ML Preparation Notes

April 29, 2021

Chapter 1

Basic Statistics

In this section, we review some basic concepts from Statistics. The material in this section is based on [4].

1.1 Covariance and Correlation

The *correlation* between two sets of data is a measure of the strength of the relationship between them. In particular, Pearson's correlation coefficient is a measure of linear relationship between two sets of data. Let X and Y be two random variables. Then Pearson's correlation coefficient $\rho(X, Y)$ is defined as:

$$\rho(X, Y) = \frac{\mathrm{Cov}(X, Y)}{\sigma_X \, \sigma_Y} \tag{1.1}$$

Two important facts about the Pearson's correlation coefficient [3]:

1. $-1 \leq \rho(X, Y) \leq 1$
2. $|\rho(X, Y)| = 1$ iff there exists $a \neq 0$ and b such that $Y = aX + b$.

1.2 Tail Inequalities and the Law of Large Numbers

We first look at Markov and Chebyshev inequalities and then apply them to random samples.

The Markov inequality tells us how much probability mass can be at large values given the mean of the distribution.

Theorem 1.1. *Let X be a random variable that takes on positive values only. Then for $t > 0$,*

$$\mathrm{P}(X \geq t) \leq \frac{\mathrm{E}[X]}{t}. \tag{1.2}$$

Proof. Let us assume that X is a continuous r.v. with pdf f_X. By definition,

$$\begin{aligned} \mathrm{E}[X] &= \int_0^\infty x f_X(x) dx \\ &= \int_0^t x f_X(x) dx + \int_t^\infty x f_X(x) dx \\ &\geq t \int_t^\infty f_X(x) dx. \end{aligned}$$

Since $t > 0$, dividing both sides of the last inequality yields the Markov inequality. □

This inequality is useful when $t > \mathrm{E}[X]$; when $t \leq \mathrm{E}[X]$, it merely bounds $\mathrm{P}(X \geq t)$ by 1.

The Chebyschev inequality involves both the mean and the variance of the distribution. It bounds the probability of how far a random variable can be from its mean as a function of the variance.

Theorem 1.2. *Let X be a random variable for which* $\mathrm{Var}(X)$ *exists. Then*

$$\mathrm{P}(|X - \mathrm{E}[X]| > t) \leq \frac{\mathrm{Var}(X)}{t^2}. \tag{1.3}$$

Proof. Define $Y := (X - \mathrm{E}[X])^2$ so that $\mathrm{E}[Y] = \mathrm{Var}(X)$. Applying the Markov inequality to the r.v. Y, we obtain:

$$\mathrm{P}(Y \geq t^2) \leq \frac{\mathrm{Var}(X)}{t^2}.$$

But $\mathrm{P}(Y \geq t^2) = \mathrm{P}(|X - \mathrm{E}[X]| > t)$ and so this proves the Chebyschev inequality too. □

To talk about the law of large numbers, one has to talk about the notion of a sequence of random variables converging to a real number. We say that a sequence $Z_1, Z_2, \ldots$ of random variables converges to the number b if the distribution of Z_n as $n \to \infty$ becomes more and more concentrated around this single number.

Definition 1.1 (Convergence in Probability)**.** A sequence $Z_1, Z_2, \ldots$ of random variables converges in probability to the number b if for every $\epsilon > 0$, the following condition holds:

$$\lim_{n\to\infty} \mathrm{P}(|Z_n - b| < \epsilon) = 1. \tag{1.4}$$

This fact is denoted by $Z_n \xrightarrow{p} b$.

Theorem 1.3 (The Law of Large Numbers)**.** *Let $X_1, \ldots, X_n$ be a random sample from a distribution with mean μ and finite variance σ^2. Then the sample mean $\bar{X}_n$ converges in probability to μ.*

Proof. We have $\mathrm{E}[\bar{X}_n] = \mu$ and $\mathrm{Var}(\bar{X}_n) = \sigma^2/n$. Use the Chebyschev inequality to obtain:

$$\mathrm{P}(|\bar{X}_n - \mu| \leq t) \geq 1 - \frac{\sigma^2}{nt^2}.$$

For every fixed $t > 0$, take limits as $n \to \infty$ to obtain: $\lim_{n\to\infty} \mathrm{P}(|\bar{X}_n - \mu| \leq t) = 1$. Hence $\bar{X} \xrightarrow{p} \mu$. □

If we know that a sequence of random variables converges in probability, what can we say about continuous functions of that sequence of random variables?

Theorem 1.4 (Continuous Functions of Random Variables)**.** *Suppose a sequence $Z_1, Z_2, \ldots$ converges in probability to b and that g is a function continuous at b. Then $g(Z_n) \xrightarrow{p} g(b)$.*

Proof. We have to show $\lim_{n\to\infty} \mathrm{P}(|g(Z_n) - g(b)| < \epsilon) = 1$. Since g is continuous at b, given $\epsilon > 0$ there exists $\delta > 0$ such that $|z_n - b| < \delta$ implies $|g(z_n) - g(b)| < \epsilon$. Now consider the events $\mathcal{E}_1 = \{|Z_n - b| < \delta\}$ and $\mathcal{E}_2 = \{|g(Z_n) - g(b)| < \epsilon\}$. By the continuity of g, $\mathcal{E}_1$ implies $\mathcal{E}_2$, and we have that $\mathrm{P}(\mathcal{E}_2) \geq \mathrm{P}(\mathcal{E}_1)$. Since $\lim_{n\to\infty} \mathrm{P}(\mathcal{E}_1) = 1$, we must have $\lim_{n\to\infty} \mathrm{P}(\mathcal{E}_2) = 1$. □

Theorem 1.5 (Chernoff Bounds)**.** *Let X be a random variable with moment generating function ψ_X. Then for any real t,*

$$\mathrm{P}(X \geq t) \leq \min_{s>0} e^{-st}\psi_X(s). \tag{1.5}$$

Proof. Fix $s > 0$. Since the map $x \longrightarrow e^x$ is 1-1 and increasing, we have that $X \geq t$ if and only if $e^{sX} \geq e^{st}$. Therefore by the Markov inequality, we obtain:

$$\mathrm{P}(X \geq t) = \mathrm{P}(e^{sX} \geq e^{st}) \leq \frac{\mathrm{E}[e^{sX}]}{e^{st}} = e^{-st}\psi_X(s).$$

This holds for any $s > 0$ and, in particular, when we minimize $e^{-st}\psi_X(s)$ subject to the condition that $s > 0$. □

Example 1.1. For each positive integer n, let X_n be a nonnegative random variable with finite mean μ_n. Show that if $\lim_{n\to\infty} \mu_n = 0$, then $X_n \xrightarrow{p} 0$.

