Probability and Statistics

Adam Boult (www.bou.lt)

July 9, 2020

Contents

Preface

This is a live document, and is full of gaps, mistakes, typos etc.

Part I

Probability

Chapter 1

Events, the probability function and the Kolgomorov axioms

1.1 Events

1.1.1 Elementary events

We have a sample space, Ω consisting of elementary events.

All elementary events are disjoint sets.

1.1.2 Non-elementary events

We have a σ-algebra over Ω called F. A σ-algebra takes a set a provides another set containing subsets closed under complement. The power set is an example.

All events E are subsets of Ω

$\forall E \in FE \subseteq \Omega$

1.1.3 Mutually exclusive events

Events are mutually exclusive if they are disjoint sets.

1.1.4 Complements

For each event E, there is a complementary event E^C such that:

$E \vee E^C = \Omega$

$E \wedge E^C = \varnothing$

This exists by construction in the measure space.

1.1.5 Union and intersection

As events are sets, we can define algebra on sets. For example for two events E_i and E_j we can define:

- $E_i \wedge E_j$
- $E_i \vee E_j$

1.2 Kolmogorov axioms

1.2.1 The probability function

For all events E in F, the probability function P is defined.

1.2.2 Measure space

This gives us the following measure space:

(Ω, F, P)

1.2.3 First Kolmogorov axiom

First axiom

The probability of all events is a non-negative real number.

$\forall E \in F[(P(E) \geq 0) \wedge (P(E) \in \mathbb{R})]$

1.2.4 Second Kolmogorov axiom

The probability of one of the elementary events occuring is 1.

The probability of the outcome set is 1.

$P(\Omega) = 1$

1.2.5 Third Kolmogorov axiom

The probability of union for mutually exclusive events is:

$P(\cup_{i=1}^{\infty} E_i) = \sum_{i=1}^{\infty} P(E_i)$

1.3 Basic results

1.3.1 Probability of null

$P(\Omega) = 1$

$P(\Omega \vee \varnothing) = 1$

$P(\Omega) + P(\varnothing) = 1$

$P(\varnothing) = 0$

1.3.2 Monotonicity

Consider $E_i \subseteq E_j$:

$E_j = E_i \vee E_k$

$P(E_j) = P(E_i \vee E_k)$

Disjoint so:

$P(E_j) = P(E_i) + P(E_k)$

We know that $P(E_k) \geq 0$ from axiom 1 so:

$P(E_j) \geq P(E_i)$

1.3.3 Bounds of probabilities

As all events are subsets of the sample space:

$P(\Omega) \geq P(E)$

$1 \geq P(E)$

From axiom 1 then know:

$\forall E \in F[0 \leq P(E) \leq 1]$

1.3.4 Union and intersection for null and universal

$P(E \wedge \varnothing) = P(\varnothing) = 0$

$P(E \vee \Omega) = P(\Omega) = 1$

$P(E \vee \varnothing) = P(E)$

$P(E \wedge \Omega) = P(E)$

1.3.5 Separation rule

Firstly:

$P(E_i) = P(E_i \wedge \Omega)$

$P(E_i) = P(E_i \wedge (E_j \vee E_j^C))$

$P(E_i) = P((E_i \wedge E_j) \vee (E_i \wedge E_j^C))$

As the latter are disjoint:

$P(E_i) = P((E_i \wedge E_j) + (E_i \wedge E_j^C))$

1.3.6 Addition rule

We know that:

$P(E_i \vee E_j) = P((E_i \vee E_j) \wedge (E_j \vee E_j^C))$

By the distributive law of sets:

$P(E_i \vee E_j) = P((E_i \wedge E_j^C) \vee E_j)$

$P(E_i \vee E_j) = P((E_i \wedge E_j^C) \vee (E_j \wedge (E_i \vee E_i^C))$

By the distributive law of sets:

$P(E_i \vee E_j) = P((E_i \wedge E_j^C) \vee (E_j \wedge E_i) \vee (E_j \wedge E_i^C))$

As these are disjoint:

$P(E_i \vee E_j) = P(E_i \wedge E_j^C) + P(E_j \wedge E_i) + P(E_j \wedge E_i^C)$

From the separation rule:

$P(E_i \vee E_j) = P(E_i) - P(E_i \wedge E_j) + P(E_j \wedge E_i) + P(E_j) - P(E_j \wedge E_i)$

$P(E_i \vee E_j) = P(E_i) + P(E_j) - P(E_i \wedge E_j)$

1.3.7 Probability of complements

From the addition rule:

$P(E_i \vee E_j) = P(E_i) + P(E_j) - P(E_i \wedge E_j)$

Consider E and E^C:

$P(E \vee E^C) = P(E) + P(E^C) - P(E \wedge E^C)$

We know that E and E^C are disjoint, that is:

$E \wedge E^C = \varnothing$

Similarly by construction:

$E \vee E^C = \Omega$

So:

$P(\Omega) = P(E) + P(E^C) - P(\varnothing)$

$1 = P(E) + P(E^C)$

1.4 Other

1.4.1 Odds

Given a set of outcomes for a variable, the odds of the outcome are defined as:

$o_f = \dfrac{P(E)}{P(E^C)}$

For example, the odds of rolling a 6 are $\dfrac{1}{5}$.

1.4.2 Discrete and continous probability

We know that:

$\sum_y P(X \wedge Y) = P(X)$

So for the continuous case

$P(X) = \int_{-\infty}^{\infty} P(X \wedge Y)dy$

This behaves like the probability for a single event, or multiple events with one fewer event if there were more than 2 events to start with.

Chapter 2

Conditional probability and Bayes' theorem

2.1 Introduction

2.1.1 Conditional probability

We define conditional probability

$$P(E_i|E_j) := \frac{P(E_i \wedge E_j)}{P(E_j)}$$

We can show this is between 0 and 1.

$$P(E_j) = P(E_i \wedge E_j) + P(\bar{E}_i \wedge E_j)$$

$$P(E_i|E_j) := \frac{P(E_i \wedge E_j)}{P(E_i \wedge E_j) + P(\bar{E}_i \wedge E_j)}$$

We know:

$$P(x_i|y_j) := \frac{P(x_i \wedge y_j)}{P(y_j)}$$

$$P(y_j|x_i) := \frac{P(x_i \wedge y_j)}{P(x_i)}$$

So:

$$P(x_i|y_j)P(y_j) = P(y_j|x_i)P(x_i)$$

$$P(x_i|y_j) = \frac{P(y_j|x_i)P(x_i)}{P(y_j)}$$

Note that this is undefined when $P(y_j) = 0$

Note that for the same event,

$P(x_i|x_j) = \frac{P(x_i \wedge x_j)}{P(x_j)}$

$P(x_i|x_j) = 0$

For the same outcome:

$P(x_i|x_i) = \frac{P(x_i \wedge x_i)}{P(x_i)}$

$P(x_i|x_i) = \frac{P(x_i)}{P(x_i)}$

$P(x_i|x_i) = 1$

2.1.2 Bayes' theorem

From the definition of conditional probability we know that:

$P(E_i|E_j) := \frac{P(E_i \wedge E_j)}{P(E_j)}$

$P(E_j|E_i) := \frac{P(E_i \wedge E_j)}{P(E_i)}$

So:

$P(E_i \wedge E_j) = P(E_i|E_j)P(E_j)$

$P(E_i \wedge E_j) = P(E_j|E_i)P(E_i)$

So:

$P(E_i|E_j)P(E_j) = P(E_j|E_i)P(E_i)$

2.1.3 Independent variables

Events are independent if:

$P(E_i|E_j) = P(E_i)$

Note that:

$P(E_i \wedge E_j) = P(E_i|E_j)P(E_j)$

And so for independent events:

$P(E_i \wedge E_j) = P(E_i)P(E_j)$

2.1.4 Conjugate priors

If the prior $P(\theta)$ and the posterior $P(\theta|X)$ are in the same family of distributions (eg both Gaussian), then the prior and posterior are conjugate distributions

Chapter 3

Entropy

3.1 Entropy

3.1.1 Information

Criteria

Self information measures surprise of outcome. also called a surprisal.

When we observe an outcome we get information. We can develop a measure for how much information is associated with a specific measurement.

Rule 1: Information is always positive

Rule 2: If $P(x) = 1$, the the information for $I(P(x)) = 0$.

Rule 3: If two events are independent, then their information is additive.

- $P(C) = P(A)P(B)$
- $I(P(C)) = I(P(A)P(B))$
- $I(P(A)) + I(P(B)) = I(P(A)P(B))$

Choice of function

A function which satisifes this is $I(P(A)) = -\log(P(A))$

Any base can be used. 2 is most common, information is in units of bit then.

3.1.2 Entropy

Introduction

Entropy measures the expected amount of information produced by a source.

$H(P(x)) = E(I(P(x))$

Entropy is similar to variance, is the sense that both measure uncertainty.

Entropy, however, has no references to specific values of x. If all values were multiplied by 100, or if parts of the distribution were cut up and swapped, entropy would be unaffected.

For a probability function $p(z)$, its entropy is :

$H(p) = -\int p(z) \ln p(z) dz.$

This is a measure of the spread of a distribution.

Negative infinity means no uncertainty

For a multivariate gaussian $H = d/2ln(2\pi e|\Sigma)$.

Part II

Variables

Chapter 4

Variables

4.1 Variables

4.1.1 Random variables

Defining variables

We have a sample space, Ω. A random variable X is a mapping from the sample space to the real numbers:

$X : \Omega \to \mathbb{R}$

We can then define the set of elements in Ω. As an example, take a coin toss and a die roll. The sample space is:

$\{H1, H2, H3, H4, H5, H6, T1, T2, T3, T4, T5, T6\}$

A random variable could give us just the die value, such that:

$X(H1) = X(T1) = 1$

We can define this more precisely using set-builder notation, by saying the following is defined for all $c \in \mathbb{R}$:

$\{\omega | X(\omega) \leq c\}$

That is, for any number random variable map X, there is a corresponding subset of Ω containing the ωs in Ω which map to less than c.

Multiple variables

Multiple variables can be defined on the sample space. If we rolled a die we could define variables for

- Whether it was odd/even
- Number on the die
- Whether it was less than 3

With more die we could add even more variables

Derivative variables

If we define a variable X, we can also define another variable $Y = X^2$.

4.1.2 Probability mass functions

$P(X = x) = P(\omega | X(\omega) = x)$

For discrete probability, this is a helpful number. For example for rolling a die.

This is not helpful for continuous probability, where the chance of any specific outcome is 0.

4.1.3 Cumulative distribution functions

Definition

Random variables all valued as real numbers, and so we can write:

$P(X \leq x) = P(\omega | X(\omega) \leq x)$

Or:

$F_X(x) = \int_{-\infty}^{x} f_X(u)du$

$F_X(x) = \sum_{x_i \leq x} P(X = x_i)$

Partitions

$P(X \leq x) + P(X \geq x) - P(X = x) = 1$

Interval

$P(a < X \leq b) = F_X(b) - F_X(a)$

4.1.4 Probability density functions

Definition

If continuous, probability at any point is 0. We instead look at probability density.

Derived from cumulative distribution function:

$F_X(x) = \int_{-\infty}^{x} f_X(u)du$

The density function is $f_X(x)$.

Conditional probability distributions

For probability mass functions:

$$P(Y = y|X = x) = \frac{P(Y = y \wedge X = x)}{P(X = x)}$$

For probability density functions:

$$f_Y(y|X = x) = \frac{f_{X,Y}(x, y)}{f_X(x)}$$

4.2 Multiple variables

4.2.1 Joint and marginal probability

Joint probability

$P(X = x \wedge Y = y)$

Marginal probability

$P(X = x) = \sum_y P(X = x \wedge Y = y)$

$P(X = x) = \sum_y P(X = x|Y = y)P(Y = y)$

4.2.2 Independence and conditional independence

Independence

x is independent of y if:

$\forall x_i \in x, \forall y_j \in y(P(x_i|y_j) = P(x_i)$

If $P(x_i|y_j) = P(x_i)$ then:

$P(x_i \wedge y_j) = P(x_i).P(y_j)$

This logic extends beyond just two events. If the events are independent then:

$P(x_i \wedge y_j \wedge z_j) = P(x_i).P(y_j \wedge z_k) = P(x_i).P(y_j).P(z_k)$

Note that because:

$$P(x_i|y_j) = \frac{P(x_i \wedge y_j)}{P(y_j)}$$

If two variables are independent

$$P(x_i|y_j) = \frac{P(x_i)P(y_j)}{P(y_j)}$$

$P(x_i|y_j) = P(x_i)$

Conditional independence

$P(A \wedge B|X) = P(A|X)P(B|X)$

This is the same as:

$P(A|B \wedge X) = P(A|X)$

Chapter 5

Moments

5.1 Moments

5.1.1 Functionals of probabilities

$\phi(P) \in \mathbb{R}$ is a functional on $P(X)$.

Examples include the expectation and variance.

We can define derivatives on these functionals.

$\phi(P) \approx \phi(P^0) + D_\phi(P - P^0)$

Where D_ϕ is linear.

5.1.2 Expected value

Definition

For a random variable (or vector of random variables), x, we define the expected value of $f(x)$ as :

$E[f(x)] := \sum f(x_i)P(x_i)$

The expected value of random variable x is therefore this where $f(x) = x$.

$E(x) = \sum_i x_i P(x_i)$

Linearity of expectation

We can show that $E(x + y) = E(x) + E(y)$:

$E[x+y] = \sum_i \sum_j (x_i + y_j) P(x_i \wedge y_j)$

$E[x+y] = \sum_i \sum_j x_i [P(x_i \wedge y_j)] + \sum_i \sum_j [y_j P(x_i \wedge y_j)]$

$E[x+y] = \sum_i x_i \sum_j [P(x_i \wedge y_j)] + \sum_j y_j \sum_i [P(x_i \wedge y_j)]$

$E[x+y] = \sum_i x_i P(x_i) + \sum_j y_j P(y_j)$

$E[x+y] = E[x] + E[y]$

Expectations of multiples

Expectations

$E(cx) = \sum_i cx P(x_i)$

$E(cx) = c \sum_i x P(x_i)$

$E(cx) = cE(x)$

Expectations of constants

$E(c) = \sum_i c_i P(c_i)$

$E(c) = cP(c)$

$E(c) = c$

Conditional expectation

If Y is a variable we are interested in understanding, and X is a vector of other variables, we can create a model for Y given X.

This is the conditional expectation.

$E[Y|X]$

$E[P(Y|X)Y]$

In the continuous case this is

$E(Y|X) = \int_{-\infty}^{\infty} yP(y|X)dy$

We can then identify an error vector.

$\epsilon := Y - E(Y|X)$

So:

$Y = E(Y|X) + \epsilon$

Here Y is called the dependent variable, and X is called the dependent variable.

Iterated expectation

$E[E[Y]] = E[Y]$
$E[E[Y|X] = E[Y]$

5.1.3 Variance

Definition

The variance of a random variable is given by:
$Var(x) = E((x - E(x))^2)$
$Var(x) = E(x^2 + E(x)^2 - 2xE(x))$
$Var(x) = E(x^2) + E(E(x)^2) - E(2xE(x))$
$Var(x) = E(x^2) + E(x)^2 - 2E(x)^2$
$Var(x) = E(x^2) - E(x)^2$

Variance of a constant

$Var(c) = E(c^2) - E(c)^2$
$Var(c) = c^2 - c^2$
$Var(c) = 0$

Variance of multiple

$Var(cx) = E((cx)^2) - E(cx)^2$
$Var(cx) = E(c^2x^2) - [\sum_i cxP(x_i)]^2$
$Var(cx) = c^2E(x^2) - c^2[\sum_i xP(x_i)]^2$
$Var(cx) = c^2[E(x^2) - E(x)^2]$
$Var(cx) = c^2Var(x)$

Link between variance of expectation

$E(x)^2 + Var(x) = E(x)^2 + E((x - E(x))^2)$
$E(x)^2 + Var(x) = E(x)^2 + E(x^2 + E(x)^2 - 2xE(x))$
$E(x)^2 + Var(x) = E(x)^2 + E(x^2) + E(E(x)^2) - E(2xE(x))$

$E(x)^2 + Var(x) = E(x)^2 + E(x^2) + E(x)^2 - 2E(x)E(x))$

$E(x)^2 + Var(x) = E(x^2)$

Covariance

$Var(x+y) = E((x+y)^2) - E(x+y)^2$

$Var(x+y) = E(x^2 + y^2 + 2xy) - E(x+y)^2$

$Var(x+y) = E(x^2) + E(y^2) + E(2xy) - E(x+y)^2$

$Var(x+y) = E(x^2) + E(y^2) + E(2xy) - [E(x) + E(y)]^2$

$Var(x+y) = E(x^2) + E(y^2) + E(2xy) - E(x)^2 - E(y)^2 - 2E(x)E(y)]$

$Var(x+y) = [E(x^2) - E(x)^2] + [E(y^2) - E(y)^2] + E(2xy) - 2E(x)E(y)$

$Var(x+y) = Var(x) + Var(y) + 2[E(xy) - E(x)E(y)]$

We then define:

$Cov(x,y) := E(xy) - E(x)E(y)$

Noting that:

$Cov(x,x) = E(xx) - E(x)E(x)$

$Cov(x,x) = Var(x)$

So:

$Var(x+y) = Var(x) + Var(y) + 2Cov(x,y)$

$Var(x+y) = Cov(x,x) + Cov(x,y) + Cov(y,x) + Cov(y,y)$

$Cov(x,c) = E(xc) - E(x)E(c)$

$Cov(x,c) = cE(x) - cE(x)$

$Cov(x,c) = 0$

5.1.4 Moments

Moments

The nth moment of variable X is defined as:

$E[X^n] = \sum_i x_i^n P(x_i)$

The mean is the first moment.

Central moments

The nth central moment of variable X is defined as:

$\mu_n = E[(X - E[X])^n] = \sum_i (x_i - E[X])^n P(x_i)$

The variance is the second central moment.

Standardised moments

The nth standardised moment of variable X is defined as:

$$\frac{E[(X - E[X])^n]}{(E[(X - E[X])^2]^{\frac{n}{2}}} = \frac{\mu_n}{\sigma^n}$$

Kertosis

Kertosis is the third standardised moment.

Skew

Skew is the fourth standardised moment.

5.1.5 Covariance matrix

With multiple events, covariance can be defined between each pair of events, including the event with itself.

The covariance between 2 variables is:

$Cov(x_i, x_j) := E(x_i x_j) - E(x_i)E(x_j)$

Which is equal to:

$Cov(x_i, x_j) = E[x_i - E(x_i)][x_j - E(x_j)]$

We can therefore generate a covariance matrix through:

$\sum = E[(X - E[X])(X - E[X])^T]$

5.1.6 Jensen's inequality

If ϕ is convex then:

$\phi(E[X]) \geq E[\phi(X)])$

5.2 Other

5.2.1 Markov's inequality and Chebyshev's inequality

Lemma 1

$E[I_{X \geq a}] = P(X \geq a)$

Consider the indicator function.

$I_{X \geq a}$

This is equal to 0 if X is below a and 1 otherwise.

We can take expectations of this.

$E[I_{X \geq a}] = P(X \geq a).1 + P(X < a).0 = P(X \geq a)$

$E[I_{X \geq a}] = P(X \geq a)$

Lemma 2

$aI_{X \geq a} \leq X$

While X is below a the left side is equal to 0, which holds.

