorem 7.** *The tail of the standard Gaussian $\mathcal{N}(0,1)$ distribution satisfies the bound for any $t \geq 0$,*

$$\frac{1}{\sqrt{2\pi}} \int\limits_t^{\infty} e^{\frac{-x^2}{2}} dx \;\leq\; \min\left\{ \frac{1}{2} e^{\frac{-t^2}{2}} \,,\, \frac{1}{\sqrt{2\pi\, t^2}} e^{\frac{-t^2}{2}} \right\}$$

*Proof.* Consider

$$R \;:=\; \frac{\frac{1}{\sqrt{2\pi}} \int\limits_t^{\infty} e^{\frac{-x^2}{2}} dx}{e^{-\frac{t^2}{2}}} \;=\; \frac{1}{\sqrt{2\pi}} \int\limits_t^{\infty} e^{\frac{-(x^2-t^2)}{2}} dx \;=\; \frac{1}{\sqrt{2\pi}} \int\limits_t^{\infty} e^{\frac{-(x-t)(x+t)}{2}} dx$$

For the first bound, let $y = x - t$,

$$R \;=\; \frac{1}{\sqrt{2\pi}} \int\limits_0^{\infty} e^{\frac{-y(y+2t)}{2}} dy \;\leq\; \frac{1}{\sqrt{2\pi}} \int\limits_0^{\infty} e^{\frac{-y^2}{2}} dy \;=\; \frac{1}{2}$$

For the second bound, note that

$$R \;\leq\; \frac{1}{\sqrt{2\pi}} \int\limits_t^{\infty} e^{\frac{-2t(x-t)}{2}} dx \;=\; \frac{1}{\sqrt{2\pi}} e^{t^2} \int\limits_t^{\infty} e^{-tx} dx \;=\; \frac{1}{\sqrt{2\pi}} e^{t^2} \frac{e^{-t^2}}{t} \;=\; \frac{1}{\sqrt{2\pi t^2}}$$

□

## 16.1. The Chernoff Method

More generally, if the random variables $\{X_i\}$ are bounded or *sub-Gaussian* (meaning the tails of the probability distribution decay at least as fast as Gaussian tails), then the tails of the average converge exponentially fast in $n$. The key to this sort of result is the so-called *Chernoff bounding method*, based on Markov's inequality and the exponential function (non-decreasing, non-negative). If $Z$ is any real-valued random variable and $s > 0$, then

$$\mathbb{P}(Z > t) \;=\; \mathbb{P}(e^{sZ} > e^{st}) \;\leq\; e^{-st}\, \mathbb{E}[e^{sZ}] \,.$$

We can choose $s > 0$ to minimize this upper bound. In particular, if we define the function

$$\psi^*(t) \;=\; \max_{s>0} \bigl\{ st - \log \mathbb{E}[e^{sZ}] \bigr\} \,,$$

then $\mathbb{P}(Z > t) \le e^{-\psi^*(t)}$.

Exponential bounds of this form can be obtained explicitly for many classes of random variables. One of the most important is the class of sub-Gaussian random variables. A random variable $X$ is said to be sub-Gaussian if there exists a constant $c > 0$ such that $\mathbb{E}[e^{sX}] \le e^{cs^2/2}$ for all $s \in \mathbb{R}$.

**Theorem 8.** *Let $X_1, X_2, ..., X_n$ be independent sub-Gaussian random variables such that $\mathbb{E}[e^{s(X_i - \mathbb{E}[X_i])}] \le e^{cs^2/2}$ for a constant $c > 0$ and $i = 1, \dots, n$. Let $S_n = \sum_{i=1}^n X_i$. Then for any $t > 0$, we have*

$$\mathbb{P}(|S_n - \mathbb{E}[S_n]| \ge t) \le 2\, e^{-t^2/(2nc)}$$

*and equivalently if $\widehat{\mu} := \frac{1}{n} S_n$ we have*

$$\mathbb{P}(|\widehat{\mu} - \mu| \ge t) \le 2\, e^{-nt^2/(2c)}$$

*Proof.*

$$\begin{aligned}
\mathbb{P}(\sum_{i=1}^n X_i - \mathbb{E}[X_i] \ge t) &\le e^{-st}\mathbb{E}\left[e^{s\left(\sum_{i=1}^n X_i - E[X_i]\right)}\right] \\
&= e^{-st}\mathbb{E}\left[\prod_{i=1}^n e^{s(X_i - \mathbb{E}[X_i])}\right] \\
&= e^{-st}\prod_{i=1}^n \mathbb{E}\left[e^{s(X_i - \mathbb{E}[X_i])}\right] \\
&= e^{-st} e^{ncs^2/2} \\
&= e^{-t^2/(2nc)}
\end{aligned}$$

where the last step follows by taking $s = t/(nc)$. □

To apply the result above we need to verify that the sub-Gaussian condition, $\mathbb{E}\left[e^{s(X_i - \mathbb{E}[X_i])}\right] \le e^{cs^2/2}$, holds for some $c > 0$. As the name suggests, the condition holds if the tails of the probability distribution decay like $e^{-t^2/2}$ (or faster).

**Theorem 9.** *If $\mathbb{P}(|X_i - \mathbb{E}[X_i]| \ge t) \le ae^{-bt^2/2}$ holds for constants $a \ge 1$, $b > 0$ and all $t \ge 0$, then*

$$\mathbb{E}[e^{s(X_i - \mathbb{E}[X_i])}] \le e^{4as^2/b} .$$

*Proof.* Let $X$ be a zero-mean random variable satisfying $\mathbb{P}(|X| \ge t) \le ae^{-bt^2/2}$. First note since $X$ has mean zero, Jensen's inequality implies $\mathbb{E}[e^{sX}] \ge e^{s\mathbb{E}X} = 1$ for all $s \in \mathbb{R}$. Thus, if $X_1$ and $X_2$ are two independent copies of $X$, then

$$\mathbb{E}[e^{s(X_1 - X_2)}] = \mathbb{E}[e^{sX_1}]\mathbb{E}[e^{-sX_2}] \ge \mathbb{E}[e^{sX_1}] = \mathbb{E}[e^{sX}] .$$

Thus, we can write

$$\mathbb{E}[e^{sX}] \;\le\; \mathbb{E}[e^{s(X_1 - X_2)}] \;=\; 1 + \sum_{\ell \ge 1} \frac{s^\ell\, \mathbb{E}[(X_1 - X_2)^\ell]}{\ell\,!} .$$

Also, since $\mathbb{E}[(X_1 - X_2)^\ell] = 0$ for $\ell$ odd (since symmetry $X_1 - X_2$ has a symmetric distribution), we have

$$\mathbb{E}[e^{sX}] \;\le\; 1 + \sum_{\ell \ge 1} \frac{s^{2\ell}\, \mathbb{E}[(X_1 - X_2)^{2\ell}]}{(2\ell)!} .$$

Next note that since $x^{2\ell}$ is convex in $x$, by Jensen's inequality we have

$$(X_1/2 - X_2/2)^{2\ell} = ((X_1 + (-X_2))/2)^{2\ell} \leq (X_1^{2\ell} + (-X_2)^{2\ell})/2 = (X_1^{2\ell} + X_2^{2\ell})/2 \, ,$$

which yields

$$\mathbb{E}[(X_1 - X_2)^{2\ell}] \; = \; \mathbb{E}[2^{\ell}(X_1/2 - X_2/2)^{2\ell}] \leq \; 2^{2\ell-1}\big(\mathbb{E}[X_1^{2\ell}] + \mathbb{E}[X_2^{2\ell}]\big) = 2^{2\ell}\mathbb{E}[X^{2\ell}] \, .$$

Next note that $\mathbb{E}[X^{2\ell}] = \int_0^\infty \mathbb{P}(X^{2\ell} > t)\, dt$ and by the change of variables $t = x^{2\ell}$ we have

$$\mathbb{E}[X^{2\ell}] \;\; = \;\; 2\ell \int_0^\infty x^{2\ell-1}\mathbb{P}(|X| > x)\, dx \;\leq\; 2\ell a \int_0^\infty x^{2\ell-1} e^{-bx^2/2}\, dx \, .$$

Now substitute $x = \sqrt{2y/b}$ to get

$$\mathbb{E}[X^{2\ell}] \;\; \leq \;\; (2/b)^{\ell}\ell a \int_0^\infty y^{\ell-1}e^{-y}\, dy \;\; = \;\; (2/b)^{\ell} a\, \ell! \, ,$$

where we recognize that $\int_0^\infty y^{\ell-1}e^{-y}\, dy = \Gamma(\ell) = (\ell-1)!$, the gamma function. So we have $\mathbb{E}[(X_1 - X_2)^{2\ell}] \leq 2^{3\ell}b^{-\ell}a\,\ell! \leq (8a/b)^{\ell}\ell!$ since $a \geq 1$. Now plugging this into the bound for $\mathbb{E}[e^{sX}]$ above, we see that each term in the sum is bounded by $s^{2\ell}(8a/b)^{\ell}\ell!/(2\ell)!$. Since $(2\ell)! \geq 2^{\ell}(\ell!)^2$ each term can be bounded by $(4as^2/b)^{\ell}/\ell!$, and so $\mathbb{E}[e^{sX}] \;\leq\; e^{4as^2/b}$. □

The simplest result of this form is for bounded random variables.

**Theorem 10.** *(Hoeffding's Inequality). Let $X_1, X_2, ..., X_n$ be independent bounded random variables such that $X_i \in [a_i, b_i]$ with probability 1. Let $S_n = \sum_{i=1}^n X_i$. Then for any $t > 0$, we have*

$$\mathbb{P}(|S_n - \mathbb{E}[S_n]| \geq t) \leq 2\, e^{-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}}$$

*Proof.* We prove a special case. The more general result above also has slightly better constants, and its proof is later in the notes. Here, assume that $a \leq X_i \leq b$ with probability 1 for all $i$. Then the following bound

$$\mathbb{P}(|S_n - \mathbb{E}[S_n]| \geq t) \leq 2\, e^{-\frac{t^2}{2\sum_{i=1}^n (b_i - a_i)^2}}$$

follows from Theorem 9 above by noting that if $a \leq X_1, X_2 \leq b$ with probability 1, then $\mathbb{E}[(X_1 - X_2)^{2\ell}] \leq (b-a)^{2\ell}$. □

If the random variables $\{X_i\}$ are binary-valued, then this result is usually referred to as the *Chernoff Bound*. Another proof of Hoeffding's Inequality, which relies Markov's inequality and some elementary concepts from convex analysis, is given in the next section. Note that if the random variables in the average $\widehat{\mu} = \frac{1}{n}\sum_{i=1}^n X_i$ are bounded according to $a \leq X_i \leq b$. Let $c = (b-a)^2$. Then Hoeffding's Inequality implies

$$\mathbb{P}(|\widehat{\mu} - \mu| \geq t) \;\; \leq \;\; 2\, e^{-\frac{2nt^2}{c}} \tag{16.1}$$

In other words, the tails of the distribution of the average are tending to zero at an exponential rate in $n$, much faster than indicated by Chebeyshev's Inequality.

**Example 11.** *Let us revisit the synthia experiments. The biologist has collected* $n$ *observations,* $x_1, \ldots, x_n$*, each corresponding to the fraction of cells that survived in a given experiment. Her estimator of the survival rate is* $\frac{1}{n}\sum_{i=1}^n x_i$*. How confident can she be that this is an accurate estimator of the true survival rate? Let us model her observations as realizations of* $n$ *iid random variables* $X_1, \ldots, X_n$ *with mean* $p$ *and define* $\widehat{p} = \frac{1}{n}\sum_{i=1}^n X_i$*. We say that her estimator is probability approximately correct with non-negative parameters* $(\epsilon, \delta)$ *if*

$$\mathbb{P}(|\widehat{p} - p| > \epsilon) \ \leq \delta$$

*The random variables are bounded between* $0$ *and* $1$ *and so the value of* $c$ *in (16.1) above is equal to* $1$*. For desired accuracy* $\epsilon > 0$ *and confidence* $1 - \delta$*, how many experiments will be sufficient? From (16.1) we equate* $\delta = 2\,\exp(-2n\epsilon^2)$ *which yields* $n \geq \frac{1}{2\epsilon^2}\log(2/\delta)$*. Note that this requires no knowledge of the distribution of the* $\{X_i\}$ *apart from the fact that they are bounded. The result can be summarized as follows. If* $n \geq \frac{1}{2\epsilon^2}\log(2/\delta)$*, then the probability that her estimate is off the mark by more than* $\epsilon$ *is less than* $\delta$*.*

## 16.2. Azuma-Hoeffding Inequality

Hoeffding's inequality can be generalized in a few ways. First, using Doob's inequality we have the stronger bound

$$\mathbb{P}(\max_{1\leq k\leq n} |S_k - \mathbb{E}[S_k]| \geq t) \leq 2\,e^{-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}}$$

Second, we can extend the inequality to martingale sequences. A martingale sequence of random variables $S_0, S_1, \ldots, S_n$ satisfies $\mathbb{E}[S_{k+1}|S_1, \ldots, S_k] = S_k$ for all $k = 0, 1, \ldots, n$. Note that sums of zero-mean and independent random variables are martingales.

**Theorem 11.** *(Azuma's Inequality). Let* $S_0, S_1, \ldots, S_n$ *be martingale sequence of random variables such that for all* $i$ $S_i - S_{i-1} \in [a_i, b_i]$ *with probability* $1$*. Then for any* $t > 0$*, we have*

$$\mathbb{P}(S_n - S_0 \geq t) \leq 2\,e^{-\frac{t^2}{2\sum_{i=1}^n (b_i - a_i)^2}}$$

**Example.** Suppose you makea bet each day. If you bet \$$b$ and have a 50/50 chance of receiving \$$2b$ or losing your money. Let $S_i$ denote your net gain on day $i$ and let $Y_i \in \{-1, +1\}$ be an indicator of the outcome for your bet on day $i$. Here are two strategies.

**Independent Betting:** Always bet \$b. Then the net gain is $S_n = b\sum_{i=1}^n Y_i$

**Recursive Betting:** On day $i$ bet \$$pS_{i-1}$ for some $p \in [0, 1]$. Then the change of wealth on day $i$ can be expressed recursively as $S_i = S_{i-1} + pS_{i-1}Y_i$. This is a martingale.

## 16.3. KL-Based Tail Bounds

It is possible to derive tighter bounds by optimizing the exponent. In particular, if the random variables belong to the exponential family, then the resulting exponent turns out to be a KL-divergence. Below we will work this out for the case of Bernoulli random variables. This results in a tail bound that is as good or better than the subGaussian/Hoeffing bounds above for $[0, 1]$-bounded random variables.

Let $x$ be a non-negative random variable. By Markov's inequality, for any $\lambda > 0$

$$\begin{aligned} \mathbb{P}(x \geq \epsilon) &= \mathbb{P}(e^{\lambda x} \geq e^{\lambda \epsilon}) \\ &\leq e^{-\lambda \epsilon}\, \mathbb{E}[e^{\lambda x}] \\ &= \exp\big(-\big(\lambda \epsilon - \log \mathbb{E}[e^{\lambda x}]\big)\big) \\ &\leq e^{-\psi^*(\epsilon)} \end{aligned}$$

where

$$\psi^*(\epsilon) \;:=\; \sup_{\lambda \in \mathbb{R}} \big(\lambda \epsilon - \log \mathbb{E}[e^{\lambda x}]\big)$$

If $x_1, \ldots, x_n$ are i.i.d. non-negative random variables, then

$$\begin{aligned} \mathbb{P}\left(\frac{1}{n}\sum_{i=1}^n x_i \geq \epsilon\right) &= \mathbb{P}\left(\sum_{i=1}^n x_i \geq n\epsilon\right) \\ &\leq e^{-n\lambda\epsilon}\, \mathbb{E}[e^{\lambda \sum_{i=1}^n x_i}] \\ &= e^{-n\lambda\epsilon}\, \mathbb{E}[e^{n\lambda x_1}] \\ &= \exp\big(-n\big(\lambda\epsilon - \log \mathbb{E}[e^{\lambda x}]\big)\big) \\ &\leq e^{-n\psi^*(\epsilon)} \end{aligned}$$

Now suppose that $x_i$ is Bernoulli with mean $p$. Then $\mathbb{P}\big(\frac{1}{n}\sum_{i=1}^n x_i - p \geq \epsilon\big) \leq \exp(-n\psi^*(p+\epsilon))$. Now consider

$$\begin{aligned} \psi^*(p+\epsilon) &= \sup_{\lambda \in \mathbb{R}} \big(\lambda(p+\epsilon) - \log \mathbb{E}[e^{\lambda x}]\big) \\ &= \sup_{\lambda \in \mathbb{R}} \big(\lambda(p+\epsilon) - \log(1-p+pe^{\lambda})\big) \end{aligned}$$

Setting the derivative of the argument to zero

$$(p+\epsilon) \;=\; \frac{pe^{\lambda}}{1-p+pe^{\lambda}}$$

and solving for $\lambda$ yields

$$\lambda \;=\; \log\left(\frac{(1-p)(p+\epsilon)}{p(1-(p+\epsilon))}\right) ,$$

so $\psi^*(p+\epsilon) = (p+\epsilon)\log(\frac{p+\epsilon}{p}) + (1-(p+\epsilon))\log(\frac{1-(p+\epsilon)}{1-p}) \;=\; \mathrm{KL}(p+\epsilon, p)$. Thus we have

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^n x_i - p \geq \epsilon\right) \;\leq\; \exp(-n\,\mathrm{KL}(p+\epsilon, p)) ,$$

and a similar derivation yields

$$\mathbb{P}\left(p - \frac{1}{n}\sum_{i=1}^n x_i \geq \epsilon\right) \leq \exp(-n\,\mathrm{KL}(p-\epsilon, p)) .$$

These bounds can be used to construct confidence intervals as follows. Let $\widehat{p} := \frac{1}{n}\sum_{i=1}^n x_i$ and consider the bound from above $\mathbb{P}(\widehat{p} - p \geq \epsilon) \;\leq\; \exp(-n\,\mathrm{KL}(p+\epsilon, p))$. In other words, if we choose $\delta$ so that $\mathrm{KL}(p+\epsilon, p) = \log(1/\delta)/n$, then $\widehat{p} - p \leq \epsilon$ with probability at least $1-\delta$. Observe that $\mathrm{KL}(p+\epsilon, p)$ is increasing in $\epsilon \geq 0$. On the event $\frac{1}{n}\sum_{i=1}^n x_i - p \leq \epsilon$ we have $\mathrm{KL}(\widehat{p}, p) \leq \mathrm{KL}(p+\epsilon, p)$. Therefore, we can construct a $(1-\delta)$-confidence upper bound on $p$ as

$$U(\widehat{p}, \delta) \;:=\; \sup\big\{q \geq \widehat{p} \,:\, \mathrm{KL}(\widehat{p}, q) \leq \log(1/\delta)/n\big\} .$$

Similarly, we can construct a $(1-\delta)$-confidence lower bound on $p$ as

$$L(\widehat{p}, \delta) \;:=\; \inf\big\{q \leq \widehat{p} \,:\, \mathrm{KL}(\widehat{p}, q) \leq \log(1/\delta)/n\big\} .$$

## 16.4. Proof of Hoeffding's Inequality

Let $X$ be any random variable and $s > 0$. Note that $\mathbb{P}(X \geq t) = \mathbb{P}(e^{sX} \geq e^{st}) \leq e^{-st}\mathbb{E}[e^{sX}]$ , by using Markov's inequality, and noting that $e^{sx}$ is a non-negative monotone increasing function. For clever choices of $s$ this can be quite a good bound.

Let's look now at $\sum_{i=1}^{n} X_i - \mathbb{E}[X_i]$. Then

$$\begin{aligned} \mathbb{P}(\sum_{i=1}^{n} X_i - \mathbb{E}[X_i] \geq t) &\leq e^{-st}\mathbb{E}\left[e^{s\left(\sum_{i=1}^{n} X_i - E[X_i]\right)}\right] \\ &= e^{-st}\mathbb{E}\left[\prod_{i=1}^{n} e^{s(X_i - \mathbb{E}[X_i])}\right] \\ &= e^{-st}\prod_{i=1}^{n}\mathbb{E}\left[e^{s(X_i - \mathbb{E}[X_i])}\right] , \end{aligned}$$

where the last step follows from the independence of the $X_i$'s. To complete the proof we need to find a good bound for $\mathbb{E}\left[e^{s(X_i - E[X_i])}\right]$.

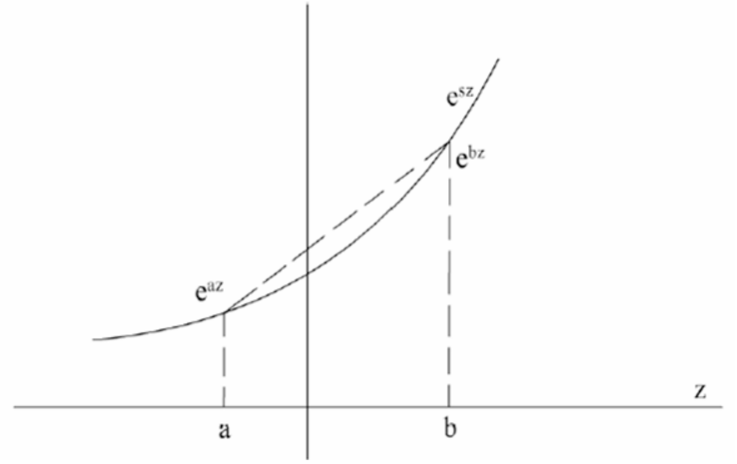


Figure 16.1: Convexity of exponential function.

**Lemma 16.4.1.** *Let $Z$ be a r.v. such that $\mathbb{E}[Z] = 0$ and $a \leq Z \leq b$ with probability one. Then*

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

This upper bound is derived as follows. By the convexity of the exponential function (see Fig. 16.1),

$$e^{sz} \leq \frac{z-a}{b-a}e^{sb} + \frac{b-z}{b-a}e^{sa}, \text{ for } a \leq z \leq b .$$

Thus,

$$\begin{aligned}
\mathbb{E}[e^{sZ}] &\leq \mathbb{E}\left[\frac{Z-a}{b-a}\right]e^{sb} + \mathbb{E}\left[\frac{b-Z}{b-a}\right]e^{sa} \\
&= \frac{b}{b-a}e^{sa} - \frac{a}{b-a}e^{sb}\ , \ \text{since } \mathbb{E}[Z]=0 \\
&= (1-\lambda+\lambda e^{s(b-a)})e^{-\lambda s(b-a)}\ , \ \text{where } \lambda = \frac{-a}{b-a}
\end{aligned}$$

Now let $u = s(b-a)$ and define

$$\phi(u) \equiv -\lambda u + \log(1-\lambda+\lambda e^u)\ ,$$

so that

$$\mathbb{E}[e^{sZ}] \leq (1-\lambda+\lambda e^{s(b-a)})e^{-\lambda s(b-a)} = e^{\phi(u)}\ .$$

We want to find a good upper-bound on $e^{\phi(u)}$. Let's express $\phi(u)$ as its Taylor series with remainder:

$$\phi(u) = \phi(0) + u\phi'(0) + \frac{u^2}{2}\phi''(v) \text{ for some } v \in [0,u]\ .$$

$$\begin{aligned}
\phi'(u) &= -\lambda + \frac{\lambda e^u}{1-\lambda+\lambda e^u} \Rightarrow \phi'(0) = 0 \\
\phi''(u) &= \frac{\lambda e^u}{1-\lambda+\lambda e^u} - \frac{\lambda^2 e^{2u}}{(1-\lambda+\lambda e^u)^2} \\
&= \frac{\lambda e^u}{1-\lambda+\lambda e^u}(1-\frac{\lambda e^u}{1-\lambda+\lambda e^u}) \\
&= \rho(1-\rho)\ ,
\end{aligned}$$

where $\rho = \frac{\lambda e^u}{1-\lambda+\lambda e^u}$. Now note that $\rho(1-\rho) \leq 1/4$, for any value of $\rho$ (the maximum is attained when $\rho = 1/2$, therefore $\phi''(u) \leq 1/4$. So finally we have $\phi(u) \leq \frac{u^2}{8} = \frac{s^2(b-a)^2}{8}$, and therefore

$$\mathbb{E}[e^{sZ}] \leq e^{\frac{s^2(b-a)^2}{8}}\ .$$

Now, we can apply this upper bound to derive Hoeffding's inequality.

$$\begin{aligned}
\mathbb{P}(S_n - \mathbb{E}[S_n] \geq t) &\leq e^{-st}\prod_{i=1}^n \mathbb{E}[e^{s(X_i-\mathbb{E}[X_i])}] \\
&\leq e^{-st}\prod_{i=1}^n e^{\frac{s^2(b_i-a_i)^2}{8}} \\
&= e^{-st}e^{s^2\sum_{i=1}^n \frac{(b_i-a_i)^2}{8}} \\
&= e^{\frac{-2t^2}{\sum_{i=1}^n (b_i-a_i)^2}} \\
&\quad \text{by choosing } \ s = \frac{4t}{\sum_{i=1}^n (b_i-a_i)^2}
\end{aligned}$$

The same result applies to the r.v.'s $-X_1, \ldots, -X_n$, and combining these two results yields the claim of the theorem.

## 16.5. Exercises

1. Let $x$ be a random variable with bounded variance $\mathbb{V}[x] < \infty$. Recall Chebyshev's inequality

$$\mathbb{P}(|x - \mathbb{E}[x]| \geq t) \;\leq\; \frac{\mathbb{V}[x]}{t^2}$$

which is obtained by applying Markov's inequality to $\mathbb{P}(|x - \mathbb{E}[x]|^2 \geq t^2)$. It bounds the probability that $|x - \mathbb{E}[x]| \geq t$, which is a two sided event. In this exercise you will derive a one-sided version of this inequality. Assume first the random variable $x$ has zero mean, meaning $\mathbb{E}[x] = 0$. Let $\sigma^2 = \mathbb{V}[x]$, its variance.

   (a) Recall the Cauchy-Schwarz inequality. For any two random variables $y$ and $z$ we have that

$$\mathbb{E}[yz] \leq \sqrt{\mathbb{E}(y^2)\mathbb{E}(z^2)}\;.$$

   Now write $t = t - \mathbb{E}[x] = \mathbb{E}[t - x] \leq \mathbb{E}[(t - x)\mathbb{1}_{\{t>x\}}]$, and use Cauchy-Schwarz to show that

$$t^2 \leq \mathbb{E}[(t - x)^2]\mathbb{P}(x < t)\;.$$

   (b) Using the fact that $\mathbb{E}[x] = 0$ manipulate the above expression to obtain inequality

$$\mathbb{P}(x \geq t) \leq \frac{\sigma^2}{\sigma^2 + t^2}\;.$$

   (c) Now make only the assumption that $x$ is a random variable for which $\mathbb{V}[x] = \sigma^2 < \infty$. Show that

$$\mathbb{P}(x - \mathbb{E}[x] \geq t) \leq \frac{\sigma^2}{\sigma^2 + t^2}\;.$$

   Note that the inequality in (1c) has the nice feature that the r.h.s. is always smaller than 1, so the bound is never trivial. **Hint:** define a suitable random variable $x$ as a function of $x$ that has zero mean, and apply the result in (b).

   (d) Use the above result to derive a two-sided version of this inequality. Namely, use the union bound to show that

$$\mathbb{P}(|x - \mathbb{E}[x]| \geq t) \leq \frac{2\sigma^2}{\sigma^2 + t^2}\;.$$

   (e) Let $z$ be a standard normal random variable and use the result of (c) to get an upper bound on $\mathbb{P}(z > 1/2)$. Noting that $z$ is symmetric around the origin we have $\mathbb{P}(z > 1/2) = \frac{1}{2}\mathbb{P}(|z| > 1/2)$. Use this and the original Chebyshev inequality to get a bound on $\mathbb{P}(z > 1/2)$. Which is a better bound? (note that we can actually compute $P(z > 1/2)$ numerically and get approximately 0.3085).