Solution. We have to show that for every $\epsilon > 0$, $\lim_{n\to\infty} \mathrm{P}(X_n \geq \epsilon) = 0$. To this end, fix $\epsilon > 0$ and $\delta > 0$. Since $\lim_{n\to\infty} \mu_n = 0$, there exists $n_{\epsilon\delta} \in \mathbf{N}$ such that for all $n \geq n_{\epsilon\delta}$, $\mu_n < \epsilon \cdot \delta$. Combining this with the Markov inequality, we obtain that for all $n \geq n_{\epsilon\delta}$

$$\mathrm{P}(X_n \geq \epsilon) \leq \frac{\mu_n}{\epsilon} < \delta.$$

This is precisely what we need to show in order to prove that $\lim_{n\to\infty} \mathrm{P}(X_n \geq \epsilon) = 0$. ■

▷

Example 1.2. Let $Z_1, Z_2, \ldots$ be a sequence of random variables, and suppose that, for $n = 1, 2, \ldots$, the distribution of Z_n is as follows:

$$\mathrm{P}(Z_n = n^2) = \frac{1}{n} \text{ and } \mathrm{P}(Z_n = 0) = 1 - \frac{1}{n}.$$

Show that $\lim_{n\to\infty} \mathrm{E}[Z_n] = \infty$ but $Z_n \xrightarrow{p} 0$.

Solution. Now $\mathrm{E}[Z_n] = n^2 \cdot \frac{1}{n} = n$ and clearly $\mathrm{E}[Z_n] \to \infty$ as $n \to \infty$. Since $\lim_{n\to\infty} \mathrm{P}(Z_n = 0) = 1$, it follows that for any $\epsilon > 0$, $\lim_{n\to\infty} \mathrm{P}(Z_n > \epsilon) = 0$. This shows that $Z_n \xrightarrow{p} 0$. ■

▷

1.3 The Central Limit Theorem

The CLT is a formal statmenet of how normal distributions can approximate distributions of general sums or averages of iid random variables. To start with, consider the case when $X_1, \ldots, X_n$ are iid random variables with $X_i \sim N(\mu_i, \sigma_i^2)$ for $1 \leq i \leq n$. Let $Y = a_1X_1 + \cdots + a_nX_n + b$, where $a_i \neq 0$ for at least one $1 \leq i \leq n$. Then Y is normally distributed as the following shows.

Theorem 1.6. *Let $X_1, \ldots, X_n$ be iid normally distributed random variables with X_i having mean μ_i and variance σ_i^2, for $1 \leq i \leq n$. Define $Y = a_1X_1 + \cdots + a_nX_n + b$, where $a_i \neq 0$ for at least one $1 \leq i \leq n$. Then Y is normally distributed with mean $a_1\mu_1 + \cdots + a_n\mu_n + b$ and variance $a_1^2\mu_1^2 + \cdots + a_n^2\mu_n^2$.*

Proof. Let $\psi_i(t)$ be the mgf of X_i for $1 \le i \le n$, and let $\psi(t)$ denote the mgf of $a_1X_1 + \cdots + a_nX_n + b$. Then

$$\begin{aligned}
\psi(t) &= \mathrm{E}[e^{t(a_1X_1+\cdots+a_nX_n+b)}] \\
&= e^{tb} \cdot \prod_{i=1}^{n} \mathrm{E}[e^{a_iX_it}] \\
&= e^{tb} \cdot \prod_{i=1}^{n} \psi_i(a_it) \\
&= e^{tb} \cdot \prod_{i=1}^{n} \exp\left(a_i\mu_it + \frac{1}{2}a_i^2\sigma_i^2t^2\right) \\
&= \exp\left(\left(b + \sum_i \mu_i\right)t + \frac{1}{2}\left(\sum_i a_i^2\sigma_i^2\right)t^2\right).
\end{aligned}$$

Note that $\psi(t)$ is the mgf of a normal distribution with mean $b + \sum_i \mu_i$ and variance $\sum_i a_i^2\sigma_i^2$. Hence $Y \sim N(b + \sum_i \mu_i, \sum_i a_i^2\sigma_i^2)$. □

Theorem 1.7 (Central Limit Theorem for IID RVs)**.** *Let $X_1, \ldots, X_n$ be iid random variables from a distribution with mean μ and variance $\sigma^2 < \infty$. Then for each fixed number x,*

$$\lim_{n\to\infty} \mathrm{P}\left(\frac{\bar{X}_n - \mu}{\sigma/\sqrt{n}} \le x\right) = \Phi(x), \tag{1.6}$$

where Φ denotes the cdf of the standard normal distribution.

We can interpret Equation 1.6 as follows: If we have a large random sample $X_1, \ldots, X_n$ from an arbitrary distribution (whether discrete or continuous), the random variable $(\bar{X}_n - \mu)/(\sigma/\sqrt{n})$ is distributed approximately as the standard normal. Consequently, $\bar{X}_n$ is distributed approximately as $N(\mu, \sigma^2/n)$ and $\sum_{i=1}^n X_i$ is distributed approximately as $N(n\mu, n\sigma^2)$.

1.4 Maximum Likelihood Estimation

Let the random variables $X_1, \ldots, X_n$ form a random sample from a distribution with pdf $f(x \mid \theta)$. Recall that this means that $X_i \overset{\text{iid}}{\sim} f(\cdot \mid \theta)$ for all $1 \le i \le n$. Let $f_n(\vec{x} \mid \theta)$ denote the value of the joint pdf of the random vector $(X_1, \ldots, X_n)'$ at the point $\vec{x} = (x_1, \ldots, x_n)'$. The *likelihood function* is the joint pdf of the observations of a random sample viewed as a function of θ for a given set of values of the sample. The maximum likelihood estimate of θ is that value of θ for which $f_n(\vec{x} \mid \theta)$ is maximized.

Example 1.3. Suppose that $X_1, \ldots, X_n$ form a random sample from a distribution with pdf $f(x \mid \theta)$ defined as follows:

$$f(x \mid \theta) = \begin{cases} \theta x^{\theta-1} & \text{for } 0 < x < 1 \\ 0 & \text{otherwise.} \end{cases}$$

Let us assume that $\theta > 0$. The joint distribution of the vector $(X_1, \ldots, X_n)'$ at the point $\vec{x} = (x_1, \ldots, x_n)'$ is

$$f_n(\vec{x} \mid \theta) = \prod_{i=1}^{n} \theta x_i^{\theta-1},$$

where we assume that $0 < x_i < 1$ for all i. Taking logs, we obtain that $\log f_n(\vec{x} \mid \theta) = n \log \theta + (\theta - 1) \sum_i \log x_i$. Take the derivative wrt θ and set to 0:

$$\frac{\partial \log f_n(\vec{x} \mid \theta)}{\partial \theta} = \frac{n}{\theta} + \sum_i \log x_i \overset{\text{set}}{=} 0$$

to obtain that $\theta = n / \sum_i \log \frac{1}{x_i}$. $\triangleright$

Example 1.4. Suppose that $X_1, \ldots, X_n$ form a random sample from a distribution with pdf $f(x \mid \theta)$ defined as follows:

$$f(x \mid \theta) = \frac{1}{2} e^{-|x-\theta|} \quad \text{for } -\infty < x < \infty.$$