While X is equal to a the left side is equal to X, which holds.

While X is above a the left side is equal to a, which holds.

Markovs inequality

$P(X \geq a) \leq \frac{\mu}{a}$

From above:

$aI_{X \geq a} \leq X$

We can take expectations of both sides:

$E[aI_{X \geq a}] \leq E[X]$

$aP(X \geq a) \leq E[X]$

$P(X \geq a) \leq \frac{\mu}{a}$

Chebyshevs inequality

We know from Markovs inequality that:

$P(X \geq a) \leq \frac{\mu}{a}$

Lets take the variable X to be $(X - \mu)^2$

$P((X - \mu)^2 \geq a) \leq \frac{E[(X - \mu)^2]}{a}$

$P((X - \mu)^2 \geq a) \leq \frac{\sigma^2}{a}$

$P(|X - \mu| \geq \sqrt{a}) \leq \frac{\sigma^2}{a}$

Take a to be a multiple k^2 of the variance σ^2.

$a = k^2\sigma^2$

$P(|X - \mu| \geq k\sigma) \leq \frac{\sigma^2}{k^2\sigma^2}$

$P(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}$

5.2.2 Characteristic functions

Transformations

Summary

Cumulative probability function

$F = \int_{-\infty}^{\infty} xP(x)$

Moment generating function

$F = \int_{-\infty}^{\infty} e^{tx}P(x)$

Characteristic function

$F = \int_{-\infty}^{\infty} e^{itx}P(x)$

Moment generating function

Take random variable X. This has moments we wish to calculate.

We can transform our function in other forms which maintain all of the required information. For example we could also use the cumulative probability function

to calculate moments. We now look for an alternative form of the probability density function which allows us to easily calculate moments.

One method is to use the probability density function and the definitions of moments, but there are other options. For example, consider the function:

$E[e^{tX}]$

Which expands to:

$E[e^{tX}] = \sum_{j=1}^{\infty} \frac{t^j E[X^j]}{j!}$

By taking the mth derivative of this, we get

$E[X^m] + \sum_{j=m+1}^{\infty} \frac{t^j E[X^j]}{j!}$

We can then set $t = 0$ to get

$E[X^m]$

Alternatively, see that differentiating m times gets us

$E[X^m e^{tX}]$

If we can get this function, we can then easily generate moments.

The function we need to get is:

$E[e^{tX}]$

In the discrete case this is:

$E[e^{tX}] = \sum_{i=1} e^{tx_i} p_i$

In the continuous case:

$E[e^{tX}] = \int_{-\infty}^{\infty} e^{tx} P(x) dx$

Characteristic function

It may not be possible to calculate the integral for the moment generating function. We now look for an alternative formula with which we can generate the same moments.

Consider

$E[e^{itX}]$

As this can be broken down into sinusoidal functions it can more readily be integrated.

This expands to

$E[e^{itX}] = \sum_{j=1}^{\infty} \frac{i^j t^j E[X^j]}{j!}$

By taking the mth derivative we get.

$E[X^m] i^m + \sum_{j=m+1}^{\infty} \frac{t^j E[X^j]}{j!}$

By setting $t = 0$ we then get:

$E[X^m] i^m$

Alternatively see that differentiating m times gets us

$E[(iX)^m e^{itX}]$

So we can get the moment by differentiating m times, and multiplying by i^{-m}.

Inverses of these functions

Moment generating function

Characteristic function

Moments of constants added to variables

$\phi_{X+c}(t) = E[e^{it(X+c)}]$

$\phi_{X+c}(t) = E[e^{itX} e^{i} tc]$

$\phi_{X+c}(t) = e^{itc} E[e^{itX}]$

$\phi_{X+c}(t) = e^{itc} \phi_X(t)$

$\phi_X(t) = e^{-itc} \phi_{X+c}(t)$

Moments of constants multiplied by events

$\phi_{cX}(t) = E[e^{itcX}]$

$\phi_{cX}(t) = \phi_X(ct)$

Taylor series of a characteristic function

$\phi_X(t) = E[e^{itX}]$

$\phi_X(t) = \sum_{j=0}^{\infty} \frac{\phi_X^j(a)(t-a)}{j!}$

Around $a = 0$

$\phi_X(t) = \sum_{j=0}^{\infty} \frac{\phi_X^j(0)(t)}{j!}$

The characteristic function is now given in terms of its moments.

We know:

$\phi_X^j(0) = E[X^j]i^j$

So:

$\phi_X(t) = \sum_{j=0}^{\infty} \frac{E[X^j]i^j(t)^j}{j!}$

$\phi_X(t) = \sum_{j=0}^{\infty} \frac{E[X^j](it)^j}{j!}$

We know:

$\frac{E[X^0](it)^0}{0!} = E[1] = 1$

$\frac{E[X^1](it)^1}{1!} = E[X](it) = it\mu_X$

$\frac{E[X^2](it)^2}{2!} = \frac{-E[X^2]t^2}{2} = \frac{-(\mu_X + \sigma_X^2)t^2}{2}$

So:

$\phi_X(t) = 1 + it\mu_X - \frac{(\mu_X + \sigma_X^2)t^2}{2} + \sum_{j=3}^{\infty} \frac{E[X^j](it)^j}{j!}$

Chapter 6

Parametric distributions

6.1 Discrete

6.1.1 Degenerate distribution

6.1.2 Discrete uniform distribution

There is a set s such that:

$P(x \in s) = p$

$P(x \notin s) = 0$

Moments of the uniform distribution

The mean is the mean of the set s.

If the set is all numbers of the real line between two values, a and b, then:

The mean is $\frac{1}{2}(a + b)$.

The variance is $\frac{(b-a)^2}{12}$ in the continuous case.

The variance is $\frac{(b-a+1)^2-1}{12}$ in the discrete case.

6.1.3 Bernoulli distribution

The outcome of a Bernoulli trial is either 0 or 1. We can describe it as:

$P(1) = p$

$P(0) = 1 - p$

With a single parameter p.

Moments of the Bernoulli distribution

The mean of a Bernoulli trial is $E[X] = (1-p)(0) + (p)(1) = p$.

The variance of a Bernoulli trial is $E[(X-\mu)^2] = (1-p)(0-\mu)^2 + (p)(1-\mu)^2 = (1-p)p^2 + p(1-p)^2] = p(1-p)$.

6.1.4 Binomial distribution

If we repeat a Bernoulli trials with the same parameter and sum the results, we have the binomial distribution.

We therefore have two parameters, p and n.

$P(X = x) = \binom{n}{x} p^x (1-p)^{n-x}$

Moments of the binomial distribution

The mean is np, which can be seen as the trials are independent.

Similarly, the variances can be addeded together giving $np(1-p)$.

6.1.5 Poisson distribution

Definition

We can use the Poisson distribution to model the number of indepedent events that occur in an a time period.

For a very short time period the chance of us observing an event is a Bernoulli trial.

$P(1) = p$

$P(0) = 1 - p$

Chance of no observations

Let's consider the chance of repeatedly getting 0: $P(0;t)$.

We can see that: $P(0;t+\delta t) = P(0;t)(1-p)$.

And therefore:

$P(0;t+\delta t) - P(0;t) = -pP(0;t))$

By setting $p = \lambda\delta t$:

$$\frac{P(0;t+\delta t) - P(0;t)}{\delta t} = -\lambda P(0;t))$$

$$\frac{\delta P(0;t)}{\delta t} = -\lambda P(0;t)$$

$P(0;t) = Ce^{-\lambda t}$

If $t = 0$ then $P(0;t) = 0$ and so $C = 1$.

$P(0;t) = e^{-\lambda t}$

Deriving the Poisson distribution

6.1.6 Multinomial distribution

Binomial recap

The mass function for the binomial case is:

$$f(x) = \frac{n!}{x!(n-x)!}p^k(1-p)^{n-k}$$

The multinomial distribution

This generalises the binomial distribution where there are more than 2 outcomes.

$$f(x_1,...,x_n) = \frac{n!}{\prod_i x_i!}\prod_i p_i^{x_i}$$

6.2 Continous distributions

6.2.1 Exponential distribution

6.2.2 Weibull distribution

6.2.3 Power law

$P(X) = \frac{\alpha - 1}{a}(\frac{x}{a})^{-\alpha}$

Where a is the lower bound.

$P(X) = 0$ for $X < a$.

Moments of the power law

$E[X^m] = \frac{\alpha - 1}{\alpha - 1 - m}a$

If $m \geq \alpha - 1$ then this is not well defined.

Higher order moments, such that the variance, cannot be identified.

6.2.4 Logistic distribution

The logistic distribution has the cumulative distribution function:

$$F(x) = \frac{1}{1 + e^{-\frac{x-\mu}{s}}}$$

6.2.5 Lvy distribution

Definition

The Lvy distribution is a continuous probability distribution.

The marginal probability is:

$$P(X) = \sqrt{\frac{c}{2\pi}}\frac{e^{-\frac{c}{2(x-\mu)}}}{(x-\mu)^{\frac{3}{2}}}$$

6.3 Gaussian distributions

6.3.1 Gaussian

$$f_x = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma}}$$

6.3.2 The error function and the complementary error function

6.3.3 Multivariable Gaussian distribution

Definition

For univariate:

$x \sim N(\mu, \sigma^2)$

We define the multivariate gaussian distribution as the distribution where any linear combination of components are gaussian.

For multivariate:

$X \sim N(\mu, \Sigma)$

Where μ is now a vector, and Σ is the covariance matrix.

Density function is :

$$f_x = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}$$

For normal gaussian it is:

$$f_x = \frac{1}{\sqrt{2\pi|\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2)}$$

This is the same wher $n = 1$.

Singular Gaussians

Need det $|\Sigma|$ and Σ^{-1}. These rely on the covariance matrix not being degenerate.

If the covariance matrix is degenerate we can instead use the pseudo inverse, and the pseudo determinant.

6.4 Extreme value distributions

6.4.1 Type-I - Gumbel distribution

The probability function is:

$$f(x) = \frac{1}{\beta} e^{-(\frac{x-\mu}{\beta} + e^{-\frac{x-\mu}{\beta}})}$$

We can use:

$$z = \frac{x-\mu}{\beta}$$

To get:

$$f(x) = \frac{1}{\beta} e^{-(z+e^{-z})}$$

Link to the logistic function

The difference between two draws from a Gumbel distribution is drawn from the logistic function.

6.4.2 Type-II - Frechet distribution

6.4.3 Type-III - Reversed Weibull distribution

6.5 Mixture models

6.5.1 Gaussian Mixture Models

Mixture models

We have a latent variable which is part of the process

The variable is distributed according to parametric distribution, but parameters are different for differnet latent classes.

There are K latent classes, and so K sets of parameters.

The population is weighted into the K classes.

We have a distribution, but we have different parameters for the distribution for different populations.

For example we could observe the height of men and women, where both are normally distributed but with different parameters.

Where there is a normal distribution, this is a Gaussian mixture model.

If there is more than one variable to observe, this is a multivariate Gaussian mixture model.

Gaussian Mixture Models (GMM)

In a Gaussian Mixture Model each non latent variable has a normal distriubtion with a mean and variance. For multiple variables there is a covariance matrix.

6.6 Other

6.6.1 Laplace distribution

6.6.2 Dirac distribution

6.6.3 Empirical distribution

6.6.4 Split-normal distribution

Part III

Sampling and statistics

Chapter 7

Independent and identically distributed variables

7.1 Identically Independently Distributed variables (IID)

7.2 Convergence

7.2.1 IID

Identically distributed

x is identically distributed to y if:

$\forall i(\exists x_i \to P(x_i) = P(y_i))$

Covariance matrix of IID variables

For IID varaibles, the covariance matrix is:

$\Sigma = \sigma^2 I$

7.2.2 Lvy's continuity theorem

7.2.3 Weak law of large numbers

The sample mean is:

$\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$

The variance of this is:

$Var[\bar{X}_n] = Var[\frac{1}{n}\sum_{i=1}^{n} X_i]$

$Var[\bar{X}_n] = \frac{1}{n^2} nVar[X]$

$Var[\bar{X}_n] = \frac{\sigma^2}{n}$

We know from Chebyshevs inequality:

$P(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}$

Use $\bar{X}_n$ as X:

$P(|\bar{X}_n - \mu| \geq \frac{k\sigma}{\sqrt{n}}) \leq \frac{1}{k^2}$

Update k so $k := \frac{k\sqrt{n}}{\sigma}$

$P(|\bar{X}_n - \mu| \geq k) \leq \frac{\sigma^2}{nk^2}$

As n increases, the chance that the sample mean lies outside a given distance from the population mean approaches 0.

7.2.4 Central limit theorem

Generalise weak law of large numbers

Characteristic function of summed IID events

$Z = \sum_{i=1}^{n} Y_i$

$\phi_Z(t) = E[e^{itZ}]$

$\phi_Z(t) = E[e^{it\sum_{i=1}^{n} Y_i}]$

$\phi_Z(t) = E[e^{itY}]^n$

$\phi_Z(t) = \phi_Y(t)^n$

Taylor series: first moments dominate with means

$Z = \sum_{i=1}^{n} Y_i$

$$Y = \frac{X}{n}$$

$$\phi_Z(t) = \phi_Y(t)^n$$

$$\phi_Z(t) = \phi_{\frac{X}{n}}(t)^n$$

$$\phi_Z(t) = \phi_X(\frac{t}{n})^n$$

$$\phi_X(t) = 1 + it\mu_X - \frac{(\mu_X + \sigma_X^2)t^2}{2} + \sum_{j=3}^{\infty} \frac{E[X^j](it)^j}{j!}$$

$$\phi_X(\frac{t}{n}) = 1 + i\frac{t\mu_X}{n} - \frac{(\mu_X + \sigma_X^2)(\frac{t}{n})^2}{2} + \sum_{j=3}^{\infty} \frac{E[X^j](i\frac{t}{n})^j}{j!}$$

$$\phi_X(\frac{t}{n}) = 1 + i\frac{t\mu_X}{n} - \frac{(\mu_X + \sigma_X^2)t^2}{2n^2} + \sum_{j=3}^{\infty} \frac{E[X^j](i\frac{t}{n})^j}{j!}$$

Eliminating the imaginary term

We want μ to be 0.

$$Z = \sum_{i=1}^{n} Y_i$$

$$Y = \frac{X - \mu_X}{n}$$

$$\phi_Y(t) = 1 + it\mu_Y - \frac{(\mu_Y + \sigma_Y^2)t^2}{2} + \sum_{j=3}^{\infty} \frac{E[Y^j](it)^j}{j!}$$

$$\mu_Y = E[\frac{X - \mu_X}{n}] = \mu_X - \mu_X n = 0$$

$$\phi_Y(t) = 1 - \frac{\sigma_Y^2 t^2}{2} + \sum_{j=3}^{\infty} \frac{E[Y^j](it)^j}{j!}$$

$$\sigma_Y^2 = E[(\frac{X - \mu_X}{n})^2]$$

$$\sigma_Y^2 = E[\frac{X^2 + \mu_X^2 - 2X\mu_X}{n^2}]$$

$$\sigma_Y^2 = \frac{E[X^2] + E[\mu_X^2] - E[2X\mu_X]}{n^2}] \; \sigma_Y^2 = \frac{E[X^2] - \mu_X^2}{n^2}]$$

$$\sigma_Y^2 = \frac{\sigma_X^2}{n^2}$$

$$\phi_Y(t) = 1 - \frac{\sigma_X^2 t^2}{2n^2} + \sum_{j=3}^{\infty} \frac{E[(\frac{X - \mu}{n})^j](it)^j}{j!}$$

$$\phi_Z(t) = \phi_Y(t)^n$$

$$\phi_Z(t) = [1 - \frac{\sigma_X^2 t^2}{2n^2} + \sum_{j=3}^{\infty} \frac{E[(\frac{X-\mu}{n})^j](it)^j}{j!}]^n$$

$$\phi_Z(t) = [1 - \frac{\sigma_X^2 t^2}{2n^2}]^n$$

Eliminating σ^2

$$Z = \sum_{i=1}^{n} Y_i$$

$$Y = \frac{X - \mu_X}{\sigma n}$$

$$\phi_Y(t) = 1 + it\mu_Y - \frac{(\mu_Y + \sigma_Y^2)t^2}{2} + \sum_{j=3}^{\infty} \frac{E[Y^j](it)^j}{j!}$$

$$\mu_Y = E[\frac{X - \mu_X}{\sigma_X n}] = \mu_X - \mu_X \sigma_X n = 0$$

$$\phi_Y(t) = 1 - \frac{\sigma_Y^2 t^2}{2} + \sum_{j=3}^{\infty} \frac{E[Y^j](it)^j}{j!}$$

$$\sigma_Y^2 = E[(\frac{X - \mu_X}{\sigma n})^2]$$

$$\sigma_Y^2 = E[\frac{X^2 + \mu_X^2 - 2X\mu_X}{\sigma^2 n^2}]$$

$$\sigma_Y^2 = \frac{E[X^2] + \mu_X^2 - 2E[X]\mu_X}{\sigma^2 n^2}$$

$$\sigma_Y^2 = \frac{E[X^2] - \mu_X^2}{\sigma^2 n^2}$$

$$\sigma_Y^2 = \frac{\sigma_X^2}{\sigma^2 n^2}$$

$$\sigma_Y^2 = \frac{1}{n^2}$$

$$\phi_Y(t) = 1 - \frac{t^2}{2n^2} + \sum_{j=3}^{\infty} \frac{E[(\frac{X-\mu}{\sigma n})^j](it)^j}{j!}$$

$$\phi_Z(t) = \phi_Y(t)^n$$

$$\phi_Z(t) = [1 - \frac{t^2}{2n^2} + \sum_{j=3}^{\infty} \frac{E[(\frac{X-\mu}{\sigma n})^j](it)^j}{j!}]^n$$

$$\phi_Z(t) = [1 - \frac{t^2}{2n^2}]^n$$

Preparing for exponential expansion

We know that

$[1 + \frac{x}{n}]^n = e^x$

As $n \to \infty$.

With:

$Z = \sum_{i=1}^{n} Y_i$

$Y = \frac{X - \mu_X}{\sigma n}$

We have:

$\phi_Z(t) = [1 - \frac{t^2}{2n^2}]^n$

With:

$Z = \sum_{i=1}^{n} Y_i$

$Y = \frac{X - \mu_X}{\sigma \sqrt{n}}$

We have:

$\phi_Z(t) = [1 - \frac{t^2}{2n}]^n$

Which tends towards

$\phi_Z(t) = e^{-\frac{1}{2}t^2}$

Rescaling

The average of random variables, less their mean, and divided by their standard deviation multiplied by the square root of the sample size, follows a normal distribution as n increases.

What does this say about the actual distribution of sample averages?