2. Hide-and-Seek. Consider the following problem. You are given two coins, one is fair, but the other one is fake and flips heads with probability $1/2 + \varepsilon$, where $\varepsilon > 0$. However, you don't know the value of $\varepsilon$. You would like to identify the fake coin quickly.

   Consider the following strategy. Flip both coins $n$ times and compute the proportion of heads of each coin (say $\widehat{p}_1$ and $\widehat{p}_2$ for coins 1 and 2, respectively). Now deem the coin for which the proportion of heads is larger to be the fake coin. What is the probability we'll make a mistake? Suppose without loss of generality that coin 1 is the fake.

(a) We'll make a mistake if $\widehat{p}_1 < \widehat{p}_2$. That is

$$\mathbb{P}(\widehat{p}_1 - \widehat{p}_2 < 0) \ .$$

Noting that $n(\widehat{p}_1 - \widehat{p}_2)$ is the sum of $2n$ independent random variables use Hoeffding's inequality to show that the probability of making an error is bounded by $e^{-n\varepsilon^2}$.

(b) Now suppose you have $m$ coins, where only one coin is a fake. Similar to what we have done for the two coins we can flip each coins $n$ times, and compute the proportion of times each coin flips heads, denoted by $\widehat{p}_1, \ldots, \widehat{p}_m$. What is the probability of making an error then?

Suppose without loss of generality that the first coin is fake. The probability of making an error is given by

$$\mathbb{P}(\widehat{p}_1 < \widehat{p}_2 \text{ or } \widehat{p}_1 < \widehat{p}_3 \text{ or } \cdots \text{ or } \widehat{p}_1 < \widehat{p}_m) = \mathbb{P}\left(\bigcup_{i=2}^{m} \{\widehat{p}_1 < \widehat{p}_i\}\right) \ .$$

Use Hoeffding's inequality an the union bound to see that the probability of making an error smaller than $(m-1)e^{-n\varepsilon^2}$.

(c) Implement the above procedure with $\varepsilon = 0.1$, $m = 2$ or $m = 100$, and the following values of $n = 10, 100, 500, 1000$. For each choice of parameters $m$ and $n$ repeat the procedure $N = 10000$ times and compute the proportion of runs where the procedure failed to identify the correct coin. Compare these with the bounds you got. How good are the bounds you derived?

# Lecture 17: Probably Approximately Correct (PAC) Learning

Suppose we have training examples of the form $\{\boldsymbol{x}_i, y_i\}$, where $\boldsymbol{x}_i$ are $d$-dimensional features and $y_i$ are scalar and bounded labels/responses. Let $\mathcal{F}$ denote a collection of prediction rules. That is, each $f \in \mathcal{F}$ is a function that maps from features to labels. The aim of Probably Approximately Correct (PAC) learning is to use the training data to select an $\widehat{f}$ from $\mathcal{F}$ so that it's predictions are probably almost as good as the best possible predictor in $\mathcal{F}$.

Perhaps the most natural approach to this task is to choose $\widehat{f}$ to minimize the errors made on the training data. This is called *empirical risk minimization* (ERM). To elaborate, let us introduce some terminology and notation.

**feature space:** $\mathcal{X}$, **feature:** $\boldsymbol{x} \in \mathcal{X}$

**label space:** $\mathcal{Y}$, **label:** $y \in \mathcal{Y}$

**predictor:** $f : \mathcal{X} \to \mathcal{Y}$, **collection of predictors:** $\mathcal{F}$

**loss function:** $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$

For example, a common learning task is binary classification, wherein $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \{-1, +1\}$, and the loss function is the $0/1$-loss defined as follows. Let $y$ be a true label and $\widehat{y}$ be the prediction (e.g., $\widehat{y} = f(\boldsymbol{x})$ for some predictor $f$). The $0/1$-loss is $1$ if $\widehat{y} \neq y$ and $0$ otherwise. We will assume throughout these notes that the losses are bounded by a constant $c$ (e.g., $c = 1$).

The basic premise of the PAC learning framework is that the training examples are i.i.d. samples from a unknown probability distribution $P$, written mathematically as $(\boldsymbol{x}_i, y_i) \overset{i.i.d.}{\sim} P$. The goal of learning is to select a predictor that minimizes the *expected loss* or *risk*:

$$\min_{f \in \mathcal{F}} \mathbb{E}_{(\boldsymbol{x},y)\sim P}[\ell(y, f(\boldsymbol{x}))] \ .$$

ERM is the optimization:

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{n} \ell(y_i, f(\boldsymbol{x}_i)) \ .$$

A predictor that achieves the minimum is denoted by $\widehat{f}$ and is called an empirical risk minimizer. Note that for large $n$, the average $\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(\boldsymbol{x}_i)) \overset{a.s.}{\to} \mathbb{E}_{(\boldsymbol{x},y)\sim P}[\ell(y, f(\boldsymbol{x}))]$, since the losses $\ell(y_i, f(\boldsymbol{x}_i))$ are i.i.d. So, ERM seems like a sensible approach to trying to select an $f$ that comes close to minimizing the risk.

## 17.1. Analysis of Empirical Risk Minimization

To simplify the notation, let us denote the risk and empirical risk as follows:

$$\begin{aligned} R(f) &= \mathbb{E}_{(\boldsymbol{x},y)\sim P}[\ell(y, f(\boldsymbol{x}))] \\ \widehat{R}(f) &= \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(\boldsymbol{x}_i)) \end{aligned}$$

Note that $\mathbb{E}[\widehat{R}(f)] = R(f)$. Markov's inequality provides a (weak) upper bound on the deviation of the empirical risk from the true risk. Assuming the losses are bounded in $[0, c]$

$$\mathbb{P}(|\widehat{R}(f) - R(f)| > t) \;\leq\; \frac{\mathbb{E}[|\widehat{R}(f) - R(f)|^2]}{t^2} \;\leq\; \frac{c^2}{4nt^2} \,,$$

since the maximum variance of $c$-bounded random variables $c^2/4$. This bound can be improved using Chernoff's bounding technique

$$\mathbb{P}(\widehat{R}(f) - R(f) > t) \;=\; \inf_{\lambda>0} \mathbb{P}(e^{\lambda(\widehat{R}(f)-R(f))} > e^{\lambda t}) \;\leq\; e^{-2nt^2/c^2} \,,$$

and by the union bound $\mathbb{P}(|\widehat{R}(f) - R(f)| > t) \leq 2\, e^{-2nt^2/c^2}$. For example, in the case of $0/1$-loss, the losses are i.i.d. binary random variables and we may take $c = 1$.

Recall that the empirical risk minimizer is $\widehat{f} = \arg\min_{f\in\mathcal{F}} \widehat{R}(f)$. If $\widehat{R}(f) \approx R(f)$ for all $f \in \mathcal{F}$, then the minimizer of $\widehat{R}$ should be "close to" the minimizer of $R$. The bound above shows that $\widehat{R}(f)$ is close to $R(f)$ for a specific function $f$. To guarantee that it is close for all $f \in \mathcal{F}$ we must consider $\mathbb{P}\Big(\bigcup_{f\in\mathcal{F}}\Big\{|\widehat{R}(f) - R(f)| > t\Big\}\Big)$, the probability that $\widehat{R}$ deviates significantly from $R$ for one or more of the functions. To bound this probability, *we will assume that $\mathcal{F}$ is finite and denote the number of functions in $\mathcal{F}$ by $|\mathcal{F}|$*. Then we have

$$\begin{aligned}\mathbb{P}\left(\bigcup_{f\in\mathcal{F}}\Big\{|\widehat{R}(f) - R(f)| > t\Big\}\right) \;&\leq\; \sum_{f\in\mathcal{F}} \mathbb{P}(|\widehat{R}(f) - R(f)| > t) \\ &\leq\; 2|\mathcal{F}|e^{-2nt^2/c^2}\end{aligned}$$

Let $\delta = 2|\mathcal{F}|e^{-2nt^2/c^2}$. The bound above says that $\widehat{R}$ is uniformly close to $R$ over $\mathcal{F}$, with probability at least $1 - \delta$.

Now let $f^\star = \arg\min_{f\in\mathcal{F}} \mathbb{E}_{(\boldsymbol{x},y)\sim P}[\ell(y, f(\boldsymbol{x}))]$, which is the best predictor in $\mathcal{F}$. This is the $f$ we would choose if we knew the data distribution $P$. ERM tries to select a predictor that is approximately as good as $f^\star$ using only the training examples. Consider the following inequalities which hold with probability at least $1 - \delta$:

$$\begin{aligned}R(\widehat{f}) \;&\leq\; \widehat{R}(\widehat{f}) + t \\ &\leq\; \widehat{R}(f^*) + t \,,\ \text{since } \widehat{f} \text{ minimizes } \widehat{R} \\ &\leq\; R(f^*) + 2t \,.\end{aligned}$$

This shows that the risk of $\widehat{f}$ is at most $2t$ larger than $R(f^*)$, the minimum risk, with probability at least $1 - \delta$. In other words, $\widehat{f}$ is probably approximately correct. Let us also define

$$\epsilon \;:=\; 2t \;=\; \sqrt{\frac{2c^2 \log(2|\mathcal{F}|/\delta)}{n}} \,.$$

Then we have $R(\widehat{f}) - R(f^*) \leq \epsilon$ with probability at least $1 - \delta$, and we say that $\widehat{f}$ is $(\epsilon, \delta)$-PAC.

Notice that the approximation error decreases with $n$ and increases with the size of $\mathcal{F}$ (although only logarithmically in $|\mathcal{F}|$). Another way to view these results is to consider the expected risk of $\widehat{f}$. Note that $R(\widehat{f})$ is a random variable, since $\widehat{f}$ is random (it depends on the random set of training examples). Taking the expectation over the training examples, we have

$$\begin{aligned}\mathbb{E}[R(\widehat{f})] \;&\leq\; (1-\delta)\left(R(f^*) + \sqrt{\frac{2c^2 \log(2|\mathcal{F}|/\delta)}{n}}\right) \;+\; \delta \max_{f\in\mathcal{F}} R(f) \\ &\leq\; R(f^*) + \sqrt{\frac{2c^2 \log(2|\mathcal{F}|/\delta)}{n}} \;+\; c\,\delta \,.\end{aligned}$$

Now since the second term is at least $O(\frac{1}{\sqrt{n}})$, take $\delta = \frac{1}{\sqrt{n}}$ to obtain

$$\begin{aligned}\mathbb{E}[R(\widehat{f})] &\leq R(f^*) + \sqrt{\frac{2c^2 \log(2|\mathcal{F}|\sqrt{n})}{n}} + \frac{c}{\sqrt{n}} \\ &= R(f^*) + O\left(\sqrt{\frac{\log|\mathcal{F}| + \log n}{n}}\right) .\end{aligned}$$

These bounds show that if the number of samples $n = O(\log|\mathcal{F}|)$, then the class $\mathcal{F}$ is PAC-learnable.

## 17.2. Exercises

1. Consider a classification setting with training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where $\boldsymbol{x}_i \in [0,1]^d$ and $y_i \in \{-1,+1\}$.

   (a) First consider $d = 1$ and a discrete set of classifiers $\mathcal{F}_m$ of the form $f_{2j}(x) = 2\mathbb{1}_{\{x \geq j/m\}} - 1$ and $f_{2j+1} = -2\mathbb{1}_{\{x \geq j/m\}} + 1$ , for $j = 0, 1, \ldots, m$ and an integer $m \geq 1$. Also consider any classifier of form $f_{t,\sigma}(x) = \sigma(2\mathbb{1}_{\{x \geq t\}} - 1)$, with $t \in [0,1]$ and $\sigma \in \{-1,+1\}$. How large must $m$ be so that for fixed $\varepsilon > 0$, $t$, and $\sigma$

   $$\min_{f \in \mathcal{F}_m} \int |f(x) - f_{t,\sigma}(x)|\, dx \leq \varepsilon .$$

   (b) Let $R(f) = \mathbb{E}[\mathbb{1}_{\{f(\boldsymbol{x}) \neq y\}}]$ and $\widehat{R}(f) = \frac{1}{n}\sum_{i=1}^n \mathbb{1}_{\{f(\boldsymbol{x}_i) \neq y_i\}}$. Derive a PAC generalization bound which states that with probability at least $1 - \delta$

   $$R(f) \leq \widehat{R}(f) + \sqrt{\frac{\log(|F_m|/\delta)}{n/2}} , \text{ for all } f \in \mathcal{F}_m ,$$

   where $|\mathcal{F}_m|$ is the cardinality of $\mathcal{F}_m$. Use this to obtain a generalization error bound for the minimum empirical risk minimizer $\widehat{f} = \arg\min_{f \in \mathcal{F}_m} \widehat{R}(f)$. Simplify the bound so that it depends only on $\widehat{R}(\widehat{f})$, $m$, $n$, and $\delta$.

   (c) Now consider the general case with $d \geq 1$. Specify the design of a discrete set of classifiers $\mathcal{F}_\varepsilon$ such that for any linear classifer $g$ on $[0,1]^d$

   $$\min_{f \in \mathcal{F}_\varepsilon} \int_{\boldsymbol{x} \in [0,1]^d} |f(\boldsymbol{x}) - g(\boldsymbol{x})|\, d\boldsymbol{x} \leq \varepsilon .$$

   Derive a generalization error bound for this case expressed in terms of $\widehat{R}(\widehat{f})$, $\varepsilon$, $n$, $d$ and $\delta$.

2. Recall the histogram classifier studied in Lecture 2.4. Let $\mathcal{F}_M$ denote the set of all histogram classifiers with $M$ equal-volume bins $B_1, \ldots, B_M$.

   (a) Show that $\widehat{f}_n^H$ defined in Lecture 2.4 minimizes the empirical risk.

   (b) Derive a bound on $\mathbb{E}[R(\widehat{f}_n^H)] - \min_{f \in \mathcal{F}_M} R(f)$ using the empirical risk bounds developed in Section 17.1.

# Lecture 18: Learning in Infinite Model Classes

Let $R(f) = \mathbb{P}(f(\boldsymbol{x}) \neq y)$ and let $\widehat{R}(f) = \frac{1}{n}\sum_{i=1}^n \mathbb{1}_{\{f(\boldsymbol{x}_i)\neq y_i\}}$. The PAC bound for a finite model class $\mathcal{F}$ may be stated as

$$\mathbb{P}\Big(\max_{f\in\mathcal{F}} |\widehat{R}(f) - R(f)| \geq \epsilon\Big) \ \leq\ 2|\mathcal{F}|e^{-2n\epsilon^2} \ ,$$

where $|\mathcal{F}|$ is the number of models in the class. We can also state this as a generalization bound. For any $\delta > 0$ and for every $f \in \mathcal{F}$, with probability at least $1-\delta$

$$R(f) \ \leq\ \widehat{R}(f) \ +\ \sqrt{\frac{\log(|\mathcal{F}|/\delta)}{2n}}.$$

Since this holds for every model in $\mathcal{F}$, it holds for the the empirical risk minimizer $\widehat{f} = \arg\min_{f\in\mathcal{F}} \widehat{R}(f)$. So if $\min_{f\in\mathcal{F}} \widehat{R}(f)$ is small and if $n$ is large compared to $\log(|\mathcal{F}|/\delta)$, then $R(\widehat{f}) = \mathbb{P}(\widehat{f}(\boldsymbol{x}) \neq y)$ is probably small too.

Now we will generalize this sort of result to cases in which the model class is infinite. Linear classifiers are a prime example of such a class. Let $\mathcal{F}$ denote the set of all linear classifiers of the form

$$f_{\boldsymbol{w}}(\boldsymbol{x}) \ =\ \begin{cases} +1 & \text{if } \boldsymbol{w}^T\boldsymbol{x} + b \geq 0 \\ -1 & \text{if } \boldsymbol{w}^T\boldsymbol{x} + b < 0 \end{cases}$$

for some $\boldsymbol{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$. There are an infinite number of choices for the weight $\boldsymbol{w}$ and bias $b$, and so $|\mathcal{F}| = \infty$.

However, suppose we have a training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$. Each linear classifier defines a hyperplane (separator) in $\mathbb{R}^d$. Consider any one of the hyperplanes. It splits $\mathbb{R}^d$ into two halfspaces. Note that we can move the hyperplane until it just touches one or more of the points, without changing the binary labeling it produces. In particular, if $n \geq 2d$ there must be at least $d$ points in one of the halfspaces, and we can move the hyperplane until it touches $d$ points. Since $d$ points (in general position) define a hyperplane in $\mathbb{R}^d$, these $d$ points define two of the possible labelings that can be realized by linear classifiers (two because we can assign $+1$ to either one of the halfspaces it generates). In effect, this particular hyperplane represents all the linear classifiers that produce the same labelings of the dataset. Since there are $\binom{n}{d}$ unique subsets of $d$ points, and there are at least $2\binom{n}{d}$ possible ways linear classifiers can label the dataset. In fact, the total number is larger than this, since we also must consider cases where fewer than $d$ points define the hyperplane separator. The total number of unique labelings of $n$ points in $d$ dimensions using hyperplanes is $\mathcal{S}(\mathcal{F}, n) := 2\sum_{k=0}^d \binom{n-1}{k}$. In other words, there are effectively at most $\mathcal{S}(\mathcal{F}, n)$ unique linear classifiers for $n$ points in $\mathbb{R}^d$; each may be *represented* by a specific hyperplane and pair of linear classifiers [9], as we defined above. The quantity $\mathcal{S}(\mathcal{F}, n)$ is called the *shatter coefficient* of $\mathcal{F}$ (we will discuss this further later).

Since the shatter coefficient is the effective size of the class, and it is finite since $n$ is finite, it is tempting to apply the standard PAC bound for finite classes. However, there is a subtle issue. The finite set representative linear classifiers depends on the specific locations of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. This means that set of representative classifiers is data-dependent and the argument used for the standard PAC bound assumed the set of classifiers to be fixed and deterministic (not depending on the training examples). The problem is that if we consider a representative classifier $f$ that depends on $\{\boldsymbol{x}_i\}$, then the errors $\mathbb{1}_{\{f(\boldsymbol{x}_i)\neq y_i\}}$ are no longer independent random variables.

## 18.1. Rademacher Complexity

Let $\mathcal{F}$ be an infinite model class. Our goal is to derive a bound of the form

$$\mathbb{P}\Big(\sup_{f\in\mathcal{F}} |\widehat{R}(f) - R(f)| \geq \epsilon\Big) \;\leq\; B(n,\epsilon)\;,$$

for some function $B(n,\epsilon)$. Bounds of this type are called uniform deviation bounds, since they bound the largest deviations over all possible $f \in \mathcal{F}$. For the class of linear models discussed above, we will show that

$$B(n,\epsilon) = 8\mathcal{S}(\mathcal{F},n)e^{-n\epsilon^2/32}$$

will suffice. This shows that the shatter coefficient, which is the number of linear classifiers needed to represent all possible labelings of the dataset, indeed plays the role that $|\mathcal{F}|$ played in the case of finite classes. *Rademacher complexity* is a standard approach to construct uniform deviation bounds.

Let $\ell_1(f),\dots,\ell_n(f)$ be iid bounded random variables satisfying $\ell_i(f) \in [0,1]$ and indexed by functions $f \in \mathcal{F}$. In the next section, we will view $\ell_i(f)$ as the prediction error using $f$ on the $i$th training example. In other words, $\ell_i(f)$ could represent the 0-1 loss $\mathbb{1}_{\{f(\boldsymbol{x}_i)\neq y_i\}}$ or any other bounded loss function. Denote the ensemble and empirical mean by $R(f) = \mathbb{E}[\ell_1(f)]$ and $\widehat{R}_n(f) = \frac{1}{n}\sum_{i=1}^n \ell_i(f)$, respectively. We will use the following lemma.

**Lemma 18.1.1.** *(McDiarmid's Bounded Difference Inequality).* Let $g : \mathbb{R}^n \to \mathbb{R}$ be a function satisfying

$$\sup_{\ell_1,\dots,\ell_n,\ell_i'} |g(\ell_1,\dots,\ell_{i-1},\ell_i,\ell_{i+1},\dots,\ell_n) - g(\ell_1,\dots,\ell_{i-1},\ell_i',\ell_{i+1},\dots,\ell_n)| \;\leq\; c_i$$

for some constant $c_i \geq 0$ for $i = 1,\dots,n$. Then if $\ell_1,\dots,\ell_n$ are independent random variables

$$\mathbb{P}(g(\ell_1,\dots,\ell_n) - \mathbb{E}[g(\ell_1,\dots,\ell_n)] \;\geq\; t) \;\leq\; \exp\left(-\frac{2t^2}{\sum_{i=1}^n c_i^2}\right)\,.$$

*Proof.* For short hand we will let $G = g(\ell_1,\dots,\ell_n)$ and define $V_i = \mathbb{E}[G|\ell_1,\dots,\ell_i] - \mathbb{E}[G|\ell_1,\dots,\ell_{i-1}]$. Then $G - \mathbb{E}[G] = \sum_{i=1}^n \mathbb{E}[G|\ell_1,\dots,\ell_i] - \mathbb{E}[G|\ell_1,\dots,\ell_{i-1}] = \sum_{i=1}^n V_i$. Note that $\mathbb{E}[V_i|\ell_1,\dots,\ell_{i-1}] = 0$ and the absolute value of $V_i|\ell_1,\dots,\ell_{i-1}$ is less than or equal to $c_i$ by assumption. Therefore, since it is a bounded random variable, its moment generating function satisfies $\mathbb{E}[e^{sV_i}|\ell_1,\dots,\ell_{i-1}] \leq e^{s^2c_i^2/8}$. Now for any $t > 0$ we have

$$\begin{aligned}
\mathbb{P}(G - \mathbb{E}[G] \geq t) \;&=\; \mathbb{P}\left(\sum_{i=1}^n V_i \geq t\right)\\
&=\; \mathbb{P}(e^{s\sum_{i=1}^n V_i} \geq e^{st}) \;\leq\; e^{-st}\mathbb{E}[e^{s\sum_{i=1}^n V_i}]\;\text{, Markov's inequality}\\
&=\; e^{-st}\,\mathbb{E}\Big[e^{s\sum_{i=1}^{n-1} V_i}\mathbb{E}[e^{sV_n}|\ell_1,\dots,\ell_{n-1}]\Big]\\
&\leq\; e^{-st}\,e^{s^2c_n^2/8}\,\mathbb{E}[e^{s\sum_{i=1}^{n-1} V_i}]\\
&\;\vdots\\
&\leq\; e^{-st}\,e^{s^2\sum_{i=1}^n c_i^2/8}
\end{aligned}$$

The result follows by taking $s = 4t/\sum_{i=1}^n c_i^2$. □

The function $g(\ell_1, \ldots, \ell_n) := \sup_{f\in\mathcal{F}} R(f) - \widehat{R}_n(f)$ satisfies the bounded differences assumption with $c_i = 1/n$. This is verified as follows. Suppose that $\ell_j = \ell_j'$ for all $j \in \{1, \ldots, n\}$ except a certain index $i$. Then

$$\begin{aligned}
|g(\ell_1, \ldots, \ell_n) - g(\ell_1', \ldots, , \ell_n')| &\leq \Big|\sup_{f\in\mathcal{F}} n^{-1}\sum_j (\ell_j - \mathbb{E}\ell_j) - \sup_{f\in\mathcal{F}} n^{-1}\sum_j (\ell_j' - \mathbb{E}\ell_j')\Big| \\
&\leq \Big|\sup_{f\in\mathcal{F}} \Big(n^{-1}\sum_j (\ell_j - \mathbb{E}\ell_j) - n^{-1}\sum_j (\ell_j' - \mathbb{E}\ell_j')\Big)\Big| \\
&\leq \Big|\sup_{f\in\mathcal{F}} (\ell_i - \mathbb{E}\ell_i)/n - (\ell_i' - \mathbb{E}\ell_i')/n\Big| \;\leq\; 1/n \;,
\end{aligned}$$

since $\ell_i, \ell_i' \in [0, 1]$. Note that this also holds for $-g(\ell_1, \ldots, \ell_n) = \sup_{f\in\mathcal{F}} \widehat{R}_n(f) - R(f)$. Everything that follows holds for $g$ and $-g$.

Therefore, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$,

$$\sup_{f\in\mathcal{F}} R(f) - \widehat{R}_n(f) \;\leq\; \mathbb{E}\Big[\sup_{f\in\mathcal{F}} R(f) - \widehat{R}_n(f)\Big] \;+\; \sqrt{\frac{\log(1/\delta)}{2n}} \;.$$

The next step is to bound $\mathbb{E}\Big[\sup_{f\in\mathcal{F}} R(f) - \widehat{R}_n(f)\Big]$. Since $R(f)$ depends on the underlying (and unknown) data distribution, our approach will aim to eliminate this unknown quantity by a symmetrization step. We will bound the difference between $R(f)$ and $\widehat{R}_n(f)$ by the difference between two independent versions of $\widehat{R}_n(f)$. Intuitively, this makes sense since both are equal to $R(f)$ in expectation and the difference between two independent empirical risks will tend to have even larger deviations.

To this end, introduce a "ghost sample" $\boldsymbol{\ell}' = \{\ell_1'(f), \ldots, \ell_n'(f)\}$ independent of $\boldsymbol{\ell} = \{\ell_1(f), \ldots, \ell_n(f)\}$ and distributed identically. Let $\widehat{R}_n'(f)$ denote the empirical mean of the ghost sample. Then by Jensen's inequality

$$\mathbb{E}_{\boldsymbol{\ell}}\Big[\sup_{f\in\mathcal{F}} R(f) - \widehat{R}_n(f)\Big] \;=\; \mathbb{E}_{\boldsymbol{\ell}}\Big[\sup_{f\in\mathcal{F}}\Big(\mathbb{E}_{\boldsymbol{\ell}'}\Big[\widehat{R}_n'(f) - \widehat{R}_n(f)\,\Big|\,\ell_1(f), \ldots, \ell_n(f)\Big]\Big)\Big] \;\leq\; \mathbb{E}_{\boldsymbol{\ell},\boldsymbol{\ell}'}\Big[\sup_{f\in\mathcal{F}} \widehat{R}_n'(f) - \widehat{R}_n(f)\Big] \;.$$

Above we are using the simple fact that the supremum of an average is less than or equal to the average of the supremum.

Now let $\boldsymbol{\sigma} = \{\sigma_1, \ldots, \sigma_n\}$ be independent Rademacher random variables with $\mathbb{P}(\sigma_i = \pm 1) = 1/2$, independent of $\ell_i$ and $\ell_i'$. Then we have

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\ell},\boldsymbol{\ell}'}\Big[\sup_{f\in\mathcal{F}} \widehat{R}_n'(f) - \widehat{R}_n(f)\Big] &= \mathbb{E}_{\boldsymbol{\ell},\boldsymbol{\ell}'}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \ell_i'(f) - \ell_i(f)\right] \\
&= \mathbb{E}_{\boldsymbol{\ell},\boldsymbol{\ell}',\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \sigma_i(\ell_i'(f) - \ell_i(f))\right] \text{, by symmetry} \\
&\leq \mathbb{E}_{\boldsymbol{\ell},\boldsymbol{\ell}',\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \sigma_i\ell_i'(f) + \sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \sigma_i\ell_i(f)\right] \\
&= 2\,\mathbb{E}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \sigma_i\ell_i(f)\right] .
\end{aligned}$$

The *Rademacher Complexity* of the class $\mathcal{F}$ with loss function $\ell$ is

$$\mathfrak{R}_n(\ell(\mathcal{F})) \;:=\; 2\,\mathbb{E}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \sigma_i\ell_i(f)\right] ,$$

where the expectation is taken with respect to $\{\sigma_i\}$ and $\{\ell_i\}$. The Rademacher complexity measures how easy it is to find a function in $\mathcal{F}$ that correlates with random sign patterns [2]. Richer classes of functions have a higher complexity. If we take the expectation only over $\{\sigma_i\}$, holding $\{\ell_i\}$ fixed, then we have the so-called *empirical* Rademacher complexity

$$\widehat{\mathfrak{R}}_n(\ell(\mathcal{F})) \;:=\; 2\,\mathbb{E}\left[\sup_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^{n}\sigma_i\ell_i(f)\middle|\{\ell_i\}\right].$$

By McDiarmid's inequality, $\mathfrak{R}_n(\ell(\mathcal{F})) \le \widehat{\mathfrak{R}}_n(\ell(\mathcal{F})) + 2\sqrt{\frac{\log(1/\delta)}{2n}}$; the factor of 2 appears due to the factor of 2 in the definition of the Rademacher complexity. The empirical Rademacher complexity is sometimes useful because we can easily construct a Monte Carlo estimate of it. Putting everything together, we have the following result.