Suppose that θ is unknown and that $-\infty < \theta < \infty$. In this case, the joint probability distribution is easily seen to be:

$$f_n(\vec{x} \mid \theta) = \frac{1}{2^n} e^{-\sum_i |x_i - \theta|}.$$

Take logs to obtain: $\log f_n(\vec{x} \mid \theta) = \log \frac{1}{2^n} - \sum_i |x_i - \theta|$. Maximizing f_n is equivalent to minimizing $\sum_i |x_i - \theta|$. This is equivalent to obtaining a point on the real line that minimizes the sum of the distances to the points $x_1, \ldots, x_n$. This happens when θ is the median of $x_1, \ldots, x_n$. $\triangleright$

Example 1.5. Suppose that $X_1, \ldots, X_n$ form a random sample from the uniform distribution on the interval $[\theta_1, \theta_2]$, where both θ_1 and θ_2 are unknown ($-\infty < \theta_1 < \theta_2 < \infty$). In this case, the log pdf of the joint distribution of $(X_1, \ldots, X_n)'$ is given by

$$\log f_n(\vec{x} \mid \theta_1, \theta_2) = \log \prod_{i=1}^{n} \frac{1}{\theta_2 - \theta_1} = -n \log(\theta_2 - \theta_1).$$

Maximizing the likelihood is equivalent to minimizing $\log(\theta_2 - \theta_1)$. The minimum possible value of θ_2 is $\max\{x_1, \ldots, x_n\}$ and the maximum possible value of θ_1 is $\min\{x_1, \ldots, x_n\}$. $\triangleright$

Example 1.6. Suppose that a certain large population contains k different types of individuals ($k \geq 2$), and let θ_i denote the proportion of people of type i, for $1 \leq i \leq k$. Here, $0 \leq \theta_i \leq 1$ and $\sum_{i=1}^{k} \theta_i = 1$. Suppose also that in a random sample of n individuals from this population there are exactly n_i individuals of type i so that $n = n_1 + \cdots + n_k$.

In this setting, for $1 \leq i \leq k$, define X_i to be the number of individuals of type i in a random sample of size n. Then the probability that $\bigwedge_{i=1}^{k} X_i = n_i$ is given by $\theta_1^{n_1} \cdots \theta_k^{n_k}$. The log pdf of the joint distribution is given by:

$$\log f_k((n_1, \ldots, n_k) \mid \theta_1, \ldots, \theta_k, n) = \sum_{i=1}^{k} n_i \log \theta_i.$$

Note that there are actually $k - 1$ variables here since we may write $\theta_k = 1 - \sum_{i=1}^{k-1} \theta_i$. Differentiating wrt θ_i for $1 \leq i \leq k - 1$, we obtain:

$$\frac{\partial \log f_k}{\partial \theta_i} = \frac{n_i}{\theta_i} - \frac{n_k}{\theta_k}.$$

Setting this to 0, we get that $\theta_i / \theta_k = n_i / n_k$. Sum this up from $1 \leq i \leq k - 1$, to obtain:

$$\frac{1 - \theta_k}{\theta_k} = \frac{n - n_k}{n_k},$$

which yields $\theta_k = n_k / n$. Substitute this in $\theta_i / \theta_k = n_i / n_k$ to obtain $\theta_i = n_i / n$. $\triangleright$

Example 1.7 (Nonexistence of an MLE)**.** An obvious disadvantage of the technique of maximum likelihood estimation is when the maximum does not exist. Consider again Example 1.5 where we let $\theta_1 = 0$ and $\theta = \theta_2$ The pdf of the uniform distribution is defined as:

$$f(x \mid \theta) = \begin{cases} \frac{1}{\theta} & 0 \leq x \leq \theta \\ 0 & \text{otherwise.} \end{cases}$$

Let us modify the definition of the pdf so that we use strict inequalities $0 < x < \theta$ above. Given a sample $x_1, \ldots, x_n$, the log pdf is $-n \log \theta$ as before and the MLE technique would require us to minimize $\log \theta$. However, in this case, there is no $\theta > \max\{x_1, \ldots, x_n\}$ that mimimizes $\log \theta$ and the MLE does not exist. ▷

Example 1.8 (Non-uniqueness of an MLE)**.** Consider a random sample $X_1, \ldots, X_n$ from the uniform distribution over the interval $[\theta, \theta + 1]$. In this case, the joint pdf is given by:

$$f_n(\vec{x} \mid \theta) = \begin{cases} 1 & \text{for } \theta \leq x_i \leq \theta + 1 \quad (1 \leq i \leq n) \\ 0 & \text{otherwise.} \end{cases}$$

In this case, the condition $\theta \leq x_i \leq \theta + 1$ for $1 \leq i \leq n$ may be written using the two conditions:

$$\theta \leq \min\{x_1, \ldots, x_n\} \text{ and } \max\{x_1, \ldots, x_n\} - 1 \leq \theta.$$

Any value of θ in the interval $[\max\{x_1, \ldots, x_n\} - 1, \min\{x_1, \ldots, x_n\}]$ is valid, but there is no unique value of θ. ▷

1.5 Bayesian Statistics and MCMC

This section is based on Chapters 12–15 from [8]. Bayes' rule gives us a recipe for calculating the posterior probability density.

$$\mathrm{P}(\Theta \mid \text{data}) = \frac{\mathrm{P}(\text{data} \mid \Theta) \cdot \mathrm{P}(\Theta)}{\mathrm{P}(\text{data})}. \tag{1.7}$$

Consider a case in which we have a sample of N data points $x_1, \ldots, x_N$. We assume that the likelihood is a Poisson distribution with mean λ and that the prior for λ is a log-normal$(1, 1)$ distribution. To calculate the probability of the data P(data), we must evaluate the integral:

$$\mathrm{P}(\text{data}) = \int_0^\infty \prod_{i=1}^{N} \frac{\lambda^{x_i} e^{-\lambda}}{x_i!} \cdot \frac{1}{\sqrt{2\pi}\lambda} e^{-\frac{1}{2}(\log \lambda - 1)^2} d\lambda. \tag{1.8}$$

While this integral is not too difficult, it explains the problem of calculating posteriors analytically. As the number of parameters (the length of Θ) increases, calculating the probability of the data requires evaluating integrals in higher dimensional spaces. This is why we use alternative methods to derive approximate versions of the posterior.

In certain special cases, the posterior distribution can be easily derived. This occurs, for example, when the prior distribution is from a so-called *conjugate prior* family. A conjugate prior family is a set of distributions defined wrt a likelihood function. If one chooses the prior to be from this family, then the posterior is guaranteed to be from the same family. In such cases, one can dispense with computing integrals to find out the posterior.