$Z = \sum_{i=1}^{n} Y_i$

$Y_i = \frac{X_i - \mu_X}{\sigma_X \sqrt{n}}$

$\sum_{i=1}^{n} Y_i$

$Y = \frac{X}{n}$

Let's create Q.

$$Q = \frac{Z\sigma_X}{\sqrt{n}} + \mu_X$$

$$Q = \frac{(\sum_{i=1}^{n} Y_i)\sigma_X}{\sqrt{n}} + \mu_X$$

$$Q = \frac{(\sum_{i=1}^{n} (\frac{X_i - \mu_X}{\sigma_X \sqrt{n}}))\sigma_X}{\sqrt{n}} + \mu_X$$

$$Q = \sum_{i=1}^{n} (\frac{X_i - \mu_X}{n}) + \mu_X$$

$$Q = \sum_{i=1}^{n} (\frac{X_i - \mu_X}{n} + \frac{\mu_X}{n})$$

$$Q = \sum_{i=1}^{n} (\frac{X_i}{n})$$

This is the sample average.

$$\phi_Q(t) = \phi_{\frac{Z\sigma_X}{\sqrt{n}} + \mu_X}(t)$$

$$\phi_Q(t) = \phi_Z(\frac{t\sigma_X}{\sqrt{n}}) e^{it\mu_X}$$

$$\phi_Z(\frac{t\sigma_X}{\sqrt{n}}) = e^{-\frac{1}{2}(\frac{t\sigma_X}{\sqrt{n}})^2}$$

$$\phi_Z(\frac{t\sigma_X}{\sqrt{n}}) = e^{-\frac{1}{2}\frac{t^2\sigma_X^2}{n}}$$

$$\phi_Q(t) = e^{-\frac{1}{2}\frac{t^2\sigma_X^2}{n}} e^{it\mu_X}$$

Normal distribution

We name the normal distribution this function when $n = 1$

$$N(\mu_X, \sigma_X^2) = e^{-\frac{1}{2}\frac{t^2\sigma_X^2}{n}} e^{it\mu_X}$$

$$N(\mu_X, \sigma_X^2) = e^{-\frac{1}{2}t^2\sigma_X^2} e^{it\mu_X}$$

Getting the probability distribution function

$$\phi_X(t) = e^{-\frac{1}{2}t^2\sigma_X^2} e^{it\mu_X}$$

$$\phi_X(t) = e^{-\frac{1}{2}t^2\sigma_X^2} [\cos(t\mu_X) + i\sin(t\mu_X)]$$

7.2.5 Convergence in distribution (converge weakly)

7.2.6 Convergence in probability and o-notation

Introduction

Converges in probability

$P(distance(X_n, X) > \epsilon) \to 0$

For all ϵ.

$X_n \to^P X$

Little o notation

Little o notation is used to describe convergence in probability.

$X_n = o_p(a_n)$

mean that

$\frac{X_n}{a_n}$

Converges to 0 and n approaches something

Can be wrtiten:

$\frac{X_n}{a_n} = o_p(1)$

Big O notation

Big O notation is used to describe boundedness.

$X_n = O_p(a_n)$

means that:

If something is little o, it is big O.

7.2.7 Almost sure convergence

X_n converges almost surely to X if:

$d(X_n, X) \to 0$

Where $d(X_n, X)$ is a distance metric.

$X_n \to^{as} X$

Chapter 8

Statistics

8.1 Creating statistics

8.1.1 Creating statistics

We take a sample from the distribution.

$x = (x_1, x_2, ..., x_n)$

A statistic is a function on this sample.

$S = S(x_1, x_2, ..., x_n)$.

8.2 Moments of statistics

8.2.1 Bias from single and joint estimation

Bias from single estimation

$\mathbf{x}_i$ and $\mathbf{z}_i$ are not independent, so we cannot estimate just $y_i = \mathbf{x}_i\theta$.

Bias from joint estimation

We could estimate our equation with a single ML algorithm.

$y_i = f(\mathbf{x}_i, \theta) + g(\mathbf{z}_i) + \epsilon_i$

For example, using LASSO.

However this would introduce bias into our estimates for θ.

Bias from iterative estimation

We could iteratively estimate both θ and $g(\mathbf{z}_i)$.

For example iteratvely doing OLS for θ and random forests for z_i.

This would also introduce bias into θ.

8.3 Asymptotic properties of statistics

8.3.1 Asymptotic distributions

$f(\hat{\theta}) \to^d G$

Where G is some distribution.

8.3.2 Asymptotic mean and variance

8.3.3 Asymptotic normality

Many statistics are asymptotically normally distribution.

This is a result of the central limit theorem.

For example:

$\sqrt{n}S \to^d N(s, \sigma^2)$

Confidence intervals for asymptotically normal statistics

We have the mean and variance, and know the distribution. This allows us to calculare confidence intervals.

8.4 Order statistics

8.4.1 Order statistics

Defining order statistics

The kth order statistic is the kth smallest value in a sample.

$x_{(1)}$ is the smallest value in a sample, the minimum.

$x_{(n)}$ is the largest value in a sample, the maximum.

Probability distributions of order statistics

The probability distribution of order statistics depends on the underlying probability distribution.

Probability distribution of sample maximum

If we have:

$Y = \max \mathbf{X}$

The probability distribution is:

$P(Y \leq y) = P(X_1 \leq y, X_2 \leq y, ..., X_n \leq y)$

If these are iid we have:

$P(Y \leq y) = \prod_i P(X_i \leq y)$

$F_y(y) = F_X(y)^n$

The density function is:

$f_y(y) = nF_X(y)^{n-1} f_x(y)$

Probability distribution of the sample minimum

If we have:

$Y = \min \mathbf{X}$

The probability distribution is:

$P(Y \leq y) = P(X_1 \geq y, X_2 \geq y, ..., X_n \geq y)$

If these are iid we have:

$P(Y \leq y) = \prod_i P(X_i \geq y)$

$F_y(y) = [1 - F_X(y)]^n$

The density function is:

$f_y(y) = -n[1 - F_X(y)]^{n-1} f_x(y)$

8.5 Bootstrapping

8.5.1 Bootstrapping

If we have a sample of n, we can create bootstrap samples by drawing with replacement for other sets with n members.

8.5.2 Variance of bootstrap estimators

8.6 Jackknifing

8.6.1 The jackknife

We have a statistic:

$S(x_1, x_2, ..., x_n)$

We may want to estimate moments for this statistic, but are unable to do so.

The jackknife estimator

The jackknife is an approach for getting moments for statistics.

We start by creating n statistics each leaving out one observation.

$\bar{S}_i(x_1, x_2, ...x_{i-1}, x_{i+1}, ..., x_n)$

We define:

$\bar{S} = \frac{1}{n} \sum_i \bar{S}_i$

Moments of the jackknife estimator

We want to know the variance.

$Var\bar{S} = \frac{n-1}{n} \sum_i (\bar{S}_i - \bar{S})^2.$

8.6.2 The infintesimal jackknife

The jackknife as a weighting

In the jackknife we calculate the statistic leaving one observation out.

This is the same as weighting observations and giving one a weighting of 0 and the others 1.

The infintesimal jackknife

For the infintesimal jackknife we reduce the weight not to 0, but by an infintesimal amount.

8.6.3 Variance of jackknife estimators

8.7 Pivotal quantity

8.7.1 Introduction

A pivotal quantity is a statistic whose distribution does not depend on the parameters of the underlying distribution.

For example, the z statistic if the underyling distribution is a normal distribution.

Chapter 9

Sampling from probability distributions

9.1 Markov Chain Monte Carlo (MCMC) methods

9.2 Metropolis-Hastings algorithm

9.2.1 Direct sampling

Density estimation through direct sampling

There is distribution $P(x)$ which we want to know more about.

If the function was closed, we could estimate it by using values of x.

Limitations of direct sampling

However if the function does not have such a form, we cannot do that.

We can't plug in values, because the function is complex.

Sometimes we may know a function of the form:

$f(x) = cP(x)$

That is, a multiple of the function.

This can happen from Bayes' theorem:

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

We may be able to estimate $P(x|y)$ and $P(y)$, but not $P(x)$

This means be have

$P(y|x) = cP(x|y)P(y)$

9.2.2 The Metropolis-Hastings algorithm

The Metropolis-Hastings algorithm

The Metropolis-Hastings algorithm creates a set of samples x such that the distribution of the samples approaches the goal distribution.

Initialisation

The algorithm takes an arbitrary starting sample x_0. It then must decide which sample to consider next.

Generation

It does this using a Markov chain. That is, there is a map $g(x_j, x_i)$.

This distribution is generally a normal distribution around x_i, making the process a random walk.

Acceptance

Now we have a considered sample, we can either accept or reject it. It is this step that makes the end distribution approximage the function.

We accept if $\frac{f(x_j)}{f(x_i)} > u$, where u is a random variable between 0 and 1, generated each time.

We can calculate this because we know this function.

Properties

9.3 Gibb's sampling

9.3.1 Gibb's sampling

Introduction

As with Metropolis-Hastings, we want to generate samples for $P(X)$ and use this to approximate its form.

We do this by using the conditional distribution. If X is a vector then we also have:

$P(x_j|x_0, ..., x_{j-1}, x_{j+1}, ..., x_n)$

We use our knowledge of this distribution.

Start with vector x_0.

This has components $x_{0,j}$

To form the next vector x_1 we loop through each component.

$P(x_{1,0}|x_{0,0}, x_{0,1}, ..., x_{0,n})$

We use this to form $x_{1,0}$

However after th the first component we update this so it uses the updated variables.

$P(x_{1,k}|x_{1,0}, ..., x_{1,k-1}, x_{0,k}, ..., x_{0,n}$

This means we only need to know the conditional distributions.

9.4 Acceptance-rejection sampling

9.4.1 Introduction

Used to sample from propability distribution function.

Useful when can't use direct sampling, because no closed form.

MORE GENERALLY FRAME THESE FIRST AS SAMPLING FROM PROBABILITY FUNCTION.

Generate pairs of (x, y). If $y < P(x)$ then keep x.

Metropolis-Hastings and Gibb's sampling are extensions of this.

Chapter 10

Likelihood functions

10.1 Likelihood functions

10.1.1 Likelihood function

We want to estimate parameters. One way of looking into this is to look at the likelihood function:

$L(\theta; X) = P(X|\theta)$

The likelihood function shows the chance of the observed data being generated, given specific parameters.

If this has high peaks then it provides information that θ is located in this region.

10.1.2 IID

For multiple events, the likelihood function is:

$L(\theta; X) = P(X|\theta)$

$L(\theta; X) = P(A_1 \wedge B_2 \wedge C_3 \wedge D_4|\theta)$

If the events are independent, that is the chance of a flip doesnt depend on any other outcomes, then:

$L(\theta; X) = P(A_1|\theta).P(B_2|\theta).P(C_3|\theta).P(D_4|\theta)...$

If the events are identically distributed, the chance of flipping a head doesnt change across flips (for example the heads side doesnt get heavier over time) then:

$L(\theta; X) = P(A|\theta).P(B|\theta).P(C|\theta).P(D|\theta)...$

$L(\theta; X) = \prod_{i=1}^{n} P(X_i|\theta)$

10.2 Score functions

10.2.1 The score

The score is defined as the differential of the log-likelihood function with respect to θ.

$V(\theta, X) = \frac{\delta}{\delta\theta} l(\theta; X)$

$V(\theta, X) = \frac{1}{\prod_{i=1}^{n} P(X_i|\theta)} \frac{\delta}{\delta\theta} L(\theta; X)$

10.2.2 Expectation of the score

The expectation of the score, given the true value of θ is:

$E[V(X|\theta)] = \int V(X|\theta) dX$

$E[V(X|\theta)] = E[\frac{1}{\prod_{i=1}^{n} P(X_i|\theta)} \frac{\delta}{\delta\theta} L(\theta; X)]$

$E[V(X|\theta)] = \int \frac{1}{\prod_{i=1}^{n} P(X_i|\theta)} \frac{\delta}{\delta\theta} L(\theta; X)$

$E[\frac{1}{\prod_{i=1}^{n} P(X_i|\theta)}]$

$\int \frac{1}{\prod_{i=1}^{n} P(X_i|\theta)} P(\theta) d\theta$

We can show that the expected value of this is 0.

10.2.3 Variance of the score

The variance of the score is:

$var[\frac{\delta}{\delta\theta} l(\theta; X)]$

$var[\frac{1}{\prod_{i=1}^{n} P(X_i|\theta)}]$

10.3 Fisher information

10.3.1 Fisher information

The Fisher information is the variance:

$E[(\frac{\delta}{\delta\theta}\log f(X,\theta))^2|\theta]$

$E[\frac{\delta^2}{\delta\theta^2}\log f(X,\theta)|\theta]$

Same as expectation of score squared, because centred around 0.

10.3.2 Fisher information matrix

We have k parameters.

$I(\theta)_{ij} = E[(\frac{\delta}{\delta\theta_i}\log f(X,\theta))(\frac{\delta}{\delta\theta_j}\log f(X,\theta))|\theta]$

10.3.3 Observed Fisher information matrix

The Fisher information matrix contains informatio about the population

The observed Fisher infoirmation is the negative of the Hessian of the log likelihood.

We have:

- $l(\theta|\mathbf{X}) = \sum_i \ln P(\mathbf{x}_i|\theta)$
- $J(\theta^*) = -\nabla\nabla^T l(\theta|mathbfX)|_{\theta=\theta^*}$

The Fisher information is the expected value of this.

$I(\theta) = E[J(\theta)]$

10.4 Orthogonality

10.4.1 Orthogonality

Two variables are called orthogonal if their entry in fisher info matrix is 0

This means that the parameters can be calculated separately. MLE estimates are separate

This can be written as a moment condition

δ

10.5 Quasi-likelihood function

10.5.1 Quasi-likelihood function

Chapter 11

Privacy

11.1 Differential privacy

Part IV

Stochastic processes

Chapter 12

Stochastic processes

12.1 Introduction to processes

12.1.1 Stochastic processes

In a stochastic process we have a mapping from a variable (time) to a random variable.

Discrete and continuous time

Time could be discrete, or continuous.

Temperature over time is a stochastic process, as is the number of cars sold each day.

Discrete and continous state space

The state space for temperature is continous, the number of people on the moon is discrete.

12.1.2 Stochastic evolution

We can describe processes by their evolution.

$p(x_t|x_{t-1}...)$

12.1.3 Gaussian processes

12.1.4 Moments of stochastic processes

12.1.5 Autocovariance and autocorrelation

Autocovariance

$AC(a, b) = cov(X_a, X_b)$

Autocorrelation

The autocorrelation between two time periods is their covariance, normlised by their variances

$$AC(a, b) = \frac{E[(X_a - \mu_a)(X_b - \mu_b)]}{\sigma_a \sigma_b}$$

This is also called serial correlation.

12.1.6 Martingale property

For a process with the Martingale property, the expected value of all future variables is the current state.

This only restricts expectations.

$E(X_{n+1}|X_0, ..., X_n) = X_n$

12.2 Stationarity

12.2.1 Weak- and wide-sense stationarity

Unconditional probabilities don't change over time.

So GDP would not be stationary, but random noise would. A random walk is not stationary, because the variance increases over time.

Order of integration

How many differences to make it stationary?

Weak-sense stationary

Mean and autocovarinance don't change over time.

Wide-sense stationary

All moments are the same.

12.2.2 Unit roots

12.2.3 White noise

Variables at each time are indepdendent.

12.2.4 Orders of integration

How many diffs do you need to do to get a stationary process?

If something is first order integrated it is $I(1)$.

12.2.5 Trend stationary

If we can remove the trend as a function, eg linear or non-linear growth, and the rest is stationary, then the process is trend stationary

12.3 Ergodic processes

12.3.1 Ergodic processes

Sample moments must converge to generating momements. Not guaranteed.

Eg process with path dependence. 50

Generating average is 50, but sample will only convergen to 100 or 0

12.4 Processes decomposition (from uni forecasting?)

12.4.1 Wold's theorem

12.5 Pseudo random numbers

12.5.1 Seeds

12.5.2 Period

12.6 Brownian motion

12.6.1 Brownian motion

brownian motion in stats. given we start at a, what is chance be end up at b? normal. do 1d then multi d

12.7 Wold's theorem

12.7.1 Introduction

12.8 Introduction

12.8.1 Seasonal and non-seasonal trends

We can model the process as:

$y_t = \mu_t + f(t) + \epsilon_t$

12.8.2 Identifying the order of integration using Augmented Dickey-Fuller

The Dickey-Fuller test with deterministic time trend was:

$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \epsilon_t$

The Augmented Dickey-Fuller model adds lags for the differences.

$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \sum_i^p \delta_i \Delta y_{t-i} + \epsilon_t$

12.8.3 Cyclical fluctuations

We can have shocks having effects over time.

This is separate to trends.

12.8.4 Identifying serial correlation using the Durbin-Watson statistic

12.8.5 Introduction to forecasting

We observe a series of observations:

$x_1, x_2, ..., x_t)$

What can we say about x_{t+1}?

If the data was drawn iid then the past data then we would just want to identify moments.

However if the data is not iid, for example because it is increasing in time, then this is not the best way.

Regression formation

We can model

$x_t = \alpha + \epsilon_t$

12.9 Autoregressive model

12.9.1 Autoregressive models (AR)

AR(1)

Our basic model was:

$x_t = \alpha + \epsilon_t$

We add an autoregressive component by adding a lagged observation.

$x_t = \alpha + \beta x_{t-1} + \epsilon_t$

AR(p)

AR(p) has p previous dependent variables.

$x_t = \alpha + \sum_{i=1}^{p} \beta_i x_{t-i}$

Propagation of shocks

A shock bumps up the output variable, which bumps up output variables forever, at a decreasing rate.

12.9.2 Testing for stationarity with Dickey-Fuller (DF) and Augmented Dicky-Fuller (ADF)

Stationarity

Unit roots

Integration order

Dickey-Fuller

The Dickey-Fuller test tests if there is a unit root.

The AR(1) model is:

$y_t = \alpha + \beta y_{t-1} + \epsilon_t$

We can rewrite this as:

$\Delta y_t = \alpha + (\beta - 1) y_{t-1} + \epsilon_t$

We test if $\beta - 1) = 0$.

If the coefficient on the last term is 1 we have a random walk, and the process is non-stationary.

If the last term is < 1 then we have a stationary process.

Variation: Removing the drift

If our model has no intercept it is:

$y_t = \beta y_{t-1} + \epsilon_t$

$\Delta y_t = (\beta - 1) y_{t-1} + \epsilon_t$

Variation: Adding a deterministic trend

If our model has a time trend it is:

$y_t = \alpha\beta y_{t-1} + \gamma t + \epsilon_t$

$\Delta y_t = \alpha + (\beta - 1)y_{t-1} + \gamma t + \epsilon_t$

Augmented Dickey-Fuller

We include more lagged variables.

$y_t = \alpha + \beta t + \sum_i^p \theta_i y_{t-i} + \epsilon_t$

If no unit root, can do normal OLS?

12.9.3 Autoregressive Conditional Heteroskedasticity (ARCH)

Variance of the AR(1) model

The standard AR(1) model is:

$y_t = \alpha + \beta y_{t-1} + \epsilon_t$

The variance is:

$Var(y_t) = Var(\alpha + \beta y_{t-1} + \epsilon_t)$

$Var(y_t)(1 - \beta^2) = Var(\epsilon_t)$

Assuming the errors are IID we have:

$$Var(y_t)) = \frac{\sigma^2}{1 - \beta^2}$$

This is independent of historic observations, which may not be desirable.

Conditional variance

Consider the alternative formulation:

$y_t = \epsilon_t f(y_{t-1})$

This allows for conditional heteroskedasticity.