**Theorem 12.** *With probability at least* $1-\delta$

$$\sup_{f\in\mathcal{F}} R(f) - \widehat{R}_n(f) \;\le\; \mathfrak{R}_n(\ell(\mathcal{F})) \;+\; \sqrt{\frac{\log(1/\delta)}{2n}}$$

*and*

$$\sup_{f\in\mathcal{F}} R(f) - \widehat{R}_n(f) \;\le\; \widehat{\mathfrak{R}}_n(\ell(\mathcal{F})) \;+\; 3\sqrt{\frac{\log(2/\delta)}{2n}}$$

Note that $1/\delta$ replaced by $2/\delta$ because we are union bounding over the original Rademacher bound and the McDiarmid bound on the empirical Rademacher bound. As mentioned above, we could have just as easily considered $\widehat{R}_n(f) - R(f)$ and obtained the same result. So we can also state two-sided bounds such as

$$\sup_{f\in\mathcal{F}} |R(f) - \widehat{R}_n(f)| \;\le\; \mathfrak{R}_n(\ell(\mathcal{F})) \;+\; \sqrt{\frac{\log(2/\delta)}{2n}}$$

with probability at least $1-\delta$. Note that the inequalities above can be stated as generalization bounds: For any $\delta > 0$ and for all $f \in \mathcal{F}$, with probability at least $1-\delta$

$$R(f) \;\le\; \widehat{R}_n(f) \;+\; \mathfrak{R}_n(\ell(\mathcal{F})) \;+\; \sqrt{\frac{\log(1/\delta)}{2n}}\,.$$

In particular, the bound holds for $\widehat{f}$ that minimizes $\widehat{R}$.

## 18.2. Generalization Bounds for Classification with 0/1 Loss

Now let us specialize the results above to the case of binary classification with $0/1$ loss. The Rademacher complexity above $\mathfrak{R}_n(\ell(\mathcal{F}))$ depends implicitly on the choice of loss. Specifically, consider the $0/1$ loss, $\ell_i(f) = \mathbb{1}_{\{f(\boldsymbol{x}_i)\neq y_i\}}$. Define the empirical Rademacher complexity for $\mathcal{F}$ (without a loss) to be

$$\widehat{\mathfrak{R}}_n(\mathcal{F}) \;:=\; \mathbb{E}_\sigma\Big[\sup_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^{n}\sigma_i f(\boldsymbol{x}_i)\Big]\,.$$

The expectation $\mathfrak{R}_n(\mathcal{F}) := \mathbb{E}[\widehat{\mathfrak{R}}_n(\mathcal{F})]$, where the expectation is with respect to $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n$, is what is normally referred to as the Rademacher complexity of the class $\mathcal{F}$. For $\ell$ chosen to be $0/1$ loss, we can relate these complexities as follows.

$$
\begin{aligned}
\widehat{\mathfrak{R}}_n(\ell(\mathcal{F})) &= 2\mathbb{E}_\sigma\Big[\sup_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^n \sigma_i \mathbb{1}_{\{f(\boldsymbol{x}_i)\neq y_i\}}\Big] \\
&= 2\mathbb{E}_\sigma\Big[\sup_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^n \sigma_i\Big(\frac{1-y_if(\boldsymbol{x}_i)}{2}\Big)\Big] \\
&= \mathbb{E}_\sigma\Big[\frac{1}{n}\sum_{i=1}^n \sigma_i + \sup_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^n \sigma_i(-y_i)f(\boldsymbol{x}_i)\Big] \\
&= \mathbb{E}_\sigma\Big[\sup_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^n \sigma_i y_i f(\boldsymbol{x}_i)\Big] \;=\; \widehat{\mathfrak{R}}_n(\mathcal{F})\,.
\end{aligned}
$$

Also for $0/1$ loss, $R(f) = \mathbb{P}(f(\boldsymbol{x}) \neq y)$ and $\widehat{R}(f) = \frac{1}{n}\sum_{i=1}^n \mathbb{1}_{\{f(\boldsymbol{x}_i)\neq y_i\}}$. Using the results above, for any $\delta > 0$ and for all $f \in \mathcal{F}$, with probability at least $1-\delta$

$$
R(f) \;\leq\; \widehat{R}_n(f) \;+\; \mathfrak{R}_n(\mathcal{F}) \;+\; \sqrt{\frac{\log(1/\delta)}{2n}}\,. \tag{18.1}
$$

## 18.3. Exercises

1. The Hamming distance between two sequences $\boldsymbol{\ell} = (\ell_1,\ldots,\ell_n)$ and $\boldsymbol{\ell}' = (\ell_1',\ldots,\ell_n')$ is defined to be the number of coordinates where $\ell_i \neq \ell_i'$. Denote this by $d_H(\boldsymbol{\ell},\boldsymbol{\ell}')$. For any set $A$ of such sequences, the distance of $\boldsymbol{\ell}$ to $A$ is $d_H(\boldsymbol{\ell}, A) := \min_{\boldsymbol{\ell}'\in A} d_H(\boldsymbol{\ell},\boldsymbol{\ell}')$. If $\ell_1,\ldots,\ell_n$ are independent random variables, prove that
$$
\mathbb{P}\Big(d_H(\boldsymbol{\ell},A) - \mathbb{E}[d_H(\boldsymbol{\ell},A)] \geq t\Big) \;\leq\; 2e^{2t^2/n}
$$

2. Give a formal proof of the bound $\mathfrak{R}_n(\mathcal{F}) \leq \widehat{\mathfrak{R}}_n(\mathcal{F}) \;+\; 2\sqrt{\frac{\log(1/\delta)}{2n}}$.

3. This exercise considers an application of the Rademacher complexity bounds to histogram classifiers. Consider a classification setting with training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where $\boldsymbol{x}_i \in [0,1]^d$ and $y_i \in \{-1,+1\}$.

   (a) Let $\mathcal{F}$ be the set of histogram classifiers with $m$ bins $B_1,\ldots,B_m$. Any $f \in \mathcal{F}$ can be written as $f(\boldsymbol{x}) = \sum_{j=1}^m L_j(f)\,\mathbb{1}_{\{\boldsymbol{x}\in B_j\}}$, where $L_j(f) \in \{-1,+1\}$ is the label $f$ assigns to points in bin $B_j$. Give an expression for the histogram classifier $\widehat{f}$ that minimizes the error on the training set.

   (b) Show that the empirical Rademacher complexity can be bounded as follows
   $$
   \widehat{\mathfrak{R}}_n(\mathcal{F}) \;\leq\; \frac{1}{n}\sum_{j=1}^m \sqrt{n_j}\,,
   $$
   where $n_j$ is the number of training examples falling in bin $j$.

   (c) Use $\widehat{\mathfrak{R}}_n(\mathcal{F})$ to bound the generalization error $\mathbb{P}(y \neq \widehat{f}(\boldsymbol{x}))$. Explain how this bound can provide guidance on selecting the number of bins $m$.

# Lecture 19: Vapnik-Chervonenkis Theory

Recall the set of linear classifiers in $\mathbb{R}^d$

$$\mathcal{F} \;=\; \left\{ f : \mathbb{R}^d \to \{-1,+1\}\,,\; f(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{w}^T\boldsymbol{x} + b),\; \boldsymbol{w} \in \mathbb{R}^d,\; b \in \mathbb{R} \right\}.$$

There are an infinite number of choices for the weight $\boldsymbol{w}$ and bias $b$, and so $|\mathcal{F}| = \infty$. However, for any training dataset consisting of $n$ examples there are at most $\mathcal{S}(\mathcal{F}, n) := 2\sum_{k=0}^{d}\binom{n-1}{k}$ possible ways linear classifiers can label the dataset [9]. The quantity $\mathcal{S}(\mathcal{F}, n)$ is called the *shatter coefficient* of $\mathcal{F}$.

**Remark:** Throughout this lecture we will exclusively consider the $0/1$ loss function in all our analysis and results.

## 19.1. Shatter Coefficient and VC Dimension

Vapnik-Chervonenkis (VC) theory is based on a generalization of this idea. The main intuition behind VC theory is that, although a collection of classifiers may be infinite, using a finite set of training data to select a good rule effectively reduces the number of different classifiers we need to consider. We can measure the effective size of a class $\mathcal{F}$ using the *shatter coefficient*. Suppose we have a training set $D_n = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ for a binary classification problem with labels $y_i \in \{-1, +1\}$. Because there are only two possible labels for each $\boldsymbol{x}_i$, each classifier in $\mathcal{F}$ produces a binary label sequence

$$\Big(f(x_1), \ldots, f(x_n)\Big) \in \{-1,+1\}^n\,.$$

There at are at most $2^n$ distinct sequences, but often not all sequences can be generated by functions in $\mathcal{F}$. Let $\mathcal{S}(\mathcal{F}, n)$ be the maximum number of labeling sequences the class $\mathcal{F}$ induces over $n$ training points in the feature space $\mathcal{X}$. More formally,

**Definition 19.1.1.** The **shatter coefficient** of class $\mathcal{F}$ is defined as

$$\mathcal{S}(\mathcal{F}, n) = \max_{x_1,\ldots,x_n \in \mathcal{X}} \left|\left\{ (f(x_1), \ldots, f(x_n)) \in \{-1,+1\}^n,\; f \in \mathcal{F} \right\}\right|,$$

where $|\cdot|$ denotes the number of elements in the set.

Clearly $\mathcal{S}(\mathcal{F}, n) \le 2^n$, but often it is much smaller. The class of linear classifiers is a canonical example.

The shatter coefficient $\mathcal{S}(\mathcal{F}, n)$ is a measure of the "effective size" of $\mathcal{F}$ with respect to a training set of size $n$. The sample complexity of selecting a classifier from a set of size $N$ is $O(\log N)$, because of the union bound. Thus, $\log \mathcal{S}(\mathcal{F}, n)$ measures the "effective dimension" of $\mathcal{F}$.

**Definition 19.1.2.** The Vapnik-Chervonenkis (VC) dimension is defined as the largest integer $k$ such that $\mathcal{S}(\mathcal{F}, k) = 2^k$. The VC dimension of a class $\mathcal{F}$ is denoted by $V(\mathcal{F})$.

Note that the VC dimension is not a function of the number of training data. We have the following result, presented here without a proof.

**Lemma 19.1.3.** *Sauer's Lemma:*

$$S(\mathcal{F}, n) \le (n+1)^{V(\mathcal{F})}\,.$$

**Example 19.1.4. Linear classifiers in $d$ dimensions.** *Note that if $n \le d+1$, then every possible labeling sequence can be realized by a linear classifier, but it is not possible if $n > d+1$. Thus, the VC dimension of linear classifiers in $\mathbb{R}^d$ is $d+1$. Sauer's Lemma shows that $\mathcal{S}(\mathcal{F}, n) \le (n+1)^{(d+1)}$. Recall that $\mathcal{S}(\mathcal{F}, n) = 2\sum_{k=0}^{d}\binom{n-1}{k}$, which indeed is less than Sauer's bound.*

## 19.2. The VC Inequality

The main result in VC theory is the following theorem, which yields generalization bounds in terms of the shatter coefficient and VC dimension.

**Theorem 19.2.1.** *(Vapnik-Chervonenkis '71): Let $\mathcal{F}$ be a class of binary classifiers with shatter coefficient $\mathcal{S}(\mathcal{F}, n)$. For any $\epsilon > 0$*

$$\mathbb{P}\left(\sup_{f\in\mathcal{F}} |\widehat{R}_n(f) - R(f)| \geq \epsilon\right) \;\leq\; 2\mathcal{S}(\mathcal{F}, n)e^{-n\epsilon^2/8} \,,$$

*or equivalently for any $\delta > 0$, with probability at least $1-\delta$*

$$\sup_{f\in\mathcal{F}} |\widehat{R}_n(f) - R(f)| \;\leq\; \sqrt{\frac{8(\log \mathcal{S}(\mathcal{F}, n) + \log(2/\delta))}{n}} \,.$$

Using Sauer's bound, we can state a generalization bound of the in terms of the VC dimension $V(\mathcal{F})$. For any $\delta > 0$ and every $f \in \mathcal{F}$, with probability at least $1-\delta$

$$R(f) \;\leq\; \widehat{R}_n(f) \;+\; \sqrt{\frac{8(V(\mathcal{F})\log(n+1) + \log(1/\delta))}{n}} \,.$$

*Proof.* We will use the following Rademacher complexity bound from 18.1. For any $\delta > 0$, with probability at least $1-\delta$

$$\sup_{f\in\mathcal{F}} \left(R(f) - \widehat{R}_n(f)\right) \;\leq\; \mathfrak{R}_n(\mathcal{F}) \;+\; \sqrt{\frac{\log(1/\delta)}{2n}} \,.$$

Let $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ denote a dataset and let $\mathcal{F}_D = \Big\{(f(x_1), \ldots, f(x_n)) \;:\; f \in \mathcal{F}\Big\}$, the set of all labelings of the datapoints that can be generated using classifiers in $\mathcal{F}$. The shatter coefficient $S(\mathcal{F}, n)$ bounds the cardinality of $\mathcal{F}_D$. We can bound the Rademacher complexity as follows.

$$\mathfrak{R}_n(\mathcal{F}) \;=\; \mathbb{E}_D\left[\mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \sigma_i f(\boldsymbol{x}_i)\right]\right] \;=\; \mathbb{E}_D\left[\mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}_D} \frac{1}{n}\sum_{i=1}^n \sigma_i f(\boldsymbol{x}_i)\right]\right] .$$

We next apply the following lemma (which we will prove later).

**Lemma 19.2.2.** *(Massart's Lemma) Let $A \subset \mathbb{R}^n$, with $|A| < \infty$. Set $r = \max_{u\in A}\|u\|_2$ and let $u = (u_1, \ldots, u_n)^T$. Then*

$$\mathbb{E}_\sigma\left[\frac{1}{n}\sup_{u\in A}\sum_{i=1}^n \sigma_i u_i\right] \;\leq\; \frac{r\sqrt{2\log|A|}}{n} \,.$$

In our setting, take $A = \mathcal{F}_D$ and notice that for every sequence $u \in \mathcal{F}_D$ we have $\|u\|_2^2 = \sum_{i=1}^n (\pm 1)^2 = n$. So applying the lemma we have

$$\mathfrak{R}_n(\mathcal{F}) \;\leq\; \mathbb{E}_D\left[\frac{\sqrt{2n\log|\mathcal{F}_D|}}{n}\right] \;\leq\; \mathbb{E}_D\left[\frac{\sqrt{2n\log S(\mathcal{F}, n)}}{n}\right] \;=\; \sqrt{\frac{2\log S(\mathcal{F}, n)}{n}} \,.$$

Thus we have

$$\sup_{f\in\mathcal{F}} \left(R(f) - \widehat{R}_n(f)\right) \;\leq\; \sqrt{\frac{2\log S(\mathcal{F}, n)}{n}} \;+\; \sqrt{\frac{\log(1/\delta)}{2n}} \,.$$

Observe that for any $a, b \geq 0$ we have $\sqrt{a} + \sqrt{b} \leq \sqrt{a+b} + \sqrt{a+b} = 2\sqrt{a+b}$. Therefore,

$$\sup_{f \in \mathcal{F}} \Big( R(f) - \widehat{R}_n(f) \Big) \;\leq\; \sqrt{\frac{8\big(\log S(\mathcal{F}, n) + \log(1/\delta)\big)}{n}} \;.$$

The two-sided version follows by repeating the same argument for $\sup_{f \in \mathcal{F}} \Big( \widehat{R}_n(f) - R(f) \Big)$ and union bounding over the two cases. □

### 19.2.1. Proof of Massart's Lemma

**Lemma 19.2.3.** *(Massart's Lemma) Let $A \subset \mathbb{R}^n$, with $|A| < \infty$. Set $r = \max_{u \in A} \|u\|_2$. Then*

$$\mathbb{E}_\sigma \left[ \frac{1}{n} \sup_{u \in A} \sum_{i=1}^n \sigma_i u_i \right] \;\leq\; \frac{r\sqrt{2 \log |A|}}{n} \;.$$

*Proof.* For all $t \geq 0$ we have

$$\begin{aligned}
\exp\Big( t\, \mathbb{E}_\sigma \Big[ \sup_{u \in A} \sum_{i=1}^n \sigma_i u_i \Big] \Big) \;&=\; \exp\Big( \mathbb{E}_\sigma \Big[ t \sup_{u \in A} \sum_{i=1}^n \sigma_i u_i \Big] \Big) \\
&\leq\; \mathbb{E}_\sigma \Big[ \exp\Big( t \sup_{u \in A} \sum_{i=1}^n \sigma_i u_i \Big) \Big] \;, \text{ by Jensen's inequality} \\
&=\; \mathbb{E}_\sigma \Big[ \sup_{u \in A} \exp\Big( t \sum_{i=1}^n \sigma_i u_i \Big) \Big] \;, \text{ since } \exp \text{ is strictly increasing} \\
&\leq\; \sum_{u \in A} \mathbb{E}_\sigma \Big[ \exp\Big( t \sum_{i=1}^n \sigma_i u_i \Big) \Big] \;, \text{ by the union bound} \\
&=\; \sum_{u \in A} \prod_{i=1}^n \mathbb{E}_{\sigma_i} \Big[ \exp\Big( t \sigma_i u_i \Big) \Big] \;, \text{ since } \sigma_i \text{ are iid } \;.
\end{aligned}$$

Recall that we encountered expectations of exponential functions this form in the Chernoff bounding method. In particular, Lemma 16.4.1 shows that since the random variable $\sigma_i u_i$ has mean zero and is bounded $|\sigma_i u_i| \leq |u_i|$, we have $\mathbb{E}_{\sigma_i}[e^{t\sigma_i u_i}] \leq e^{t^2 (2u_i)^2/8}$. Thus, we have

$$\begin{aligned}
\exp\Big( t\, \mathbb{E}_\sigma \Big[ \sup_{u \in A} \sum_{i=1}^n \sigma_i u_i \Big] \Big) \;&\leq\; \sum_{u \in A} \prod_{i=1}^n \mathbb{E}_{\sigma_i} \Big[ \exp\Big( t \sigma_i u_i \Big) \Big] \;\leq\; \sum_{u \in A} \prod_{i=1}^n e^{t^2 (2u_i)^2/8} \\
&=\; \sum_{u \in A} \exp\Big( \frac{t^2 \|u\|_2^2}{2} \Big) \;\leq\; \sum_{u \in A} \exp\Big( \frac{t^2 r^2}{2} \Big) \;=\; |A| \exp\Big( \frac{t^2 r^2}{2} \Big)
\end{aligned}$$

Now take the log of both sides and divide by $t$ to obtain

$$\mathbb{E}_\sigma \Big[ \sup_{u \in A} \sum_{i=1}^n \sigma_i u_i \Big] \;\leq\; \frac{\log |A|}{t} + \frac{t r^2}{2} \;.$$

Choosing $t$ to minimize this bound yields the result. □

## 19.3. Exercises

1. Consider a binary classification problem with features in $[0,1]^d$ and binary labels $+1$ and $-1$. For classifiers, let's use the set of all linear (hyperplane) classifiers in $\mathbb{R}^d$.

   (a) How many training examples are sufficient to learn linear classifier that (with large probability) has a probability of error at most $\epsilon > 0$ larger than that of the best possible linear classifier?

   (b) Suppose that the Bayes optimal classifier $f^*$ (i.e., the classifier that minimizes the probability of error) is nonlinear and that the minimum probability of error achievable using a linear classifer is $0 < \gamma < 1$ larger than the probability of error of the Bayes classifier. How many samples would suffice to learn a linear classifier with probability of error at most $\eta > \gamma$ in this case?

2. Show the following monotonicity property of VC-dimension: if $\mathcal{F}$ and $\mathcal{F}'$ are hypothesis classes with $\mathcal{F} \subset \mathcal{F}'$, then the VC-dimension of $\mathcal{F}'$ is greater than or equal to that of $\mathcal{F}$.

3. Suppose that $\mathcal{F}$ is the set of all axis aligned rectangles in $\mathbb{R}^d$. How many training examples are needed to learn an $(\epsilon, \delta)$-PAC classifier in $\mathcal{F}$.

# Lecture 20: Learning with Continuous Loss Functions

Let $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ be iid training examples and let $\ell$ be a loss function. Consider the empirical risk function $\widehat{R}_n(f) = \frac{1}{n}\sum_{i=1}^n \ell(y_i, f(\boldsymbol{x}_i))$ and its expectation $R(f) = \mathbb{E}[\ell(y, f(\boldsymbol{x})]$. Assume the losses are bounded in $[0,1]$. Theorem 12 states that with probability at least $1-\delta$

$$\sup_{f\in\mathcal{F}} R(f) - \widehat{R}_n(f) \ \leq\ \mathfrak{R}_n(\ell(\mathcal{F})) \ +\ \sqrt{\frac{\log(1/\delta)}{2n}}$$

where the Rademacher complexity with respect to $\ell$ is

$$\mathfrak{R}_n(\ell(\mathcal{F})) \ =\ 2\,\mathbb{E}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \sigma_i \ell(y_i, f(\boldsymbol{x}_i))\right].$$

We will apply these bounds to continuous loss functions like

$$\begin{aligned} \text{hinge:} \qquad & \ell(y, f(\boldsymbol{x})) = \max(0, 1 - yf(\boldsymbol{x})) \\ \text{logistic:} \qquad & \ell(y, f(\boldsymbol{x})) = \log(1 + \exp(-yf(\boldsymbol{x}))) \end{aligned}$$

If the losses are bounded in $[0,1]$, then $\mathfrak{R}_n(\ell(\mathcal{F})) \leq 1$. Observe that $\mathfrak{R}_n(\ell(\mathcal{F})) = 1$ leads to a vacuous bound since if the losses are bounded in $[0,1]$, then trivially $R(f) \leq 1$. So generalization bounds based on Rademacher complexity are only meaningful if $\mathfrak{R}_n(\ell(F)) < 1$. In fact, the bounds are only interesting if $\mathfrak{R}_n(\ell(\mathcal{F}))$ decays as $n$ grows. Why might this sort of decay happen? Recall that one interpretation of $\mathfrak{R}_n(\ell(\mathcal{F}))$ is that it measures how well functions in $\mathcal{F}$ can correlate with a random sequence of ±1 values. As $n$ grows, it becomes more and more difficult to match such a sequence. Even if $\mathcal{F}$ is infinite it may not be possible to do this for large $n$. To see this, start with a finite set $\mathcal{F}_0$ with a certain Rademacher complexity. Suppose that we create a number of additional functions that are small perturbations of each $f \in \mathcal{F}_0$. Let $\mathcal{F}_1$ denote the new larger set. Since each $f \in \mathcal{F}_1$ is very close to one of the $f \in \mathcal{F}_0$, the arguments $y_i f(\boldsymbol{x}_i)$ of the losses above will not change much, and we have $\mathfrak{R}_n(\ell(\mathcal{F}_1)) \approx \mathfrak{R}_n(\ell(\mathcal{F}_0))$. In other words, increasing the set does not necessarily increase the Rademacher complexity.

## 20.1. Generalization Bounds for Continuous Loss Functions

The $0/1$ loss is natural for binary classification since its expectation is the probability of misclassification. However, minimizing the empirical risk is difficult due to the discontinuous nature of $0/1$ loss. This is the main reason we work with continuous loss functions like the hinge loss or logistic loss. These can be easily minimized using gradient descent procedures. We will assume that $y_i = \pm 1$ and thus the sign of the $f(\boldsymbol{x}_i)$ is the predicted label. Note that the hinge and logistic losses are functions of $z = yf(\boldsymbol{x})$. So we will express the loss as $\ell(yf(\boldsymbol{x}))$, a function of a single scalar argument $z = yf(\boldsymbol{x})$.

The bounds in Theorem 12 apply in such cases (assuming the losses are bounded). However, we would like to be able to compute or bound the Rademacher complexity $\mathfrak{R}_n(\ell(\mathcal{F}))$, so that we can determine its dependence on $n$. To the end, we will first bound $\mathfrak{R}_n(\ell(\mathcal{F}))$ in terms of

$$\mathfrak{R}_n(\mathcal{F}) \ =\ 2\,\mathbb{E}\Big[\sup_{f\in\mathcal{F}} \sum_{i=1}^n \sigma_i f(\boldsymbol{x}_i)\Big]$$

using the following lemma, and then bound $\mathfrak{R}_n(\mathcal{F})$. It is often easier to bound $\mathfrak{R}_n(\mathcal{F})$ than bounding $\mathfrak{R}_n(\ell(\mathcal{F}))$ directly.

**Lemma 20.1.1.** *Assume the loss $\ell$ is L-Lipschitz: $|\ell(z) - \ell(z')| \leq L\,|z - z'|$. Then*

$$\mathfrak{R}_n(\ell(\mathcal{F})) \;=\; 2\,\mathbb{E}\sup_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^{n}\sigma_i\ell\big(y_i f(\boldsymbol{x}_i)\big) \;\leq\; 2L\,\mathbb{E}\sup_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^{n}\sigma_i y_i f(\boldsymbol{x}_i) \;=\; 2L\,\mathbb{E}\sup_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^{n}\sigma_i f(\boldsymbol{x}_i)$$

The hinge and logistic losses are $1$-Lipschitz functions. We will prove the lemma in a bit, but first let us apply it to an interesting case.

### 20.1.1. Application to Linear Classifiers

**Theorem 20.1.2.** *Consider linear classifiers of the form $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x}$, with $\|\boldsymbol{w}\|_2 \leq 1$ and $\|\boldsymbol{x}\|_2 \leq 1$. Let $B_1^d = \{\boldsymbol{x}\in\mathbb{R}^d : \|\boldsymbol{x}\|_2 \leq 1\}$ and define*

$$\mathcal{F}_{\text{lin}} := \left\{f : B_1^d \to \mathbb{R}\,,\; f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x}\,,\; \|\boldsymbol{w}\|_2 \leq 1\right\}.$$

*Assume that the loss $\ell$ is L-Lipschitz. Then*

$$\mathfrak{R}_n(\ell(\mathcal{F}_{\text{lin}})) \;\leq\; 2L\,\mathfrak{R}_n(\mathcal{F}_{\text{lin}}) \;\leq\; \frac{2L}{\sqrt{n}}\,.$$

*Proof.* First inequality follows from Lemma 20.1.1. Let $\|\cdot\|$ denote the Euclidean norm $\|\cdot\|_2$. Now consider

$$\begin{aligned}
\mathfrak{R}_n(\mathcal{F}_{\text{lin}}) \;&=\; \mathbb{E}\Big[\sup_{f\in\mathcal{F}_{\text{lin}}}\frac{1}{n}\sum_{i=1}^{n}\sigma_i f(\boldsymbol{x}_i)\Big] \;=\; \mathbb{E}\Big[\sup_{\|\boldsymbol{w}\|\leq 1}\frac{1}{n}\sum_{i=1}^{n}\sigma_i\boldsymbol{w}^T\boldsymbol{x}_i\Big] \\
&\leq\; \mathbb{E}\Big[\sup_{\|\boldsymbol{w}\|\leq 1}\frac{1}{n}\|\boldsymbol{w}\|\,\Big\|\sum_{i=1}^{n}\sigma_i\boldsymbol{x}_i\Big\|\Big]\,,\text{ by Cauchy-Schwartz inequality} \\
&\leq\; \frac{1}{n}\mathbb{E}\Big[\Big\|\sum_{i=1}^{n}\sigma_i\boldsymbol{x}_i\Big\|\Big] \;=\; \frac{1}{n}\mathbb{E}\Big[\sqrt{\Big\|\sum_{i=1}^{n}\sigma_i\boldsymbol{x}_i\Big\|^2}\Big] \;\leq\; \frac{1}{n}\sqrt{\mathbb{E}\Big[\Big\|\sum_{i=1}^{n}\sigma_i\boldsymbol{x}_i\Big\|^2\Big]}\,,\text{ by Jensen's inequality} \\
&\leq\; \frac{1}{n}\sqrt{\mathbb{E}\left[\sum_{i,j=1}^{n}\sigma_i\sigma_j\boldsymbol{x}_i^T\boldsymbol{x}_j\right]} \;=\; \frac{1}{n}\sqrt{\sum_{i=1}^{n}\|\boldsymbol{x}_i\|^2} \;=\; \frac{1}{\sqrt{n}}\,.
\end{aligned}$$

□

To conclude, we have shown the following result (which also holds for logistic loss).