Example 1.9 (Beta-Binomial)**.** Let $X_1, \dots, X_n \overset{\text{iid}}{\sim} \text{Bernoulli}(\theta)$ and suppose that $\theta \sim \text{Beta}(\alpha, \beta)$. Suppose that one observes $X_i = x_i$ for $1 \le i \le n$. Then the posterior distribution of θ is given by:

$$f(\theta \mid X_1, \dots, X_n) = \frac{f(X_1, \dots, X_n \mid \theta) \cdot f(\theta)}{f(X_1, \dots, X_n)}.$$

In the above expression, $f(X_1, \dots, X_n)$ is a constant for a given set of values of the random variables X_i. The likelihood, when $X_i = x_i$ for $1 \le i \le n$, is given by:

$$\begin{aligned} f(x_1, \dots, x_n \mid \theta) &= \prod_{i=1}^{n} \theta^{x_i} (1-\theta)^{1-x_i} \\ &= \theta^{\sum_i x_i} \cdot (1-\theta)^{n - \sum_i x_i}. \end{aligned}$$

Now we may write:

$$\begin{aligned} f(\theta \mid X_1, \dots, X_n) &= \frac{f(X_1, \dots, X_n \mid \theta) \cdot f(\theta)}{f(X_1, \dots, X_n)} \\ &\propto \theta^{\sum_i x_i} \cdot (1-\theta)^{n - \sum_i x_i} \cdot \theta^{\alpha - 1} \cdot (1-\theta)^{\beta - 1} \\ &\propto \theta^{\alpha + \sum_i x_i - 1} \cdot (1-\theta)^{\beta + n - \sum_i x_i - 1}. \end{aligned}$$

The last expression is the kernel of the beta distribution with parameters $\alpha + \sum_i x_i$ and $\beta + n - \sum_i x_i$. Since the expression on the right must be a valid probability distribution, the proportionality constant must be that of the beta distribution with these parameters, that is,

$$\frac{\Gamma(\alpha + \beta + n)}{\Gamma(\alpha + \sum_i x_i) \cdot \Gamma(\beta + n - \sum_i x_i)}.$$

$\triangleright$

Thus if the prior is a beta distribution and the likelihood is a bernoulli distribution, then the posterior is a beta distribution also. Thus the set of beta distributions is a conjugate family wrt the bernoulli likelihood.

Chapter 2

Timeseries

The material in the section follows [2]. A *time series* is a sequence of random variables $\{X_1, X_2, \ldots\}$. A complete model of such a time series would require specifying the joint probability distributions of all random vectors $(X_1, \ldots, X_n)'$ for all $n \geq 1$. In practice, this might be impossible to do unless the time series is generated by some simple well-understood mechanism. Instead one typically specifies the first- and second-order moments of the joint distibutions, that is, $\mathrm{E}[X_t]$ and $\mathrm{E}[X_t X_{t+h}]$ for all $t \geq 1$ and for all $h \geq 0$.

The next important concepts are those of stationarity and the auto-correlation function. Roughly speaking, a time series $\{X_t\}_{t=-\infty}^{\infty}$ is stationary if its "statistical properties" are similar to those of the time-shifted series $\{X_{t+h}\}_{t=-\infty}^{\infty}$ for every integer "lag" h. By statistical properties, we mean the first- and second-order moments of $\{X_t\}$.

Formally, a time series $\{X_t\}$ is weakly stationary if

1. $\mathrm{E}[X_t]$ is independent of t

2. $\mathrm{Cov}(X_t, X_{t+h})$ is independent of t for every fixed lag h.

In contrast, a time series $\{X_t\}$ is strictly stationary if the random vectors $(X_{t_1}, \ldots, X_{t_n})'$ and $(X_{t_1+h}, \ldots, X_{t_n+h})'$ have the same joint distributions for all sets of indices $\{t_1, \ldots, t_n\}$, for all $h \geq 0$ and all $n \geq 1$. This is written as:

$$(X_{t_1}, \ldots, X_{t_n})' \overset{d}{=} (X_{t_1+h}, \ldots, X_{t_n+h})'$$

Strict stationarity implies the following:

1. The random variables X_t are identically distributed.

2. Pairs of random variables (X_t, X_{t+h}) have the same distribution as (X_1, X_{1+h}) (set $n = 2$).

3. Strict stationarity implies weak stationarity: $\mathrm{E}[X_t] = \mathrm{E}[X_1]$ and $\mathrm{Cov}(X_t, X_{t+h}) = \mathrm{Cov}(X_1, X_{1+h})$ for all $t \geq 1$ and all $h \geq 0$. Both terms are independent of t.

4. Weak stationarity does *not* imply strong stationarity. We show this by an example. Let $Z_i \overset{\text{iid}}{\sim} N(0,1)$ for all i. Define X_t as:

   $$X_t = \begin{cases} Z_t & \text{if } t \text{ is even} \\ 2Z_t & \text{if } t \text{ is odd.} \end{cases}$$

   Then $\mathrm{E}[X_t] = 0$ for all t. Also,

   $$\mathrm{Cov}(X_t, X_{t+h}) = \begin{cases} 0 & \text{if } h > 0 \\ 1 & \text{if } h = 0. \end{cases}$$

This follows from the assumed independence of X_t and X_{t+h} when $h > 0$. However, X_ts do not have the same distribution, a requirement of strict stationarity.

The autocovariance function of a stationary time series $\{X_t\}$ at lag h is defined as $\mathrm{Cov}(X_{t+h}, X_t) = \mathrm{Cov}(X_h, X_0)$. The autocorrelation of $\{X_t\}$ at lag h is defined as

$$\frac{\mathrm{Cov}(X_{t+h}, X_t)}{\mathrm{Var}(X_t)} = \frac{\mathrm{Cov}(X_h, X_0)}{\mathrm{Var}(X_0)}.$$

Example 2.1 (iid Noise)**.** Suppose $\{X_t\}$ is iid noise with zero mean and $\mathrm{E}[X_t^2] = \sigma^2 < \infty$. Then $\mathrm{E}[X_t] = 0$ for all t and for every fixed lag h and all t:

$$\mathrm{Cov}(X_t, X_{t+h}) = \begin{cases} 0 & \text{if } h > 0 \\ \sigma^2 & \text{if } h = 0. \end{cases}$$

Thus iid noise is stationary. We denote such a series as $\{X_t\} \sim \mathrm{IID}(0, \sigma^2)$.

Example 2.2 (White Noise)**.** A sequence $\{X_t\}$ of *uncorrelated* random variables with zero mean and finite second moment σ^2 is called *white noise*. Clearly, the covariance function at lag h is the same as that of iid noise and, as such, white noise is stationary. We denote such a series as $\{X_t\} \sim \mathrm{WN}(0, \sigma^2)$. Unlike iid noise, the components of white noise need not be independent (recall that independence implies zero correlation but not the other way around). In particular, every $\mathrm{IID}(0, \sigma^2)$ sequence is a $\mathrm{WN}(0, \sigma^2)$ sequence but not the other way around.

Example 2.3 (Random Walk). A *random walk* $\{S_t\}$ is a sequence obtained by cumulatively summing iid random variables. A random walk with zero mean is obtained by defining $S_0 = 0$ and

$$S_t = X_1 + \cdots + X_t$$

for all $t > 0$, where $\{X_t\} \sim \mathrm{IID}(0, \sigma^2)$. In this case, $\mathrm{E}[S_t] = 0$ and $\mathrm{E}[S_t^2] = t\sigma^2 < \infty$ for all t. For all lags $h \geq 0$ and all t,

$$\begin{aligned} \mathrm{Cov}(S_t, S_{t+h}) &= \mathrm{Cov}(S_t, S_t + X_{t+1} + \cdots + X_{t+h}) \\ &= \mathrm{Cov}(S_t, S_t) + \mathrm{Cov}(S_t, X_{t+1}) + \cdots + \mathrm{Cov}(S_t, X_{t+h}) \\ &= t\sigma^2. \end{aligned}$$

The last equality follows since $\mathrm{Cov}(S_t, X_{t+i}) = 0$ for all $i \geq 1$. Hence $\mathrm{Cov}(S_t, S_{t+h})$ depends on t and $\{S_t\}$ is not stationary.