12.10 Moving average models

12.10.1 Moving Average models (MA)

We add previous error terms as input variables

MA(q) has q previous error terms in the model

Unlike AR models, the effects of any shocks wear off after q terms.

This is harder to fit the OLS, the error terms themselves are not observed.

12.11 Autoregressive Moving Average models

12.11.1 Autoregressive Moving Average models (ARMA)

We include both AR and MA

Estimted using Box-Jenkins

12.11.2 Autoregressive Integrated Moving Average models (ARIMA)

Uses differences to remove non statiority

Also estiamted with box-jenkins

12.11.3 Seasonal ARIMA

12.12 Forecasting

12.12.1 Monte carlo simulations

12.12.2 N-step ahead

12.12.3 Consensus forecasting

12.13 Introduction to multiple time series

12.13.1 Testing for cointegration with Johansen

12.14 Vector Autoregression (VAR)

12.14.1 Vector Autoregression (VAR)

We consider a vector of observables, not just one

Autoregressive (AR) model for a vector.

VAR(p) looks p back.

The AR(p) model is:

$y_t = \alpha + \sum_{i=1}^{p} \beta y_{t-i} + \epsilon_t$

VAR(p) generalises this to where y_t is a vector. We define VAR(p) as:

y_t

$y_t = c + \sum_{i=1}^{p} A_i y_{t-i} + \epsilon_t$

12.14.2 VAR impulse response

12.14.3 Bayesian VAR

12.15 Structural models

12.15.1 Autoregressive Distributed Lag (ARDL) model

Include lagged y and lagged x (and current x)

12.16 ARMAX

12.16.1 ARMAX

12.16.2 Error Correction Model

Static model

Like PAM we start with static estimator.

The ECM

The ECM does a regression with first differences, and includes lagged error terms.

We start with a basic first-difference model.

$\Delta y_t = \Delta x_t$

We could also expand this to include laggs for both x and y. Here we don't.

We know that long term $y_t = \theta x_t$. We use the error from this in a first difference model.

$\Delta y_t = \alpha \Delta x_t + \beta(y_{t-1} - \theta x_{t-1})$

Page on identifying error terms

Also, page on Vector Error Correction Model (VECM)

12.16.3 Partial Adjustment Model

Estimating a static model

We start by estimating a static model.

$y_t = \alpha + \theta x_t + \gamma_t$

Equilibrium

We then use this form an equilibrium for y_t, y_t^*.

$y_t^* = \hat{\alpha} + \hat{\theta} x_t$

The process depends on the difference from this equilibrium.

$y_t - y_{t-1} = \beta(y_t^* - y_{t-1}) + \epsilon_t$

$y_t - y_{t-1} = \beta(\hat{\alpha} + \hat{\theta}x_t - y_{t-1}) + \epsilon_t$

$y_t = \beta\hat{\alpha} + \beta\hat{\theta}x_t + (1-\beta)y_{t-1} + \epsilon_t$

$y_t = \alpha y_{t-1} + (1-\beta)(y_t^* - y_{t-1}) + \epsilon$

The higher β, the slower the adjustment.

If stationary, can we can use OLS.

12.16.4 ARIMAX

12.16.5 SARIMA

Chapter 13

Markov processes

13.0.1 Introduction

Markov property

For a process with the Markov property, only the current state matters for all probability distributions.

$P(x_{t+n}|x_t) = P(x_{t+n}|x_t, x_{t-1}...)$

13.1 Markov chains

13.1.1 Finite state Markov chains

Transition matrices

This shows the probability for moving between discrete states.

We can show the probability of being in a state by multiplying the vector state by the transition matrix.

Mv

Time-homogenous Markov chains

For time-homogenous Markov chains the transition matrix is independent of time.

For these we can calculate the probability of being in any given state in the future:

$M^n v$

This becomes independent of v as we tend to infinity. The initial starting state does not matter for long term probabilities.

How to find steady state probability?

$Mv = v$

The eigenvectors! With associated eigenvector 1. There is only one eigenvector. We can find it by iteratively multiplying any vector by M.

13.1.2 Random walks

13.1.3 Infinite state Markov chains

Markov model description We can represent the transition matrix as a series of rules to reduce the number of dimensions $P(x_t|y_{t-1}) = f(x, y)$

can represent states as number, rather than atomic. could be continuous, or even real.

in more complex, can use vectors.

13.2 Hidden Markov Models

13.2.1 Introduction

As well as the Markov process X, we have another process Y which depends on X.

13.3 Dynamic Bayesian networks

13.3.1 Introduction

Chapter 14

Multivariate time series

14.1 Multiple time series

14.1.1 Cointegration

If we have multiple variables, we can explore the order of integration of linear combinations.

If two series have time trends, a linear combination of them could remove this.

14.1.2 Exogeneity

Contemporaneous exogeneity

$Cov(x_{it}, u_{it}) = 0$

Strict exogeneity

$Cov(x_{is}, u_{it}) = 0)$

This is stronger than contemporeous, all periods.

Shocks don't affect future outcomes.

Sequential exogeneity

Sequential exogeneity: a bit looser than strict exogeneity. only holds when $s \leq t$.

So shocks can affect, but only in future.

14.1.3 Introduction

Weak stationary processes can be decomposed to a deterministic and a stochastic component.

Chapter 15

Sampling from processes

15.1 Introduction

Chapter 16

Bayesian networks

16.1 Bayesian networks

16.1.1 Bayesian networks

Part V

Stochastic methods

Chapter 17

Integration

17.1 Introduction

Chapter 18

Optimisation

18.1 Random search

18.1.1 Random search

We start with a random set of parameters, x.

We then loop through the following:

- We define a search space local to our current selection.
- We randomly select a point from this space.
- We compare the new point to our current point. If the new point is better we move to that.

18.1.2 Random optimisation

This is similar to random search, however we use a multivariate Gaussian distribution around our current point rather than a hypersphere.

18.1.3 Simulated annealing

Introduction

We can use a version of Metropolis-Hastings to find the global maximum of a function $f(x)$.

We start with an arbitrary point x_0.

We move randomly from this to identiy a candidate point x_c.

We accept this with probability depending on the the relationship between x_0 and x_c.

This process will converge on the global maximum.

Hyperparameter

There is a hyperparameter for selection. At the extreme this becomes a greedy function.

18.2 Bayesian optimisation

18.2.1 Bayesian optimisation

Introduction

If we have sampled from the hyperparameter space we know something about the shape.

Can we use this to inform where we should next look?

The shape of the function is $y = f(\mathbf{x})$

We have observations $\mathbf{X}$ and $\mathbf{y}$.

So what's our posterior, $P(y|\mathbf{X}, \mathbf{y})$?

Exploration and exploitation

The can be a tradeoff between:

- Exploring - which gives us a better shape for $y = f(x)$; and
- Exploiting - which gives us a better estimate for the global optimum.

The surrogate function

We do not know $y = f(x)$, but we model it as:

$z(x) = y(x) + \epsilon$

We can then maximise z

Proposing new candidates

We want an algorithm which maps from our history of observations to a new candidate.

There are different approaches:

- Probability of improvement - Choosing one with the highest chance of a more optimal value
- Expected improvement - Choosing one with the biggest expected increase in the optimal value
- Entropy search - choosing one which reduces uncertainty about the global maximum.

18.3 Evolutionary algorithms

18.3.1 Evolutionary algorithms

Initialisation

We generate a set of candidate parameter values, x.

Evaluate using the fitness function

We evaluate each of these against a fitness function (the function we are optimising).

We assign fitness values to each individual.

Crossover and mutation

We generate a second generation. We select "parents" randomly using the fitness values as weightings.

The values of the new individual are a function of the values of the parents, and noise (mutation).

We do this for each member in the next generation.

We iterate this process across successive generations.

18.4 Differential evolution

18.4.1 Differential evolution

18.5 Particle swarms

18.5.1 Particle swarms

Chapter 19

Stochastic calculus

19.1 Introduction

19.1.1 Ito integrals

19.1.2 Stochastic differential equations

Chapter 20

Lossy compression

20.1 Lossy compression

Part VI

Exploratory data analysis

Chapter 21

Distance metrics and outliers

21.1 Measuring distance between vectors

21.1.1 L_p norms

L_p norms can be used to measure the distance between two metrics.

If we have data points v and w the distance is:

$$||v - w|| = (\sum_i |v_i - w_i|^p)^{\frac{1}{p}}$$

If $p = 2$ we have the Euclidian norm. If $p = 1$ we have the Manhatten norm.

21.1.2 Dot product

Given two vectors we can calculate:

$$\frac{a.b}{||a||||b||}$$

If the two vectors are identical, this is 1. If they are orthogonal this is 0. If they are opposite, this is -1.

21.1.3 Kernels

This is a generalisation of the dot product function, where we want to find similarity between two vectors.

If we have data points v and w the distance is:

$K(v, w)$

21.1.4 Mahalanobis distance

We have a point. How far away is this from the mean.

For a single dimension: number of standard deviations.

What about multidimensional data?

Could do sd for all distances, but correlations between variables. If two variables are highly correlated, it's not really twice as far.

We use this:

$D_M(\mathbf{x}) = \sqrt{(\mathbf{x} - \mu)^T S^{-1} (\mathbf{x} - \mu)}$

21.2 Measuring distance between matrices

21.2.1 Frobenius norm

If we have matrices A and A the distance is:

$||A - B|| = \sqrt{\sum_i \sum_j |a_{ij} - b_{ij}|^2}$

This is a Euclidian norm.

21.3 Measuring distance between time series

21.3.1 Dynamic time warping

We may want to examine the similarity between two sequences.

We want to match a sample from one sequence to a sample from the other sequence.

Simply matching at the same time point is naive, as samples may move at different speeds, or have offsets.

21.4 Finding neighbours

21.4.1 Nearest Neighbour Search (NSS)

21.4.2 Finding neighbours

Say we have a distance function and a sample. How can we identify the k-nearest neighbours?

We can find the distance for all points, sort this and take the top k observations.

21.4.3 k-Nearest Neighbour Search

Chapter 22

Association rules

22.1 Association rules

22.1.1 Association rules

The data

We have a transaction dataset, D.

This includes transactions of items in I.

Any subset of I is an itemset.

A subset of size k is a k-itemset.

Transactions are a k-itemset with a unique id, tid.

The set of all transactions is T.

A tidset is a subset of T.

Forming a lattice

We have a total order on the items.

An itemset ab is greater than a, for examplle.

The two points of the lattice are the nullset, and I.

Mappings

We have a mapping from I to T called t.

We have another mapping from T to I called i.

Frequency

We define the frequency of an itemset as the number of transactions it appears in.

We can write the frequency of A as $\sum A$.

22.1.2 Strong rules

Strong rules

We use frequent patterns to generate strong rules, R.

An example of a strong rule is $a \rightarrow b$.

We can look at this by comparing the support of a to $a \wedge b$.

$supp(A \rightarrow B) = P(A \wedge B)$

$conf(A \rightarrow B) = P(B|A)$

$$conf(A \rightarrow B) = \frac{P(A \wedge B)}{P(A)}$$

22.1.3 Support

Support

We define the support of an itemset as the proportion of transactions which contain the itemset.

$supp(A) = \frac{\sum A}{n}$.

We can also consider this as:

$supp(A) = P(A)$

22.1.4 Frequent patterns

Frequent patterns

A frequent itemset is one where the support is above a minimum.

We know that if an itemset is frequent, then all its subsets are also frequent.

We look for frequent patterns, F, between the items I.

An example of a frequent pattern is $\{a, b\}$.

22.1.5 Confidence

Confidence

The confidence of a frequent pattern is defined as:

$$conf(A \to B) = \frac{supp(A \wedge B)}{supp(A)}$$

22.1.6 Finding strong rules using the Apriori algorithm

Finding frequent patterns

We can use search algorithms to find frequent patterns. Starting at the empty set.

Apriori algorithm

Breadth first search to generate candidate set of itemsets with support above some value.

Start with a 1-itemset, and increase k once done.

Once we have found a frequent pattern, we can immediately identify other frequent patterns associated with it.

We can do this by looking at confidence, not support.

22.1.7 Interest

Interest

An alternative measure for finding rules is to use interest.

$$Interest(A \to B) = \frac{P(A \wedge B)}{P(A)P(B)}$$

$$Interest(A \to B) = \frac{supp(A \wedge B)}{P(supp(A))P(supp(B))}$$

If this is 1, then they are independent.

If this is greater than 1, they are positively dependent.

If this is less than 1, they are negatively dependent.

22.1.8 Quantitative association rules

Quantitative association rules

The search space is infinite in size. For example continuous age.

We choose intervals instead.

Chapter 23

Data cleaning

23.1 Precleaning

23.1.1 Precleaning data formats (float32 for nums)

23.1.2 Standardising file types

23.2 Joining data sets

23.2.1 Consistent variable naming

23.2.2 Concatenating data

23.2.3 Joining data

23.3 Cleaning categorial data

23.3.1 One Hot Encoding

23.4 Checking for consistency

23.4.1 Cross-consistency

23.5 Data shaping

23.5.1 Wide and long data

Introduction

23.5.2 Collapsing data

23.6 Cleaning text data

23.6.1 Bag-of-words

Generally remove punctuation

23.6.3 Feature hashing

23.7 Dropping variables

23.7.1 Sensitive information

23.7.2 Dropping unnecessary information, like names and derived variables

23.8 Dropping unnecessary information, like names and derived variables

23.8.1 Creating interactive terms

23.9 Deciling continuous data

Chapter 24

Summary statistics and visualisation for one variable

24.1 Basis statistics for a single variable

24.1.1 N

The is the size of the sample.

24.1.2 Sample range

Minimum

This is the smallest value in the sample.

Maximum

This is the largest value in the sample.

Range

This is the difference between the maximum and minimum.

Median

This is the value whereby 50% of the sample can be found below the value.

Percentiles

The xth percentile is the value by which $x\%$ of the values can be found below it.

Interquartile range

This is the differnence between the 25th percentile and the 75th percentile.

24.1.3 Sample mode

The is the most common value in the sample.

24.2 Sample moments

24.2.1 Sample mean

We previously defined the population mean is defined as $\mu = E[X]$.

The sample mean is defined as $\bar{x} = \frac{1}{n}\sum_i x_i$.

Centred mean

We can subtract the mean from each entry in the sample. This will leave a new mean of 0. This is convenient for many calculations.

24.2.2 Sample variance

We previously defined the population variance as $\sigma^2 = E[(X-\mu)^2]$.

We define the sample variance as $\sigma^2 = \frac{1}{n}\sum_i (x_i - \bar{x})^2$.

We can calculate this using matrices:

$M = X - \bar{x}$

$\sigma^2 = \frac{1}{n} M^T M.$

Centred variance

If $\bar{x} = 0$ then:
$\sigma^2 = \frac{1}{n} X^T X.$

24.3 Other

24.3.1 Standard error

24.3.2 Standard deviation

24.3.3 Sample size

24.4 Updating statistics

24.4.1 Updating the mean

$$\bar{x}_{n+1} = \frac{n\bar{x}_n + x_{n+1}}{n+1}$$

24.4.2 Updating the variance

If it is centred:
$\sigma_n^2 = \frac{1}{n} X_n^T X_n$
So:
$$\sigma_{n+1}^2 = \frac{n\sigma_n^2 + x_{n+1}^t x_{n+1}}{n+1}$$

24.5 Visualising a single continous variable

24.5.1 Box and whisker plots

24.5.2 Density plot

Chapter 25

Summary statistics and visualisation for multiple variables

25.1 Statistics for two variables

25.1.1 Sample covariance

We previously defined the population covariance as $\sigma_{XY} = E[(X - \mu_X)^T(Y - \mu_Y)]$.

We define the sample covariance as $\sigma_{XY} = \frac{1}{n}\sum_i (x_i - \bar{x})(y_i - \bar{y})$.

We can calculate this using matrices:

$M = X - \bar{x}$

$N = Y - \bar{y}$

$\sigma_{XY} = \frac{1}{n} M^T N.$

25.1.2 Sample correlation

$\rho_{XY} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y}$

25.1.3 Covariance matrix

If we have n variables we can have a $n \times n$ matrix Σ where:

$\Sigma_{ij} = \sigma_{ij} = \frac{1}{n}(X_i - \bar{x}_i)^T(X_j - \bar{x}_j)$

25.1.4 Centred covariance

If $\bar{x} = \bar{y} = 0$ then:

$\sigma_{XY} = \frac{1}{n}X^TY$

25.1.5 Correlation matrix

Here each entry is the correlation rather than the covariance.

25.2 Correlation coefficients

25.2.1 Pearson correlation coefficient

The Pearson correlation coefficient is definited as the covariance normalised by the individual variances.

It is between -1 (total negative linear correlation), 0 (no linear correlation) and 1 (total negative linear correlation).

$p_{X,Y} = \frac{cov(X,Y)}{\sigma_X\sigma_Y}$

25.2.2 Spearman rank correlation

For each of 2 variables we create a ranking of them.

From X and Y we then have R_X and R_Y.

We then calculate the Pearson correlation coefficient between the rankings.

$r_S = \frac{cov(R_X,R_Y)}{\sigma_{R_X}\sigma_{R_Y}}$

25.2.3 Kendall rank correlation

Concordant and discordant pairs

$$\tau = \frac{n_{concordant} - n_{discordant}}{\binom{n}{2}}$$

25.2.4 General correlation coefficient

25.3 Updating statistics

25.3.1 Updating the covariance

If it is centred:

$$\sigma^n_{XY} = \frac{1}{n} X^T_n Y_n$$

So:

$$\sigma^{n+1}_{XY} = \frac{n\sigma^n_{XY} + x^t_{n+1} y_{n+1}}{n+1}$$

25.4 Visualising multiple continuous variables

25.4.1 Time series

25.4.2 Scatter plots (with size as variable)

25.4.3 Q-Q plots

Plot quartiles of variables against each other.

25.5 Visualising a single class variable

25.5.1 Bar and column charts

25.5.2 Pie charts

25.6 Visualising multiple class variables

25.6.1 Stacked bar and column charts

25.7 Visualising class and continous variables

25.7.1 Multiple box and whiskers

25.7.2 Scatter plots with colour

25.8 Visualising geographic data

25.9 Visualising time series

25.9.1 Heat maps

25.9.2 Sparklines

Chapter 26

Testing population means with Z-tests and T-tests

26.1 Z-test

26.1.1 Z-test for variable significance

The standard score

We may want to see how different a mean statistic is from a specific value.

The standard score allows us to measure this, by taking this distance and standardising by the standard deviation.

$$z = \frac{\bar{x} - x_0}{\sigma}$$

This requires us to know the standard deviation, which is in general not known.

If the sample size is large, we know this converges to the normal distribution through the central limit theorem.

The Z-test

We can see how likely our statistic was to be produced if it was drawn from a normal distribution with mean x_0 and standard deviation s_0.

P-values

This is the chance of the statistic being produced by chance.

26.2 t-test

26.2.1 T-test for variable significance

T-statistic

In practice we don't know the population standard deviation and so must estimate it instead.

We use the standard deviation on the sample.

$$t = \frac{\bar{x} - x_0}{s_0}$$

Student's t-distribution

As we have used the sample standard deviation we have lost a degree of freedom, and can no longer model the variable as a normal distribution, as we did for the z-statistic.