**Corollary 1.** *Assume $y_i \in [-1,1]$ and $\|\boldsymbol{x}_i\|_2 \leq 1$ for $i = 1,\ldots,n$, and let $\widehat{\boldsymbol{w}}$ be a solution to the convex optimization*

$$\min_{\boldsymbol{w}:\|\boldsymbol{w}\|_2\leq 1}\sum_{i=1}^{n}\left(1 - y_i\boldsymbol{w}^T\boldsymbol{x}_i\right)_+\,.$$

*Then with probability at least $1-\delta$*

$$\mathbb{P}(y \neq \mathit{sign}(\widehat{\boldsymbol{w}}^T\boldsymbol{x})) \;\leq\; \frac{1}{n}\sum_{i=1}^{n}\left(1 - y_i\widehat{\boldsymbol{w}}^T\boldsymbol{x}_i\right)_+ \;+\; \frac{2}{\sqrt{n}} \;+\; \sqrt{\frac{2\log 1/\delta}{n}}\,.$$

In this case, the losses are bounded in $[0,2]$ rather than $[0,1]$; this follows from the Cauchy-Schwartz inequality. Therefore, the last term in the bound is $\sqrt{\frac{2\log 1/\delta}{n}}$ instead of $\sqrt{\frac{\log 1/\delta}{2n}}$ (this follows from the application McDiramid's inequality). Although here we specialized our discussion to the hinge loss, similar arguments can be used for other Lipschitz losses such as the logistic loss, which is also 1-Lipschitz.

## 20.2. Proof of Lemma 20.1.1

**Lemma 20.2.1.** *(Lipschitz Property of Rademacher Complexity). Suppose $\{\phi_i\}$ and $\{\psi_i\}$ are two sets of functions on domain $\mathcal{F}$ such that for each $i$ and $f, f' \in \mathcal{F}$,*

$$|\phi_i(f) - \phi_i(f')| \leq |\psi_i(f) - \psi_i(f')| \quad \text{(A1)}$$

*Let $\sigma = \{\sigma_1, \sigma_2, \dots\}$ be a sequence of i.i.d. Rademacher random variables. Then*

$$\mathbb{E}_\sigma\Big[\sup_f \sum_{i=1}^n \sigma_i \phi_i(f)\Big] \;\leq\; \mathbb{E}_\sigma\Big[\sup_f \sum_{i=1}^n \sigma_i \psi_i(f)\Big] \,.$$

*Proof.* The proof is similar to one given in [19].

$$\begin{aligned}
\mathbb{E}_{\sigma_1,\dots,\sigma_n}\Big[\sup_f \sum_{i=1}^n \sigma_i\phi_i(f)\Big] \;&=\; \mathbb{E}_{\sigma_2,\dots,\sigma_n}\mathbb{E}_{\sigma_1}\Big[\sup_f \Big\{\sigma_1\phi_1(f) + \sum_{i=2}^n \sigma_i\phi_i(f)\Big\}\Big] \\
&=\; \mathbb{E}_{\sigma_2,\dots,\sigma_n}\Big[\frac{1}{2}\sup_f \Big\{\phi_1(f) + \sum_{i=2}^n \sigma_i\phi_i(f)\Big\} \;+\; \frac{1}{2}\sup_{f'}\Big\{-\phi_1(f') + \sum_{i=2}^n \sigma_i\phi_i(f')\Big\}\Big] \\
&=\; \mathbb{E}_{\sigma_2,\dots,\sigma_n}\Big[\sup_{f,f'}\Big\{\frac{\phi_1(f)-\phi_1(f')}{2} + \frac{\sum_{i=2}^n \sigma_i\phi_i(f) + \sum_{i=2}^n \sigma_i\phi_i(f')}{2}\Big\}\Big] \\
&=\; \mathbb{E}_{\sigma_2,\dots,\sigma_n}\Big[\sup_{f,f'}\Big\{\frac{|\phi_1(f)-\phi_1(f')|}{2} + \frac{\sum_{i=2}^n \sigma_i\phi_i(f) + \sum_{i=2}^n \sigma_i\phi_i(f')}{2}\Big\}\Big] \text{ , (\#)} \\
&\leq\; \mathbb{E}_{\sigma_2,\dots,\sigma_n}\Big[\sup_{f,f'}\Big\{\frac{|\psi_1(f)-\psi_1(f')|}{2} + \frac{\sum_{i=2}^n \sigma_i\phi_i(f) + \sum_{i=2}^n \sigma_i\phi_i(f')}{2}\Big\}\Big] \text{ , by A1} \\
&=\; \mathbb{E}_{\sigma_2,\dots,\sigma_n}\Big[\sup_{f,f'}\Big\{\frac{\psi_1(f)-\psi_1(f')}{2} + \frac{\sum_{i=2}^n \sigma_i\phi_i(f) + \sum_{i=2}^n \sigma_i\phi_i(f')}{2}\Big\}\Big] \text{ , (\#)} \\
&=\; \mathbb{E}_{\sigma_2,\dots,\sigma_n}\mathbb{E}_{\sigma_1}\Big[\sup_f\Big\{\sigma_1\psi_1(f) + \sum_{i=2}^n \sigma_i\phi_i(f)\Big\}\Big] \text{ , reverse of step in 2nd line above.}
\end{aligned}$$

To see that step (#) holds, note that if $\phi_1(f) < \phi_1(f')$ (or $\psi_1(f) < \psi_1(f')$), then swapping $f$ and $f'$ increases the difference term while leaving the others fixed. Now continue from the last line above by repeating the same argument with respect to $\sigma_2$. This yields

$$\mathbb{E}_{\sigma_1,\dots,\sigma_n}\Big[\sup_f \sum_{i=1}^n \sigma_i\phi_i(f)\Big] \;\leq\; \mathbb{E}_{\sigma_1,\dots,\sigma_n}\Big[\sup_f\Big\{\sum_{i=1}^2 \sigma_i\psi_i(f) + \sum_{i=3}^n \sigma_i\phi_i(f)\Big\}\Big]$$

Continuing this process for $\sigma_3, \dots, \sigma_n$ yields the claimed inequality.

□

**Corollary 2.** *Consider a finite collection of stochastic processes* $z_1(f), z_2(f), \ldots, z_n(f)$ *indexed by* $f \in f$. *Let* $\sigma_1, \ldots, \sigma_n$ *be independent Rademacher random variables (i.e., each* $\sigma_i$ *takes values* $-1$ *and* $+1$ *with probabilities* $1/2$*). Then for any* $L-$*Lipschitz function* $\ell$ *(i.e.,* $|\ell(z) - \ell(z')| \leq L|z - z'|$, $\forall z, z'$*)*

$$\mathbb{E}\left[\sup_{f\in f}\sum_{i=1}^{n}\sigma_i\,\ell\big(z_i(f)\big)\right] \;\leq\; L\,\mathbb{E}\left[\sup_{f\in f}\sum_{i=1}^{n}\sigma_i\,z_i(f)\right].$$

*Proof.* Apply the lemma above with $\phi_i(f) = \varphi\big(z_i(f)\big)$, $\psi_i(f) = Lz_i(f)$.

□

The following 2-sided generalization of the corollary above, sometimes called the "contraction" property of Rademacher complexity, can be found in [6] (note the extra factor of 2 due to the absolute value).

**Corollary 3.** *Consider a finite collection of stochastic processes* $z_1(f), z_2(f), \ldots, z_n(f)$ *indexed by* $f \in f$. *Let* $\sigma_1, \ldots, \sigma_n$ *be independent Rademacher random variables (i.e., each* $\sigma_i$ *takes values* $-1$ *and* $+1$ *with probabilities* $1/2$*). Then for any* $L-$*Lipschitz function* $\ell$ *(i.e.,* $|\ell(z) - \ell(z')| \leq L|z - z'|$, $\forall z, z'$*)*

$$\mathbb{E}\left[\sup_{f\in f}\left|\sum_{i=1}^{n}\sigma_i\,\ell\big(z_i(f)\big)\right|\right] \;\leq\; 2L\,\mathbb{E}\left[\sup_{f\in f}\left|\sum_{i=1}^{n}\sigma_i\,z_i(f)\right|\right].$$

## 20.3. Exercises

1. Prove that the hinge and logistic loss functions are 1-Lipschitz.

2. Let $\rho_1, \rho_2 > 0$. Derive a generalization bound like the one in (1) for the class

$$\mathcal{F}_{\text{lin}}(\rho_1, \rho_2) \;=\; \Big\{f : f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x},\; \|\boldsymbol{w}\|_2 \leq \rho_1,\; \|\boldsymbol{x}\|_2 \leq \rho_2\Big\}$$

3. Derive a generalization bound like the one in (1) for the class

$$\mathcal{F}_{\text{lin},1,\infty} \;=\; \Big\{f : f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x},\; \|\boldsymbol{w}\|_1 \leq 1,\; \|\boldsymbol{x}\|_\infty \leq 1\Big\}$$

   Hint: Use the fact that $|\boldsymbol{w}^T\boldsymbol{x}| \leq \|\boldsymbol{w}\|_1\|\boldsymbol{x}\|_\infty$ and Massart's inequality.

4. Consider quadratic prediction functions of the form $f(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{W}\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{w}$, with $\|\boldsymbol{W}\|_2 \leq 1$ and $\|\boldsymbol{w}\|_2 \leq 1$, where $\|\boldsymbol{W}\|_2$ is the spectral norm of $\boldsymbol{W}$. Assuming $\|\boldsymbol{x}\|_2 \leq 1$, derive a generalization bound analogous to the one in Corollary 1.

# Lecture 21: Introduction to Function Spaces

Let $\mathcal{F}$ be a class of functions. Given a training dataset $\{(\boldsymbol{x}_i, y_i)\}$, we can pose the empirical risk minimization problem

$$\min_{f\in\mathcal{F}} \sum_{i=1}^{n} \ell(y_i, f(\boldsymbol{x}_i)) \ .$$

The solution is a function $f \in \mathcal{F}$ that minimizes the sum of losses on the training data (i.e., that fits the training data best). Learning linear classifiers is a good example. The*function space* of all (homogeneous) linear functions on $\mathbb{R}^d$ is

$$\mathcal{F} \ = \ \Big\{ f \ : \ f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} \, , \ \boldsymbol{w} \in \mathbb{R}^d \Big\}.$$

We can limit this further by restricting the norm of $\boldsymbol{w}$. Define the*function class*

$$\mathcal{F}_B \ = \ \Big\{ f \ : \ f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} \, , \ \|\boldsymbol{w}\| \le B \Big\}.$$

In this case we have

$$\min_{f\in\mathcal{F}_B} \sum_{i=1}^{n} \ell(y_i, f(\boldsymbol{x}_i)) \ \equiv \ \min_{\boldsymbol{w}:\|\boldsymbol{w}\|\le B} \sum_{i=1}^{n} \ell(y_i, \boldsymbol{w}^T\boldsymbol{x}_i)$$

and we can solve the optimization using gradient descent methods. The problem is also equivalent to the regularized optimization

$$\min_{\boldsymbol{w}\in\mathbb{R}^d} \sum_{i=1}^{n} \ell(y_i, \boldsymbol{w}^T\boldsymbol{x}_i) \ + \ \lambda_B \|\boldsymbol{w}\|^2$$

for a certain regularization parameter $\lambda_B > 0$.

We can generalize this to other function classes as follows. Let $\|f\|$ denote the *norm* of the function $f$. Norms map functions to real numbers, and satisfy: $\|f\| \ge 0$, $\|f + g\| \le \|f\| + \|g\|$, and if $\|f\| = 0$, then $f = 0$. There are many ways to define norms on functions, which we will discuss in more detail below. Norms based on the integrals of $f$ or its $k$th derivatives $f^{(k)}$ are common. For example, $\|f\| := \sum_{k=0}^{K} \big( \int |f^{(k)}(x)|^2 \, dx \big)^{1/2}$. Given a norm, we can define a function space $\mathcal{F} = \{f \ : \ \|f\| < \infty\}$ and classes of functions as $\mathcal{F}_B = \{f \ : \ \|f\| \le B\}$ and then consider learning with this class by solving optimizations of the form

$$\min_{f\in\mathcal{F}_B} \sum_{i=1}^{n} \ell(y_i, f(\boldsymbol{x}_i)) \ \text{ or } \ \min_{f\in\mathcal{F}} \sum_{i=1}^{n} \ell(y_i, f(\boldsymbol{x}_i)) \ + \ \lambda_B \|f\|^2 \ .$$

This leads to a number of questions:

1. What sorts of norms and function classes are useful in machine learning?
2. Can we derive generalization bounds for classes defined in terms of function norms?
3. If $\mathcal{F}$ is not defined in terms of a finite number of parameters, can we efficiently solve the optimizations?

## 21.1. Constructions of Function Classes

### 21.1.1. Parameteric Classes

The simplest way to construct a function class is in terms of a set of parameters or weights. The linear functions in the class above are one example, and polynomial functions and neural networks others. For example, a single output, two-layer neural network has the form $f(\boldsymbol{x}) = \sum_{k=1}^{K} v_k \phi(\boldsymbol{w}_k^T \boldsymbol{x} + b_k)$, where the activation function $\phi$ is fixed (e.g., Rectified Linear Unit) and the input and output weights, $\boldsymbol{w}_k$ and $v_k$, and the biases $b_k$ are the learnable parameters. This gives us the neural network class

$$\mathcal{F} \ = \ \left\{ f \ : \ f(\boldsymbol{x}) = \sum_{k=1}^{K} v_k \phi(\boldsymbol{w}_k^T \boldsymbol{x} + b_k), \ \boldsymbol{w}_k \in \mathbb{R}^d, \ v_k, b_k \in \mathbb{R} \right\} .$$

We could further limit this class by placing constraints on the size of the weights and biases.

### 21.1.2. Atomic Classes

Consider a family of parameterized functions $\{\phi_{\boldsymbol{w}}\}_{\boldsymbol{w} \in \mathcal{W}}$ where each $\phi_{\boldsymbol{w}}$ has the same functional form parameterized by the choice of $\boldsymbol{w}$. The set $\mathcal{W}$ could be a certain subset of $\mathbb{R}^d$. We call the functions *atoms* since we can take weight combinations of them to synthesize more complex functions. The neurons in a neural network are an example of atoms. Fourier basis functions of the form $\phi_{\boldsymbol{w}}(\boldsymbol{x}) := e^{i\boldsymbol{w}^T\boldsymbol{x}}$, where $i = \sqrt{-1}$, are another. We can define function classes in terms of atoms. For example, if $\mathcal{W}$ is finite or countably infinite we can define a class like

$$\mathcal{F}_B \ = \ \left\{ f \ : \ f(\boldsymbol{x}) = \sum_{\boldsymbol{w} \in \mathcal{W}} v(\boldsymbol{w}) \phi_{\boldsymbol{w}}(\boldsymbol{x}) \ , \ v(\boldsymbol{w}) \in \mathbb{R}, \sum_{\boldsymbol{w} \in \mathcal{W}} |v(\boldsymbol{w})|^2 \leq B \right\} .$$

This can be viewed as a generalization of the parametric class idea to include models with an infinite number of parameters. We can even consider continuously infinite weighted combinations using integrals, such as

$$\mathcal{F}_B \ = \ \left\{ f \ : \ f(\boldsymbol{x}) = \int v(\boldsymbol{w}) \phi_{\boldsymbol{w}}(\boldsymbol{x}) \, d\boldsymbol{w} \ , \ \int |v(\boldsymbol{w})|^2 \, dw \leq B \right\} .$$

A classic example of this is the Fourier transform. Let $f$ be a function satisfying $\int |f(\boldsymbol{x})|^2 \, dx < \infty$. Such functions can be represented as $\frac{1}{(2\pi)^d} \int F(\boldsymbol{w}) e^{i\boldsymbol{w}^T\boldsymbol{x}} \, d\boldsymbol{w}$, where the function $F(\boldsymbol{w}) = \int f(\boldsymbol{x}) e^{-i\boldsymbol{w}^T\boldsymbol{x}} d\boldsymbol{x}$, the Fourier transform of $f$. “Infinite width" neural networks of the form $f(\boldsymbol{x}) = \int v(\boldsymbol{w}) \phi(\boldsymbol{w}^T\boldsymbol{x} + b) \, d\boldsymbol{w}$ are another popular example.

### 21.1.3. Nonparametric Classes

Given a function norm $\|f\|$ we can define the class $\mathcal{F}_B = \{f \ : \ \|f\| \leq B\}$. For example, consider continuous functions on the interval $[0, 1]$. We may define a norm to be $\|f\|_{C^0} = \sup_{x \in [0,1]} |f(x)|$. This defines a class of functions without an explicit parameterization. If we consider functions that have continuous derivatives $f^{(1)}, \ldots, f^{(k)}$, then we could define a class based on the norm $\|f\|_{C^k} = \sum_{j=1}^{k} \sup_{x \in [0,1]} |f^{(j)}(x)|$. Note that

$$\mathcal{F}_B^k := \left\{ f \ : \ \|f\|_{C^k} \leq B \right\} \ \subset \ \left\{ f \ : \ \|f\|_{C^0} \leq B \right\} =: \mathcal{F}_B^0 \, .$$

This shows that certain nonparametric classes are larger than others. A common approach to work with nonparametric classes in practice is to approximate the functions in such classes with parameteric or atomic models, revealing bridges between the different ways we may formulate model classes. A classical example of this is the Weierstrauss theorem.

**Weierstrauss (1885):** If $f$ is a continuous function on $[0, 1]$, then for any continuous $f : [0, 1] \to \mathbb{R}$ and any $\epsilon > 0$ there exists a polynomial $p$ such that

$$\sup_{x \in [0,1]} |p(x) - f(x)| < \epsilon.$$

## 21.2. Exercises

Suppose we want to interpolate data $\{(x_i, y_i)\}_{i=1}^n$, where $x_i \in [0, 1]$ and $y_i \in \mathbb{R}$. It is possible to interpolate these data using a polynomial of degree $n-1$, but this interpolation will typically widely fluctuate between the data points. Suppose instead we interpolate with a degree $d > n-1$ polynomial function. There an infinite number of such "overparameterized" polynomials that interpolate the data, so it is natural to choose the one with minimum norm. The norm could simply be the Euclidean norm of the polynomial coefficients or it could be another function of the coefficients.

1. Let $\mathcal{F}_d$ denote the set of all a degree $d$ polynomials. Consider the optimization

$$\min_{f \in \mathcal{F}_d} \left\{ \sum_{i=1}^n \big(y_i - f(x_i)\big)^2 \; + \; \lambda \int_0^1 |f(x)|^2 \, dx \right\}$$

   where $\lambda > 0$. The polynomial functions are parametric of the form $f(x) = \sum_{k=0}^d w_k x^k$. Reformulate this as an optimization over the polynomial coefficients $\boldsymbol{w} = [w_0,\, w_1,\, \cdots ,\, w_d]^T$. Show that the solution has the form

$$\widehat{\boldsymbol{w}} = (\boldsymbol{V}^T\boldsymbol{V} + \lambda \boldsymbol{A})^{-1}\boldsymbol{V}^T\boldsymbol{y} \; .$$

   Give explicit expressions for the elements of the matrices $\boldsymbol{V}$ and $\boldsymbol{A}$.

2. Now consider the optimization

$$\min_{f \in \mathcal{F}_d} \left\{ \sum_{i=1}^n \big(y_i - f(x_i)\big)^2 \; + \; \lambda \int_0^1 (f''(x))^2 \, dx \right\}$$

   where $f''$ is the second derivative of $f$ and $\lambda > 0$. Reformulate this as an optimization over the polynomial coefficients $\boldsymbol{w} = [w_0,\, w_1,\, \cdots ,\, w_d]^T$. Show that the solution has the form

$$\widehat{\boldsymbol{w}} = (\boldsymbol{V}^T\boldsymbol{V} + \lambda \boldsymbol{B})^{-1}\boldsymbol{V}^T\boldsymbol{y} \; .$$

   Give an explicit expression for the matrix $\boldsymbol{B}$.

3. If we take $\lambda = 0^+$, then the optimization above corresponds to

$$\min_{f \in \mathcal{F}_d} \int_0^1 (f''(x))^2 \, dx \;\text{ subject to }\; f(x_i) = y_i, \; i = 1, \ldots, n \; .$$

   Explain/discuss the behavior of the solution as $d$ increases.

# Lecture 22: Banach and Hilbert Spaces

## 22.1. Review of Vector Spaces

We start from an a brief review of vector spaces in this section, and then introduce normed vector spaces, complete normed vector spaces (Banach spaces), and then Banach spaces with an inner product (Hilbert Spaces). Examples are provided along the discussion.

**Definition 22.1.1.** A vector space $\mathcal{F}$ is a set of elements (vectors) together with addition and scalar multiplication operators satisfying the following axioms. For any $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in \mathcal{F}$ and any scalars $\alpha, \beta \in \mathbb{R}$:[a]

- If $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{F}$, then $\boldsymbol{u} + \boldsymbol{v} \in \mathcal{F}$
- $\boldsymbol{u} + \boldsymbol{v} = \boldsymbol{v} + \boldsymbol{u}$ (commutativity of addition)
- $\boldsymbol{u} + (\boldsymbol{v} + \boldsymbol{w}) = (\boldsymbol{u} + \boldsymbol{v}) + \boldsymbol{w}$ (associativity of addition)
- $\exists$ null vector $\mathbf{0} \in \mathcal{F}$ such that $\boldsymbol{v} + \mathbf{0} = \boldsymbol{v}$ (identity element of addition)
- $\exists - \boldsymbol{v} \in \mathcal{F}$ such that $\boldsymbol{v} + (-\boldsymbol{v}) = \mathbf{0}$ (inverse element of addition)
- If $\boldsymbol{u} \in \mathcal{F}$, then $\alpha \boldsymbol{u} \in \mathcal{F}$
- $\alpha(\beta \boldsymbol{v}) = (\alpha\beta)\boldsymbol{v}$ (compatibility of scalar multiplication with field multiplication)
- $1\boldsymbol{v} = \boldsymbol{v}$ where $1$ denotes the multiplicative identity in $\mathbb{R}$ (identity element of scalar multiplication)
- $\alpha(\boldsymbol{u} + \boldsymbol{v}) = \alpha\boldsymbol{u} + \alpha\boldsymbol{v}$ (distributivity of scalar multiplication with respect to vector addition)
- $(\alpha + \beta)\boldsymbol{v} = \alpha\boldsymbol{v} + \beta\boldsymbol{v}$ (distributivity of scalar multiplication with respect to field addition)

[a] We could also work with complex valued functions and the field of complex numbers $\mathbb{C}$ or other fields. The default field considered in this note is $\mathbb{R}$ unless otherwise mentioned.

Note that many other properties can be derived from above axioms. For instance, $0\boldsymbol{v} = \mathbf{0}$ can be derived by noticing $0 + 0 = 0 \implies (0 + 0)\boldsymbol{v} = 0\boldsymbol{v} \implies 0\boldsymbol{v} + 0\boldsymbol{v} = 0\boldsymbol{v} \implies 0\boldsymbol{v} + 0\boldsymbol{v} + (-0\boldsymbol{v}) = 0\boldsymbol{v} + (-0\boldsymbol{v}) \implies 0\boldsymbol{v} + (0\boldsymbol{v} + (-0\boldsymbol{v})) = 0\boldsymbol{v} + (-0\boldsymbol{v}) \implies 0\boldsymbol{v} + \mathbf{0} = \mathbf{0} \implies 0\boldsymbol{v} = \mathbf{0}$. The abstract definition of vector space can be easily understood with the following examples.

**Example 22.1.2.**

- $\mathbb{R}$ *with* $\boldsymbol{v} \in \mathbb{R}$.
- $\mathbb{R}^d$ *with* $\boldsymbol{v} = [v_1, v_2, \ldots, v_d]^\top$ *and each* $v_i \in \mathbb{R}$.
- $\mathbb{R}^\infty$ *with* $\boldsymbol{v} = [v_1, v_2, \ldots,]^\top$ *and each* $v_i \in \mathbb{R}$.
- $C([0,1])$ *with* $\boldsymbol{v}$ *being any real-valued continuous function defined on* $[0,1]$.

- $P_d([0,1])$ *with* $\boldsymbol{v}$ *being any polynomial of degree* $d$ *or smaller defined on* $[0,1]$.

We can also define a subspace of a vector space, which is a subset that is closed under linear combinations.

**Definition 22.1.3.** A non-empty subset $\mathcal{S} \subseteq \mathcal{F}$ is a subspace of $\mathcal{F}$ if $\alpha \boldsymbol{u} + \beta \boldsymbol{v} \in \mathcal{S}$ for all $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{S}$ and scalars $\alpha, \beta \in \mathbb{R}$.

One should notice that $\mathbf{0} \in \mathcal{S}$ since we can always set $\alpha = \beta = 0$. Examples of subspaces are provided as follows.

**Example 22.1.4.**

- $\{\boldsymbol{v} : \boldsymbol{v} = [v_1, v_2, \ldots, v_k, 0, \ldots, 0] \in \mathbb{R}^d\}$ *is a subspace of* $\mathbb{R}^d$.
- $P_d([0,1])$ *is a subspace of* $C([0,1])$.

If $\mathcal{S}, \mathcal{T}$ are subspaces of $\mathcal{F}$, it's easy to check that the following two subsets are also subspaces:

- $\mathcal{S} \cap \mathcal{T} = \{\boldsymbol{v} : \boldsymbol{v} \in \mathcal{S}, \boldsymbol{v} \in \mathcal{T}\}$
- $\mathcal{S} + \mathcal{T} = \{\boldsymbol{v} : \boldsymbol{v} = \boldsymbol{u} + \boldsymbol{w}, \boldsymbol{u} \in \mathcal{S}, \boldsymbol{w} \in \mathcal{T}\}$

One can also define an *affine* subspace, with respect to a fixed vector $\boldsymbol{w} \in \mathcal{F}$, as follows,

$$\mathcal{S}_{\boldsymbol{w}} = \{\boldsymbol{v} : \boldsymbol{v} = \boldsymbol{u} + \boldsymbol{w}, \boldsymbol{u} \in \mathcal{S}, \boldsymbol{w} \in \mathcal{F}\}.$$

Every subspace is an affine subspace (with $\boldsymbol{w} = \mathbf{0}$), yet the converse is not true (an affine subspace need not to contain $\mathbf{0}$).

We use dimension to measure the "size" of a vector (sub-)space. Before getting into that, we first introduce the notions of linear dependence and linear independence.

**Definition 22.1.5.** A set of vectors $\{\boldsymbol{v}_j\}$ is said to be linearly dependent if at least one vector $\boldsymbol{v}_i$ in the set can be written as the linear combination of the others, i.e.,

$$\boldsymbol{v}_i = \sum_{j \neq i} \alpha_j \boldsymbol{v}_j.$$

If no vector in the set can be written in this way, the set of vectors are said to be linear independent.