Example 2.4 (First-Order Moving Average)**.** Consider the series defined by

$$X_t = Z_t + \theta Z_{t-1}, \quad t = 0, \pm 1, \pm 2, \ldots, \tag{2.1}$$

where $Z_t \sim \mathrm{WN}(0, \sigma^2)$ and θ is a real-valued constant. Now $\mathrm{E}[X_t] = 0$ and

$$\begin{aligned} \mathrm{E}[X_t^2] &= \mathrm{E}[(Z_t + \theta Z_{t-1})^2] \\ &= \mathrm{E}[Z_t^2 + 2\theta Z_t Z_{t-1} + \theta^2 Z_{t-1}^2] \\ &= \sigma^2 + 2\theta\, \mathrm{E}[Z_t Z_{t-1}] + \theta^2 \sigma^2 \\ &= (1 + \theta^2)\sigma^2. \end{aligned}$$

The last equality follows since, Z_t and Z_{t-1} being uncorrelated, satisfy $\mathrm{Cov}(Z_t, Z_{t-1}) = 0$. Recall that $\mathrm{Cov}(Z_t, Z_{t-1}) = \mathrm{E}[Z_t Z_{t-1}] - \mathrm{E}[Z_t]\, \mathrm{E}[Z_{t-1}]$ and that being uncorrelated is sufficient for the expectation of the product of two random variables to be equal to the product of their expectations. One can easily verify that for all t

$$\operatorname{Cov}(X_t, X_{t+h}) = \begin{cases} (1+\theta^2)\sigma^2 & \text{if } h = 0 \\ \theta\sigma^2 & \text{if } h = \pm 1 \\ 0 & \text{if } |h| \geq 2. \end{cases} \tag{2.2}$$

Thus the conditions of weak stationarity hold and the sequence $\{X_t\}$ is weakly stationary. The autocorrelation function is given by:

$$\rho(h) = \frac{\operatorname{Cov}(X_0, X_h)}{\operatorname{Var}(X_0)} = \begin{cases} 1 & \text{if } h = 0 \\ \frac{\theta}{1+\theta^2} & \text{if } h = \pm 1 \\ 0 & \text{if } |h| \geq 2. \end{cases} \tag{2.3}$$

Example 2.5 (First-Order Autoregression)**.** Let $\{X_t\}$ be a stationary series satisfying the equation:

$$X_t = \phi X_{t-1} + Z_t \quad t = 0, \pm 1, \pm 2, \ldots, \tag{2.4}$$

where $\{Z_t\} \sim \mathrm{WN}(0, \sigma^2)$, $|\phi| < 1$ and Z_t is uncorrelated with X_s for all $s < t$. As can be seen, $\mathrm{E}[X_t] = 0$ and for $h > 0$:

$$\begin{aligned} \operatorname{Cov}(X_{t+h}, X_t) &= \operatorname{Cov}(\phi X_{t+h-1} + Z_{t+h}, X_t) \\ &= \phi \operatorname{Cov}(X_{t+h-1}, X_t) + \operatorname{Cov}(Z_{t+h}, X_t) \\ &= \phi \operatorname{Cov}(X_{t+h-1}, X_t). \end{aligned}$$

From this, one can show that $\operatorname{Cov}(X_{t+h}, X_t) = \phi^h \operatorname{Cov}(X_t, X_t)$. By assumption, $\{X_t\}$ is stationary and hence $\operatorname{Cov}(X_{t+h}, X_t) = \phi^h \operatorname{Cov}(X_0, X_0)$ Next suppose that $h < 0$. Let $s = t + h = t - |h|$ so that $t = s + |h|$. Then $\operatorname{Cov}(X_{t+h}, X_t) = \operatorname{Cov}(X_s, X_{s+|h|}) = \phi^{|h|} \operatorname{Cov}(X_s, X_s) = \phi^{|h|} \operatorname{Cov}(X_0, X_0)$. The autocorrelation function at lag h is

$$\rho(h) = \frac{\phi^{|h|} \operatorname{Cov}(X_0, X_0)}{\operatorname{Cov}(X_0, X_0)} = \phi^{|h|}.$$

One can also obtain a closed-form expression for $\operatorname{Cov}(X_0, X_0)$. Since $\operatorname{Cov}(X_0, X_0) = \operatorname{Cov}(X_t, X_t)$ and

$$\operatorname{Cov}(X_t, X_t) = \operatorname{Cov}(\phi X_{t-1} + Z_t, \phi X_{t-1} + Z_t) = \phi^2 \operatorname{Cov}(X_0, X_0) + \operatorname{Cov}(Z_t, Z_t),$$

we obtain that $\operatorname{Cov}(X_0, X_0) = \sigma^2/(1 - \phi^2)$.

Chapter 3

Trees, Bagging and Random Forests

Random forests are built using decision trees. Decision trees are easy to build, easy to use and interpret but they are prone to overfitting. Overfitting can be minimized using techniques such as *cost complexity pruning*. Nevetheless this tendency to overfit causes decision trees to have *high variance*. This means that if we fit decision trees to different data sets from the same underlying distribution, we are likely see very different trees. In particular, if we randomly split a given dataset into train and test and fit a tree to each, the results will likely be quite different. This tendency to have a high variance can be reduced by using a general procedure known as *bootstrap aggregation* or *bagging*.

The general principle behind bagging is easy enough to see. Let $X_1, \dots, X_n$ be iid random variables from a distribution with mean μ and variance σ^2. If we use the sample mean $\bar{X}_n$ as an estimate of μ, the variance is σ^2/n. As n increases, the variance decreases. Therefore a natural way to reduce the variance of a predictor is to use several data sets from the population, fit a predictor to each of these data sets and then take the average of these predictions. Of course, in practice, we do not have access to multiple data sets. So what is done is that several random samples are selected from a single training data set. This procedure is called bootstrapping. Given a training data set with n data points, one creates a bootstrapped data set by sampling n times from the training data *with replacement*. One generates multiple bootstrapped data sets and then trains decision trees on each of these data sets. Let's assume that there are M such bootstrapped data sets and that $\hat{f}_i$ is the decision tree obtained by training on the ith set. In case of regression problems, the predicted response given a data point $\mathbf{x}$ is:

$$\hat{f}_{\text{bag}}(\mathbf{x}) = \frac{1}{M} \sum_{i=1}^{M} \hat{f}_i(\mathbf{x}).$$

If the problem is a classiffication problem, the predicted class is the one that is predicted most often by the trees.