We now have a distribution with an addition parameter, the number of degrees of freedom.

The number of degrees of freedom is $n - 1$.

As the sample size tends towards infinity, the distribution tends towards the normal distribution.

Student's t-test

Confidence interval

26.2.2 Welch's t-test

Alternative to student.

Part VII

Estimating generative probability distributions

Chapter 27

Non-parametric estimation of probability distributions

27.1 Histograms

27.1.1 Histograms

27.2 Kernels

27.2.1 Kernel density estimation

27.2.2 Smoothing kernel estimation

Smoothed kernels

We have $K(x - x_i)$

We can smooth this to:

$K_h(x - x_i) = \frac{1}{h} K(\frac{x - x_i}{h})$

Where $h > 0$ is the smoothing bandwidth.

$f(x) = \frac{1}{n} \sum_{i=1}^{n} K_h(x - x_i)$

Chapter 28

Bayesian parameter estimation

28.1 Bayesian parameter estimation

28.1.1 Bayesian parameter estimation

Bayes rule

We want to generate the probability distribution of θ given the evidence X.

We can transform this using Bayes rule.

$$P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)}$$

Here we have:

- Our prior - $P(\theta)$
- Our likelihood function - $P(X|\theta)$
- Our posterior - $P(\theta|X)$

Normal priors and posteriors

If our prior is a normal distribution then:

$$P(\theta) = \frac{1}{\sqrt{(2\pi)^n|\Sigma_0|}} e^{-\frac{1}{2}(x-\mu)^T\Sigma_0^{-1}(x-\mu)}$$

Similarly, if our likelihood function $P(X|\theta)$ is a normal distriubtion then:

$$P(X|\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

We can now plug these into Bayes rule:

$$P(\theta|X) = \frac{1}{P(X)} \frac{1}{\sqrt{2\pi\sigma_0^2}} e^{-\frac{(\theta-\mu_0)^2}{2\sigma_0^2}} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$P(\theta|X) \propto e^{-\frac{1}{2}[\frac{(\theta-\mu_0)^2}{\sigma_0^2} + \frac{(x-\mu)^2}{\sigma^2}]}$$

We can then set this an a new Gaussian:

$$P(\theta|X) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}^{\frac{1}{2}}} e^{-\frac{1}{2}[\frac{(\theta-\mu_0)^2}{\sigma_0^2} + \frac{(x-\mu)^2}{\sigma^2}]}$$

28.1.2 Empirical Bayes

Bayes rule

We can calculate the posterior probability for θ, but we need a prior $P(\theta)$.

$$P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)}$$

Empirical Bayes

With empirical Bayes we get our prior from the data.

We have $P(X|\theta)$

And $P(\theta|\rho)$

We observe X and want to estimate θ.

$$P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)} = \frac{P(X|\theta)}{P(X)} \int P(\theta|\rho)P(\rho)d\rho$$

28.1.3 Prior and posterior predictive distributions

Prior predictive distribution

Our prior predictive distribution for X depends on our prior for θ.

$P(\mathbf{x}) = \int_\Theta P(\mathbf{x}|\theta)P(\theta)d\theta$

Posterior predictive distribution

Once we have calculated $P(\theta|X)$, we can calculate a posterior probability distribution for X.

$P(\mathbf{x}|\mathbf{X}) = \int_\Theta P(\mathbf{x}|\theta)P(\theta|\mathbf{X})d\theta$

28.1.4 Bayesian risk

Risk and Bayes risk.

Chapter 29

Point estimates of probability distributions

29.1 Point estimates for parameters

29.1.1 Estimators

When we take statistics we are often concerned with inferring properties of the underlying probability function.

As the properties of the probability distribution function affect the chance of observing the sample, we can analyse samples to infer properties of the underlying distribution.

There are many properties would could be interested in. This includes moments and parameters of a specific probability distribution function.

An estimator is a statistic which is our estimate of one of these values.

Emphasise that statistics and estimators are different things. A statistic may be terrible estimator, but be useful for other purposes.

29.1.2 Sufficient statistics

We can make estimates of a population parameter using statistics from the same.

A statistic is sufficient if it contains all the information needed to estimate the parameter.

We can describe the role of a parameter as:

$P(x|\theta, t)$

t is a sufficient statistic for θ if:

$P(x|t) = P(x|\theta, t)$

29.2 Properties of point estimators

29.2.1 Estimator error and bias

Error of an estimator

The error of an estimator is the difference between it and the actual parameter.

$Error_\theta[\hat{\theta}] = \hat{\theta} - \theta$

Bias of an estimator

The bias of an estimator is the expected error.

$Bias_\theta[\hat{\theta}] := E_\theta[\hat{\theta} - \theta]$

$Bias_\theta[\hat{\theta}] := E_\theta[\hat{\theta}] - \theta$

29.2.2 Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) of an estimator

Mean squared error

Mean squared error

$MSE = E[(\hat{\theta} - \theta)^2] = E[((\hat{\theta} - E[\hat{\theta}]) + (E[\hat{\theta}] - \theta))^2]$

$MSE = E[(\hat{\theta} - \theta)^2] = E[(\hat{\theta} - E[\hat{\theta}])^2 + (E[\hat{\theta}] - \theta)^2 + 2(E[\hat{\theta}] - \theta)(\hat{\theta} - E[\hat{\theta}])]$

$MSE = E[(\hat{\theta} - \theta)^2] = E[(\hat{\theta} - E[\hat{\theta}])^2] + E[(E[\hat{\theta}] - \theta)^2] + E[2(E[\hat{\theta}] - \theta)(\hat{\theta} - E[\hat{\theta}])]$

$MSE = E[(\hat{\theta} - \theta)^2] = Var(\hat{\theta}) + (E[\hat{\theta}] - \theta)^2 + 2(E[\hat{\theta}] - \theta)E[\hat{\theta} - E[\hat{\theta}]]$

$MSE = E[(\hat{\theta} - \theta)^2] = Var(\hat{\theta}) + Bias(\hat{\theta})^2$

Root Mean Square Error (RMSE)

This is the square root of the MSE.

It is also called the Root Mean Square Deviation (RMSD)

29.2.3 Asymptotic properties of estimators

29.2.4 Consistency and efficiency of estimators

Consistency

A statistic $\hat{\theta}$ is a consistent estimator for θ if its error tends to 0.

That is:

$\hat{\theta} \rightarrow^p \theta$

We can show that an estimator is consistent if we can write:

$\hat{\theta} - \theta$ as a function of n, causing it to tend to 0.

Efficiency

Efficiency measures the speed at which a consistent estimator tends towards the true value.

The speed of this convergence is the efficiency. could be fairly efficient plus biased too p Measured as:

$$e(\hat{\theta}) = \frac{\frac{1}{I(\theta)}}{Var(\hat{\theta})}$$

If an estimator as an efficiency of 1 and is unbiased, it is efficient.

Relative efficiency

We can measure the relative efficiency of two consistent estimators:

The relative efficiency is the variance of the first estimator, divided by the variance of the second.

Root-n estimators

An estimator is root-n consistent if it is consistent and its variance is:

$O(\frac{1}{n})$

n^δ-convergent

A consistent estimator is n^δ-consistent if its variance is:
$O(\frac{1}{n^{2\delta}})$

29.2.5 Cramr-Rao lower bound

For an unbiased estimator, the variance cannot be below the Cramer-Rao lower bound.

$$Var(\hat{\theta}) \geq \frac{1}{I(\theta)}$$

Where $I(\theta)$ is the Fisher information.

We can prove this.

We have the score:

$$V = \frac{\delta}{\delta\theta} \ln f(X, \theta)$$

$$V = \frac{1}{f(X,\theta)} \frac{\delta}{\delta\theta} f(X, \theta)$$

The expectation of the score is 0:

$$E[V] = E[\frac{1}{f(X,\theta)} \frac{\delta}{\delta\theta} f(X, \theta)]$$

$$E[V] = \int \frac{1}{f(X,\theta)} \frac{\delta}{\delta\theta} f(X, \theta) dx$$

29.2.6 Bias-Variance trade-off

Bias-variance trade-off. if we care about $E[(y - xt)^2]$ then we may not want an unbiased estimator. by adding some bias we could reduce the variance a lot.

29.3 Sort

29.3.1 Testing estimators

Assessing estimators of parametric models: do monte carlo simulations

29.3.2 Loss

loss functions for point estimates. point estimate confidence interval h3

29.3.3 Estimator properties

best asymptotically normal (BAN) estimators AKA consistently asymptotically normal efficience (CANE)

these are root n consistent!

29.3.4 Feasible and infeasible estimators

Feasible uses known terms. Infeasible uses those that aren't

Eg Ω is infeasible, unless we assume its form, making it feasible.

29.3.5 Bias etc

pages: + Cramer rao + Minimum-Variance Unbiased Estimators (MVUE)

Unbiased estimators for some kernel value. Can use used to estimate population moments.

29.3.6 Rao-Blackwell theorem

29.3.7 One step and k-step estimators

in cramer rao stuff?

29.3.8 Delta method

in bias section?

We can consider X_n to be a sequence. We are interest in asymptotic properties of this sequence.

29.3.9 Fat tails

section on fat tails + can't estimate pop mean from sample mean + method of moments requires non-fat tails + correlation/covariance with fat tails.

Chapter 30

Maximum Likelihood Estimation (MLE)

30.1 Maximising the likelihood function

30.1.1 Maximising the likelihood function

We have a likelihood function of the data.

$L(\theta; X) = P(X|\theta)$

We choose values for θ which maximise the likelihood function.

$argmax_\theta P(X|\theta)$

That is, for which values of θ was the observation we saw most likely?

This is a mode estimate.

30.1.2 IID

$L(\theta; X) = \prod_i P(x_i|\theta)$

30.1.3 Logarithms

We can take logarithms, which preserve stationary points. As logarithms are defined on all values above 0, and all probabilities are also above zero (or zero), this preserves solutions.

The non-zero stationary points of:

$\ln L(\theta; X) = \ln \prod_i P(x_i|\theta)$

$\ln L(\theta; X) = \sum_i \ln P(x_i|\theta)$

30.1.4 Example: Coin flip

Lets take our simple example about coins. Heads and tails are the only options, so $P(H) + P(T) = 1$.

$P(H|\theta) = \theta$

$P(T|\theta) = 1 - \theta$

$\ln L(\theta; X) = \sum_i \ln P(x_i|\theta)$

If we had 5 heads and 5 tails we would have:

$\ln L(\theta; X) = 5 \ln(\theta) + 5 \ln(1 - \theta)$

So $P(H) = \frac{1}{2}$ is the value which makes our observation most likely.

30.2 Properties of the MLE estimator

30.2.1 Asymptotic normality of the MLE

30.3 Results for specific distributions

30.3.1 MLE of the Gaussian distribution

The parameters are the population means and covariance matrix.

The MLE estimator for the mean is the sample mean.

The MLE estimator for the covariance matrix is the unadjusted sample covariance.

30.3.2 MLE of the Poisson distribution

30.3.3 MLE of the Bernoulli and binomial distributions

30.4 Other

30.4.1 Restricted Maximum Likelihood

We can partition out Likelihood functions, and include a part only with variance.

30.4.2 Targeted Maximum Likelihood Estimation

30.4.3 Scores

Existing score: rename Maximum Likelihood score

MLE bad if true theta not at where score is 0

Eg if one sided tails, true theta is not at MLE condition.

Can we find other scores?

30.4.4 Orthogonality

Score of one parameter depends on other parameters

If we misestimate one, then estimate another, will be bad answer

We want the score not to change around bad estimates

We want nuisance parameter bias not to affect score

separate page for orthogonality for sets of parameters. eg nuisance; of interest

Chapter 31

Maximum A-Priori (MAP) estimation

31.1 Maximum A-Priori Estimation

31.1.1 Maximum A-Priori (MAP) estimation

Mode estimate

$Argmax_\theta p(\theta|X)$

Using Bayes theorem:

$$P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)}$$

So:

$$P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)}$$

$$Argmax_\theta p(\theta|X) = Argmax_\theta \frac{p(X|\theta)P(\theta)}{P(X)}$$

The denominator isn't affected so:

$Argmax_\theta p(\theta|X) = Argmax_\theta p(X|\theta)P(\theta)$

If $P(\theta)$ is a constant then this is the same as the MLE estimator.

Other

$Argmax_\theta p(\theta|X)$

Mode estimate

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{p(X)}$$

$$Argmax_\theta \frac{p(X|\theta)p(\theta)}{p(X)}$$

θ doesn't change denominator so can instead use:

$Argmax_\theta p(X|\theta)p(\theta)$

It is the same as maximum likelihood estimator if $p(\theta)$ is a constant.

31.1.2 MAP of the Gaussian distribution

Chapter 32

The Method Of Moments (MOM)

32.1 Method of Moments

32.1.1 Method of moments

Introduction

If we have k parameters to estimate, we can solve this if we have k equations.

We generate these

First, we link each first k moments to functions of the parameters.

Then we replace the momenets with sample estimates.

Estimation

The moments of this population distribution are:

$\mu_i = E[X^i] = g_i(\theta_1, ..., \theta_k)$

We have a sample.

$X = [X_1, ..., X_n]$

We now define the method of moments estimator

$\hat{\mu}_i = \frac{1}{n} \sum_{j=1}^{n} x_j^i$

Chapter 33

The Generalised Method of Moments (GMM)

33.1 Generalised Method of Moments (GMM)

33.1.1 Difference from Method Of Moments (MOM)

More conditions than data.

33.1.2 Generalised Method of Moments (GMM)

We have a function on the output and a parameter:

$g(y, \theta)$

A moment condition is that the expectation of such a function is 0.

$m(\theta) = E[g(y, \theta)] = 0$

To do GMM, we estimate this using:

$\hat{m}(\theta) = \frac{1}{n} \sum_i g(y_i, \theta)$

We define:

$\Omega = E[g(y, \theta) g(y, \theta)^T]$

$G = E[\Delta_\theta g(y, \theta)]$

And then minimise the norm:

$||\hat{m}(\theta)||^2_W = \hat{m}(\theta)^T W \hat{m}(\theta)$

Where W is a positive definite matrix for the norm.

Ω^{-1} is most efficient. But we don't know this. It depends on θ.

We can estimate it if IID:

$\hat{W}(\hat{\theta}) = (\frac{1}{n}\sum_i g(y,\hat{\theta})g(y,\hat{\theta})^T)^{-1}$

33.1.3 Two-step feasible GMM

Estimate using $\mathbf{W} = \mathbf{I}$

Consistent, but not efficient.

33.1.4 Moment conditions

OLS:

$E[x(y - x\theta)] = 0$

WLS

$E[x(y - x\theta)/\sigma^(x)] = 0$

IV

$E[z(y - x\theta)] = 0$

MLE

$E[\Delta_\theta \ln f(x,\theta)] = 0$

33.1.5 New GMM

$m(\theta_0) = E[g(\mathbf{x}_i, \theta_0]$

We replace this with sample moment

$\hat{m}(\theta) = \frac{1}{n}\sum_i g(\mathbf{x}_i, \theta)$

We have the "score"

$\nabla_\theta g(\mathbf{x}_i, \theta_0)$

Information

$G = E[\nabla_\theta g(\mathbf{x}_i, \theta_0)]$

Variance-covariance loss matrix

$\Omega = E[g(\mathbf{x}_i, \theta_0)g(\mathbf{x}_i, \theta_0)^T]$

We want to minimise moment loss

$||\hat{m}(\theta)||^2_W = \hat{m}(\theta)^T W \hat{m}(\theta)$

$\hat{\theta} = argmin_\theta(\frac{1}{n}\sum_i g(\mathbf{x}_i, \theta))^T \hat{W}(\frac{1}{n}\sum_i g(\mathbf{x}_i, \theta))$

33.1.6 Asymptotic

CLT means normal.

They are consistent IF moment condition is true.

There is an explicit formula for variance.

$\sqrt{n}(\hat{\theta} - \theta_0) \rightarrow^d N[0, (G^T W G)^{-1} G^T W \Omega W^T G (G^T W^T G)^{-1}]$

If we choose $W \propto \Omega^{-1}$ then:

$\sqrt{n}(\hat{\theta} - \theta_0) \rightarrow^d N[0, (G^T \Omega^{-1} G)^{-1}]$

Problem: we need to estimate Ω and G.

Ω: estimate from sample. allows us to choose estimator, but still leaves variance unidentified.

Do the above from OLS? This is where robust etc stuff comes from

If it is specified. Moment conditions are equal to the number of moments, then W doesn't matter. This is normal Method of Moments.

Estimating the weighting matrix

33.1.7 Iterated GMM

33.1.8 Moment-covariance matrix

33.1.9 Bias and variance of the GMM estimator

page on Bias and variance of the GMM estimator (cluster assumption should be part of moment condition?) part of later calculation of weighting?

Can do robust, hac, clustering as part of GMM too.

Chapter 34

M-estimators

34.1 M-estimators

34.1.1 Introduction

page setting out linear stuff to come

OLS, generalised linea rmodels etc are m-estimators, as are gmm

h3 on parametric

With maximum likelihood estimation we maximise a function.

We could choose other functions to maximise or minimise.

$\sum_i f(x_i, \theta)$

If $f(x_i, \theta)$ is differentiable wrt to θ this can be solved by finding the stationay point.

This is a ϕ type.

Otherwise it is a ρ type.

page on influence funcitons there

Generalisation of MLE.

$m_\theta = m_\theta(x, \theta)$

Z-estimator is where this is met, through diff

$\frac{\delta}{\delta\theta} m_\theta = z_\theta(\theta, x) = 0$

M-estimator for mean

$m_\theta(\theta) = -(x - \theta)^2$

$z_\theta(\theta) = x - \theta$

Chapter 35

Estimating population moments

35.1 Plug-in estimators

35.1.1 Estimating the population mean

35.1.2 Estimating the population variance

35.1.3 Estimating the population standard deviation

Chapter 36

Testing generative parameter estimates with Z-tests and T-tests

Chapter 37

Choosing parametric probability distributions

37.1 Choosing the form of a model

37.1.1 Sample sizes

If you're modelling house prices using just size, getting a large sample size won't help too much

Can improve low bias models*** Sample size

Is data size an issue? can artificially restrict training data size and then evaluate error

Training: + zero error for low m + increases error as m increases, as degrees of freedom/m falls

cv: + error decreases as data set increases, more accurate theta The two curves converge towards each other for v large m

When are large datasets useful?

When all features available:

Predicting house price using just size, won't benefit from more data...

If choosing correct word in sentence (to, too, two), more helpful

If human expert can do it, then more data probably helpful

Expert realtor probably couldn't do much with just size, but speaker could answer other q

Could expert do it?

Low bias algorithms do well with more data

More data good if large number of parameters, or lot of hidden units.

37.2 Choosing the form of a model

37.2.1 Overfitting

Role of lambda: high makes impact of more variables lower =¿ high bias

Low makes impacts of more variables strong =¿ high variance

Can trade off using cut off. only make positive if above 0.7

How to use? difficult, as lambda within cost!

Can do similarly to d:

Run for a range of lambda (eg 0, 0.01, 0.02, 0.04, 0.08: 10), then pick from cross validation set

Low lambda always has low cost for training set, but not for cv set..