**Theorem 22.1.6.** *A set of vectors* $\{\boldsymbol{v}_j\}$ *is linearly independent if and only if*

$$\sum_j \alpha_j \boldsymbol{v}_j = \mathbf{0} \implies \alpha_j = 0, \ \forall j.$$

*Proof.* We prove the theorem by contradiction on both directions.

"$\Longrightarrow$" Suppose there exists $\alpha_i \neq 0$ such that $\sum_j \alpha_i \boldsymbol{v}_j = \mathbf{0}$, we then have $\boldsymbol{v}_i = \sum_{j \neq i} -\alpha_j \boldsymbol{v}_j / \alpha_i$. And this contradicts with the definition of linear independence.

"$\Longleftarrow$" Suppose the set of vectors is linear dependent, we then have $\boldsymbol{v}_i = \sum_{j \neq i} \alpha_j \boldsymbol{v}_j$ for some $\boldsymbol{v}_i$. Rearranging the equility gives $\sum_{j \neq i} \alpha_j \boldsymbol{v}_j - \boldsymbol{v}_i = \mathbf{0}$, which contradicts with the statement $\alpha_j = 0, \ \forall j$ (the coefficient of $\boldsymbol{v}_i$ is $-1 \neq 0$). □

**Definition 22.1.7.** A set of linearly independent vectors $\{\boldsymbol{u}_i\}$ in $\mathcal{F}$ is a basis for $\mathcal{S} \subseteq \mathcal{F}$ if every $\boldsymbol{v} \in \mathcal{S}$ can be written as

$$\boldsymbol{v} = \sum_i \alpha_i \boldsymbol{u}_i.$$

If a basis $\{\boldsymbol{u}_i\}$ is finite, then $\mathcal{S}$ is finite-dimensional. Otherwise, $\mathcal{S}$ is infinite-dimensional.

Another way to say that a set of linearly independent vectors $\{\boldsymbol{u}_i\}$ forms a basis for $\mathcal{S}$ is that $\operatorname{span}(\{\boldsymbol{u}_i\}) = \mathcal{S}$. The span is the set of all vectors that can be formed by linear combinations of $\{\boldsymbol{u}_i\}$. Unless otherwise mentioned, we will always be working with bases that are countable, i.e., finite or countably infinite. Examples of bases are provided as follows.

**Example 22.1.8.**

- $\mathbb{R}^d$ *is* $d$*-dimensional with basis* $\{\boldsymbol{e}_i\}_{i=1}^d$*, where* $\boldsymbol{e}_i$ *is a* $d$*-dimensional vector with* $1$ *on the* $i$*-th entry and* $0$ *elsewhere.*
- $\mathbb{R}^\infty$ *is infinite-dimensional with basis* $\{\boldsymbol{u}_i\}_{i=1}^\infty$*, where* $\boldsymbol{u}_i$ *is a infinite-dimensional vector with* $1$ *on the* $i$*-th entry and* $0$ *elsewhere.*
- $P_d([0,1])$ *is* $(d+1)$*-dimensional with basis* $\{\boldsymbol{u}_i(x)\}_{i=0}^d$*, where* $\boldsymbol{u}_i(x) = x^i$*.*

## 22.2. Normed Vector Spaces and Banach Spaces

We equip a vector space with a norm and introduce the normed vector space. A norm can be thought as the formalization of "length/size" in vector spaces.

**Definition 22.2.1.** A normed vector space $(\mathcal{F}, \|\cdot\|)$ is a vector space $\mathcal{F}$ equipped with functional mapping $\|\cdot\| : \mathcal{F} \to \mathbb{R}$ satisfying the following properties. For any $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{F}$ and scalar $\alpha \in \mathbb{R}$:

- $\|\boldsymbol{v}\| \geq 0$ (nonnegativity)
- $\|\boldsymbol{v}\| = 0 \iff \boldsymbol{v} = \mathbf{0}$ (positivity on non-zero vectors)
- $\|\alpha \boldsymbol{v}\| = |\alpha| \|\boldsymbol{v}\|$ (absolutely scalable)
- $\|\boldsymbol{u} + \boldsymbol{v}\| \leq \|\boldsymbol{u}\| + \|\boldsymbol{v}\|$ (triangle inequality)

It is important to specify the norm $\|\cdot\|$ when defining a normed vector space $(\mathcal{F}, \|\cdot\|)$. Different norms on the same vector space induce different normed spaces with different properties. Some examples of normed vector spaces are provided as follows.

**Example 22.2.2.**

- $\mathbb{R}^d$ *is a normed vector space with norm* $\|\boldsymbol{v}\|_p = (\sum_{i=1}^d |v_i|^p)^{1/p}$ *when* $p \geq 1$.[8]
- $C([0,1])$ *is a normed vector space with norm* $\|f\|_{\mathrm{L}^\infty} = \sup_{x\in[0,1]}|f(x)|$ *or norm* $\|f\|_{\mathrm{L}^1} = \int_0^1 |f(x)|dx$.
- $C^1([0,1])$ *is a normed vector space with norm* $\|f\| = \sup_{x\in[0,1]}|f(x)| + \sup_{x\in[0,1]}|f'(x)|$.[9]
- $\mathrm{BV}([0,1])$ *is a normed vector space with norm* $\|f\| = |f(0)| + \mathrm{TV}(f)$ *where*

$$\mathrm{TV}(f) = \sup_{P\in\mathcal{P}} \sum_{i=0}^{n_P-1} |f(x_{i+1}) - f(x_i)|$$

*and* $\mathcal{P}$ *is the set of all partitions* $[0,1]$ *and* $0 = x_0 \leq \cdots \leq x_{n_P} = 1$ *are the boundaries of the partition* $P$.[10]

Equipped with a norm $\|\cdot\|$, one can easily define a metric $d(\boldsymbol{u}, \boldsymbol{v}) = \|\boldsymbol{u} - \boldsymbol{v}\|$ to measure the distance between two vectors. This allows us to analyze sequences of vectors and their limits. Several related definitions are provided as follows.

**Definition 22.2.3.** Let $(\mathcal{F}, \|\cdot\|)$ be a normed vector space. A sequence $\{\boldsymbol{v}_n\}_{n\geq 1}$ in $\mathcal{F}$ is said to converge to $\boldsymbol{v} \in \mathcal{F}$ if

$$\lim_{n\to\infty} \|\boldsymbol{v}_n - \boldsymbol{v}\| = 0.$$

**Definition 22.2.4.** A set $\mathcal{S} \subseteq \mathcal{F}$ is closed if and only if every convergent sequence in $\mathcal{S}$ has its limit point in $\mathcal{S}$.

**Definition 22.2.5.** Let $(\mathcal{F}, \|\cdot\|)$ be a normed vector space. A sequence $\{\boldsymbol{v}_n\}_{n\geq 1}$ in $\mathcal{F}$ is Cauchy if for any $\epsilon > 0$, there exists $N(\epsilon) \in \mathbb{N}$ such that for any $m, n \geq N(\epsilon)$, we have

$$\|\boldsymbol{v}_m - \boldsymbol{v}_n\| < \epsilon.$$

The definition of Banach space is provided as follows.

[8] It does not form a norm when $0 \leq p < 1$.

[9] $C^1([0,1])$ stands for all continuously differentiable functions defined on $[0,1]$.

[10] We add the $|f(0)|$ term in $\|f\|$ to make it a norm. $\mathrm{TV}(f)$ itself is a semi-norm since it doesn't satisfy the second property, i.e., positivity on non-zero vectors.

**Definition 22.2.6.** A normed vector space $(\mathcal{F}, \|\cdot\|)$ is said to be *complete* if every Cauchy sequence in $\mathcal{F}$ converges to limits in $\mathcal{F}$. A complete normed vector space is called a *Banach space*.

Examples of Banach spaces and non-Banach spaces are provided as follows.

**Example 22.2.7.**

- $\mathbb{R}$ *with absolute-value norm is a Banach space and* $\mathbb{R}^d$ *with* $p$*-norm* ($p \geq 1$) *is a Banach space.*
- $C([0,1])$ *with norm* $\|f\|_{\mathrm{L}^\infty} = \sup_{x\in[0,1]}|f(x)|$ *is a Banach space.*
- $C([0,1])$ *with norm* $\|f\|_{\mathrm{L}^1} = \int_0^1|f(x)|dx$ *is not complete (and thus not a Banach space).*

Some brief explanations of the above examples are provided as follows.

- The proof of completeness of $\mathbb{R}$ is a consequence of the Bolzano-Weierstrass theorem. The proof of completenesso of $\mathbb{R}^d$ can be done by checking the convergence of each coordinate and use the completeness result of $\mathbb{R}$.
- Let $(f_n)$ be a Cauchy sequence in $C([0,1])$. For any $x \in [0,1]$, $(f_n(x))$ is Cauchy in $\mathbb{R}$; we then define $f : [0,1] \to \mathbb{R}$ such that $f(x) = \lim_{n\to\infty} f_n(x)$. We next show that $f_n \to f$ under $\mathrm{L}^\infty$ norm and $f \in C([0,1])$ as below.

  $f_n \to f$: Fix any $\epsilon > 0$. Since $(f_n)$ is Cauchy, there exists $N(\epsilon) \in \mathbb{N}$ such that for any $m, n \geq N(\epsilon)$, we have $\|f_n - f_m\|_{\mathrm{L}^\infty} \leq \epsilon$. As a result, for any $n \geq N(\epsilon)$, we have

  $$\|f_n - f\|_{\mathrm{L}^\infty} = \sup_{x\in[0,1]}|f_n(x) - f(x)| = \sup_{x\in[0,1]}\lim_{m\to\infty}|f_n(x) - f_m(x)| \leq \sup_{x\in[0,1]}\lim_{m\to\infty}\|f_n - f_m\|_{\mathrm{L}^\infty} \leq \epsilon.$$

  $f \in C([0,1])$: Consider any $n \geq N(\epsilon/3)$ so that $\|f_n - f\|_{\mathrm{L}^\infty} \leq \epsilon/3$ and any fixed $x \in [0,1]$. Since $f_n$ is continuous, there must exists a $\delta(\epsilon/3) > 0$ such that for any $|y - x| \leq \delta(\epsilon/3)$, we have $|f_n(y) - f_n(x)| \leq \epsilon/3$. As a result, we have

  $$|f(y) - f(x)| \leq |f(y) - f_n(y)| + |f_n(y) - f_n(x)| + |f_n(x) - f(x)| \leq \epsilon,$$

  and thus $f \in C([0,1])$.
- Showing that $C([0,1])$ with norm $\|f\|_{\mathrm{L}^1} = \int_0^1|f(x)|dx$ is not complete is left as an exercise.

## 22.3. Hilbert Spaces

Hilbert spaces generalize the familiar concept of Euclidean space. A Hilbert space is a type of Banach space equipped with an additional geometric structure called an inner product, which allows the definition of length and angle.

**Inner product:** Let $\mathcal{H}$ be a vector space. A *inner product* is a mapping from $\mathcal{H} \times \mathcal{H}$ to $\mathbb{R}$ satisfying the following for any $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathcal{H}$ and any scalar $\alpha, \beta$:

1. symmetry: $\langle \mathbf{u}, \mathbf{v} \rangle = \langle \mathbf{v}, \mathbf{u} \rangle$;
2. linearity: $\langle \alpha\mathbf{u} + \beta\mathbf{v}, \mathbf{w} \rangle = \alpha\langle \mathbf{u}, \mathbf{w} \rangle + \beta\langle \mathbf{v}, \mathbf{w} \rangle$;
3. positive-definite: $\langle \mathbf{v}, \mathbf{v} \rangle > \mathbf{0}$ if $\mathbf{v} \neq \mathbf{0}$.

The inner product induces the norm

$$\|\mathbf{v}\|= \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle}. \tag{22.1}$$

Obviously, the norm induced by inner product is nonnegative and for any $\alpha \in \mathbb{R}$, $\|\alpha\mathbf{v}\|= |\alpha|\|\mathbf{v}\|$ . Thus, to show the equation 22.1 is a valid norm, we only need to show that it also satisfies triangle inequality. To prove this property, let us introduce Cauchy-Schwarz inequality first.

**Cauchy-Schwarz Inequality:** For any $\mathbf{u}, \mathbf{v} \in \mathcal{H}$, $|\langle \mathbf{u}, \mathbf{v} \rangle| \leq \|\mathbf{u}\|\|\mathbf{v}\|$.

*Proof.* For any $\alpha$ we have $0 \leq \langle \mathbf{u} - \alpha\mathbf{v}, \mathbf{u} - \alpha\mathbf{v} \rangle = \|\mathbf{u}\|^{\mathbf{2}} - \mathbf{2}\alpha\langle \mathbf{u}, \mathbf{v} \rangle + \alpha^{\mathbf{2}}\|\mathbf{v}\|^{\mathbf{2}}$. Let $\alpha = \frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\|\mathbf{v}\|^{\mathbf{2}}}$. Then we have $0 \leq \|\mathbf{u}\|^{\mathbf{2}} - \frac{|\langle \mathbf{u}, \mathbf{v} \rangle|^{\mathbf{2}}}{\|\mathbf{v}\|^{\mathbf{2}}}$, hence leads to the desired result. □

Cauchy-Schwarz inequality leads to triangle inequality directly: $\|\mathbf{u} + \mathbf{v}\|\leq \|\mathbf{u}\|+\|\mathbf{v}\|$.

**Hilbert Space:** Let $\mathcal{H}$ be vector space equipped with an inner product and associated norm. If the space is complete with respect to the norm, then it is called a *Hilbert Space*.

**Parallelogram Law:** If $\mathcal{H}$ is a Hilbert space, then for any $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{H}$,

$$\|\boldsymbol{u} + \boldsymbol{v}\|^2 + \|\boldsymbol{u} - \boldsymbol{v}\|^2 \ = \ 2\big(\|\boldsymbol{u}\|^2 + \|\boldsymbol{v}\|^2\big) \ .$$

In fact a Banach space is a Hilbert space if and only if the parallelogram law holds. This means that if the parallelogram law fails to hold for certain a $\boldsymbol{u}$ and $\boldsymbol{v}$, then the space is not a Hilbert space.

**Example 22.3.1.** $\mathbb{R}^n$ *equipped with inner product* $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \sum_i u_i v_i$ *for any* $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$ *is a Hilbert space.*

**Example 22.3.2.** $L^2[0, 1]$ *equipped with inner product* $\langle f, g \rangle = \int f(x)g(x)\, dx$ *for any* $f, g \in L^2[0, 1]$ *is a Hilbert space.*

**Example 22.3.3.** $\mathcal{H} = \{$*all polynomial functions on* $[0, 1]\}$ *equipped with inner product* $\langle f, g \rangle = \int f(x)g(x)\, dx$ *for any* $f, g \in \mathcal{H}$ *is a Hilbert space. Note* $\mathcal{H}$ *is a subspace of* $L^2[0, 1]$.

There are many interesting properties of Hilbert space. Specifically geometric intuition plays an important role in many aspects of Hilbert space theory. Analogs of the Pythagorean theorem holds in a Hilbert space.

**Orthogonality:** Consider a Hilbert space $\mathcal{H}$. Two vectors $\mathbf{u}, \mathbf{v} \in \mathcal{H}$ are orthogonal if $\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{0}$, denoted by via $\mathbf{u} \perp \mathbf{v}$. A vector $\mathbf{u} \in \mathcal{H}$ is orthogonal to an subspace $\mathcal{S} \in \mathcal{H}$ if $\mathbf{u} \perp \mathbf{v}$, for every $\mathbf{v} \in \mathcal{S}$.

**Pythagorean Theorem:** If $\mathbf{u} \perp \mathbf{v}$, then $\|\mathbf{u}+\mathbf{v}\|^{\mathbf{2}} = \|\mathbf{u}\|^{\mathbf{2}} + \|\mathbf{v}\|^{\mathbf{2}}$ .

**Parallelogram Law:** For any $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{H}$,

$$\|\boldsymbol{u}+\boldsymbol{v}\|^2 + \|\boldsymbol{u}-\boldsymbol{v}\|^2 = 2\big(\|\boldsymbol{u}\|^2 + \|\boldsymbol{v}\|^2\big) .$$

## 22.4. Exercises

1. Verify that $C[0,1]$ is a vector space.
2. Verify that $P_d[0,1]$ is a subspace of $C[0,1]$.
3. A basis $\{\boldsymbol{u}_j\}$ is said to be *orthonormal* if $\langle \boldsymbol{u}_i, \boldsymbol{u}_j \rangle = 0$ for all $i \neq j$. Construction an orthonormal basis for the space of linear functions on $[0,1]$, i.e., $P_1[0,1]$.
4. Consider $C[0,1]$ and show that $\sup_{x\in[0,1]} |f(x)|$ and $\big(\int_0^1 f^2(x)dx\big)^{1/2}$ are norms.
5. Show that $C[0,1]$ with the norm $\|f\|_{L^1} = \int_0^1 |f(x)|dx$ is not a Banach space. Hint: Consider the sequence of functions
$$f_n(x) = \begin{cases} 0 & 0 \le x < \frac{1}{2} - \frac{1}{n}, \\ nx + (1 - \frac{n}{2}) & \frac{1}{2} - \frac{1}{n} \le x < \frac{1}{2}, \\ 1 & \frac{1}{2} \le x \le 1. \end{cases}$$
6. Verify that $\langle f, g \rangle = \int_0^1 f(x)g(x)\,dx$ is a a valid inner product and that it generates the space $L^2[0,1]$.
7. Explain why $P_d[0,1]$ is a subspace of $L^2[0,1]$.

# Lecture 23: Reproducing Kernel Hilbert Spaces

A Reproducing Kernel Hilbert Space (RKHS) are a special type of Hilbert spaces that is especially important in machine learning, because they enable practical and computationally efficient learning methods in infinite dimensional function spaces. Before moving on, let us introduce a bit of notation. We write $f$ to refer to a function/vector in a vector space and $f(\boldsymbol{x})$ to refer to the value of $f$ at the point $\boldsymbol{x}$. We also sometimes write $f(\cdot)$ to refer to the function/vector. We will also be working with functions of two points or vectors. For example, an inner product is such a function and we write $\langle \cdot, \cdot \rangle$ to refer to this function and $\langle f, g \rangle$ to refer to its value for the pair $f, g$.

**Reproducing Kernel Hilbert Space:** A Hilbert Space $\mathcal{H}$ of functions on a domain $\mathcal{X}$ is said to be a *Reproducing Kernel Hilbert Space* (RKHS) if there is a function $k$ defined on $\mathcal{X} \times \mathcal{X}$ satisfying two properties:

1. $k(\cdot, \boldsymbol{x}) \in \mathcal{H}$ for each $\boldsymbol{x} \in \mathcal{X}$
2. $\langle f, k(\cdot, \boldsymbol{x}) \rangle = f(\boldsymbol{x})$ for each $f \in \mathcal{H}$

Such a function is called a *reproducing kernel.*

The domain $\mathcal{X}$ could be $\mathbb{R}^d$, for example. Notice that the reproducing kernel satisfies

$$\langle k(\cdot, \boldsymbol{x}'), k(\cdot, \boldsymbol{x}) \rangle = k(\boldsymbol{x}, \boldsymbol{x}') \ .$$

**Example 23.0.1.** *Consider the Hilbert space $\mathbb{R}^d$ equipped with inner product $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \boldsymbol{u}^T\boldsymbol{v}$, denoted $\ell_d^2$. Here the domain is $\mathcal{X} = \{1, \ldots, d\}$. Define the kernel $k(i, j) = 1$ if $i = j$ and $0$ otherwise. Clearly, $k(\cdot, j) \in \ell_d^2$ for $j = 1, \ldots, d$. And $k$ satisfies the reproducing property $u(j) = \langle \boldsymbol{u}, k(\cdot, j) \rangle = \sum_{i=1}^d u(i)k(i, j)$. This shows that $d$-dimensional Euclidean space is an RKHS.*

**Example 23.0.2.** *Let $f$ be a univariate function and let $f^{(1)}$ denote its first derivative. Let the domain $\mathcal{X} = [0, 1]$ and $\|f\|_{L^2} = \big(\int_0^1 |f(u)|^2 du\big)^{1/2}$. Consider the normed vector space*

$$\mathcal{H}^1[0, 1] = \{f : [0, 1] \to \mathbb{R}\, ,\ f(0) = 0,\ \|f^{(1)}\|_{L^2} < \infty\}$$

*with inner product $\langle f, g \rangle = \int f^{(1)}(u)g^{(1)}(u)\, du$. This is an RKHS with reproducing kernel $k(x, x') = \min(x, x')$. To see this, write $\min(x, x') = \int_0^x \mathbb{1}_{\{u \in [0, x']\}}\, du$. Observe that for $x$ fixed $k^{(1)}(u, x) = \mathbb{1}_{\{u \in [0, x]\}}$ as a function of $u$ and thus*

$$\langle f, k(\cdot, x) \rangle \ = \ \int_0^1 f^{(1)}(u)\mathbb{1}_{\{u \in [0, x]\}}\, du \ = \ \int_0^x f^{(1)}(u)\, du \ = \ f(x) \ .$$

## 23.1. Constructing an RKHS

We can also construct an RKHS by starting with a positive semidefinite kernel function, defined as follows.

**Positive semidefinite (psd) kernel:** A symmetric bivariate function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive semidefinite (psd) if for all integers $n \geq 1$ and all $\{x_i\}_{i=1}^n \subset \mathcal{X}$, the $n \times n$ matrix $K_{ij} = k(x_i, x_j)$ is psd.

Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be any psd kernel. The kernel defines a unique Hilbert space $\mathcal{H}$ of functions on $\mathcal{X}$, with the reproducing property

$$\langle f, k(\cdot, x)\rangle_{\mathcal{H}} = f(x) \quad \forall f \in \mathcal{H},\ x \in \mathcal{X}.$$

Consider functions of the form

$$f(\cdot) = \sum_{i=1}^{n} \alpha_i k(\cdot, x_i)$$

where $\{x_i\}_{i=1}^n \subset \mathcal{X}$ and $\{\alpha_i\}_{i=1}^n \subset \mathbb{R}$. It is easily verified that the set of all such functions is a vector space, which we will denote by $\widetilde{\mathcal{H}}$. Now let $f, \widetilde{f} \in \widetilde{\mathcal{H}}$, which we have the forms

$$f(\cdot) = \sum_{j=1}^{n} \alpha_j k(\cdot, x_j) \ \text{ and } \ \widetilde{f}(\cdot) = \sum_{j=1}^{\widetilde{n}} \widetilde{\alpha}_j k(\cdot, x_j)\ .$$

Define the inner product on $\widetilde{\mathcal{H}}$ as

$$\langle f, \widetilde{f}\rangle_{\widetilde{\mathcal{H}}} \ := \ \sum_{i=1}^{n} \sum_{j=1}^{\widetilde{n}} \alpha_i \widetilde{\alpha}_j k(x_i, \widetilde{x}_j)$$

This is a valid inner product since it satisfies the following,

1. Symmetry. $\langle f, \widetilde{f}\rangle_{\widetilde{\mathcal{H}}} = \langle \widetilde{f}, f\rangle_{\widetilde{\mathcal{H}}}$.
2. Linearity. $\langle af + bg, \widetilde{f}\rangle_{\widetilde{\mathcal{H}}} = a \cdot \langle f, \widetilde{f}\rangle_{\widetilde{\mathcal{H}}} + b \cdot \langle g, \widetilde{f}\rangle_{\widetilde{\mathcal{H}}}$
3. $\langle f, f\rangle_{\widetilde{\mathcal{H}}} \geq 0$ with equality iff $f = 0$. Note this is true because $k$ is positive semidefinite.

Furthermore, this definition satisfies

$$\langle f, k(\cdot, x)\rangle_{\widetilde{\mathcal{H}}} = \sum_{j=1}^{n} \alpha_j k(x, x_j) = f(x).$$

The final step in the construction is to complete $\widetilde{\mathcal{H}}$. We complete $\widetilde{\mathcal{H}}$ by including limits of all Cauchy sequences in $\widetilde{\mathcal{H}}$ and thus we get $\mathcal{H}$, which is the RKHS.

Lastly, we prove the uniqueness of $\mathcal{H}$. Recall the notion of orthogonality in Hilbert spaces.

**Orthogonality:** Consider a Hilbert space $\mathcal{H}$. Two vectors $f, g \in \mathcal{H}$ are orthogonal if $\langle f, g\rangle = 0$, denoted by via $f \perp g$. A vector $f \in \mathcal{H}$ is orthogonal to an subspace $\mathcal{S} \in \mathcal{H}$ if $f \perp g$, for every $g \in \mathcal{S}$. Let $\mathcal{S}^{\perp}$ denote the set of all vectors in $\mathcal{H}$ that are orthogonal to $\mathcal{S}$. Then any vector $f \in \mathcal{H}$ may be decomposed as $f = f_{\mathcal{S}} + f_{\mathcal{S}^{\perp}}$, where $f_{\mathcal{S}}$ is the component in $\mathcal{S}$ and $f_{\mathcal{S}^{\perp}}$ is the component in $\mathcal{S}^{\perp}$. This is denoted by $\mathcal{H} = \mathcal{S} \oplus \mathcal{S}^{\perp}$.

Suppose $\mathcal{H}_1$ is some other RKHS with $k$ as its kernel. Then $k(\cdot, x) \in \mathcal{H}_1$ for $\forall x \in \mathcal{X}$. Since $\mathcal{H}_1$ is complete,

$$\mathcal{H} = \text{closure}(\{f : f(\cdot) = \sum \alpha_j k(\cdot, x_j)\}) \subset \mathcal{H}_1$$

Thus $\mathcal{H}$ is a subspace of $\mathcal{H}_1$ and $\mathcal{H}_1 = \mathcal{H} \bigoplus \mathcal{H}^{\perp}$. Then for $g \in \mathcal{H}^{\perp}$, since $k(\cdot, x) \in \mathcal{H}$, we have

$$0 = \langle k(\cdot, x), g\rangle = g(x), \quad \forall x$$

so $g = 0$, which means $\mathcal{H}^{\perp} = \{0\}$. Therefore, $\mathcal{H}_1 = \mathcal{H}$.

Any RKHS also has a unique kernel. To see this suppose $k_1$ and $k_2$ generate the same $\mathcal{H}$. Then the reproducing property implies that

$$\langle f,\, k_1(\,\cdot,\, x) - k_2(\,\cdot,\, x)\rangle_{\mathcal{H}} = 0 \quad \text{for all } f \in \mathcal{H},\, x \in \mathcal{X}.$$

Now, let $f(\cdot) = k_1(\,\cdot,\, x')$ for any $x' \in \mathcal{X}$. Then we have

$$0 = \langle f,\, k_1(\,\cdot,\, x) - k_2(\,\cdot,\, x)\rangle_{\mathcal{H}} = \langle k_1(\,\cdot,\, x'),\, k_1(\,\cdot,\, x) - k_2(\,\cdot,\, x)\rangle_{\mathcal{H}} \;=\; k_1(x',\, x) - k_2(x',\, x)$$

which implies that $k_1 = k_2$.

## 23.2. Examples of PSD Kernels

**Ex.1:** Linear kernel. Let $\mathcal{X} = \mathbb{R}^d$, consider the kernel $k(x_1, x_2) = \langle x_1, x_2\rangle = x_1^T x_2$, then we have

$$f(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i) = (\sum_{i=1}^{n} \alpha_i x_i^T)x$$

also it's easy to show this kernel is psd because we have $K_{ij} = k(x_i, x_j) = x_i^T x_j$, if we define $X = [x_1, \cdots, x_n]$ then

$$K = X^T X$$

which is psd since $\alpha^T K \alpha = ||X\alpha||_2^2 \geq 0$ for any $\alpha$. Using the linear kernel is equivalent to simple linear regression on $\mathbb{R}^d$, which shows that the linear kernel generates an RKHS of dimension $d$.