Using bootstrapped samples to fit decision trees has one clear benefit in that estimating the test error of the bagged model becomes very easy. In particular, one does not need to use cross-validation. This follows from the observation that each bootstrapped data set omits approximately 1/3 of the data points from the training set. The data points that are left out of a given bootstrapped sample are called the *out-of-bag* samples. In order to estimate the test error of the tree fit to this bootstrapped sample, one simply needs to find out the error on the out-of-bag samples. To see why, on average, 1/3 of the data points are left out, consider a training data set $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ with n elements. The probability that the ith datum is *not* selected in our bootstrapped sample S is

$$\mathrm{P}((\mathbf{x}_i, y_i) \notin S) = \left(1 - \frac{1}{n}\right)^n \stackrel{\text{def}}{=} p_i.$$

define $Z_i = 1$ if $(\mathbf{x}_i, y_i) \in S$ and otherwise 0. Then $\sum_i Z_i$ is the number of elements of the training set that are not in S. The expectation of $\sum_i Z_i$ is given by:

$$\mathrm{E}\left[\sum_i Z_i\right] = \sum_i (p_i \cdot 1 + (1 - p_i) \cdot 0) = n \cdot \left(1 - \frac{1}{n}\right)^n \approx n \cdot e^{-1} = 0.37n.$$

All of this is good but training decision trees on bootstrapped data sets still produce predictors that are correlated. This situation corresponds to having random variables $X_1, \ldots, X_n$ that are identically distributed with a positive pairwise correlation ρ. The variance of the sample mean in this situation is:

$$\begin{aligned}
\mathrm{Var}(\bar{X}_n) &= \mathrm{Var}\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) \\
&= \frac{1}{n^2}\left(\sum_{i=1}^{n} \mathrm{Var}(X_i) + 2 \sum_{1 \le i < j \le n} \mathrm{Cov}(X_i, X_j)\right) \\
&= \frac{1}{n^2}\left(n\sigma^2 + 2 \cdot \binom{n}{2}\rho\sigma^2\right) \\
&= \frac{\sigma^2}{n} + \frac{n-1}{n}\rho\sigma^2 \\
&= \rho\sigma^2 + (1 - \rho)\frac{\sigma^2}{n}.
\end{aligned}$$

As n increases, the second term vanishes, but the first remains and the correlation limits the advantages of averaging.

Random forests try to reduce this correlation by making use of a small tweak when splitting nodes of the trees (Figure 3.1). As in bagging, one builds a number of decision trees on bootstrapped data sets. The difference is that each time a split in a tree is considered, a *random subset* of k predictors are used from the total set of p predictors. The split is then based on only one of these k predictors that were selected. A fresh sample of k predictors is taken for each split in each tree. The initial value of k is $\sqrt{p}$ and finally decided using cross validation.

Input. A data set $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbf{R}^p$; a minimum node size n_{min}; the number of trees in the forest M; the number k of predictors to pick at each split.

Step 1. For $i = 1$ to M:

1. Draw a bootstrap sample of size n from the training data set.
2. Construct a decision tree T_i to the bootstrapped data by recursively repeating the steps for each leaf node, till the minimum node size n_{min} is reached.
    (a) Select k predictors randomly from among the set of p predictors.
    (b) Pick the best predictor to make the split.
    (c) Split the node into two child nodes.

Step 2. Output the ensemble of trees $\{T_i\}_{i=1}^M$

To make a prediction at a new point $\mathbf{x}$:

Regression. $\hat{f}(\mathbf{x}) = \frac{1}{M} \sum_{i=1}^M T_i(\mathbf{x})$.

Classification. Let $\hat{C}_i(\mathbf{x})$ be the class prediction of the ith tree. Then $\hat{C}(\mathbf{x}) = \text{majority vote } \{\hat{C}_i(\mathbf{x})\}$.

Figure 3.1: The Random Forest Algorithm.

Chapter 4

Gradient Boosted Regression

Gradient boosting is a technique where a sequence $F_0, \ldots, F_M$ of decision trees are constructed where each tree F_m in the sequence is fit to the errors of the predictor obtained from the trees that precede it. The predictor obtained from a sequence of trees $F_0, \ldots, F_{m-1}$ takes an additive form and given an input $\mathbf{x}$, the predicted output is:

$$F_0(\mathbf{x}) + \nu \cdot \sum_{j=1}^{m-1} F_j(\mathbf{x}), \tag{4.1}$$

where ν is the learning rate. This is different from ensemble methods such as random forests in that, in the latter, the prediction is the mean predicted value over all the learners in the ensemble.

This technique can be used for both regression and classification. We first look at regression as the presentation is slightly easier.

4.1 Gradient Boosting for Regression

Consider a regression problem where given $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbf{R}^p$ and $y_i \in \mathbf{R}$, we wish to find out $F\colon \mathbf{R}^p \to \mathbf{R}$ from an appropriate hypothesis class such that $F(\mathbf{x}_i) \approx y_i$ minimizing some loss function. The loss function typically used in regression is squared loss and for a single data point is defined as:

$$L(y_i, F(\mathbf{x}_i)) = \frac{1}{2}(y_i - F(\mathbf{x}_i))^2. \tag{4.2}$$

The loss for the entire data set is:

$$L(F) = \frac{1}{2}\sum_{i=1}^{n}(y_i - F(\mathbf{x}_i))^2. \tag{4.3}$$

Gradient boosting requires loss functions that are differentiable and squared loss is one such function. Although taking derivates wrt F in Equation 4.3 is trivial, it does provide a useful insight.

$$\frac{\mathrm{d}L(F)}{\mathrm{d}F} = -\sum_{i=1}^{n}(y_i - F(\mathbf{x}_i)) \tag{4.4}$$

The derivative is the *negative* of the sum of the residuals. This fact will be important as we work through the algorithm.

The algorithm itself is presented in Figure 4.1. Step 1 of the algorithm asks us to initialize the model with a constant value F_0 to be computed using:

$$F_0 = \text{argmin}_\gamma \sum_{i=1}^{n}(y_i - \gamma)^2 = \text{argmin}_\gamma \sum_{i=1}^{n}(y_i^2 - 2y_i\gamma + \gamma^2) \overset{\text{def}}{=} \text{argmin}_\gamma\, g(\gamma) \tag{4.5}$$

Differentiating wrt γ and then setting the resulting expression to 0 yields:

$$\frac{\mathrm{d}g}{\mathrm{d}\gamma} = \sum_{i=1}^{n}(-2y_i + 2\gamma) \overset{\text{set}}{=} 0 \Rightarrow \gamma = \frac{\sum_{i=1}^{n} y_i}{n} = \bar{y}_n.$$

Thus the inital prediction is simply the mean of the response. This seems like the intuitive thing to do.

Step 2 is where the sequence of trees is constructed. In each iteration, the algorithm constructs a new tree based on the residuals of the previous predictor. There are M iterations in total resulting in M trees. In practice, this is a parameter to algorithm and is determined using techniques such as cross-validation. The first step in this sequence is to compute the "pseudo residuals" r_{im} for each data point using Equation 4.8. Since our loss function is the squared loss, we see that r_{im} evaluates to:

$$r_{im} = y_i - F_{m-1}(\mathbf{x}_i).$$

This is exactly the residual as defined in linear regression. In the context of gradient boosting, the term pseudo residual is used to remind us that we are not working with linear regression.