Regularisation: add to error term the size of the term. penalised large parameters

May not fit outside sample

High bias: eg house prices and size. linear would have high bias for out of scope sample (underfitting)

High variance: making polynomial passing through all data (overfitting)

Can reduce overfitting by reducing features either manaually or using models

OR regularisation: keep all features, but reduce magnitude of theta

37.2.2 Regularisation

Make cost function include size of θ^2 values

$\min \frac{1}{2m}[\sum(h(x)-y)^2 + 1000\theta 3^2 + 1000\theta 4^2]$

or more broadly:

$\min \frac{1}{2m}[\sum ..... + \lambda \sum \theta j^2]$

Tend to not include theta 0 as convention, no regularisation

Update for linear regression is

$\theta j = \theta j - \alpha(\frac{1}{m}) * sum(h(x) - y)xj + (\lambda/m\theta j)$

$\theta j = \theta j(1 - \alpha\lambda/m) - alpha(1/m) * \sum(h(x) - y)xj$

This is the same as before, but theta j updates from a smaller θ j each time.

Normal equation needs a change

$(X'X)^{-}1X'y = \theta''$

Now is

$(X'X + \lambda I)^{-}1Xy'$

although for theta 0, lambda zero, so indentiy matrix, but first element 0

REGULARISATION FOR REGULARISATION

add to end of $J(\theta)$:

$\frac{\lambda}{2m} \sum \theta j^2$

update for θj $j > 0$: is a as linear regression, but $h(x)$ is a different function

37.3 Choosing the form of a model

37.3.1 AIC, AICc, Bayes factor, BIC

37.4 Choosing the form of a model

37.5 Kullback-Leibler divergence

Bayesian inference means we have full distribution of $p(w)$, not just moments of a specific point estimate

37.5.1 Cross entropy:

$H(P, Q) = E_P(I(Q))$

So for a discrete distribution this is:

$H(P, Q) = -\sum_x P(x) \log Q(x)$

Q is prior

P is posterior

37.5.2 Kullback-Leibler divergence

When we move from a prior to a posterior distribution, the entropy of the probability distribution changes.

$D_{KL}(P||Q) = H(P,Q) - H(P)$

KL divergence is also called the information gain.

37.5.3 Gibb's inequality

$D_{KL}(P||Q) \geq 0$

37.6 Bayesian model selection

37.7 Introduction

Part VIII

Estimating latent variable models

Chapter 38

Latent variable models

38.1 Latent variable models

38.1.1 Latent class analysis

38.2 The Expectation-Maximisation (EM) algorithm

38.2.1 The Expectation-Maximisation algorithm

Expectation-Maximisation algorithm

This is used to learn the parameters for a Gaussian Mixture Model

We cannot simply maximise the likelihood function, because this cannot be specified for a latent model.

The log likelihood function normally is:

$L(\theta; X) = p(X|\theta)$

With hidden variables it is:

$L(\theta; X, Z) = p(X|\theta) = \int p(X, Z|\theta) dZ$

1: Expectation step

We consider the expected log likelihood. We call this

$E[\log L(\theta; X, Z)]$

2: Maximisation step

38.3 Stochastic Expectation-Maximisation

Part IX

Unsupervised machine learning

Chapter 39

Dimensionality reduction with Principal Component Analysis (PCA)

39.1 Dimensionality reduction

39.1.1 Classical principal component analysis

Introduction

Principal component analysis takes a dataset X with m variables and returns a principal component matrix A with size $m \times k$.

Each new dimension is a linear function of the existing data. $Z = XA$.

Each dimension in uncorrelated, and ordered, in order of descending explanation of variability.

The problem of principal component analysis is to find these weightings A.

Classical PCA

We take the first k eigenvectors of the covariance matrix, ordered by eigenvalue.

Getting the eigenvectors using SVD

We can decompose $X = U\Sigma A^T$.

We can take the eigenvectors from A.

Choosing the number of dimension

We can choose k such that a certain percentage of the variance is retained.

39.1.2 Robust principal component analysis

Robust PCA

Robust PCA can be used to deal with corrupted data, such as corrupted image data.

Rather than data X we have $M = L_0 + S_0$ where L_0 is what we want to recover (and is low rank), and S_0 is noise (and sparce).

In video footage, L_0 can correspond to the background, while S_0 corresponds to movement.

Chapter 40

K-means and k-mediods clustering

40.1 Clustering

40.1.1 Evaluating clusterings

Davies-Bouldin index

The Davies-Bouldin index is a method for evaluating clustering algorithms, such as k-means.

It examines the distance between centroids, and the tightness of centroids.

40.1.2 k-means clustering

Introduction

K-means clustering is the most widely used unsupervised model.

In k-means clustering we identify k centroids in the feature space. We then calculate the distance from each data point to each of the centroids, and allocate the data point to the nearest centroid.

This requires a method for calculating the location of the centroids.

Identifying the centroids

We apply an iterative approach to identifying the centroids.

We first initialise by assigning centroids randomly to existing data points.

We then iteratively perform the following:

- Calculate the distances between each data point and each centroid.
- Assign each data point to the closed centroid.
- Update each centroid location to the mean of the data points allocated to it.

Calculating distances

This method requires us to calculate the distance between two points in the feature space.

For k-means we use the Euclidian distance.

Potential issues

It is possibile for a centroid to have no data assigned to it. If this happens we can eliminate the cluster, or reassign some data points.

The algorithm may only arrive at a local minima. In order to maximise the chance of an effective clustering, we can do k-means under different initialisations of the centroids in order to minimise risk of bad local optima.

Choosing k

If the points in each cluster follow a normal distribution, that's a good sign. This can be tested with Anderson-Darling.

If it's not normal, we can split the cluster into 2.

Using clusting as part of data analysis

We can choose k if output is being used in later data analysis (eg type assignment, complaint level or something)

40.1.3 k-medoids

Introduction

k-mediods is similar to k-means clustering, with two key differences:

- Centroids are now always located on data points, rather than floating freely.
- We mimimise l_1 distance, rathern than l_2.

Partitioning Around Medoids (PAM) algorithm

This is the most common approach for k-medoids.

We initialise randomly, as we do for k-means.

We then iterate the following:

- Calculate the loss for the current allocation
- For each medoid, see if swapping allocation with another (non-medoid) data point decreases the cost.
- If it does, make the swap.

Part X

Estimating discriminative probability distributions

Chapter 41

Bayesian parameter estimation of discriminative models

41.1 Introduction

41.1.1 Generative and discriminative models

Recap

For parametric models without dependent variables we have a form:

$P(y|\theta)$

And we have various ways of estimating θ.

We can write this as a likelihood function:

$L(\theta; y) = P(y|\theta)$

Discriminative models

In discriminative models we learn:

$P(y|X, \theta)$

Which we can write as a likelihood function:

$L(\theta; y, X) = P(y|X, \theta)$

Generative models

In generative models we learn:

$P(y, X|\theta)$

Which we can write as a likelihood function:

$L(\theta; y, X) = P(y, X|\theta)$

We can use the generative model to calculate dependent probabilities.

$$P(y|X, \theta) = \frac{P(y, X|\theta)P(\theta)}{P(X, \theta)}$$

$$P(y|X, \theta) = \frac{P(y, X|\theta)}{P(X|\theta)}$$

41.2 Bayesian parameter estimation

41.2.1 Bayesian parameter estimation for dependent models

Recap

For non-dependent models we had:

$$P(\theta|y) = \frac{P(y, \theta)}{P(y)}$$

$$P(\theta|y) = \frac{P(y|\theta)P(\theta)}{P(y)}$$

The bottom bit is a normalisation factor, and so we can use:

$P(\theta|y) \propto P(y|\theta)P(\theta)$

We have here:

- Our prior - $P(\theta)$
- Our posterior - $P(\theta|y)$
- Our likelihood function - $P(y|\theta)$

Bayesian regression for generative models

We know:

$$P(\theta|y, X) = \frac{P(y, \theta, X)}{P(y, X)}$$

$$P(\theta|y,X) = \frac{P(y,X|\theta)P(\theta)}{P(y,X)}$$

The bottom bit is a normalisation factor, and so we can use:

$P(\theta|y,X) \propto P(y,X|\theta)P(\theta)$

We have here:

- Our prior - $P(\theta)$
- Our posterior - $P(\theta|y,X)$
- Our likelihood function - $P(y,X|\theta)$

Bayesian regression for discriminative models

We know:

$$P(\theta|y,X) = \frac{P(y,\theta,X)}{P(y,X)}$$

$$P(\theta|y,X) = \frac{P(y|\theta,X)P(\theta,X)}{P(y,X)}$$

$$P(\theta|y,X) = \frac{P(y|\theta,X)P(\theta)P(X|\theta)}{P(y,X)}$$

We assume $P(X|\theta) = X$ and so:

$$P(\theta|y,X) = \frac{P(y|\theta,X)P(\theta)P(X)}{P(y,X)}$$

The bottom bit is a normalisation factor, and so we can use:

$P(\theta|y,X) \propto P(y|X,\theta)P(\theta)$

We have here:

- Our prior - $P(\theta)$
- Our posterior - $P(\theta|y,X)$
- Our likelihood function - $P(y|X,\theta)$

41.2.2 Prior and posterior predictive distributions for dependent variables

Prior predictive distribution

Our prior predictive distribution for $P(y|X)$ depends on our prior for θ.

$P(y|X) = \int_\Theta P(\mathbf{y}|X,\theta)P(\theta)d\theta$

Posterior predictive distribution

Once we have calculated $P(\theta|\mathbf{y}, \mathbf{X})$, we can calculate a posterior probability distribution for $P(y|X)$.

$P(y|\mathbf{x}, \mathbf{y}, \mathbf{X}) = \int_\Theta P(y|\mathbf{x}, \theta)P(\theta|\mathbf{y}, \mathbf{X})d\theta$

Chapter 42

Point variable estimates for discriminative models

42.1 Predictions and residuals

42.1.1 Predictions

Our data $(\mathbf{y}, \mathbf{X})$ is divided into $(y_i, \mathbf{x}_i)$.

We create a function $\hat{y}_i = f(\mathbf{x}_i)$.

The best predictor of y given x is:

$g(X) = E[Y|X]$

The goal of regression is to find an approximation of this function.

42.1.2 Residuals

$\epsilon_i = y_i - \hat{y}_i$

42.1.3 Residual sum of squares (RSS)

$RSS = \sum_i \epsilon_i^2$

$RSS = \sum_i (y_i - \hat{y}_i)^2$

42.1.4 Explained sum of squares (ESS)

$ESS = \sum_i (\bar{y} - \hat{y}_i)^2$

42.1.5 Total sum of squares (TSS)

$TSS = \sum_i (y_i - \bar{y})^2$

42.1.6 Relationship between prediction and probability distribution

$P(y|X, \theta)$

$\hat{y} = f(\mathbf{x})$

Through integration?

$E[y] = \int P(y|X, \theta) dy$

42.1.7 Coefficient of determination (R^2)

$R^2 = 1 - \frac{RSS}{TSS}$

42.2 Classification

42.2.1 Binary classification

Classification models are a type of regression model, where y is discrete rather than continuous.

So we want to find a mapping from a vector X to probabilities across discrete y values.

A classifier takes X and returns a vector.

For a classifier we have K classes.

42.2.2 Classification

Confusion matrix. true positve, false positive, false negative, true negative

Can use this to get

Accuracy: percentage correct

Precision: percentage of positive predictions which are correct

Recall (sensitivity): percentage of poitive cases that were predicted as positive

Specificity: percentage of negative cases preicated as negative

42.2.3 Multiclass classification

Multiclass classification

What if can be email for work, friends, family, hobby?

42.2.4 Confusion matrix

Include error types here

42.2.5 Hard and soft classifiers

A hard classifier can return a sparce vector with 1 in the relevant classification.

A soft classifier returns probabilities for each entry in the vector.

The vector represents $P(Y = k|X = x)$

42.2.6 Transforming soft classifiers into hard classifiers

We can use a cutoff.

If there are more than two classes we can choose the one with the highest score.

42.3 Loss functions for point predictions

42.3.1 Minimum Mean Square Error (MMSE)

Mean estimate.

Can do for a parameter, or for a predicted estimate for y.

Linear models

MLE is same as y^2 loss

MAP is same as y^2 loss with regularisation

42.3.2 Loss functions for soft classifiers

Hinge loss

Brier score

42.3.3 Loss functions for hard classifiers

Don't want answers outside 0 and 1.

F score

F1 score

F_1 score: $\frac{2PR}{(P+R)}$

may not just care about accuracy, eg breast cancer screening

high accurancy can result from v basic model (ie all died on titanic)

Receiver Operating Characteristic (ROC) Area Under Curve (AUC)

42.4 Other

42.4.1 Estimating other priors

Estimating $P(k|T)$ - Which variables we split by, given the tree size

Estimating $P(r|T,k)$ - The cutoff, given the tree size and the variables we are splitting by

Estimating $P(\theta|T,k,r)$

42.4.2 Maximum Likelihood Estimation (MLE) for generative and discriminative models

42.4.3 Maximum A-Priori estimation (MAP) for generative models

Bayesian regression for generative models

We know:

$$P(\theta|y,X) = \frac{P(y,\theta,X)}{P(y,X)}$$

$$P(\theta|y,X) = \frac{P(y,X|\theta)P(\theta)}{P(y,X)}$$

The bottom bit is a normalisation factor, and so we can use:

$P(\theta|y,X) \propto P(y,X|\theta)P(\theta)$

We have here:

- Our prior - $P(\theta)$
- Our posterior - $P(\theta|y,X)$
- Our likelihood function - $P(y,X|\theta)$

42.4.4 Bayesian classifier

Classification risk

We can measure the risk of a classifier. This is the chance of misclassification.

$R(C) = P(C(X) \neq Y)$

The Bayesian classifier

This is the classifer $C(X)$ which minimises the chance of misclassification.

It takes the output of the soft classifier and chooses the one with the highest chance.

Chapter 43

Using F-tests to compare regression models

43.1 Hypothesis testing

43.1.1 Power of tests

43.1.2 Type I and type II errors

43.1.3 Sensitivity tests

43.2 F-test

43.2.1 F test for equal population means

43.2.2 F test for additional variables

43.3 Other

43.3.1 Cohen's d

Chapter 44

Test sets and validation sets

44.1 Test sets

44.1.1 Overfitting

Overfitting is a risk. Instead we split to test, train. risk of using too many features. more features always improve training score, not necessarily test score.

As model gets more complex, both test and train do better. however at some point, test stops doing better, overfitting

Structural risk minimisation can address this trade off. use test and training sets. train model on train, rate it on test

Structural minimisation curve has accuracy of boths sets over complexity

To avoid overfitting:

+ reduce number of features + do a model selection + use regularisation + do cross validation

Can choose other model parameters

How to evalute model?

44.1.2 K-fold cross-validation

Can do k-fold cross validation. given algo A and dataset D, divide D into k equal sized subsets

For each subset, train the model on all other subsets and test on the other subset. average error between folds

44.2 Splitting

44.2.1 K-folds

The problem of different sample sizes (sample size for validations sets is lower, different hyper parameters could be more appropriate)

44.2.2 Leave-One-Out

44.3 Hyperparameters

44.3.1 Learning rate, batching and momentum

44.4 Search methods

44.4.1 Grid-search

44.5 Validation sets

44.5.1 Validation sets

which features? remove, add?

change lambda,regularisation

change polynomial features

Part XI

Supervised linear regression

Chapter 45

Ordinary Least Squares for prediction

45.1 Constructing a linear model

45.1.1 Defining linear models

Defining

One option for $f(X)$ is a linear model.

$f(X_i) = \hat{Y}_i = \beta_0 + \sum_{j=1}^{p} \beta_i X_{ij}$

The values for β are the regression coefficients.

So we have:

$Y_i = \beta_0 + \sum_{j=1}^{p} \beta_i X_{ij} + e(X_i) + e_i$

We define the error of the estimate as:

$\epsilon_i = Y_i - \hat{Y}_i$

$\epsilon_i = e(X_i) + e_i$

So:

$Y_i = \beta_0 + \sum_{j=1}^{p} \beta_i X_{ij} + \epsilon_i$

The linear model could be wrong for two reasons. No linear model could be appropriate, or the wrong coefficients could be provided for a linear model.

Linear regression if f is a linear function on w. NB: not linear in x necessarily. could have x^2 etc, but still linear in w.

45.1.2 Intercept

45.1.3 Modelling non-linear functions as linear

Polynomials

The function $y = x^2$ is not linear, however we can model is as linear, by including x^2 as a variable.

We can expand this, and using linear models to estimate parameters for functions such as:

$y = ax^3 + bx^2 + xz$

Logarithms and exponentials

We can also transform data using logarithms and exponents.

For example we can model

$\ln y = \theta \ln x$

45.1.4 Geometric interpretation of OLS

Best Approximation Theorem

45.2 Calculating Ordinary Least Squares (OLS) estimators

45.2.1 Normal equation

Least squares

The square error is $\sum_i (\hat{y}_i - y_i)^2$.

The differential of this with respect to $\hat{\theta_j}$ is:

$2 \sum_i \frac{\delta \hat{y}_i}{\delta \hat{\theta_j}} (\hat{y}_i - y_i)$

The stationary point is where this is zero:

$\sum_i \frac{\delta \hat{y}_i}{\delta \hat{\theta_j}} (\hat{y}_i - y_i) = 0$

Linear least squares

Here, $\hat{y_i} = \sum_j x_{ij}\hat{\theta_j}$

Therefore: $\frac{\delta \hat{y_i}}{\delta \hat{\theta_j}} = x_{ij}$

And so the stationary point is where

$\sum_i x_{ij}(\sum_j x_{ij}\hat{\theta_j} - y_i) = 0$

$\sum_i x_{ij}(\sum_j x_{ij}\hat{\theta_j}) = \sum_i x_{ij}y_i$

Normal equation

We can write this in matrix form.

$X^T X\hat{\theta} = X^T y$

We can solve this as:

$\hat{\theta} = (X^T X)^{-1}X^T y$

Perfectly correlated variables

If variables are perfectly correlated then we cannot solve the normal equation.

Intuitively, this is because for perfectly correlated variables there is no single best parameter, as changes to one parameter can be counteracted by changes to another.

45.2.2 Mean and variance of predictions

Bias

$\hat{y} = \theta x$

$E[\hat{y} - y] = E[\theta x - y]$

$y = \hat{y} + \epsilon$

$E[y - \hat{y}|X]$

$E[\epsilon|X]$

Unbiased so long as independent of error term.

Variance

$Var[\hat{y} - y] = Var[\theta x - y]$

$Var[y - \hat{y}|X]$

$Var[\epsilon|X]$

45.2.3 The Moore-Penrose pseudoinverse

For a matrix X, the pseudoinverse is $(X^*X)^{-1}X^*$.

For real matrices, this is: $(X^TX)^{-1}X^T$

The pseudoinverse can be written as X^+

Therefore θ is the pseudoinverse of the inputs, multiplied by the outputs. Or:

$\theta = X^+y$

The pseudoinverse satisfies:

$XX^+X = X$

$X^+XX^+ = X^+$

45.2.4 Leverage

Introduction

Leverage measures how much the predicted value of y_i, $\hat{y}_i$, changes as y_i changes.