**Ex.2:** Polynomial kernel. Let $\mathcal{X} = \mathbb{R}^d$ and consider the kernel

$$k(x_1, x_2) = (\langle x_1, x_2\rangle)^p = (x_1^T x_2)^p$$

for simplicity, we consider the case $p = 2$ here. Then $k(x_1, x_2) = (\sum_{j=1}^{d} x_{1j}x_{2j})^2 = \sum_{j=1}^{d} x_{1j}^2 x_{2j}^2 + 2\sum_{i<j} x_{1i}x_{1j}x_{2i}x_{2j}$. This could be rewritten as

$$k(x_1, x_2) = \langle \phi(x_1), \phi(x_2)\rangle = \phi(x_1)^T \phi(x_2)$$

where

$$\phi(x) = \begin{bmatrix} x_j^2, & j = 1, \cdots, d \\ \sqrt{2}x_i x_j, & i < j \end{bmatrix}$$

actually $\phi$ here is a so-called feature map. This shows that the polynomial kernel generates a finite dimensional RKHS, with dimension $d(d+1)/2$, the number of terms in $\phi(x)$. Then

$$f(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i) = (\sum_{i=1}^{n} \alpha_i \phi(x_i))^T \phi(x)$$

and it is also easy to show the kernel is psd since we have $K = \Phi^T \Phi$ where

$$\Phi = [\phi(x_1), \cdots, \phi(x_n)].$$

**Ex.3:** Gaussian kernel: Let $\alpha > 0$ and

$$k(x_1, x_2) = \exp(-\alpha \|x_1 - x_2\|_2^2).$$

**Ex.4:** Laplace kernel:

$$k(x_1, x_2) = \exp(-\alpha ||x_1 - x_2||_2).$$

The Gaussian and Laplace kernels each generate an RKHS of infinite dimension. The feature maps associated with these spaces involve monomials of all possible degrees, but with scaling factors on each monomial that differ depending on the choice of kernel.

## 23.3. The Representer Theorem

Now let us consider the problem of learning in an RKHS $\mathcal{H}$. The goal will be to find a function $f \in \mathcal{H}$ that fits a set of training data and has a small norm.

**Representer Theorem:** Let $\mathcal{H}$ be an RKHS with kernel $k$. Then for any data $\{(x_i, y_i)\}_{i=1}^n$ and any continuous loss function $\ell$, there exists an $f \in \mathcal{H}$ that minimizes

$$\sum_{i=1}^n \ell(y_i, f(x_i)) + \lambda \|f\|_{\mathcal{H}}^2, \qquad \lambda > 0$$

and has a representation

$$f(\cdot) = \sum_{i=1}^n \alpha_i k(\,\cdot, x_i), \qquad \alpha_1, \ldots, \alpha_n \in \mathbb{R}.$$

If the loss function is convex, then the solution is unique.

*Proof.* Let us assume that a solution exists. Let $\mathcal{H}_0 = \text{span}\{k(\,\cdot, x_i)\}_{i=1}^n$. The orthogonal complement to $\mathcal{H}_0$ is $\mathcal{H}_0^\perp = \{f \in \mathcal{H} : \; f(x_i) = 0, \, i = 1, \ldots, n\}$. To see this, note that every function in $\mathcal{H}_0$ has the form $\sum_{i=1}^n \alpha_i k(\,\cdot, x_i)$. Let $f$ be orthogonal to $\mathcal{H}_0$. Then we have

$$0 \;=\; \left\langle f, \sum_{i=1}^n \alpha_i k(\,\cdot, x_i) \right\rangle_{\mathcal{H}} \;=\; \sum_{i=1}^n \alpha_i f(x_i)\,.$$

Since this equality holds for all choices of $\alpha_1, \ldots, \alpha_n$ it holds if and only if $f(x_i) = 0$, $i = 1, \ldots, n$. Any $f \in \mathcal{H}$ may be decomposed as $f = f_0 + f_0^\perp$, where $f_0 \in \mathcal{H}_0$ and $f_0^\perp \in \mathcal{H}_0^\perp$. Note that $\sum_{i=1}^n \ell(y_i, f(x_i)) = \sum_{i=1}^n \ell(y_i, f_0(x_i))$ and that $\|f\|_{\mathcal{H}}^2 = \|f_0\|_{\mathcal{H}}^2 + \left\|f_0^\perp\right\|_{\mathcal{H}}^2$ by orthogonality. Since the loss term does not depend on $f_0^\perp$ it is clear that the overall objective is minimized with $f_0^\perp = 0$. Together these imply that a global minimizer $\widehat{f} \in \mathcal{H}_0$ which completes the proof. □

The representer theorem shows that the solution is a linear combination of the functions $k(\cdot, x_1), \ldots, k(\cdot, x_n)$. In other words, the Representer Theorem shows that the solution is a *linear-in-parameters* model. So all our results pertaining to linear modeling apply in the RKHS setting. This is often referred to as the *kernel trick*. The weights $\alpha_1, \ldots, \alpha_n$ can be found by solving a finite dimensional optimization problem as follows. Note that the norm of the solution $f$ is

$$\|f\|_{\mathcal{H}} \;=\; \Big\langle \sum_{i=1}^n \alpha_i k(\cdot, x_i)\,,\, \sum_{j=1}^n \alpha_j k(\cdot, x_j) \Big\rangle_{\mathcal{H}} \;=\; \sum_{i,j=1}^n \alpha_i \alpha_j k(x_i, x_j)\,.$$

Let $\boldsymbol{K}$ denote the $n \times n$ matrix with $i, j$th entry $k(x_i, x_j)$ and let $\boldsymbol{\alpha} \in \mathbb{R}^n$ be a vector with $i$th entry $\alpha_i$. Then we can write the norm as $\|f\|_{\mathcal{H}} = \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha}$. Thus, knowing that the solution has this form, we may equivalently solve the optimization

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^d} \sum_{i=1}^n \ell\Big(y_i, \sum_{j=1}^n \alpha_j k(x_i, x_j)\Big) \ + \ \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha} \, .$$

This can be solved, for example, using gradient descent. Note that if the loss function is convex, then this is a convex optimization and gradient descent (with a sufficiently small step size) is guaranteed to converge to a minimizer.

## 23.4. Exercises

1. Recall the RKHS
$$\mathcal{H}^1[0,1] = \{f : [0,1] \to \mathbb{R} \, , \, f(0) = 0, \, \|f^{(1)}\|_{L^2} < \infty\}$$
Consider the representor theorem in this case. Describe the nature of the function that solves the optimization.

2. $k(\boldsymbol{x}, \boldsymbol{x}')$ is a valid kernel if and only if for every $n \geq 1$ and every set $\{\boldsymbol{x}_i\}_{i=1}^n$ the matrix $\boldsymbol{K}$ with $ij$-th element $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is positive semi-definite. Use this to show that the following are valid kernels:
   (a) $k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^T \boldsymbol{x}'$
   (b) $k(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^T \boldsymbol{x}' + 1)^p$ for integers $p \geq 1$
   (c) $k(\boldsymbol{x}, \boldsymbol{x}') = f(\boldsymbol{x}) f(\boldsymbol{x}')$ for any function $f$

3. Suppose that $k_1$ and $k_2$ are valid kernels. Show that the following are also kernels.
   (a) $k(\boldsymbol{x}, \boldsymbol{x}') = k_1(\boldsymbol{x}, \boldsymbol{x}') + k_2(\boldsymbol{x}, \boldsymbol{x}')$
   (b) $k(\boldsymbol{x}, \boldsymbol{x}') = k_1(\boldsymbol{x}, \boldsymbol{x}') k_2(\boldsymbol{x}, \boldsymbol{x}')$
   Hint: Consider the Hardamard products of the eigendecompositions $\boldsymbol{K} = \sum_i \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^T$
   (c) $k(\boldsymbol{x}, \boldsymbol{x}') = p(k_1(\boldsymbol{x}, \boldsymbol{x}'))$, where $p$ is a polynomial with positive coefficients
   (d) $k(\boldsymbol{x}, \boldsymbol{x}') = \exp(k_1(\boldsymbol{x}, \boldsymbol{x}'))$
   Hint: Consider the Taylor Series of the exponential function.

4. Let $\{\boldsymbol{x}_i\}_{i=1}^n$ be points in $\mathbb{R}^d$ and assume $n \geq d$. Let $\boldsymbol{X}^T = [\boldsymbol{x}_1 \cdots \boldsymbol{x}_n]$, let $k(\boldsymbol{x}, \boldsymbol{x}') = (\langle \boldsymbol{x}, \boldsymbol{x}' \rangle + 1)^2$ and let $\boldsymbol{K}$ be the associated $n \times n$ Gram matrix with $ij$th entry $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$.
   (a) What is the rank of $\boldsymbol{K}$ and $\boldsymbol{K}_1 = \boldsymbol{X}\boldsymbol{X}^T$?
   (b) Suppose $\boldsymbol{x}_i = \boldsymbol{x}_j$ for some $i \neq j$. What is the rank of $\boldsymbol{K}$ and $\boldsymbol{K}_1 = \boldsymbol{X}\boldsymbol{X}^T$?

5. (Normalized Kernels). If $k$ is a kernel such that $k(\boldsymbol{x}, \boldsymbol{x}) > 0$ for all $\boldsymbol{x}$, then show that
$$\widetilde{k}(\boldsymbol{x}, \boldsymbol{x}) \ := \ \frac{k(\boldsymbol{x}, \boldsymbol{x}')}{\sqrt{k(\boldsymbol{x}, \boldsymbol{x}) k(\boldsymbol{x}', \boldsymbol{x}')}}$$
is also a kernel.

6. (Gaussian Kernel). Show that $k_G(\boldsymbol{x}, \boldsymbol{x}') = e^{-\|\boldsymbol{x} - \boldsymbol{x}'\|^2/\sigma^2}$ is a valid kernel. Hint: Consider the exponential kernel $k_E(\boldsymbol{x}, \boldsymbol{x}') = e^{\boldsymbol{x}^T \boldsymbol{x}'}$ and use the normalization trick above.

7. (Laplace Kernel). Show that for $\alpha > 0$

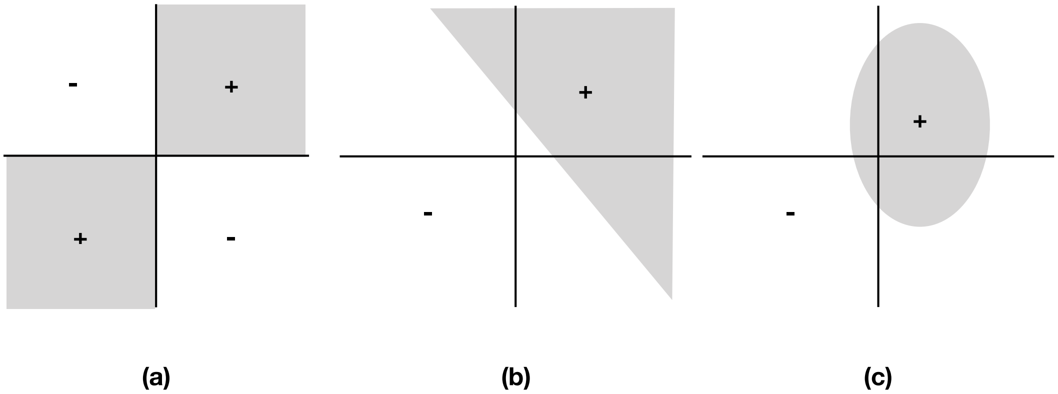
$$k(\boldsymbol{x}, \boldsymbol{x}') \;=\; e^{-\alpha\|\boldsymbol{x}-\boldsymbol{x}'\|}$$

is a kernel. Hint: Use the following fact

$$e^{-\alpha\sqrt{s}} \;=\; \int_0^\infty e^{-su}\frac{\alpha}{2\sqrt{\pi u^3}}e^{-\frac{\alpha^2}{4u}}\,du$$

The Laplacian kernel does not decay to zero as rapidly as the Gaussian kernel, and therefore is less likely to encounter numerical problems.

8. Consider the three binary classification regions in $\mathbb{R}^2$ depicted below. Is there a kernel function that can represent all of them?



(a) (b) (c)

# Lecture 24: Analysis of RKHS Methods

The representer theorem tells us that the problem of finding the function in a (possibly infinite dimensional) RKHS that minimizes training losses can be posed as a finite dimensional optimization of the form

$$\widehat{\alpha} \;=\; \arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^d} \sum_{i=1}^n \ell\Big(y_i, \sum_{j=1}^n \alpha_j k(\boldsymbol{x}_i, \boldsymbol{x}_j)\Big) \;+\; \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha}\;,$$

where $k$ is the reproducing kernel. The function $\widehat{f}(\cdot) = \sum_{i=1}^n \widehat{\alpha}_i k(\cdot, \boldsymbol{x}_i)$ is a solution to

$$\min_{f\in\mathcal{H}} \sum_{i=1}^n \ell(y_i, f(\boldsymbol{x}_i)) \;+\; \|f\|_{\mathcal{H}}^2\;.$$

The soluion $\widehat{f}$ is a weighted combination of kernel functions "centered" at each training point $\boldsymbol{x}_i$. For example, if $k$ is a radial basis kernel, like the Gaussian or Laplacian, then the solution is a weighted combination of "bump" functions at the data points. This lecture analyzes the performance and properties of kernel methods of this type.

## 24.1. Rademacher Complexity Bounds for Kernel Methods

Recall the Rademacher complexity bounds developed in Lecture 20. Let $\mathcal{F}$ be a class of functions, $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ be iid training examples, and $\ell$ be an $L$-Lipschitz loss function. Consider the empirical risk function $\widehat{R}_n(f) = \frac{1}{n}\sum_{i=1}^n \ell(y_i, f(\boldsymbol{x}_i))$ and its expectation $R(f) = \mathbb{E}[\ell(y, f(\boldsymbol{x})]$. Assume the losses are bounded in $[0, C]$. Theorem 12 states that with probability at least $1-\delta$

$$\sup_{f\in\mathcal{F}} R(f) - \widehat{R}_n(f) \;\leq\; 2L\,\mathfrak{R}_n(\mathcal{F}) \;+\; C\sqrt{\frac{\log(1/\delta)}{2n}}$$

where

$$\mathfrak{R}_n(\mathcal{F}) \;=\; \mathbb{E}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \sigma_i f(\boldsymbol{x}_i)\right]\;.$$

To apply this machinery to the RKHS setting, we will consider a constrained form of the optimization

$$\min_{f\in\mathcal{H}} \sum_{i=1}^n \ell(y_i, f(\boldsymbol{x}_i)) \text{ subject to } \|f\|_{\mathcal{H}}^2 \leq B^2\;.$$

In other words, we will consider the Rademacher complexity of the class of functions

$$\mathcal{H}_B = \{f \in \mathcal{H} \,:\, \|f\|_{\mathcal{H}} \leq B\}\;.$$

The bound above yields a *generalization bound* of the following form. For any $\delta > 0$ with probability $1-\delta$

$$R(\widehat{f}) \;\leq\; \widehat{R}(\widehat{f}) + 2L\,\mathfrak{R}_n(\mathcal{H}_B) + C\sqrt{\frac{\log 1/\delta}{2n}},$$

Here $\widehat{f}$ is the function in $\mathcal{H}_B$ that minimizes the training loss, $R(\widehat{f})$ is the test error, and $\widehat{R}(\widehat{f})$ is train error. Recall that this bound assumes that the losses are bounded in $[0, C]$. To check that this requirement is met, consider the loss as a function of $y_i f(\boldsymbol{x}_i)$. Assume that $y_i \in [-1, 1]$ and note that

$$|y_i f(\boldsymbol{x}_i)| \;\;\leq\;\; |f(\boldsymbol{x}_i)| \;=\; |\langle f, k(\cdot, \boldsymbol{x}_i)\rangle_{\mathcal{H}}| \;\leq\; \|f\|_{\mathcal{H}} \|k(\cdot, \boldsymbol{x}_i)\|_{\mathcal{H}}$$

by the Cauchy-Schwartz inequality. Since we are working with the class $\mathcal{H}_B$ we have $\|f\|_{\mathcal{H}} \leq B$. And the reproducing property yields

$$\|k(\cdot,\boldsymbol{x}_i)\|_{\mathcal{H}}^2 \;=\; \langle k(\cdot,\boldsymbol{x}_i, k(\cdot,\boldsymbol{x}_i)\rangle_{\mathcal{H}} \;=\; k(\boldsymbol{x}_i,\boldsymbol{x}_i)\;.$$

Thus, we have $\|k(\cdot,\boldsymbol{x}_i)\|_{\mathcal{H}} \leq \sup_{\boldsymbol{x}} k(\boldsymbol{x},\boldsymbol{x})$. So assuming the kernel function is bounded, we have

$$|y_i f(\boldsymbol{x}_i)| \;\leq\; B \sup_{\boldsymbol{x}} \sqrt{k(\boldsymbol{x},\boldsymbol{x})}\;.$$

Let $C$ be an upper bound on the loss function over the range $[-B\sup_{\boldsymbol{x}}\sqrt{k(\boldsymbol{x},\boldsymbol{x})}, B\sup_{\boldsymbol{x}}\sqrt{k(\boldsymbol{x},\boldsymbol{x})}]$ and assume the loss is lower bounded by $0$. For example, if $k$ is a Gaussian kernel and $\ell$ is logistic or hinge loss, then we can use $C = 1 + B$ and $L = 1$.

We can bound the Rademacher complexity of $\mathcal{H}_B$ as follows.

$$\begin{aligned}
\mathfrak{R}_n(\mathcal{H}_B) &= \frac{1}{n}\mathbb{E}\left[\sup_{f\in\mathcal{H}_B}\sum_{i=1}^n \sigma_i f(\boldsymbol{x}_i)\right] \;=\; \frac{1}{n}\mathbb{E}\left[\sup_{f\in\mathcal{H}_B}\sum_{i=1}^n \sigma_i\langle f, k(\cdot,\boldsymbol{x}_i)\rangle_{\mathcal{H}}\right] \;=\; \frac{1}{n}\mathbb{E}\left[\sup_{f\in\mathcal{H}_B}\left\langle f, \sum_{i=1}^n \sigma_i k(\cdot,\boldsymbol{x}_i)\right\rangle_{\mathcal{H}}\right] \\
&\leq \frac{1}{n}\mathbb{E}\left[\sup_{f\in\mathcal{H}_B}\|f\|_{\mathcal{H}}\|\sum_{i=1}^n \sigma_i k(\cdot,\boldsymbol{x}_i)\|_{\mathcal{H}}\right] \;=\; \frac{B}{n}\mathbb{E}\left[\|\sum_{i=1}^n \sigma_i k(\cdot,\boldsymbol{x}_i)\|_{\mathcal{H}}\right] \\
&\leq \frac{B}{n}\sqrt{\mathbb{E}\left[\|\sum_{i=1}^n \sigma_i k(\cdot,\boldsymbol{x}_i)\|_{\mathcal{H}}^2\right]} \;=\; \frac{B}{n}\sqrt{\mathbb{E}\left[\sum_{i=1}^n\sum_{j=1}^n \sigma_i\sigma_j\langle k(\cdot,\boldsymbol{x}_i), k(\cdot,\boldsymbol{x}_j)\rangle_{\mathcal{H}}\right]} \\
&= \frac{B}{n}\left[\sum_{i=1}^n k(\boldsymbol{x}_i,\boldsymbol{x}_i)\right]^{1/2} \;\leq\; \frac{B}{\sqrt{n}}\sup_{\boldsymbol{x}}\sqrt{k(\boldsymbol{x},\boldsymbol{x})},
\end{aligned}$$

where the first inequality follows by Cauchy-Schwartz inequality, the second inequality follows by the Jensen's inequality, and the last inequality holds because $\mathbb{E}[\sigma_i\sigma_j] = 0$ if $i \neq j$. Putting everything together, we have shown that for any $\delta > 0$ with probability $1 - \delta$

$$R(\widehat{f}) \;\leq\; \widehat{R}(\widehat{f}) + 2L\,\frac{B\sup_{\boldsymbol{x}}\sqrt{k(\boldsymbol{x},\boldsymbol{x})}}{\sqrt{n}} + C\sqrt{\frac{\log 1/\delta}{2n}}\;.$$

For example, if we use logistic or hinge loss and a radial kernel function like the Gaussian or Laplacian kernel, then we have

$$R(\widehat{f}) \;\leq\; \widehat{R}(\widehat{f}) + \frac{2B}{\sqrt{n}} + (1+B)\sqrt{\frac{\log 1/\delta}{2n}}\;.$$

## 24.2. Properties of Kernel Functions

The Rademacher complexity bound depends on the maximum value of the kernel function, but otherwise does not reflect particular characteristics of the kernel function. To gain insight into the differences between kernels and the RKHSs they generate, let us focus on translation-invariant kernels that only depend on the difference between $\boldsymbol{x}$ and $\boldsymbol{x}'$. We will denote translation-invariant kernels as $k(\boldsymbol{x},\boldsymbol{x}') = k(\boldsymbol{x}-\boldsymbol{x}')$. The Gaussian kernel $k(\boldsymbol{x},\boldsymbol{x}') = \exp(\alpha\|\boldsymbol{x}-\boldsymbol{x}'\|_2^2)$ is an example.

We will use the Fourier transform to study such kernels. Recall the Fourier transform of a function $f \in L^2(\mathbb{R}^d)$ is

$$F(\boldsymbol{\omega}) \;=\; \int f(\boldsymbol{x})e^{-i\boldsymbol{\omega}^T\boldsymbol{x}}d\boldsymbol{x}$$

and the inverse transform is

$$f(\boldsymbol{x}) \;=\; \frac{1}{(2\pi)^d}\int F(\boldsymbol{\omega})e^{i\boldsymbol{\omega}^T\boldsymbol{x}}\,d\boldsymbol{\omega}\;.$$

Here $i = \sqrt{-1}$, $\boldsymbol{\omega}, \boldsymbol{x} \in \mathbb{R}^d$, and $e^{i\boldsymbol{\omega}^T\boldsymbol{x}} = \cos(\boldsymbol{\omega}^T\boldsymbol{x}) + i\sin(\boldsymbol{\omega}^T\boldsymbol{x})$. The squared $L^2$ norm $\int |f(\boldsymbol{x})|^2 d\boldsymbol{x}$ can be intepreted as the total "energy" of the function $f$. The Fourier transform $F(\boldsymbol{\omega})$ indicates how much of the energy is associated with each frequency $\boldsymbol{\omega}$.

We will specifically kernels that can be expressed in terms of the parameter $\boldsymbol{\rho} = \boldsymbol{x} - \boldsymbol{x}'$ and consider the Fourier transform of the function $k(\boldsymbol{\rho})$.

**Example 24.2.1.** *Consider Gaussian kernels of the form* $k(\boldsymbol{\rho}) = \sigma^{-d}\exp\big(-\frac{\|\boldsymbol{\rho}\|_2^2}{2\sigma^2}\big)$*, for some* $\sigma > 0$ *that controls the width of the Gaussian bump. The Fourier transform of* $k$ *is*

$$K(\boldsymbol{\omega}) \;=\; \exp(-\sigma^2\|\boldsymbol{\omega}\|_2^2/2)\;.$$

*This shows that the Fourier transform decays exponentially as the frequency of oscillation* $\|\boldsymbol{\omega}\|_2$ *increases and as* $\sigma^2$ *increases. Larger values of* $\sigma^2$ *correspond to broader and smoother bumps. This tells us that solutions based on Gaussian kernels tend to be relatively smooth functions.*

**Example 24.2.2.** *Consider Laplacian of the form* $k(\boldsymbol{\rho}) = \exp(-\alpha\|\boldsymbol{\rho}\|_2)$*, for some* $\alpha > 0$ *that controls the width of this sort of bump. The Fourier transform of* $k$ *is*

$$K(\boldsymbol{\omega}) \;=\; 2^{d/2}\alpha\sqrt{\pi}\,\Gamma\big(d/2+1\big)\,\big(\alpha^2 + \|\boldsymbol{\omega}\|_2^2\big)^{-\frac{d+1}{2}}\;,$$

*where* $\Gamma$ *is the Gamma function satisfying* $\Gamma(n+1) = n!$ *when* $n$ *is a positive integer. This shows us that its Fourier transform decays less rapidly than in the Gaussian case; like* $\|\boldsymbol{\omega}\|_2^{-(d+1)}$ *which is much slower than exponential decay. This tells us that solutions based on Laplacian kernels tend to be less smooth in comparison to those based on Gaussian kernels .*

This Fourier analysis has several practical implications. As noted above, different kernels induce different spectral decays in the frequency domain. If we have prior knowledge that the function we are trying to learn has certain frequency characteristics, then we can try to match the kernel to these characteristics. For example, if we know that the true function has little or no energy above a certain frequency, then we can use this information to choose $\sigma$ or $\alpha$.

## 24.3. Take-Away Messages

Let $\mathcal{H}$ be an RKHS with kernel $k$ and consider the ball of radius $B > 0$ in $\mathcal{H}$:

$$\mathcal{H}_B \;=\; \big\{f \in \mathcal{H} \,:\, \|f\|_{\mathcal{H}} \le B\big\}\;.$$

We saw that a solution to the constrained optimization

$$\min_{f\in\mathcal{H}} \sum_{i=1}^n \ell(y_i, f(\boldsymbol{x}_i)) \text{ subject to } \|f\|_{\mathcal{H}} \le B$$

can be represented $\widehat{f}(\cdot) = \sum_{i=1}^n \alpha_i k(\cdot, \boldsymbol{x}_i)$, for some $\alpha_i$ depending on the data. This is remarkable and potentially useful, but since $\mathcal{H}$ may be contain very complex functions we should worry about possibly overfitting to training data.

The Rademacher complexity analysis shows that learning is well-posed if $B/\sqrt{n}$ is small, since this ensures the solution will generalize well to new examples. This bound cannot be improved too much, since in general we have a $n^{-1/2}$ term in the generalization bound in finite classes too. From this perspective, we see that learning in an RKHS ball is not more difficult that learning with a finite class of functions. This might seem surprising, but it crucially depends on the assumption that the norm of the solution is at most $B$. This is a restriction on what functions we consider. As we increase our training set size, we might want to allow for more functions and let $B$ grow with $n$. Also, we might want to allow $B$ to depend on the dimension of the feature space, possibly even exponentially in $d$. So there may be a lot that we are constraining or ruling out with the norm bound.

We also discussed how the Fourier transforms of translation invariant kernels can have dramatically different decay characteristics. The decay of the Fourier transform, along with the norm bound $B$, affect how rapidly varying the solution can be. Consider the RKHS balls associated with the Gaussian kernel $k(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\alpha^2\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2)$ and the Laplacian kernel $k(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\alpha\|\boldsymbol{x} - \boldsymbol{x}'\|_2)$ for some $\alpha > 0$, denoted by $\mathcal{H}_B^G$ and $\mathcal{H}_B^L$, respectively. These balls both contain the function $f = 0$, but there are many functions that may be in one ball and not the other. The Rademacher complexity analysis tells us that both will generalize well if $B/\sqrt{n}$ is small, but the two solutions may be very different functions. This means that the empirical risk of one solution may be much smaller than the other and thus lead to a smaller bound on the generalization error. A nice overview of other approaches to understanding kernel methods is given in [11].

To put the analysis to practical use, we can find solutions using different kernels (varying the kernel function and parameters) and then check the norms and the empirical risks of the various solutions. If a particular solution has small empirical risk and a small norm in its RKHS, then this indicates it may be a better solution than another that has a larger empirical risk and/or larger norm in its RKHS. Of course, another practical approach in this setting is cross-validation: “hold-out" some of the training data and then estimate the error rate of each solution using these data. The Rademacher complexity analysis is complementary to this as it (a) sheds light on the tradeoffs involved in learning good classifiers and (b) could be used as a criterion for selection that does not require splitting the data into train and validation sets (all the available data is used for training).