The second step in the sequence for Step 2 is the construction of a regression tree to fit the residuals r_{im} that were computed. Assume that the mth tree has leaf nodes R_{jm}, where $1 \leq j \leq J_m$. The next step asks us to compute the values of these leaf nodes. Differentiating wrt γ in Equation 4.9 and setting to 0, we obtain that

$$\gamma_{jm} = \frac{1}{|\{x_i \in R_{jm}\}|} \sum_{x_i \in R_{jm}} (y_i - F_{m-1}(x_i)). \tag{4.6}$$

Again this has an intuitive interpretation. The value of the jth leaf in the mth tree is the mean residual value of all the data points that trickle down to that leaf node. The final step of Step 2 is to update the mth predictor using the most recently constructed tree.

Finally, Step 3 simply returns the updated predictor from the very last iteration in Step 2. To make a prediction at a new point $\mathbf{x}$, we output:

$$F_0(\mathbf{x}) + \nu \cdot F_1(\mathbf{x}) + \cdots + \nu \cdot F_M(\mathbf{x}).$$

4.2 Gradient Boosting for Classification

We next consider gradient boosting for classification problems. Suppose that we are given a dataset $\{(\mathbf{x}_i, y_i)_{i=1}^n\}$, where $\mathbf{x}_i \in \mathbf{R}^p$ and $y_i \in \{0, 1\}$. The problem is to find a mapping $f\colon \mathbf{x} \to y$. As in logistic regression, we modify the problem slightly and do not work directly with the class labels y_i. Instead, we consider the log(odds) of the event $\mathrm{P}\{y = 1 \mid \mathbf{x}\}$. Since $-\infty < \log(\text{odds}) < +\infty$, this restatement allows us to focus on functions $F\colon \mathbf{R}^p \to \mathbf{R}$ rather than from $\mathbf{R}^p \to \{0, 1\}$. This is helpful because gradient boosting requires differentiable loss functions.

Our next step is to build an appropriate loss function. Let $p = \mathrm{P}\{y = 1 \mid \mathbf{x}\}$. Then we may write

$$\mathrm{P}\{y \mid \mathbf{x}\} = p^y \cdot (1-p)^{1-y}. \tag{4.10}$$

Consequently, the likelihood of $y_1, \ldots, y_n$ given $\mathbf{x}_1, \ldots, \mathbf{x}_n$ assuming that the data $\{(\mathbf{x}_i, y_i)_{i=1}^n\}$ are independent is

$$\mathrm{P}\{y_1, \ldots, y_n \mid \mathbf{x}_1, \ldots, \mathbf{x}_n\} = \prod_{i=1}^{n} p_i^{y_i} \cdot (1-p_i)^{1-y_i}. \tag{4.11}$$

The log-likelihood is $\sum_i [y_i \log p_i + (1-y_i)\log(1-p_i)]$. If we were to fit a model such a logistic regression, we would search for those model parameters for which the log-likelihood is a maximum. If we were to

Input. A dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, a differentiable loss function $L(y, F(\mathbf{x}))$ and the number of trees to be constructed M

Step 1. Initialize model with a constant value $F_0(\mathbf{x})$ such that for all $\mathbf{x}_j$, $1 \leq j \leq n$,

$$F_0(\mathbf{x}_j) = \text{argmin}_\gamma \sum_{i=1}^{n} L(y_i, \gamma). \tag{4.7}$$

Step 2 For $m = 1$ to M:

1. For each $i \in [1, \ldots, n]$, compute the pseudo residuals:

$$r_{im} = -\left[\frac{\partial L(y, F(\mathbf{x}))}{\partial F(\mathbf{x})}\right]_{F(\mathbf{x})=F_{m-1}(\mathbf{x}),(\mathbf{x}_i,y_i)}. \tag{4.8}$$

2. For each $i \in [1, \ldots, n]$, fit a regression tree to the r_{im} values and create terminal regions R_{jm}, where $j = 1, \ldots, J_m$, where J_m is the number of leaves in the mth tree.
3. For each $j \in [1, \ldots, J_m]$, compute an output value for leaf j in tree m:

$$\gamma_{jm} = \text{argmin}_\gamma \sum_{\mathbf{x}_i \in R_{jm}} L(y_i, F_{m-1}(\mathbf{x}_i) + \gamma). \tag{4.9}$$

4. Update $F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \nu \cdot \sum_{j=1}^{J_m} \gamma_{jm} I(\mathbf{x} \in R_{jm})$.

Step 3 Return F_M.

Figure 4.1: The Gradient Boosted Algorithm.

use a transform of the log-likelihood as a loss function, we would want to minimize that transform. The easiest transform is the negative log-likelihood. Minimizing the negative log-likelihood is equivalent to maximizing the log-likelihood (which is what we want to do). This negative log-likelihood will then be our loss function and this also goes by the name of *cross entropy.*

We re-write the loss function in terms of log(odds) because this is what our gradient boosting model will output. To simplify the resulting expressions, we consider only one data point and omit the data index i. We may then write:

$$\begin{aligned} -\left[y \log p + (1-y)\log(1-p)\right] &= -y \log p - (1-y)\log(1-p) \\ &= -y \log p + y \log(1-p) - \log(1-p) \\ &= -y \log \frac{p}{1-p} - \log\left(1 - \sigma\left(\log \frac{p}{1-p}\right)\right). \end{aligned}$$

In the last step, we used the fact that $p = \sigma(p/(1-p))$, where σ is the sigmoid function. This last step can be simplified by expanding out the sigmoid function and the loss function can then be written as:

$$-y \log \frac{p}{1-p} + \log(1 + e^{\log \frac{p}{1-p}}). \tag{4.12}$$

Note the positive sign before the logarithm.

Finally, we note that the gradient boosting engine gives us a function $F(\mathbf{x})$ that represents the log(odds). With this, we can write the loss function more clearly as:

$$L(y, F(\mathbf{x})) = -yF(\mathbf{x}) + \log(1 + e^{F(\mathbf{x})}). \tag{4.13}$$

This function is differentiable wrt $F(\mathbf{x})$ and with some manipulation, one can show that:

$$\frac{\mathrm{d}L}{\mathrm{d}F(\mathbf{x})} = -y + \sigma(F(\mathbf{x})). \tag{4.14}$$

Now this has a nice interpretation. The term $\sigma(F(\mathbf{x}))$ is the predicted probability that $y = 1$. If we were to interpret the label $y \in \{0, 1\}$ as a probability, then the derivative of the loss function wrt $F(\mathbf{x})$ is the negative of the difference of the actual probability and the predicted probability.