We have:

$\mathbf{y} = \mathbf{X}\theta + \mathbf{u}$

$\hat{\theta} = X(X^TX)^{-1}X^Ty$

$\hat{\theta} = P_Xy$

The leverage score is defined as:

$h_i = P_{ii}$

45.3 Making forecasts with OLS

45.3.1 The projection and annihilation matrices

The projection matrix

We have X.

The projection matrix is $X(X^TX)^{-1}X^T$

The projection matrix maps from actual y to predicted y

$\hat{y} = Py$

Each entry refers to the covariance between actual and fitted

$p_{ij} = \frac{Cov(\hat{y}_i, y_j}{Var(y_j)}$

The annihilation matrix

We can get residuals too:

$u = y - \hat{y} = y - py = (1 - P)y$

$1 - P$ is called the annihilator matrix

We can now use the propagation of uncertainty

$\Sigma^f = A\Sigma^x A^T$

To get:

$\Sigma^u = (I - P)\Sigma^y(I - P)$

Annihilator matrix is:

$M_X = I - X(X^TX)^{-1}X^T$

Called this because:

$M_X X = X - X(X^TX)^{-1}X^TX$

$M_X X = 0$

Is called residual maker

45.4 Frisch-Waugh-Lovell theorem

45.4.1 Introduction

If we have a partitioned linear regression model:

$\mathbf{y} = \mathbf{X}\theta + \mathbf{Z}\beta + \mu$

Use the annihilator matrix:

$M_X\mathbf{y} = M_X\mathbf{X}\theta + M_X\mathbf{Z}\beta + M_X\mu$

$M_X\mathbf{y} = M_X\mathbf{Z}\beta + M_X\mu$

We can then estimate β.

Frisch-Waugh-Lovell theorem says that this is the same estimate as the original regression.

45.5 Trimming

45.5.1 Introduction

OLS:

$\hat{\theta} = \frac{\sum_i (X_i - \mu_X)(y_i - \mu_y)}{\sum_i (x_i - \mu_X)^2}$

Trimming

$\hat{\theta} = \frac{n^{-1}\sum_i (X_i - \mu_X)(y_i - \mu_y)\mathbf{1}_i}{n^{-1}\sum_i (x_i - \mu_X)^2\mathbf{1}_i}$

Where:

$\mathbf{1}_i = \mathbf{1}(\hat{f}(z_i) \geq b)$

Where $b = b(n)$ is a trimming parameter, where:

$b \to 0$ as $n \to \infty$

45.6 Best linear predictor

45.6.1 Introduction

The best linear predictor is the one which minimises:

$E[Y - X\theta]$

Under what circumstances is this the same as OLS? When n¿¿p. When n is not, then other linear estimators (like LASSO) can be better.

45.7 Other

45.7.1 Cook's distance

Cook's distance measures the effect of deleting outliers. work out predictions if outlier was removed, sum all differneces in y hat

Outliers have a high Cook's distance.

45.7.2 Bayesian linear regression

In linear regression we have

$P(y|X, \theta, \sigma_\epsilon^2)$

For Bayesian linear regression we want:

$P(\theta, \sigma_\epsilon^2|y, X)$

We can use Bayes rule:

$P(\theta, \sigma_\epsilon^2|y, X) \propto P(y|X, \theta, \sigma_\epsilon^2)P(\theta|\sigma_\epsilon^2)P(\sigma_\epsilon^2)$

Chapter 46

Regularising linear regression for prediction

46.1 OLS predictions with many parameters

46.1.1 Too many variables

If there are more independent variables than samples then OLS will not work. There will be an infinite number of perfect fits.

For example if we regression genetic information on height with 1000 people, there will be too little data to fit using OLS.

This is due to colinearity.

We could also have too many variables through the use of derived variables. For example if we choose to use x, x^2, x^3 etc.

Optimal sparce regression

Optimal is $\lambda = \sigma 2\sqrt{2\log(pn)n}$

Relies on knowing σ, which we may not.

Instead we can use root LASSO.

Minimise the squareroot of the sum of squares loss (over n) , and use $\lambda = \sqrt{2\log(pn)/n}$

Doesn't have σ

Lasso biased, estimators 0 for many.

Post-LASSO

We can use LASSO for model selection, then use OLS on only those estimators.

46.2 Least Absolute Shrinkage and Selection Operator (LASSO)

46.2.1 Least Absolute Shrinkage and Selection Operator (LASSO)

Introduction

With LASSO we add a constraint to $\hat{\theta}$.

$\sum_i \hat{\theta}_i \leq t$

Regularisation of LLS. Sum of thetas are constrained to be below hyperparameter t

L1 regularisation

This is also known as sparce regression, because many weights are set to 0.

This now looks like:

$w_{lasso} = \arg\min ||y - Xw||_2^2 + \lambda ||w||_1$

Hyperparameter

t is a hyperparameter.

46.2.2 Optimal hyperparameter for LASSO

46.2.3 Feature scaling

46.3 Ridge regression

46.3.1 Ridge regression

Regularisation of LLS. The cost function now includes a norm on $M\theta$.

L_2 regularisation

This allows us to solve problems where there are too many features. L_1 also allows us to to do this.

Overspecified

If $n > d$ we can minimise weights subject to Xw=y. This is the same as the least norm.

Maximum A-Priori (MAP) estimator for linear regression

Maximum a priori estimation. equiv to ridge regression with a priori estimate of 0

$W_{RR} = (\lambda I + X^T X)^{-1} X^T y$

$E_[wRR] = (\lambda I + X^T X)^{-1} X^T X w$

$Var[W_{RR}] = a$

46.3.2 Elasticnet

Regularisation of LLS. Combines lasso and ridge regression.

L_1 and L_2 regularisation

46.3.3 L_p regularisation

Introduction

We can generalise this to:

$w_{l_p} = \arg\min ||y - Xw||_2^2 + \lambda ||w||_p^p$

For ridge regression there is always a solution.

For least squares there is a solution if $X^T X$ is invertible

For Lasso we must use numerical optimisation.

lasso and L_1 induces sparcity

Goal is $min||y - f(x)|| + \lambda g(w)$

Ridge regression: $g(w) = ||w||$

If $\lambda = 0$, OLS, if infinite, w goes to 0.

Normal equation changes to: $\lambda I + X^T X)^{-1} X^T y$

We can preprocess to avoid processing of 1s. shift mean of y to 0. normalise x mean 0 var 1.

46.3.4 Lava

Introduction

Alternative to ElasticNet

Each parameter is split into

$\theta_i = \rho_i + \phi_i$

There is L_2 loss on rho and L_1 loss on ϕ.

This means that large coefficients can be penalised like L_1 and small coefficients like L_2.

46.4 Tests

46.4.1 The Ramsey RESET test

The Ramsey Regression Equation Specification Error Test (RESET)

Once we have done our OLS we have $\hat{y}$.

The Ramsey RESET test is an additional stage, which takes these predictions and estimates:

$y = \theta x + \sum_{i=1}^{3} \alpha_i \hat{y^i}$

We then run an F-test on α, with the null that $\alpha = 0$.

46.4.2 The Link test

Introduction

Alternative to RESET

We have $\hat{y}$.

We regress $y = \alpha + \beta\hat{y} + \gamma\hat{y}^2$.

We test that $\gamma = 0$.

If it is not, then this suggests the model is misspecified.

46.5 Bias trade-off

46.5.1 Introduction

Trade-off between parameter accuracy and prediction accuracy.

Chapter 47

Choosing linear models for prediction

47.1 Other

47.1.1 Selection

How to restrict variables? best subset. iterate through alol and test.

not feasible for large numbers of variables. forward selection, backward selection, L1 alternatives.

removing variable does: increase bias. may reduce variance of prediction

Chapter 48

Generalised linear models

48.1 Introduction

48.1.1 Link/activation functions

48.2 Estimating parameters

48.2.1 Delta rule

Introduction

We want to train the parameters θ.

We can do this with gradient descent, by working out how much the loss function falls as we change each parameter.

The delta rule tells us how to do this.

The loss function

The error of the network is:

$E = \sum_j \frac{1}{2}(y_j - a_j)^2$

We know that $a_j = a(\theta x_j)$ and so:

$E = \sum_j \frac{1}{2}(y_j - a(\theta x_j))^2$

Minimising loss

We can see the change in error as we change the parameter:

$$\frac{\delta E}{\delta \theta_i} = \sum_j \frac{\delta E}{\delta a_j} \frac{\delta a_j}{\delta z_j} \frac{\delta z_j}{\delta \theta_i}$$

$$\frac{\delta E}{\delta \theta_i} = -\sum_j (y_j - a_j) a'(z_j) x_{ij}$$

Delta

We define delta as:

$$\delta_i = -\frac{\delta E}{\delta z_j} = \sum_j (y_j - a_j) a'(z_j)$$

So:

$$\frac{\delta E}{\delta \theta_i} = \delta_i x_{ij}$$

The delta rule

We update the parameters using gradient descent:

$$\Delta \theta_i = \alpha \delta_i x_{ij}$$

48.2.2 Maximum likelihood

48.3 Link/activation functions: Regression

48.3.1 Identity link function

The function

$$a(z) = z$$

The derivative

$$a'(z) = 1$$

Notes

This is the same as ordinary linear regression.

48.3.2 Absolute value rectification

$a(x) = |x|$

48.3.3 Rectified Linear Unit (ReLU)

The function

$a(z) = \max(0, z)$

The derivative

Its differential is 1 for values of z above 0, and 0 for values of z below 0.

The differential is undefined at $z = 0$, however this is unlikely to occur in practice.

Notes

The ReLU activation function induces sparcity.

48.3.4 Noisy ReLU

48.3.5 Leaky ReLU

48.3.6 Parametric ReLU

48.3.7 Softplus

The function

$a(z) = \ln(1 + e^z)$

The derivative

Its derivative is the sigmoid function:

$a'(z) = \dfrac{1}{1 + e^{-z}}$

Notes

The softplus function is a smooth approximation of the ReLU function.

Unlike the ReLU function, Softplus does not induce sparcity.

48.3.8 Exponential Linear Unit (ELU)

48.4 Link/activation functions: Classification

48.4.1 The binomial data generating process

Introduction

For linear regression our data generating process is:

$y = \alpha + \beta x + \epsilon$

For linear classification our data generating process is:

$z = \alpha + \beta x + \epsilon$

And set y to 1 if $z > 0$

Or:

$y = \mathbf{I}[\alpha + \beta x + \epsilon > 0]$

Probability of each class

The probability that an invididual with characteristics x is classified as 1 is:

$P_1 = P(y = 1|x)$

$P_1 = P(\alpha + \beta x + \epsilon > 0)$

$P_1 = \int \mathbf{I}[\alpha + \beta x + \epsilon > 0] f(\epsilon) d\epsilon$

$P_1 = \int \mathbf{I}[\epsilon > -\alpha - \beta x] f(\epsilon) d\epsilon$

$P_1 = \int_{\epsilon=-\alpha-\beta x}^{\infty} f(\epsilon) d\epsilon$

$P_1 = 1 - F(-\alpha - \beta x)$

Example: The logistic function

Depending on the probability distribution of *epsilon* we have different classifiers.

For the logistic case we then have

$$P(y = 1|x) = \frac{e^{\alpha+\beta x}}{1 + e^{\alpha+\beta x}}$$

48.4.2 Perceptron (step function)

The function

If the sum is above 0, $a(z) = 1$. Otherwise, $a(z) = 0$.

The derivative

This has a differential of 0 at all point except 0, where it is undefined.

Notes

This function is not smooth.

These is the activation function used in the perceptron.

Perceptron data needs to be linearly separable to train.

Even if linearly separable, doesn't necessarily get the best outcome?

48.4.3 Perceptron

Perceptron: one node neural network. is one or zero depednign if weightd inputs enough. therefore is classiication

If error, update weights

Only works if linearly separable. ie can draw linear line completely separting all inputs

Neural network has more layers

Works if data is linear

How to treat node inputs: raw, sigmoid, 0,1

For all of these want the cost function have only one solution, like least squares doe. not guaranteed for all

For logistic, want to make it convex. loss = -log(f(x)) or -log(1-f(x)) depending on correct y. this is convex

How to create node inputs: sigmoid, binary cutoff

48.4.4 Logistic function (AKA sigmoid, logit)

The function

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

The range of this activation is between 0 and 1.

The derivative

$$\sigma'(z) = \frac{e^{-z}}{(1 + e^{-z})^2}$$

$$\sigma'(z) = \sigma(z)\frac{1 + e^{-z} - 1}{1 + e^{-z}}$$

$$\sigma'(z) = \sigma(z)[1 - \sigma(z)]$$

Notes

48.4.5 Probability unit (probit)

The function

The cumulative distribution function of the normal distribution.

$\Phi(z)$

The derivative

The normal distribution:

$\Phi'(z) = \phi(z)$

48.4.6 tanH

48.4.7 ArcTan

48.4.8 Radial Basis Function (RBF) activation function

$a(x) = \sum_i a_i f(||x - c_i||)$

$a(x) = \sum_i a_i f e^{||x-c_i||^2}$

48.4.9 Linear probability model

$p = xB.$ can be outside $[0, 1]$.

48.5 Multinomial classification

48.5.1 The multinomial data generating process

Introduction

In the binomial case we had:

$z_i = \alpha + \beta x_i + \epsilon_i$

And set y_i to 1 if $z_i > 0$

In the multinomial case we have m alternatives

$z_{ij} = \alpha + \beta x_{ij} + \epsilon_{ij}$

And set $y_{ij} = 1$ if $z_{ij} > z_{ik} \forall k \neq j$

Generalised version

We can rewrite this as:

$z_{ij} = v_{ij} + \epsilon_{ij}$

Where:

$v_{ij} = \alpha + \beta x_{ij}$

In this case v does not depend on j, but in other formulations it could.

Probabilities

$P_{ij} = P(y_{ij} = 1|x_{ij})$

$P_{ij} = P(z_{ij} > z_{ik} \forall k \neq j)$

$P_{ij} = P(\epsilon_{ik} < v_{ij} - v_{ik} + \epsilon_{ij} \forall k \neq j)$

The form of the multinomial model: Intercepts

Previously we described the multinomial model

$z_{ij} = v_{ij} + \epsilon_{ij}$

Where:

$v_{ij} = \alpha + \beta x_{ij}$

The probability of j being chosen is.

$P_{ij} = P(\epsilon_{ik} < v_{ij} - v_{ik} + \epsilon_{ij} \forall k \neq j)$

Interceps in v cancel out. Therefore in the basic model there is no need to use

$v_{ij} = \alpha + \beta x_{ij}$

We can instead use:

$v_{ij} = \beta x_{ij}$

The form of the multinomial model: Conditional model

We have :

$v_{ij} = \beta x_{ij}$

What do we include in x_{ij}?

We can include observable characteristics for each product:

$v_{ij} = \alpha_j + \beta x_j$

One of the α_j must be normalised to 0, as only differences matter. We cannot tell the difference if all α are raised by the same amount.

For consistency with other models we can write this as:

$v_{ij} = \beta x_{ij}$

Even though this does not vary from individual to individual.

Here β represents average preferences for each product characteristic.

The form of the multinomial model: The multinomial model

We have differing characteristics for each individual:

$v_{ij} = \beta x_i$

However this adds a constant for each product. For this to discriminate we need varying coefficients.

$v_{ij} = \beta_j x_i$

As we only observe differences, one of the β_j must be normalised to 0.

We can rewrite this.

$v_{ij} = \sum_k \beta_k \delta_{kj} x_i$

$v_{ij} = \beta z_{ij}$

The original x_i is dense and contains data about the individual.

z_{ij} is sparce and only has entries in the {j} section.

Here β represents how the

The form of the multinomial model: Combined multinomial and conditional model

If we have observations of the characteristics of both individuals and alternatives we can write:

$v_{ij} = \beta_m m_{ij} + \beta_c c_{ij}$

$v_{ij} = \beta x_{ij}$

Here β represents both:

- Average preferences for customer characteristics (conditional)
- How preferences change as individual characteristics change (multinomial)

48.5.2 Extreme IID multinomial

IID

The probability of j being chosen is:

$P_{ij} = P(\epsilon_{ik} < v_{ij} - v_{ik} + \epsilon_{ij} \forall k \neq j)$

If these are independent then we have:

$P_{ij} = \prod_{k \neq j} P(\epsilon_{ik} < v_{ij} - v_{ik} + \epsilon_{ij})$

$P_{ij} = \prod_{k \neq j} F_\epsilon(v_{ij} - v_{ik} + \epsilon_{ij})$

We do not know ϵ_{ij} so we have to integrate over possibilities.

$P_{ij} = \int [\prod_{k \neq j} F_\epsilon(v_{ij} - v_{ik} + \epsilon_{ij})] f_\epsilon(\epsilon_{ij}) d\epsilon_{ij}$

Extreme values

We have:

$P_{ij} = \int [\prod_{k \neq j} F_\epsilon(v_{ij} - v_{ik} + \epsilon_{ij})] f_\epsilon(\epsilon_{ij}) d\epsilon_{ij}$

If ϵ is extreme value type-I this gives us:

$$P_{ij} = \frac{e^{v_{ij}}}{\sum_k e^{v_{ik}}}$$

Independence of irrelevant alternatives

Consider the ratio two probabilities:

$$\frac{P_{ij}}{P_{im}} = \frac{e^{v_{ij}}}{e^{v_{im}}}$$

This means that changes to any other products do not affect relative odds.

This can be undesirable. For example removing one option may cause unbalanced substitution.

For example raising the price of buses may cause more substitution to trains than helicopter, for a commute.

48.5.3 Estimating multinomial logit models

Estimating with individual level data.

Estimating with market share level data.

48.5.4 Nested logit

The probability of j being chosen is:

$P_{ij} = P(\epsilon_{ik} < v_{ij} - v_{ik} + \epsilon_{ij} \forall k \neq j)$

If the error terms are not IID this is more difficult to calculate.

We divide the J alternatives into nests. Within each of these we assume IID error terms, but allow variation between them.

For example we could have a nest of public/private transport. We could have a nest of types of product, and within that the firms offering the product.

The nested logit model does 2 or more seqential IID logit models. One to select the nest, and the other to select the alternative within the nest.

48.5.5 Mixed logit (random coefficients)

Introduction

In our standard model we have:

$z_{ij} = \beta x_{ij} + \epsilon_{ij}$

If we allow the parameters to vary for each individual we have:

$z_{ij} = \beta_i x_{ij} + \epsilon_{ij}$

The probability of choosing j now depends on the distribution of β.

In the IID case we had:

$$P_{ij} = \frac{e^{\beta x_{ij}}}{\sum_k e^{\beta x_{ik}}}$$

Rather than evaluate this at a single point β we integrate.

$$P_{ij} = \int \frac{e^{\beta x_{ij}}}{\sum_k e^{\beta x_{ik}}} f(\beta) d\beta$$

If β is degenerate this reduces to the standard logit model.

48.5.6 Multinomial probit

This relaxes the IID and extreme value assumption.

Errors have a normal variance-covariance matrix.

48.5.7 Softmax

The softmax function is often used in the last layer of a classification network.

It takes a vector of dimension k and returns another vector of the same size. Only, this time all numbers are between 0 and 1 and the values sum to 1.