## 24.4. Exercises

1. Suppose that instead of learning a function from point evaluations, we instead consider learning a function from generic continuous linear measurements. We can formulate this learning problem over a Hilbert space $\mathcal{H}$. We can model continuous linear measurements of a function $f \in \mathcal{H}$ by inner products of the form $\langle \nu_i, f\rangle$, $i = 1, \dots, n$, where $\nu_i \in \mathcal{H}$, $i = 1, \dots, n$ are the measurement functionals. Prove that if a solution exists to the following optimization problem

   $$\min_{f\in\mathcal{H}} \sum_{i=1}^{n} \ell(y_i, \langle \nu_i, f\rangle) + \lambda\|f\|_{\mathcal{H}}^2, \quad \lambda > 0,$$

   then the solution admits a representation of the form

   $$f = \sum_{i=1}^{n} \alpha_i \nu_i, \quad \alpha_1, \dots, \alpha_n \in \mathbb{R}.$$

   *Hint: The RKHS representer theorem is a special case of this problem when $\nu_i = k(\cdot, \boldsymbol{x}_i)$ since $\langle k(\cdot, \boldsymbol{x}_i), f\rangle = f(\boldsymbol{x}_i)$ by the reproducing property. Adapt the proof of the RKHS representer theorem.*

2. Some kernels can be associated with an explicit feature map. For example, the polynomial kernel $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\boldsymbol{x}_1^T\boldsymbol{x}_2 + 1)^p = \phi(\boldsymbol{x}_1)^T\phi(\boldsymbol{x}_2)$, where $\phi(\boldsymbol{x})$ is a $D \times 1$ vector containing all monomials of the elements in $\boldsymbol{x}$

up to and including degree $p$ (here $D$ is the number of distinct monomial terms). In this case, the solution to a learning problem can be written as $\widehat{f}(\boldsymbol{x}) = \boldsymbol{w}^T\phi(\boldsymbol{x})$ for some weight $\boldsymbol{w} \in \mathbb{R}^D$. Suppose that $\|\boldsymbol{x}_i\|_2 \leq 1$, $i = 1, \ldots, n$. Derive a Rademacher complexity bounds using the kernel approach discussed in this lecture and the linear modeling approach discussed in Lecture 20 with the model $\boldsymbol{w}^T\phi(\boldsymbol{x})$. How do the bounds compare?

3. Let $k$ be a Gaussian or Laplacian kernel and let $\mathcal{H}$ denote the corresponding RKHS.

   (a) Show that the solution to

   $$\min_{f\in\mathcal{H}} \sum_{i=1}^{n} (y_i - f(\boldsymbol{x}_i))^2 + \lambda \|f\|_{\mathcal{H}}^2$$

   has the form $\widehat{f}_\lambda(\cdot) = \sum_{i=1}^n \widehat{\alpha}_i k(\cdot, \boldsymbol{x}_i)$ where $\widehat{\boldsymbol{\alpha}} \in \mathbb{R}^n$ is given by

   $$\widehat{\boldsymbol{\alpha}} = \big(\boldsymbol{K} + \lambda \boldsymbol{I}\big)^{-1} \boldsymbol{y}$$

   where $\boldsymbol{y} \in \mathbb{R}^n$ is the vector of the labels.

   (b) Argue that, in general, the kernel matrix $\boldsymbol{K}$ with $ij$ entry equal to $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ will be full rank (i.e., positive definite). Exceptions include cases where certain $\boldsymbol{x}_i$ values are repeated, for example.

   (c) Assume $\boldsymbol{K}$ is full rank. Then we may take $\lambda = 0$ and obtain the solution $\widehat{\boldsymbol{\alpha}} = \boldsymbol{K}^{-1}\boldsymbol{y}$. What is the training error of the solution $\widehat{f}_0$? That is, what is $\sum_{i=1}^n (y_i - \widehat{f}_0(\boldsymbol{x}_i))^2$?

   (d) The training error is the same for any solution of this form, no matter the choice of kernel or its width parameter. Discuss why and how different choices might lead to different predictions on new test examples. Use insights from the Rademacher complexity analysis to explain how different choices might lead to better or worse generalization.

# Lecture 25: Neural Networks

Recall positive definite kernel $k$ generates an RKHS and the solutions to learning in an RKHS have the form

$$f(\cdot) \;=\; \sum_{i=1}^{n} \alpha_i k(\cdot, \boldsymbol{x}_i)$$

where $\{\boldsymbol{x}_i\}_{i=1}^n$ are the training examples and $\{\alpha_i\}_{i=1}^n \subset \mathbb{R}$. We can interpret such a function as a linear combination of fixed nonlinear functions $\{k(\cdot, \boldsymbol{x}_i)\}_{i=1}^n$. Two layer neural networks have a similar form and construction. Let $\sigma : \mathbb{R} \to \mathbb{R}$ be an “activation function." The most common activation function today is the Rectified Linear Unit (ReLU) defined by $\sigma(\cdot) = \max\{0, \cdot\}$. A two layer neural network is a function $f : \mathbb{R}^d \to \mathbb{R}$ of the form

$$f(\boldsymbol{x}) \;=\; \sum_{j=1}^{m} v_j \, \sigma(\boldsymbol{w}_j^T \boldsymbol{x} + b_j) \;,\; \forall \boldsymbol{x} \in \mathbb{R}^d \;,$$

where $v_j, b_j \in \mathbb{R}$ and $\boldsymbol{w}_j \in \mathbb{R}^d$ are “trainable" parameters. Like the RKHS functions, neural network functions can be viewed as a linear combination of nonlinear functions. Unlike kernel methods, the nonlinear functions in a neural network are not fixed, since $\boldsymbol{w}_j$ and $b_j$ are adjustable parameters. This is a key distinction between neural networks and kernel methods.

## 25.1. Neural Network Function Spaces

To be concrete, let us fix the activation function to be the ReLU. Also for notational convenience, we will append the bias $b_j$ to the input weight vector $\boldsymbol{w}_j$ and append a 1 to $\boldsymbol{x}$ so that each neuron is written as

$$\sigma(\boldsymbol{w}_j^T \boldsymbol{x}) \;=\; \max(0, \boldsymbol{w}_j^T \boldsymbol{x}) \;=:\; (\boldsymbol{w}_j^T \boldsymbol{x})_+$$

We will assume this for the remainder of the discussion. Define the set of neural network functions as

$$\mathcal{F} \;=\; \Big\{ f \;:\; f(\boldsymbol{x}) = \sum_{j=1}^{m} v_j (\boldsymbol{w}_j^T \boldsymbol{x})_+ ,\; m \geq 1,\; \boldsymbol{w}_j \in \mathbb{R}^{d+1},\; v_j \in \mathbb{R} \Big\} \,.$$

If $f, g \in \mathcal{F}$, then $f + g$ is also a neural network function in $\mathcal{F}$ and so $\mathcal{F}$ is a vector space (all the other properties of a vector space are easily verified).

The next step is to add a norm to this vector space. Since the weights of a neural network determine the function it represents, any norm we choose will depend on the weights in some way. One natural thing we can try is to define the norm of a neural network function as some norm on the weights of the neural network. Let us consider the Euclidean norm of the weights, the square root of the sum of squared weights. To gain some insight, consider a simple ReLU neural network with a single neuron, $f(\boldsymbol{x}) = v(\boldsymbol{w}^T \boldsymbol{x})_+$. The Euclidean norm of the weights is $(\|\boldsymbol{w}\|_2^2 + |v|^2)^{1/2}$. Note that the function $f$ tends to zero if $v$ or $\boldsymbol{w}$ tends to zero, but the Euclidean norm of the weights tends to zero only if both $v$ and $\boldsymbol{w}$ tend to zero. For example, if $\boldsymbol{w} \neq 0$ and $v = 0$, then $f = 0$ but $(\|\boldsymbol{w}\|_2^2 + |v|^2)^{1/2} \neq 0$. This shows that the Euclidean norm of the weights is not a valid norm for neural network functions.

The problem arises because of the way neural networks are parameterized, with both input and output weights for each neuron. Inspection of $f(\boldsymbol{x}) = v(\boldsymbol{w}^T \boldsymbol{x})_+$ shows that $f$ tends to the $0$ function if and only if the product

$|v|\,\|\boldsymbol{w}\|_2 \to 0$. This suggests using this product as the basis for a norm on $\mathcal{F}$. Consider a general $f \in |F$ of the form $f(\boldsymbol{x}) = \sum_{j=1}^m v_j(\boldsymbol{w}_j^T\boldsymbol{x})_+$ and consider $\|f\| := \sum_{j=1}^m \|v_j\boldsymbol{w}_j\|_2$ as a norm. Then $\|f\| = 0$ if and only if $f = 0$ and $\|\alpha f\| = \|\sum_{j=1}^m \alpha v_j(\boldsymbol{w}_j^T\boldsymbol{x})_+\| = \sum_{j=1}^m \|\alpha v_j\boldsymbol{w}_j\|_2 = |\alpha|\|f\|$ for any $\alpha \in \mathbb{R}$. Let $\widetilde{f}$ be another neural network function with $\widetilde{m}$ neurons and weights $\widetilde{\boldsymbol{w}}_j$ and $\widetilde{v}_j$. Then we have

$$\|f + \widetilde{f}\| \;\le\; \sum_{j=1}^m \|v_j\boldsymbol{w}_j\|_2 + \sum_{j=1}^{\widetilde{m}} \|\widetilde{v}_j\widetilde{\boldsymbol{w}}_j\|_2 \;=\; \|f\| + \|\widetilde{f}\| \,.$$

The inequality arises because there could neurons in $f$ and $\widetilde{f}$ with exactly the same input weights and biases but output weights of differing signs. So this is indeed a valid norm on $\mathcal{F}$, and it is often called the *path-norm* of the network. The vector space $\mathcal{F}$ of two layer ReLU neural networks equipped with the path-norm is a normed vector space and its completion is a Banach space[11].

While the path-norm may seem unusual at first glance, it is actually arises naturally in neural network training. The most common sort regularization in neural network training is called “weight decay," explained as follows. Let $L(f) = \sum_{i=1}^n \ell(y_i, f(\boldsymbol{x}_i))$, the sum of losses for neural network function $f$ on a training set. Consider the optimization

$$\min_f L(f) + \frac{\lambda}{2}\sum_{j=1}^m \left(\|\boldsymbol{w}_j\|_2^2 + |v_j|^2\right) ,$$

Neural network training is based on gradient descent. The negative gradient of the objective with respect to any weight in the network, say $v_k$ for example, is $-\frac{\partial L}{\partial v_k} - \lambda v_k$. So each step of gradient descent will take a small step in the direction $-\frac{\partial L}{\partial v_k}$ with a proportionately small amount of weight decay $-\lambda v_k$. In other words, weight decay in gradient descent training is equivalent to regularizing the sum of squared weights.

The ReLU is piecewise linear and the functions $\alpha^{-1}(\alpha x)_+$ are equivalent for all $\alpha > 0$. Let us reconsider the optimization in light of this. We can scale the input and output weights of the $j$th neuron by $\alpha_j > 0$ and $\alpha_j^{-1}$ without affecting the neural network function. Let $f_\alpha$ denote this equivalent function and consider the optimization

$$\min_{f_\alpha} L(f_\alpha) + \frac{\lambda}{2}\sum_{j=1}^m \left(\alpha_j^2\|\boldsymbol{w}_j\|_2^2 + \alpha_j^{-2}|v_j|^2\right) .$$

The loss is invariant to $\alpha$, but for any set of weights it is easy to check that the regularization term is smallest for $\alpha_j^2 = |v_j|/\|\boldsymbol{w}_j\|_2$. Thus, at a global minimum of the objective function we have

$$\frac{1}{2}\left(\|\boldsymbol{w}_j\|_2^2 + |v_j|^2\right) = \|v_j\boldsymbol{w}_j\|_2 \,.$$

This implies that solutions to the optimization

$$\min_{f_\alpha} L(f) + \lambda\sum_{j=1}^m \|v_j\boldsymbol{w}_j\|_2 \,,$$

are equivalent to those of the weight decay objective.

Another perspective that sheds some light on this choice of norm is the notion of stability. We call a function $f$ *stable* if $f(\boldsymbol{x}) \approx f(\boldsymbol{x} + \boldsymbol{\epsilon})$ for any small perturbation $\boldsymbol{\epsilon}$ and every $\boldsymbol{x}$. Stable functions have good generalization

[11] Because this space contains generalized functions (measures), like the Dirac delta, the completion is not with respect to the norm topology of the space, but with respect to the weak* topology via a Prokhorov’s theorem.

and robustness properties, since the produce similar outputs for similar inputs. Consider a single ReLU neuron function $f(\boldsymbol{x}) = v_j(\boldsymbol{w}_j^T\boldsymbol{x})_+$ and note that

$$|f(\boldsymbol{x}+\boldsymbol{\epsilon}) - f(\boldsymbol{x})| \;\leq\; \|v_j\boldsymbol{w}_j\|_2 \, \|\boldsymbol{\epsilon}\|_2 \, .$$

So it is stable if the product $v_j\boldsymbol{w}_j$ has a small norm. This is the case if both $v_j$ and $\boldsymbol{w}_j$ are small, but also if one is large and the other is much smaller. This illustrates the problem with the Euclidean norm of the weights. The Euclidean norm of the weights $(\sum_j \|\boldsymbol{w}_j\|_2^2 + |v_j|^2)^{1/2}$ is small if both $v_j$ and $\boldsymbol{w}_j$ are small, but it is not small if one is large and the other is very small (e.g., $\|\boldsymbol{w}_j\|_2 = 1$ and $v_j = 0.001$). So the Euclidean norm of the weights does not reflect the stability of neural network functions.

## 25.2. ReLU Neural Network Banach Space

To characterize the sort of functions are in the two layer ReLU neural network Banach space let us consider one-dimensional (univariate) case where $f : \mathbb{R} \to \mathbb{R}$. For this characterization we will not regularize the biases $b_j$ and so the path norm in this case is simply $\sum_{j=1}^m |v_j w_j|$. Regularizing the biases is unnecessary since if $v_j = 0$, then the neuron does not contribute to the neural network function $f$. To ensure this is still a valid norm, we can fix $|w_j| = 1$ and absorb its scale into $v_j$. With this normalization, the path norm is simply $\sum_{j=1}^m |v_j|$.

The derivative of a ReLU function is a step function, that is for any $b \in \mathbb{R}$

$$\frac{\partial\sigma(x-b)}{\partial x} \;=\; \frac{\partial \max\{0, x-b\}}{\partial x} \;=\; \begin{cases} 1 & x \geq b \\ 0 & x < b \end{cases}$$

Consider a univariate neural network function $f(x) = \sum_{j=1}^m v_j\, \sigma(w_j x + b_j)$. Its path-norm is $\sum_{j=1}^m |v_j w_j|$. We can move all the scaling of $|w_j|$ out of the ReLU funtions and write $f(x) = \sum_{j=1}^m v_j |w_j|\, \sigma\big(\frac{w_j}{|w_j|}(x + b_j/w_j)\big)$. The derivative of $f$ is

$$f'(x) \;=\; \sum_{j=1}^m v_j |w_j|\, u\Big(\frac{w_j}{|w_j|}(x + b_j/|w_j|)\Big) \, ,$$

where $u(\cdot)$ is the unit step function that is 1 when its argument is nonnegative and 0 otherwise. So $f'$ is a piecewise constant function. The *total variation* of such a function is the sum of the sizes of the changes/jumps in the function. So the total variation of $f'$ is

$$\mathrm{TV}(f') \;=\; \sum_{j=1}^m |v_j w_j| \, .$$

In other words, in the univariate case the path-norm is equal to the total variation of the derivative of $f$. The Banach space of functions with derivatives of finite total variation is called $\mathrm{BV}^2(\mathbb{R})$. This is the ReLU neural network Banach space. If a function $f$ has its second derivative $f'' \in L^1(\mathbb{R})$, then $\mathrm{TV}(f') = \int |f''(x)|\, dx$. So we can think of the path-norm as measuring the $L^1$ norm of the second derivative of the neural network function.

## 25.3. Exercises

1. Let $f$ be a univariate two layer ReLU neural network. What is its second derivative? Is it in $L^1(\mathbb{R})$?
2. The path-norm of a univariate ReLU neural network is $\sum_{j=1}^m |v_j w_j|$, which is the $\ell^1$ norm of the vector of $\{v_j w_j\}_{j=1}^m$. Since regularizing with the path-norm is equivalent to weight decay, what does suggest about the nature of solutions as we increase $\lambda$?

# Lecture 26: Neural Network Approximation and Generalization Bounds

Let $\sigma$ be the ReLU activation function, $\sigma(x) = \max\{0, x\}$, and consider two layer neural networks with neurons of the form $\sigma(\boldsymbol{w}^T\boldsymbol{x} + b)$ with $\boldsymbol{x} \in \mathbb{R}^d$. We will append the bias of each neuron to the input weight vector and append a 1 to $\boldsymbol{x}$, denoted by $\boldsymbol{x}$. Hence, we will use the notation $\sigma(\boldsymbol{w}^T\boldsymbol{x})$, with $\boldsymbol{w} \in \mathbb{R}^{d+1}$, for each neuron. Because the ReLU function is piecewise linear, the size of $\boldsymbol{w}$ can be absorbed into the output weight $v$, so let us assume that $\|\boldsymbol{w}\|_2 = 1$. Consider the space of neural network functions mapping $\mathbb{R}^d \to \mathbb{R}$

$$\left\{f \,:\, f(\boldsymbol{x}) = \sum_{j=1}^{m} v_j\sigma(\boldsymbol{w}_j^T\boldsymbol{x}),\, m \geq 1,\, \boldsymbol{w}_j \in \mathbb{R}^{d+1},\, \|\boldsymbol{w}_j\|_2 = 1,\, v_j \in \mathbb{R}\right\}.$$

The vectors in $\mathbb{R}^{d+1}$ satisfying $\|\boldsymbol{w}\|_2 = 1$ is the surface of the unit sphere in $d+1$ dimensions, denoted by $\mathbb{S}^d$. Let $\mathcal{F}$ be the space of all functions of the form

$$f(\boldsymbol{x}) \;=\; \int \sigma(\boldsymbol{w}^T\boldsymbol{x})\, d\nu(\boldsymbol{w})$$

where $\nu(\boldsymbol{w})$ is a finite measure on $\mathbb{S}^d$. The measure $\nu$ plays the role of the output weights. If we take the measure $d\nu(\boldsymbol{w}) = \sum_{j=1}^m v_j\,\delta(\boldsymbol{w} - \boldsymbol{w}_j)\,d\boldsymbol{w}$, the integral formula produces the finite width neural network

$$f(\boldsymbol{x}) \;=\; \sum_{j=1}^{m} v_j\,\sigma(\boldsymbol{w}_j^T\boldsymbol{x})\;.$$

Thus, $\mathcal{F}$ contains all functions in the vector space above. The measure $\nu$ can be split into positive and negative parts $\nu = \nu^+ + \nu^-$ and suggests the norm

$$\|f\| \;=\; \int_{\mathbb{S}^d} d\nu^+(\boldsymbol{w}) - \int_{\mathbb{S}^d} d\nu^-(\boldsymbol{w})\;.$$

Observe that for a finite width neural network $\|f\| = \sum_{j=1}^m |v_j|$.

There is a small problem with the norm defined above, due to the fact that the same function could be represented by different neural networks (and different measures $\nu$). For example, adding the neurons $v\sigma(\boldsymbol{w}^T\boldsymbol{x})$ and $-v\sigma(\boldsymbol{w}^T\boldsymbol{x})$ to any network does not change the function it represents (since the contributions of the two neurons cancel each other). To deal with this, we will define the norm of a function $f$ to be

$$\|f\| \;:=\; \inf_{\nu\,:\, f=\int\sigma(\boldsymbol{w}^T\boldsymbol{x})\,d\nu(\boldsymbol{w})}\left(\int_{\mathbb{S}^d} d\nu^+(\boldsymbol{w}) - \int_{\mathbb{S}^d} d\nu^-(\boldsymbol{w})\right)\;.$$

Taking the infimum over representations eliminates the problem of non-uniqueness.

Equipped with $\|f\|$, $\mathcal{F}$ is a Banach space written as

$$\mathcal{F} \;=\; \left\{f \,:\, f(\boldsymbol{x}) = \int\, \sigma(\boldsymbol{w}^T\boldsymbol{x})\,d\nu(\boldsymbol{w})\;,\; \|f\| < \infty\right\}.$$

Specialized to the case $d = 1$, this is the space BV$^2$ discussed in the last lecture. Roughly speaking, we can think of this as the space of functions with absolutely integrable second derivatives. For $d \geq 1$, the space $\mathcal{F}$ is also characterized in terms of second derivatives, but measured in the Radon transform domain [20, 21].

## 26.1. Approximating Functions in $\mathcal{F}$

In general, an $f \in \mathcal{F}$ is represented by an infinite width neural network. However, in practice we will always use finite width neural networks. How wide should a practical neural network be? To answer this we will quantify how well such any function in $\mathcal{F}$ can be *approximated* by a neural network of width $m$ in the following sense. Let $\mathcal{F}_m$ denote the set of all neural networks with width at most $m$ and for any $f \in \mathcal{F}$ consider

$$\min_{f_m \in \mathcal{F}_m} \|f - f_m\|_{L^2(\Omega)}$$

where $\|g\|^2_{L^2(\Omega)} := \int_\Omega |g(\boldsymbol{x})|^2 d\boldsymbol{x}$ for some bounded domain $\Omega \subset \mathbb{R}^d$. A small approximation error $\|f - f_m\|_{L^2(\Omega)}$ means that $f_m$ is a good approximation to $f$. To interpret this, consider the following. Suppose we pick $\boldsymbol{x}$ at random from a probability density $p$ supported in $\Omega$. Then

$$\mathbb{E}[|f(\boldsymbol{x}) - f_m(\boldsymbol{x})|^2] \;\leq\; M\|f - f_m\|^2_{L^2(\Omega)}$$

where $M$ is the maximum value of the density $p$. Here is the result we will prove, which is from [1] and based on a simple argument attributed to Maurey in [22].

**Theorem 26.1.1.** *Let $\Omega \subset \mathbb{R}^d$ be a bounded domain. Then there exists a constant $C_0 > 0$ such that for every $m \geq 1$ and any $f \in \mathcal{F}$ there is a width $m$ neural network satisfying*

$$\|f - f_m\|^2_{L^2(\Omega)} \;\leq\; \frac{C_0}{m} .$$

The theorem shows that the approximation error is proportional to $1/m$, which means that finite width neural networks are good approximations to any function in $\mathcal{F}$. Remarkably, the approximation rate has no dependence on the dimension $d$ of the domain. For any $\epsilon > 0$, a network of width $O(1/\epsilon)$ is sufficient for $\epsilon$ approximation error. This is unusual since for most common multivariate function approximation methods, the rate depends on the dimension like $\sqrt[d]{m}$ or equivalently an $\epsilon$ approximation error requires $O(1/\epsilon^d)$ terms. For example, suppose the domain $\Omega = [0,1]^d$ an consider a histogram approximation (piecewise constant approximation) of $f \in \mathcal{F}$. In general, we would require at least $m^d$ bins in the histogram partition to guarantee an error of $1/m$. This is the so-called "curse of dimensionality". The same is true for any kernel method. In contrast, neural network approximations in $\mathcal{F}$ are *immune* to the curse.

The reason for this immunity is simple. Consider the ball of radius $C > 0$ in $\mathcal{F}$. This contains many functions, including finite width neural networks, which must satisfy $\sum_j |v_j| \leq C$. This constraint means that the finite width networks may have many neurons with very small $v_j$, but only a few with larger $v_j$. The magnitude $|v_j|$ is the slope of the $j$th ReLU function. So functions in any ball of finite radius in $\mathcal{F}$ may only have large slopes and variation in (at most) a small number of directions, which are determined by the corresponding input weights. So the space $\mathcal{F}$ contains functions that are very smooth except possibly in a few directions. In this sense, any function in $\mathcal{F}$ is intrinsically low dimensional, but there is no single low-dimensional subspace that contains every function in $\mathcal{F}$. The key characteristic of neural networks is that they can adapt to the special directions of an underlying $f$ by learning the appropriate input weights. In this sense, the neurons are what we might call "steerable".

We conclude this section with a elementary proof of the theorem based on a probabilistic argument. We have specialized this result to the space $L^2(\Omega)$ and to ReLU neural networks. However, the arguments only hinge on the fact that the functions we aim to approximate are essentially $\ell^1$ combinations of functions that are bounded on $\Omega$. So the same result can hold for other constructions that meet these requirements, including neural networks with different activation functions.

*Proof.* Without loss of generality, assume that $\|f\| \leq C$. Let $C > 0$ and define the set of neurons

$$\mathcal{N}_C \;=\; \left\{\eta \,:\, \eta(\boldsymbol{x}) = v\,\sigma(\boldsymbol{w}^T\boldsymbol{x})\,,\, |v| \leq C\right\}.$$

Let conv$(\mathcal{N}_C)$ denote the convex hull of $\mathcal{N}_C$. This is the set of all functions of the form

$$f(\boldsymbol{x}) \;=\; \sum_{j\geq 1} \gamma_j v_j\,\sigma(\boldsymbol{w}_j^T\boldsymbol{x})$$

where $\gamma_j \geq 0$ and $\sum_{j\geq 1}\gamma_j = 1$. In other words, conv$(\mathcal{N}_C)$ contains all two layer neural networks with $\|\boldsymbol{v}\|_1 \leq C$, where $\boldsymbol{v}$ is a vector containing $\{v_j\}_{j\geq 1}$. These neural networks may be arbitrarily wide, but must satisfy this $\ell^1$ bound on the vector of output weights.

Because $|v| \leq C$, $\|\boldsymbol{w}\|_2 = 1$, and $\Omega$ is a bounded domain, every $\eta \in \mathcal{N}_C$ is bounded on $\Omega$ and therefore there exists a $B > 0$ such that $\|\eta\|_{L^2(\Omega)} \leq B$ for all $\eta \in \mathcal{N}_C$. This and the triangle inequality implies that $\|f\|_{L^2(\Omega)} \leq B$ for all $f \in$ conv$(\mathcal{N}_C)$.

Let $\mathcal{G}_C$ denote the completion of conv$(\mathcal{N}_C)$ in $L^2(\Omega)$. This means conv$(\mathcal{N})$ is dense in $\mathcal{G}_C$ (with respect to the $L^2(\Omega)$ norm) and $f \in \mathcal{G}_C$. For ease of notation, we will denote the norm in $L^2(\Omega)$ by $\|\cdot\|_{L^2}$, that is $\|f\|_{L^2}^2 = \int_\Omega |f(\boldsymbol{x})|^2 d\boldsymbol{x}$.