We now go through the gradient boosting algorithm in Figure 4.1 step by step, this time for a binary classification problem. Step 1 asks us to initialize the model with a constant value γ which is the solution to $\mathrm{argmin}_\gamma \sum_{i=1}^n L(y_i, \gamma)$. Now,

$$\sum_{i=1}^n L(y_i, \gamma) = \sum_{i=1}^n (-y_i\gamma + \log(1 + e^\gamma)) = -\gamma \sum_{i=1}^n y_i + n \cdot \log(1 + e^\gamma) \overset{\mathrm{def}}{=} g(\gamma). \tag{4.15}$$

Differentiating this wrt γ, we obtain:

$$\frac{\mathrm{d}g}{\mathrm{d}\gamma} = -\sum_{i=1}^n y_i + n \cdot \frac{e^\gamma}{1 + e^\gamma} = -\sum_{i=1}^n y_i + n \cdot \sigma(\gamma). \tag{4.16}$$

Setting the right hand expression above to 0, we obtain:

$$\gamma = \log \frac{\bar{y}_n}{1 - \bar{y}_n}. \tag{4.17}$$

Thus the initial constant solution is the log(odds) of the mean $\mathrm{P}\{y = 1 \mid \mathbf{x}\}$ in the data. Intuitively, this seems like a good initial solution to start out with.

Step 2 is where all the trees are constructed. There are M trees in total and this number has to be decided beforehand. In practice, this is a parameter to algorithm and is determined using techniques such as cross-validation. The first step in this sequence asks us to compute "residuals" r_{im} for each data point $(\mathbf{x}_i, y_i)$ and each tree. Computing the residuals amounts to computing:

$$r_{im} = -\left[\frac{\partial L(y, F(\mathbf{x}))}{\partial F(\mathbf{x})}\right]_{F(\mathbf{x}) = F_{m-1}(\mathbf{x}), (\mathbf{x}_i, y_i)} = y_i - \sigma(F_{m-1}(\mathbf{x}_i)) \tag{4.18}$$

Note that $F_{m-1}(\mathbf{x})$ gives the log(odds) of the event that $y = 1$ given the data $\mathbf{x}$. Hence $\sigma(F_{m-1}(\mathbf{x}))$ represents $\mathrm{P}\{y = 1 \mid \mathbf{x}\}$. Consequently, r_{im} is the difference between the observed probability and the predicted probability. This looks very much like the residuals as defined in linear regression. Thus the name "pseudo residuals."

In the next step in the sequence for Step 2 is the construction of a regression tree to fit these pseudo residual values. Suppose that the mth tree has J_m leaves. The third step determines an appropriate output value of each leaf of the tree just constructed. The output value for the jth leaf of this tree is:

$$\gamma_{jm} = \mathrm{argmin}_\gamma \sum_{\mathbf{x}_i \in R_{jm}} L(y_i, F_{m-1}(\mathbf{x}_i) + \gamma). \tag{4.19}$$

This asks us to find that value of γ such that, when added to the log(odds) value of the previous prediction, the sum of the losses is minimized.

We could differentiate the above expression wrt γ but this is potentially very messy. Instead, we simplify the loss function by using a second-order Taylor expansion. In this context, recall that if $f\colon \mathbf{R} \to \mathbf{R}$ is a function that is infinitely differentiable at a point x, then

$$f(x + h) \approx f(x) + \frac{f^{(1)}(x)}{1!} \cdot h + \frac{f^{(2)}(x)}{2!} \cdot h^2 + \frac{f^{(3)}(x)}{3!} \cdot h^3 + \cdots . \tag{4.20}$$

Using this, we can write a second-order approximation to our loss function:

$$\begin{aligned} L(y_i, F_{m-1}(\mathbf{x}_i) + \gamma) &\approx L(y_i, F_{m-1}(\mathbf{x}_i)) + \gamma \cdot \frac{\mathrm{d}L(y_i, F_{m-1}(\mathbf{x}_i))}{\mathrm{d}F_{m-1}} + \frac{\gamma^2}{2} \cdot \frac{\mathrm{d}^2 L(y_i, F_{m-1}(\mathbf{x}_i))}{\mathrm{d}F_{m-1}^2} \\ &= L(y_i, F_{m-1}(\mathbf{x}_i)) + \gamma \cdot (-y_i + \sigma(F_{m-1}(\mathbf{x}_i)) + \frac{\gamma^2}{2} \cdot \sigma(F_{m-1}(\mathbf{x}_i))(1 - \sigma(F_{m-1}(\mathbf{x}_i))). \end{aligned}$$

Differentiate the right-hand side wrt γ to obtain:

$$-y_i + \sigma(F_{m-1}(\mathbf{x}_i) + \gamma \cdot \sigma(F_{m-1}(\mathbf{x}_i))(1 - \sigma(F_{m-1}(\mathbf{x}_i))). \tag{4.21}$$

Set this to 0 and solve for γ:

$$\gamma = \frac{y_i - \sigma(F_{m-1}(\mathbf{x}_i)}{\sigma(F_{m-1}(\mathbf{x}_i))(1 - \sigma(F_{m-1}(\mathbf{x}_i)))} = \frac{y_i - p_i}{p_i(1 - p_i)}. \tag{4.22}$$

This is the expression for just a single data point. Taking into account all data points, we obtain:

$$\gamma = \frac{\sum_{i=1}^n (y_i - \sigma(F_{m-1}(\mathbf{x}_i))}{\sum_{i=1}^n \sigma(F_{m-1}(\mathbf{x}_i))(1 - \sigma(F_{m-1}(\mathbf{x}_i)))} = \frac{\sum_i (y_i - p_i)}{\sum_i p_i(1 - p_i)}. \tag{4.23}$$

We can now evaluate the value γ_{jm} of each leaf node of the mth tree. The fourth and final step of Step 2 is to update the prediction function F_{m-1} using the values of the leaf nodes of the tree just constructed.

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \nu \cdot \sum_{j=1}^{J_m} \gamma_{jm} I(\mathbf{x} \in R_{jm}).$$

Step 3 simply outputs the function F_M obtained after the updation in the Mth round. To make a prediction at a new point $\mathbf{x}$, we first compute the log(odds):

$$F_0(\mathbf{x}) + \nu \cdot F_1(\mathbf{x}) + \cdots + \nu \cdot F_M(\mathbf{x}).$$

Using this, we compute the probability $\sigma(\log(\text{odds}))$ that the corresponding label is a 1. If this probability exceeds a threshold, which is usually set to 0.5, then we output a 1; otherwise, we output a 0.

Chapter 5

Neural Networks

Bibliography

[1] Joseph K. Blitzstein, Jessica Hwang. *Introduction to Probability*, Second Edition, Chapman and Hall, 2019.

[2] Peter J. Brockwell, Richard A. Davis. *Introduction to Time Series and Forecasting*, Third Edition, Springer, 2016.

[3] George Casella, Roger L. Berger. *Statistical Inference*, Second Edition, Duxbury Advanced Series, 2001.

[4] Morris DeGroot, Mark J. Schervish. *Probability and Statistics*, Fourth Edition, Pearson, 2012.

[5] Trevor Hastie, Robert Tibshirani, Jerome Friedman. *The Elements of Statistical Learning*, Second Edition, Springer, 2013.

[6] Rob J. Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*, Second Edition. Online book at: `https://otexts.com/fpp2/`

[7] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. *an Introduction to Statistical Learning*, Springer, 2014.

[8] Ben Lambert. *A Student's Guide to Bayesian Statistics*, Sage Publications, 2018.