Given some $\delta > 0$ and $f \in \mathcal{G}_C$, there exists $\bar{f} \in$ conv$(\mathcal{N}_C)$ satisfying $\|f - \bar{f}\|_{L^2} \leq \delta/m$, since conv$(\mathcal{N}_C)$ is dense in $\mathcal{G}_C$. Thus, $\bar{f} = \sum_{j=1}^N \gamma_j\bar{\eta}_j$ with $\bar{\eta}_j \in \mathcal{N}_C$, $\gamma_j \geq 0$, $\sum_{j=1}^N |\gamma_j| = 1$ for some sufficiently large $N$, possibly much larger than $m$. Let $\eta_i$, $i = 1, \ldots, m$, be drawn independently from $\{\bar{\eta}_1, \ldots, \bar{\eta}_N\}$ according to the probabilities $\{\gamma_1, \ldots, \gamma_N\}$. That is, $\mathbb{P}(\eta_i = \bar{\eta}_j) = v_j$. Let $\widehat{f}_m = \frac{1}{m}\sum_{i=1}^m \eta_i$. Because $\mathbb{E}[\eta_i] = \sum_{j=1}^n \gamma_j\bar{\eta}_j = \bar{f}$, we have $\mathbb{E}[\widehat{f}_m] = \bar{f}$. Furthermore

$$\begin{aligned}
\mathbb{E}\big[\|\widehat{f}_m - \bar{f}\|_{L^2}^2\big] \;&=\; \mathbb{E}\Big[\Big\|\frac{1}{m}\sum_{i=1}^m \eta_i - \bar{f}\Big\|_{L^2}^2\Big] \;=\; \frac{1}{m}\mathbb{E}[\|\eta_1 - \bar{f}\|_{L^2}^2] \\
&=\; \frac{1}{m}\big(\mathbb{E}[\|\eta_1]\|_{L^2}^2] - \|\bar{f}\|_{L^2}^2]\big) \;\leq\; \frac{B^2 - \|\bar{f}\|_{L^2}^2}{m} \;\leq\; \frac{B^2}{m}
\end{aligned}$$

where the second and third inequalities follow from the the fact that the $\eta_i$ are iid and $\mathbb{E}[\eta_i] = \bar{f}$. Since $\widehat{f}_m$ satisfies the bound on average, the must exist at least one specific $f_m \in$ conv$(\mathcal{N}_C)$ that does as well. Then we have

$$\|f - f_m\|_{L^2} \;=\; \|f - f_m + f_m - \bar{f}\|_{L^2} \;\leq\; \|f - \bar{f}\|_{L^2} + \|\bar{f} - f_m\|_{L^2} \;\leq\; \delta/m + \frac{B^2}{m}\,.$$

Since $\delta > 0$ was arbitrary this completes the proof. □

## 26.2. Generalization Bounds for Neural Networks

Single-hidden-layer neural networks are functions mapping $\mathbb{R}^d \to \mathbb{R}$ with the following form

$$f(\boldsymbol{x}) \;=\; \sum_{j=1}^m v_j\,\sigma(\boldsymbol{w}_j^T\boldsymbol{x})\,,$$

where $\sigma$ is an activation function. Although different choices are possible we will focus on the popular Rectified Linear Unit (ReLU) activation function, defined as $\sigma(x) = \max\{0, x\}$. As above the input $\boldsymbol{x} \in \mathbb{R}^d$, $\boldsymbol{x} \in \mathbb{R}^{d+1}$ is $\boldsymbol{x}$ appended with a 1, and $\boldsymbol{w}_j \in \mathbb{R}^{d+1}$.

**Theorem 26.2.1.** *Let $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ be a set of points in $\mathbb{R}^d$. Consider the class of two layer neural networks*

$$\mathcal{F}_C = \left\{ f(\boldsymbol{x}) = \sum_{j=1}^{m} v_j \, \sigma(\boldsymbol{w}_j^T \boldsymbol{x}) \; : \; m \geq 1 \, , \, \sum_{j=1}^{m} |v_j| \, \|\boldsymbol{w}_j\| \leq C \right\} .$$

*The empirical Rademacher complexity of $\mathcal{F}_C$ satisfies*

$$\widehat{\mathfrak{R}}_n(\mathcal{F}_C(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)) \; \leq \; \frac{2C}{n} \sqrt{\sum_{i=1}^{n} \|\boldsymbol{x}_i\|^2} \, .$$

Note that this bound does not involve $m$, the number of neurons. Rather, it depends on the *size* of the neural network weights. This gives some insight on why having a large number of neurons does not negatively impact the ability of neural networks to generalize well.

*Proof.* We will take $C = 1$ in the proof and the same argument applies for any $C > 0$. Recall $\sigma_i$, $i = 1, \ldots, n$, are $\pm 1$ valued iid random variables (not to be confused with $\sigma$ which denotes the ReLU).

$$\begin{aligned}
&\widehat{\mathfrak{R}}_n(\mathcal{F}_1, (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)) \\
&= \frac{1}{n} \mathbb{E} \left[ \sup_{f \in \mathcal{F}} \sum_{i=1}^{n} \sigma_i f(\boldsymbol{x}_i) \right] \\
&= \frac{1}{n} \mathbb{E} \left[ \sup_{\{v_j, \boldsymbol{w}_j\} \,:\, \sum_{j=1}^N |v_j| \, \|\boldsymbol{w}_j\| \leq 1} \quad \sum_{i=1}^{n} \sigma_i \sum_{j=1}^{N} v_j \sigma(\boldsymbol{w}_j^T \boldsymbol{x}_i) \right] \\
&= \frac{1}{n} \mathbb{E} \left[ \sup_{\{v_j, \boldsymbol{w}_j\} \,:\, \sum_{j=1}^N |v_j| \, \|\boldsymbol{w}_j\| \leq 1} \quad \sum_{j=1}^{N} v_j \|\boldsymbol{w}_j\| \left( \sum_{i=1}^{n} \; \sigma_i \sigma(\boldsymbol{x}_i^T \boldsymbol{w}_j / \|\boldsymbol{w}_j\|) \right) \right] \\
&\leq \frac{1}{n} \mathbb{E} \left[ \sup_{\{v_j, \boldsymbol{w}_j\} \,:\, \sum_{j=1}^N |v_j| \, \|\boldsymbol{w}_j\| \leq 1} \quad \left( \sum_{j=1}^{N} |v_j| \|\boldsymbol{w}_j\| \right) \max_{1 \leq j \leq N} \left| \sum_{i=1}^{n} \; \sigma_i \sigma(\boldsymbol{x}_i^T \boldsymbol{w}_j / \|\boldsymbol{w}_j\|) \right| \right] ,
\end{aligned}$$

where in second to last step we used the following property of the ReLU: for $\alpha \geq 0$ $\sigma(\alpha z) = \alpha \sigma(z)$. The last step follows from Hölder's inequality, namely, for two vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^d$ we have

$$\boldsymbol{a}^T \boldsymbol{b} = \sum_{i=1}^{d} a_i b_i \leq \|\boldsymbol{a}\|_1 \|\boldsymbol{b}\|_\infty = \sum_{i=1}^{d} |a_i| \max_i |b_i| \, .$$

Continuing, we have

$$\begin{aligned}
\widehat{\mathfrak{R}}_n(\mathcal{F}_1, (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)) &\leq \frac{1}{n}\mathbb{E}\left[\sup_{\{\boldsymbol{w}_j\}\,:\,\|\boldsymbol{w}_j\|\leq 1} \max_{1\leq j\leq N}\left|\sum_{i=1}^{n} \sigma_i \sigma(\boldsymbol{x}_i^T \boldsymbol{w}_j)\right|\right] \\
&= \frac{1}{n}\mathbb{E}\left[\sup_{\boldsymbol{w}:\|\boldsymbol{w}\|\leq 1}\left|\sum_{i=1}^{n} \sigma_i \sigma(\boldsymbol{x}_i^T \boldsymbol{w})\right|\right] \\
&\leq \frac{2}{n}\mathbb{E}\left[\sup_{\boldsymbol{w}:\|\boldsymbol{w}\|\leq 1}\left|\sum_{i=1}^{n} \sigma_i \boldsymbol{x}_i^T \boldsymbol{w}\right|\right] \\
&\leq \frac{2}{n}\mathbb{E}\left[\left\|\sum_{i=1}^{n} \sigma_i \boldsymbol{x}_i^T\right\|\right] \\
&\leq \frac{2}{n}\sqrt{\sum_{i=1}^{n} \|\boldsymbol{x}_i\|^2}\,.
\end{aligned}$$

The first equality holds since the supremum over $\boldsymbol{w}_j$ is the same for all terms. The first inequality is the *contraction property* of Rademacher complexity (see Lemma 26.2.2 below). The second to last inequality follows by the Cauchy-Scwartz inequality.

The following 2-sided generalization of Lemma 20.2.1, sometimes called the "contraction" property of Rademacher complexity, can be found in [6, Theorem 11.6].

**Lemma 26.2.2.** *Consider $z_1(\theta), z_2(\theta), \dots, z_n(\theta)$, a collection of stochastic processes indexed by $\theta \in \Theta$. Let $\sigma_1, \dots, \sigma_n$ be independent Rademacher random variables. Then, for any $L-$Lipschitz function $\varphi$ satisfying $\varphi(0) = 0$*

$$\mathbb{E}\left[\sup_{\theta\in\Theta}\left|\sum_{i=1}^{n} \sigma_i\, \varphi\big(z_i(\theta)\big)\right|\right] \;\leq\; 2L\,\mathbb{E}\left[\sup_{\theta\in\Theta}\left|\sum_{i=1}^{n} \sigma_i\, z_i(\theta)\right|\right].$$

□

## 26.3. Exercises

1. Consider a neural network function $f(\boldsymbol{x}) = \sum_{j=1}^{m} v_j \sigma(\boldsymbol{w}_j^T \boldsymbol{x})$. Its norm is $\|f\| = \sum_{j=1}^{m} \|v_j \boldsymbol{w}_j\|_2$. Show that the space of functions with this norm is not a Hilbert space by demonstrating that it fails to satisfy the Parallelogram Law. Hint: Consider functions $f$ and $g$ with just one ReLU neuron each.

2. In engineering textbooks, the Dirac delta on $\mathbb{R}$ is a distribution (generalized function) defined as

$$\delta(x) = \begin{cases} 0 & x \neq 0 \\ \infty & x = 0 \end{cases}$$

with the property that

$$\int_{\mathbb{R}} f(x)\delta(x)\,\mathrm{d}x = f(0)$$

for some sufficiently nice $f : \mathbb{R} \to \mathbb{R}$. Use this property to morally[12] show that $\delta \notin L^2(\mathbb{R})$.

*Hint: Morally,* $\|\delta\|_{L^2}^2 = \int_{\mathbb{R}} \delta(x)\delta(x)\,\mathrm{d}x$.

3. Let $\Omega \subset \mathbb{R}^d$ be a bounded domain. Consider a set of functions $\Phi$, where each $\varphi \in \Phi$ maps from $\Omega$ to $\mathbb{R}$ and is bounded on $\Omega$. The set may be uncountably infinite. Consider the convex hull

$$\mathcal{C}_\Phi \;=\; \Big\{ f \,:\, f = \sum_{j \geq 1} \gamma_j \varphi_j \,,\, \varphi_j \in \Phi \,,\, \gamma_j \geq 0 \,,\, \sum_{j \geq 1} \gamma_j = 1 \Big\}$$

Let $\|\varphi\|_\infty = \sup_{\boldsymbol{x} \in \Omega} |\varphi(\boldsymbol{x})|$ and define the norm of $\|f\| = \sum_{j \geq 1} \gamma_j \|\varphi_j\|_\infty$. Let $\bar{\mathcal{C}}_\Phi$ denote the closure of $\mathcal{C}_\Phi$ with respect to the norm $L^2(\Omega)$. Show that there exists a constant $C > 0$ such that for every $m \geq 1$ and any $f \in \bar{\mathcal{C}}_\Phi$ there is an $m$-term function in $f_m \in \mathcal{C}_\Phi$ (i.e., at most $m$ nonzero terms) satisfying

$$\int_\Omega |f(\boldsymbol{x}) - f_m(\boldsymbol{x})|^2 d\boldsymbol{x} \;\leq\; \frac{C}{m} \,.$$

Notice there is no dependence on the dimension of $\Omega$.

4. In most cases the feature vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are bounded (meaning their norm is finite). Assume that $\|\boldsymbol{x}_i\|_2 \leq B$ show that the Rademacher complexity of $\mathcal{F}_C$ is bounded by $2C\sqrt{B^2+1}/\sqrt{n}$.

5. Let $\widehat{f} \in \mathcal{F}_C$ be a binary classifier that minimizes the logistic or hinge loss on an iid training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$. Bound the generalization error $\mathbb{P}(y \neq f(\boldsymbol{x}))$, where $(\boldsymbol{x}, y)$ drawn from the same distribution as the training data.

[12] This fact can be shown rigorously using standard duality arguments used when working with distributions, but is out of the scope of this class.

# Appendix A: Notation

$\mathbb{P}(A)$ is the probability of the event $A$ with respect to everything that is random in the definition of $A$. The symbol $\mathbb{P}$ alone means the joint distribution of everything random in the setting under consideration.

$\mathbb{P}_{XY}$ is the joint distribution of $(X, Y)$, $\mathbb{P}_X$ is the marginal distribution of $X$.

$\mathbb{E}[Z]$ is the expectation of the random variable $Z$ with respect to everything that is random in the definition of $Z$.

$\mathbb{V}[Z] = \mathbb{E}[(Z - \mathbb{E}[Z])^2]$ is the variance of the random variable $Z$ with respect to everything that is random in the definition of $Z$.

$\mathbb{E}_{Y|X}[f(X, Y)]$ is the conditional expectation of the random variable $f(X, Y)$ given $X$. This is also sometimes denoted as $\mathbb{E}[f(X, Y)|X]$.

$p(x, y)$ denotes the joint probability/mass function of $X$ and $Y$. In classification settings $Y$ is discrete valued and $X$ maybe be continuous and/or discrete valued. If both $X$ and $Y$ are discrete, then $p(x, y)$ denotes the probability that $X = x$ and $Y = x$. If $X$ is continuous, then $p(x, y) = p(x|y)p(y)$ where $p(x|y)$ is the probability density of $X$ given $Y = y$ and $p(y)$ is the probability that $Y = y$. $p(x)$ is the marginal density (or mass function) of $x$.

$\mathbb{P}(Y = 1|X = x)$ is more explicit notation for the conditional probability that $Y = 1$ given $X = x$, which is the same as $p(1|x) = p(x, 1)/p(x)$, using the notation above. It's used to make clear that the 1 refers to $Y = 1$. Another way this might be denoted is $p_{Y|X}(1|x)$ which also clarifies that it refers the situation when $Y = 1$.

$X \sim \mathbb{P}_X$ means that $X$ is a random variable with distribution $\mathbb{P}_X$. If $p(x)$ denotes the probability density function of $X$, then we may also write $X \sim p$.

If $X_1, X_2, \ldots, X_n$ are independent and identically distributed according to $\mathbb{P}_X$ with density function $p(x)$, then we write $X_1, \ldots, X_n \overset{\text{iid}}{\sim} \mathbb{P}_X$ or $X_1, \ldots, X_n \overset{\text{iid}}{\sim} p$.

For clarity, we use $X$ to denote a random variable and $x$ to denote a specific value that $X$ may take. Sometimes this becomes cumbersome. For example, we often use capital letters to denote matrices and lower case letters to denote vectors and scalars. So we just use $x$ generically and the context will dictate whether we are talking about the random variable or a specific value taken by the random variable.

In linear algebra notation, vectors and matrices are often denoted with **bold** lower or upper case symbols. For example, $\mathbf{x}$, $\mathbf{X}$, and $x$ denote a vector, matrix, and scalar, respectively.

# Appendix B: Useful Inequalities

**Markov's Inequality:** For any nonnegative (scalar) random variable $X$

$$\mathbb{P}(X \geq t) \;\leq\; \frac{\mathbb{E}[X]}{t}$$

**Chebyshev's Inequality:** Let $X$ be a random variable with mean $\mu$ and variance $\sigma^2$. Then

$$\mathbb{P}(|X-\mu| \geq t) \;\leq\; \frac{\sigma^2}{t^2}$$

**Jensen's Inequality:** Let $X$ be a random variable. For any convex function $\varphi$

$$\mathbb{E}[\varphi(X)] \geq \varphi(E[X])$$

**Cauchy-Schwarz Inequality:** Let $f$ and $g$ be functions. Then

$$\int f(x)g(x)\,dx \;\leq\; \Big(\int |f(x)|^2\,dx\Big)^{1/2}\Big(\int |g(x)|^2\,dx\Big)^{1/2}$$

If $x$ and $y$ are random variables, then

$$|\mathbb{E}[xy]|^2 \;\leq\; \mathbb{E}[x^2]\,\mathbb{E}[y^2]$$

**Hölders's Inequality:** Let $f$ and $g$ be functions, and let $p, q \geq 1$ satisfy $1/p + 1/q \leq 1$. Then

$$\int |f(x)g(x)|\,dx \;\leq\; \Big(\int |f(x)|^p\,dx\Big)^{1/p}\Big(\int |g(x)|^q\,dx\Big)^{1/q}$$

**Chernoff's Bound:** Let $z_1, z_2, ..., z_n$ be independent bounded random variables such that $z_i \in [0,1]$ with probability 1. Then for any $\epsilon > 0$

$$\mathbb{P}\Big(\Big|\frac{1}{n}\sum_{i=1}^{n}(z_i - \mathbb{E}[z_i])\Big| \geq \epsilon\Big) \;\leq\; 2e^{-2n\epsilon^2}$$

**Hoeffding's Inequality:** Let $z_1, z_2, ..., z_n$ be independent bounded random variables such that $z_i \in [a_i, b_i]$ with probability 1. Let $S_n = \sum_{i=1}^{n} z_i$. Then for any $t > 0$, we have

$$\mathbb{P}(|S_n - \mathbb{E}[S_n]| \geq t) \leq 2\,e^{-\frac{2t^2}{\sum_{i=1}^{n}(b_i-a_i)^2}}$$

# Appendix C: Convergence of Random Variables

The fact that averages of realizations of random variables converge to the corresponding expected (mean) value is central to the analysis and design of machine learning algorithms. This note discusses several forms of convergence. Let $X$ be a real-valued random variable, and let $X_1, X_2, \ldots$ be an infinite sequence of independent and identically distributed copies of $X$.

## 29.1. Law of Large Numbers

Let $\widehat{\mu}_n = \frac{1}{n}\sum_{i=1}^n X_i$, $n \geq 1$, be the empirical averages of this sequence. The law of large numbers refers to the fact that $\widehat{\mu}_n \to \mathbb{E}[X]$ as $n \to \infty$. Specifically, there is a weak and strong version of the law of large numbers.

**Weak Law of Large Numbers.** If $\mathbb{E}[|X|] < \infty$, then $\widehat{\mu}_n$ converges *in probability* to $\mathbb{E}[X]$, i.e., for every $\epsilon > 0$

$$\lim_{n\to\infty} \mathbb{P}(|\widehat{\mu}_n - \mathbb{E}[X]| \geq \epsilon) = 0 \; .$$

**Strong Law of Large Numbers.** If $\mathbb{E}[|X|] < \infty$, then $\widehat{\mu}_n$ converges *almost surely* to $\mathbb{E}[X]$, i.e.,

$$\mathbb{P}(\lim_{n\to\infty} \widehat{\mu}_n = \mathbb{E}[X]) = 1 \; .$$

Proving the laws of large numbers involves a few standard results from analysis (e.g., the Borel-Cantelli lemma). Here we will simply provide a bit of intuition based on Markov's inequality: for any $X$ then $\mathbb{P}(X^2 \geq a) \leq \frac{\mathbb{E}[X^2]}{a}$ for any $a > 0$. So, suppose that $\mathbb{E}[X^2] < \infty$, a slightly stronger moment condition than required for the laws of large numbers. Then we have

$$\mathbb{P}(|\widehat{\mu}_n - \mathbb{E}[X]| \geq \epsilon) \;=\; \mathbb{P}(|\widehat{\mu}_n - \mathbb{E}[X]|^2 \geq \epsilon^2) \;\leq\; \frac{\mathbb{E}[|\widehat{\mu}_n - \mathbb{E}[X]|^2]}{\epsilon^2} \;=\; \frac{\frac{1}{n}\mathbb{E}[|X_1 - \mathbb{E}[X]|^2]}{\epsilon^2} \;\leq\; \frac{\mathbb{E}[X^2]}{n\epsilon^2} \; ,$$

where we use the fact that the variance of a sum of independent random variables is equal to the sum of the variances of each term. Thus, $\mathbb{P}(|\widehat{\mu}_n - \mathbb{E}[X]| \geq \epsilon) \to 0$ as $n \to \infty$ for *any* $\epsilon > 0$. With a bit more work, we can show this holds for *every* $\epsilon > 0$.

## 29.2. Central Limit Theorem

The LLN tells us that empirical averages converge to expected values, but little else about the (random) behavior of averages. The most elementary characterization of this is the Central Limit Theorem (CLT), which states that the distribution of averages of random variables tends to a Gaussian distribution.

**Central Limit Theorem.** If $\mathbb{E}[X] = \mu$ and $\mathbb{E}[|X - \mu|^2] = \sigma^2 < \infty$, then $\sqrt{n}(\widehat{\mu}_n - \mu)$ converges *in distribution* to $\mathcal{N}(0, \sigma^2)$.

Notice that $\sqrt{n}\,\widehat{\mu}_n = \frac{1}{\sqrt{n}}\sum_{i=1}^n X_i$. Therefore, the variance of $\sqrt{n}\,\widehat{\mu}_n$ is

$$\mathbb{V}(\widehat{\mu}_n) \;=\; \frac{1}{n}\mathbb{V}\Big(\sum_{i=1}^n X_i\Big) \;=\; \frac{1}{n}\sum_{i=1}^n \mathbb{V}(X_i) \;=\; \sigma^2 \;,$$

since the $X_i$ are independently and identically distributed with common variance $\sigma^2$. In other words, normalizing the sum by $\sqrt{n}$ rather than $n$ *stablizes* the variance and therefore it converges to a finite-variance random variable, rather than a deterministic constant. Stating the result a slightly different way, for large values of $n$, the distribution of the empirical average $\widehat{\mu}_n$ is approximately $\mathcal{N}(\mu, \sigma^2/n)$. This provides a characterization of the random fluctuations of the empirical average (roughly Gaussian with variance $\sigma^2/n$).

The CLT suggests that it should be possible to sharpen Markov's inequality to obtain a better bound on the deviations of $\widehat{\mu}_n$. Suppose $X \sim \mathcal{N}(0, \sigma^2)$. Markov's inequality shows that $\mathbb{P}(|X| > t) = \mathbb{P}(|X|^2 > t^2) \leq \frac{\sigma^2}{t^2}$, for any $t > 0$. However, the tail of the Gaussian distribution decays exponentially, and therefore

$$\mathbb{P}(X > t) \;=\; \frac{1}{\sqrt{2\pi}}\int_t^\infty e^{\frac{-x^2}{2\sigma^2}}dx \;\leq\; \frac{1}{2}e^{\frac{-t^2}{2\sigma^2}}$$

To see this consider

$$R \;:=\; \frac{\frac{1}{\sqrt{2\pi\sigma^2}}\int_t^\infty e^{\frac{-x^2}{2\sigma^2}}dx}{e^{-\frac{t^2}{2\sigma^2}}} \;=\; \frac{1}{\sqrt{2\pi\sigma^2}}\int_t^\infty e^{\frac{-(x^2-t^2)}{2\sigma^2}}dx \;=\; \frac{1}{\sqrt{2\pi\sigma^2}}\int_t^\infty e^{\frac{-(x-t)(x+t)}{2\sigma^2}}dx$$

Let $y = x - t$, then

$$R \;=\; \frac{1}{\sqrt{2\pi\sigma^2}}\int_0^\infty e^{\frac{-y(y+2t)}{2\sigma^2}}dy \;\leq\; \frac{1}{\sqrt{2\pi\sigma^2}}\int_0^\infty e^{\frac{-y^2}{2\sigma^2}}dy \;=\; \frac{1}{2}$$

Now let us apply this reasoning to $\widehat{\mu}_n$. Assume that $\widehat{\mu}_n - \mathbb{E}[X] \sim \mathcal{N}(0, \sigma^2/n)$, which the CLT shows is approximately correct for large $n$. Markov's inequality gives the bound $\mathbb{P}(|\widehat{\mu}_n - \mathbb{E}[X]| > t) \leq \frac{\sigma^2}{nt^2}$. However, the Gaussian tail-bound shows that

$$\mathbb{P}(|\widehat{\mu}_n - \mu| > t) \;\leq\; \exp\Big(-\frac{nt^2}{2\sigma^2}\Big)\;,$$

which is exponentially smaller (as a function of $n$) than the bound given by Markov's inequality.

## 29.3. Law of the Iterated Logarithm

The LLN shows that averages converge to the expected value and the CLT characterizes the distribution of the average in the large-sample limit. The sequence of empirical means $\{\widehat{\mu}_n\}_{n\geq 1}$ is a random process that fluctuates about $\mathbb{E}[X]$. Another natural question is to quantify how large these fluctuations may be, and this is what the Law of the Iterated Logarithim (LIL) tells us.

The LLN and CLT consider partial sums of $X_1, X_2, \ldots$ normalized by $n$ or $\sqrt{n}$, respectively. Partial sums normalized by a higher power of $n$ converge to $0$ (not interesting) and partial sums normalized by a lower power than $n^{1/2}$ will not converge to a finite variance random variable (not interesting). So it is reasonable to consider normalizations between $n^{1/2}$ and $n$. In particular, normalizing by $\sqrt{n \log\log n}$ characterizes the maximal deviations of the sequence of empirical means from the expected value.

**Law of the Iterated Logarithm.**

$$\mathbb{P}\Big( \limsup_{n\to\infty} \frac{\sum_{i=1}^{n}(X_i - \mathbb{E}[X])}{\sqrt{2\sigma^2\, n \log\log n}} = 1 \Big) \;=\; 1 \,.$$

Roughly speaking, this tells us that the sequence of random variables $\{|\widehat{\mu}_n - \mu|\}_{n\geq 1}$ tends to be bounded by the function $\sqrt{\frac{2\sigma^2 \log\log n}{n}}$, which is $\sqrt{2\log\log n}$ times the standard deviation of $\widehat{\mu}_n - \mu$ (note $\sqrt{\mathbb{E}[|\widehat{\mu}_n - \mu|^2]} = \sqrt{\frac{\sigma^2}{n}}$).

# Bibliography

[1] Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

[2] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[3] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[4] CM Bishop. Pattern recognition and machine learning (information science and statistics), 2006.

[5] David Blackwell. Conditional expectation and unbiased sequential estimation. *The Annals of Mathematical Statistics*, pages 105–110, 1947.

[6] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

[7] Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509, 2006.

[8] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[9] Thomas M Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers*, pages 326–334, 1965.

[10] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[11] Felipe Cucker and Steve Smale. On the mathematical foundations of learning. *Bulletin of the American mathematical society*, 39(1):1–49, 2002.

[12] Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media, 2013.

[13] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[14] David L Donoho and Jain M Johnstone. Ideal spatial adaptation by wavelet shrinkage. *biometrika*, 81(3):425–455, 1994.

[15] Mário AT Figueiredo and Robert D Nowak. An em algorithm for wavelet-based image restoration. *IEEE Transactions on Image Processing*, 12(8):906–916, 2003.

[16] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[17] Rudolph Kalman. E. 1960. a new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82:35–45, 1960.

[18] Nicolai Meinshausen and Bin Yu. Lasso-type recovery of sparse representations for high-dimensional data. *The annals of statistics*, 37(1):246–270, 2009.

[19] Ron Meir and Tong Zhang. Generalization error bounds for bayesian mixture algorithms. *Journal of Machine Learning Research*, 4(Oct):839–860, 2003.

[20] Greg Ongie, Rebecca Willett, Daniel Soudry, and Nathan Srebro. A function space view of bounded norm infinite width relu nets: The multivariate case. In *International Conference on Learning Representations*, 2019.

[21] Rahul Parhi and Robert D Nowak. Banach space representer theorems for neural networks and ridge splines. *J. Mach. Learn. Res.*, 22(43):1–40, 2021.

[22] G Pisier. Remarques sur un résultat non publié de b. maurey. *Séminaire d'Analyse fonctionnelle (dit" Maurey-Schwartz")*, pages 1–12, 1980.

[23] Eugen Slutsky. `http://en.wikipedia.org/wiki/Slutsky's_theorem`.

[24] Aad W van der Vaart and Jon A Wellner. Weak convergence and empirical processes with applications to statistics. *Journal of the Royal Statistical Society-Series A Statistics in Society*, 160(3):596–608, 1997.

[25] Norbert Wiener, Norbert Wiener, Cyberneticist Mathematician, Norbert Wiener, Norbert Wiener, and Cybernéticien Mathématicien. *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications*, volume 113. MIT press Cambridge, MA, 1949.

[26] Stephen J Wright, Robert D Nowak, and Mário AT Figueiredo. Sparse reconstruction by separable approximation. *IEEE Transactions on signal processing*, 57(7):2479–2493, 2009.

[27] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936, 2003.