The softmax function is based on the sigmoid function.

$$a_j(z) = \frac{e^{z_j}}{\sum_i e^{z_i}}$$

48.5.8 Temperature for Softmax

48.6 Generalised Additive Models (GAMs)

48.6.1 Introduction

Part XII

Supervised machine learning

Chapter 49

Classification And Regression Trees (CART)

49.1 Simple decision trees

49.1.1 Tree traversal

49.1.2 Training decision trees with information gain

We can train a decision tree by starting out with the most simple tree - all outcomes in same node.

We can then do a greedy search to identify which split on the node is best.

We can then iterate this process on future nodes.

Training with information gain

We split nodes to increase maximum entropy.

Entropy is:

$E = -\sum_i^n p_{i=1} \log_2 p_i$

Where we are summing across all nodes.

Information gain

The gain in entropy is the original entropy - weighted by size entropy of each branch

Information gain ratio

49.1.3 Training decision trees with Gini impurity

49.1.4 Pruning decision trees

Training a decision tree until there is only one entry from the training set will result in overfitting.

We can use pruning to regularise trees.

Pruning

Reduced error pruning

From bottom, replace each node with a leaf of the most popular class. Accept if no reduction in accuracy.

Cost complexity pruning

Take full tree T_0

Iteratively find a subtree to replace with a leaf. Cost function is accuracy and number of leaves.

Remove this generating T_{i+1}

When we have just the root, choose a single tree using CV.

Growing and pruning

Generally we would split the data up. Grow the tree with one set and then prune with the other.

We can split our data up and iterate between growing and pruning.

When pruning, for each pair of leaves we test to see if they should be merged.

If our two sets are A and B we can do:

- A: Grow
- B: Prune
- B: Grow
- A: Prune

And repeat this process.

Partial regression trees

Once we have built a tree, we keep a single leaf and disard the rest.

49.1.5 Decision trees with many classes

Training decision trees with many classes

49.2 Regression trees

49.2.1 Classic regression trees

In a classical regression tree, we follow a decison process as before, but the outcome is real number.

Within each leaf, all inputs are assigned that same number.

Training

With a regression problem we cannot split nodes the same way as we did for classification.

Instead by split by the residual sum of squares.

49.2.2 Training decision trees with Mean Squared Error (MSE)

49.2.3 Mixed regression trees

In classical trees all items in a leaf are assigned the same values. In this model, all are given θ for a parametric model.

This makes the resulting trees smoother.

We have some $\hat{y}_i = f(\mathbf{x}_i, \theta) + \epsilon$

The approach generalises classic regression trees. There the estimate was $\bar{y}$. Here it's a regression.

Training

At each node we do OLS. If the R^2 of the model is less than some constant, we find a split which maximises the minimum of the two new R^2.

49.2.4 Classifying with probabilistic decision trees

Previously our decision tree classifier was binary.

We can instead adapt the mixed tree model and using a probit model at each leaf.

49.3 Bayesian trees

49.3.1 Priors of trees

Priors for simple trees

We can define a tree as a set of nodes: T.

For each node we define a splitting variable k and a splitting threshold r.

Our prior is $P(T, k, r)$.

We split this up to:

$P(T, k, r) = P(T)P(k, r|T)$

$P(T, k, r) = P(T)P(k|T)P(r|T, k)$

So we want to estimate:

- $P(T)$ - The number of nodes.
- $P(k|T)$ - Which variables we split by, given the tree size.
- $P(r|T, k)$ - The cutoff, given the tree size and the variables we are splitting by.

Priors for mixed trees

If at the leaf we have a parametric model, our prior is instead:

$P(T, k, r, \theta) = P(T)P(k|T)P(r|T, k)P(\theta|T, k, r)$

We then need to additionally estimate $P(\theta|T, k, r)$.

49.3.2 The pinball prior

We can generate a tree with a fixed number of leaves, according to our prior.

As we start the tree we assocaite the root node with a count of all leaves.

As we split a node, the remaining leaf counts are divided between the directions. If there is only one leaf left, we do no further splitting.

49.3.3 Estimating other priors

49.3.4 Bayesian CART

Our prior

Call the collective parameters of the tree $\Theta = (T, k, r)$ and θ.

Collectively our prior is defined by $P(\Theta)$ and $P(\theta)$

Bayes' theorem

We want to know the posterior given our data X.

$P(\Theta|X) = \frac{P(X|\Theta)P(\Theta)}{P(X)}$

$P(\Theta|X) \propto P(X|\Theta)P(\Theta)$

Expanded posterior

We know explore $P(X|\Theta)$

$P(X|\Theta) = \int P(X|\theta, \Theta)P(\theta)d\theta$

This means our posterior is:

$P(\Theta|X) \propto P(\Theta) \int P(X|\theta, \Theta)P(\theta)d\theta$

Estimation

This can be estimated with MCMC.

49.4 Causal trees

49.4.1 easuring treatment effects in leaves

49.4.2 Sample splitting for treatment effects

49.4.3 Honest trees

We use part of the sample to estimate Θ, and another part of the sample to estimate the treatment effect.

49.4.4 Estimating ATE using MCMC

49.5 Other

49.5.1 Training with unbalanced data

Unbalanced dataset: more of one class than others.

Can reduce sample of majorit, or synthetically generate minority.

Chapter 50

Support Vector Machines (SVMs)

50.1 Linear Support Vector Classifiers (SVCs)

50.1.1 Hard-margin SVC

Linear separators

We want to create a hyperplane to separate classes.

For classification problem (x, y)

Hyperplane is wx-b=0

Hard margin

If data is linearly separable then a hyperplane exists such that all data can be correctly classified

There are an infinit number that could work.

We select two parallel with distance between as large as possible. the region between these two is the margin

The maximum margin hyperplane is the one between the two margin planes

We can rescale the two hyperplanes to:

wx-b=1

wx-b=-1

The distance between the two parrallel hyperplanes is $\frac{2}{||w||}$

So we minimise $||w||$ conditional an all points being correctly classified

$y_i(wx_i - b) \geq 1$

We select w and b to solve this.

50.1.2 Support vectors

Support vectors are those that make up the classifer boundry.

50.1.3 Estimating the SVC using quadratic equations

50.1.4 Soft-margin SVC

Soft margin

Soft margin

Data may not be linearly separable, so we introduce a hinge loss function

$Max(0, 1 - y_i(wx - b))$

We then minimise

$\lambda||w||^2 + [\frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i(wx_i - b)]$

This introduces λ as a parameter.

50.1.5 Regularising the SVC

50.2 Multiple classes

50.2.1 Support vector classifiers for multiple classes

50.3 Non-linear support vector classifiers

50.3.1 The dot product SVC

50.3.2 The kernel trick

We can use kernels as an alternative to the dot product.

50.3.3 The radial basis function (RBF) SVC

Chapter 51

Other machine learning classifiers

51.1 Regularising classifiers

51.1.1 Label smoothing

Regualrising of classifiers assume some incorrect (similar for regression?)

51.2 Variational Bayes

51.2.1 Introduction

51.3 Interpreting black box models

51.3.1 Introduction

partial dependence plots can be used on black box models

Interpretation: if its interpretable then you can adjust an interpretable part manually part way?

Transparency of models: Sparse linear models are more transparent. Decomposition: Can each part of the model have input, output, parameters which can be interpreted? Complex feature selection means loss of this. Complex models. Boosting. Loses Can we say formal things about performance? We can for linear models (?), but not for others For agents, we can't validate their behaviour.

We can for manually defined rules. We can for interpretive models. Post-hoc interpretability

LIME Local Interpretable Model-Agnostic Explanations. Page on explainable models, h3 in that on locally explainable models

Explaining models: We may want to understand how it works. Black box algorithms are hard to understand.

This is important if the algorithm is used in high stakes cases, or where data is different to the static case used for training.

We can create explainable models (sparse linear models)

Take black box models and make them explainable.

SLAM algorithms?

Local explanation? Saliency maps?

Why do we care about transparency?

51.4 Confidence intervals of black box models

51.4.1 Introduction

ensemble confidence intervals non-parametric confidence intervals semi-parametric confidence intervals

one-sided, 2 sided confidence intervals

jackknife for bagging confidence interval

Chapter 52

The Naive Bayes classifier

52.1 Naive Bayes

52.1.1 The Naive Bayes posterior

Bayes theorem

Consider Bayes' theorem

$$P(y|x_1, x_2, ..., x_n) = \frac{P(x_1, x_2, ..., x_n|y)P(y)}{P(x_1, x_2, ..., x_n)}$$

Here, y is the label, and $x_1, x_2, ..., x_n$ is the evidence. We want to know the probability of each label given evidence.

The denominator, $P(x_1, x_2, ..., x_n)$, is the same for all, so we only need to identify:

$$P(y|x_1, x_2, ..., x_n) \propto P(x_1, x_2, ..., x_n|y)P(y)$$

The assumption of Naive Bayes

We assume each x is independent. Therefore:

$$P(x_1, x_2, ..., x_n|y) = P(x_1|y)P(x_2|y)...P(x_n|y)$$

$$P(y|x_1, x_2, ..., x_n) \propto P(x_1|y)P(x_2|y)...P(x_n|y)P(y)$$

52.1.2 The Naive Bayes classifier

Calculating the Naive Bayes estimator

With the Naive Bayes assumption we have:

$P(y|x_1, x_2, ..., x_n) \propto P(x_1|y)P(x_2|y)...P(x_n|y)P(y)$

We now choose y which maximises this.

This is easier to calculate, as there is less of a sample restriction.

This is used when evidence is also in classes, as the chance of any individual outcome on a continuous probability is 0.

Estimating $P(y)$

We can easily calculate $P(y)$, by looking at the frequency across the sample.

Estimating $P(x_1|y)$

Normally, $P(x_1|y) = \frac{n_c}{n_y}$, where:

- n_c is the number of instances where the evidence is c and the label is y.
- n_y is the number of instances where the label is y.

Regularising the Naive Bayes estimator

To reduce the risk of specific probabilities being zero, we can adjust them, so that:

$P(x_1|y) = \frac{n_c + mp}{n_y + m}$, where:

- p is the prior probability. If this is unknown, use $\frac{1}{k}$, where k is the number of classes.
- m is a parameter called the equivilant sample size.

52.1.3 Text classification using Naive Bayes

Naive Bayes and text classification

Naive Bayes can be used to classify text documents. The x variables can be appearances of each word, and y can be the document classification.

Chapter 53

The K-Nearest Neighbours (KNN) classifier

53.1 K-Nearest Neighbours (KNN)

53.1.1 K-nearest neighbours

K-nearest neighbours is a non-parametric classifer. For a point with an unknown class we identify the classes of the K-nearest neighbours and assign the most common class.

For this we need to find the distance between observations. We can do this using norms.

Weightings of the dependent variables is important here.

Practicality

Requires space to store

N samples, d features

times: O(n.d)

53.1.2 Choosing K for K-Nearest Neighbours

Chapter 54

Discriminant analysis

54.1 Discriminant analysis

54.1.1 Linear Discriminant Analysis (LDA)

Assume data is mixed gaussian. Use this to estimate classes.

54.1.2 Kernel Fisher discriminant analysis

Use LDA with kernel feature spaces.

Chapter 55

Non-parametric regression

55.1 Kernel regression

55.1.1 Kernel regression

Introduction

For parametric regression we have:

$y = f(X)$

Where the form of $f(X)$ is fixed, such as for linear regression.

For non-parametric regression we have:

$y = m(X)$

Where $m(X)$ is not fixed.

We can estimate $m(X)$ using kernel regression.

$$m(X) = \frac{\sum_{i=1}^{n} K_h(x - x_i) y_i}{\sum_{i=1}^{n} K_h(x - x_i)}$$

We know this because we have:

$$E(y|X) = \int y f(y|x) dy = \int y \frac{f(x, y)}{f(x)} dy$$

We then use kernel density estimation for both.

55.2 Splines

55.2.1 Multivariate Adaptive Regression Splines (MARS)

A linear model looks like:

$\hat{y} = c + \sum_i x_i \theta_i$

MARS instead produces a linear model for subsets of X.

$\hat{y} = c + \sum_i B_j(x_i, a_j)\theta_i$

Where:

- $B_j = max(0, x_i - a_j)$; or
- $B_j = -max(0, a_j - x_i)$

This is trained using a forward pass and a backward pass.

Forward pass

Backward pass

55.2.2 Bayesian splines

55.3 Other

55.3.1 Local regression

55.3.2 LOWESS

55.3.3 LOESS

55.3.4 Kernel regression

Quantile regression

In other supervised?

Normally we return a central estimate, commonly the mean.

Quantile regression returns an estimate of the ith quartile instead.

Goal is to find xth quartile of variance.

Linear quantile regression

Tree quantile regression

55.3.5 Principal component regression

Do PCA on X.

Do OLS with this.

Transform parameters by reversing PCA procedure on parameters.

55.3.6 Partial least squares regression

This expands on principal component regression.

Both X and Y are mapped to new spaces.

Chapter 56

Ensemble methods

56.1 Combining learners

56.1.1 Majority voting

Combining classifiers

If we generate different classifiers then each can give different predictions for the same input.

If we have different predictions how should we proceed?

Using one model or multiple models

It is not obvious that we should want to use more than one model. If one model was superior to another then there may be no benefit to using the information from the additional model.

However if the errors in different models are varied, then combining multiple models can lead to better performance as each individual model can have unique information.

Majority voting

One approach to using different predictions is to use majority voting.

If we have a collection of hard classifiers then we choose the classification with the most votes.

Condorcet's Jury Theorem

Consider a collection of classifiers. For each classifier there is a chance p_i of the classification being correct.

If the voters are independent

If voters are independent, and the chance of one vote being right is greater than 0.5, then the more voters, the better.

A weak model can still be useful, if it is independent.

56.1.2 Averaging regression predictions

If we have multiple predictions, we can take an average of these, possibly weighted.

We have m regressors $g_j(\mathbf{x}_i)$.

Our output is:

$h(\mathbf{x}_i) = \sum_j w_j g_j(\mathbf{x}_i)$

56.1.3 Stacking and SuperLearner

Introduction

With stacking we take the predictions from each of our classifiers, and then train a new model using these predictions as inputs.

Hard and soft inputs

Hard classification (0 or 1) and soft classification (between 0 and 1) can be used as inputs.

Cross validation

We can select hyper-parameters using cross validation, however there is an issue of using cross validation twice on the same data. Once for the underyling classifiers, and again for the stacked model.

SuperLearner

We have m models.

Part 1: Train each of them on all data.

Part 2: Split the data into k sets

For each set, associate all other data as training

For each fold:

- Fit each model on the training data
- Predict on other
- Create weighted predictor. choose weightings to minimise error

Part 3: Use these weightings on the original (unrestricted data) model

56.2 Generating learners with boosting

56.2.1 L_2 boosting

56.2.2 Adaboost

Introduction

Boosting is a way to create multiple learners for use in an ensemble predictor.

The goal is to create many predictors, which may not be themselves very accuracte, but have a high degree of independence.

AdaBoost

AdaBoost is a popular algorithm for boosting.

It works by:

- Creating a set of weak learners using different restrictions on features in the training data.
- Choosing the weak learner that most reduces the error of the combined learners, and give it a weighting which most reduces the error of the combined learners.
- Creating a new weighting for the dataset, where ones poorly predicted (by the combination of learners) are given high weights.
- Repeating the process a fixed number of times.

56.2.3 Gradient boosting

Gradient boosting does not iterative change the weights for the learners. Instead, it trains on different errors.

While AdaBoost trains to reduce the absolute error for each weak classifier, gradient boosting trains on the difference between the actual classiffication and the current classification.

56.3 Generating learners with Bootstrapped AGGregation (bagging)

56.3.1 Bagging

Introduction

Bagging, or Bootstrap AGGregation, is a way of generating weak learners.

Bootstrapping

Bootstraping refers to taking samples with repacement from the training set. This is how the datasets for each of the weak learners are formed.

How to do bagging

We take samples from the training set, with replacement, and train each of these separately. This gives us our weak learners.

56.4 Ensemble methods for trees

56.4.1 Gradient tree boosting

This applies gradient boosting to tree.

56.4.2 Multiple Additive Regression Trees (MART)

56.4.3 Bayesian Additive Regression Trees (BART)

56.4.4 Extra trees

56.4.5 Random forests

These use bagging techniques with random trees.

At each node, rather than sample the whole data we sample a random selection.

Get d dimensions, and sample m of them at each node.

Choose $m \leq \sqrt{d}$

56.4.6 Regression forests

56.4.7 Causal forests

56.4.8 Bayesian causal forests

56.5 Bootstrapping moments of ensemble statistics

56.5.1 Bootstrapping mean and variance

Sample size selection

56.5.2 Bootstrapping confidence intervals

Need to know estimate is unbiased for this.

Part XIII

Supervised neural networks

Chapter 57

Multi-layer perceptrons

57.1 Multi-Layer Perceptron (MLP)

57.1.1 Hidden layers

In the perceptron we have input vector x, and output:

$a = a(wx)$

We can augment the perceptron by adding a hidden layer.

Now the output on the activation function is an input to a second layer. By using different weights, we can create a second vector of inputs to the second layer.

The parameters of a feed forward model

Θ^j is a matrix of weights for mapping layer j to $j+1$. So we have Θ^1 and Θ^2.

If we have s units in the hidden layer, n features and k classes:

- The dimension of Θ^1 is $(n+1) \times s$
- The dimension of Θ^2 is $(s+1) \times k$

These include the offsets for each layer.

The activation function of a multi-layer perceptron

For a perceptron we had $a = f(wx)$. Now we have:

$a_i^j = f(a_{j-1}\Theta_{j-1})$

We refer to the value of a node as a_i^j, the activation of unit i in layer j.

Initialising parameters

We start by randomly initialisng the value of each θ.

We do this to prevent each neuron from moving in sync.

57.1.2 Dummies in neural networks

57.1.3 Back propagation

Adapting the delta rule

To arrive at the delta rule we considered the cost function:

$E = \sum_j \frac{1}{2}(y_j - a_j)^2$

And used the chain rule:

$\frac{\delta E}{\delta \theta_i} = \frac{\delta E}{\delta a_j}\frac{\delta a_j}{\delta z_j}\frac{\delta z_j}{\delta \theta_i}$

This gave us:

$\Delta\theta_i = \alpha \sum_j (y_j - a_j)a'(z_j)x_{ij}$

Or, setting $\delta_i = -\frac{\delta E}{\delta z_j} = \sum_j (y_j - a_j)a'(z_j)$

$\Delta\theta_i = \alpha\delta_j x_{ij}$

Let's update the rule for multiple layers:

$\frac{\delta E}{\delta \theta_{li}} = \frac{\delta E}{\delta a_{lj}}\frac{\delta a_{lj}}{\delta z_{lj}}\frac{\delta z_{lj}}{\delta \theta_{li}}$

Previously $\frac{\delta z_{lj}}{\delta \theta_{li}} = x_i$. We now use the more general a_{li}. For the first layer, these will be the same.

We can then instead write:

$\Delta\theta_i = \alpha\delta_{lj}a_{li}$

Calculating delta values

Now we need a way of calculating the value of δ_{lj} for all neurons.

$$\delta_i = -\frac{\delta E}{\delta z_{lj}}$$

If this is an output node, then this is simply $\sum_j (y_j - a_j) a'(z_j)$

If this is not an output node, then the impact of change in the parameter will affect the results through all intermediate neurons.

In this case:

$$\frac{\delta E}{\delta z_{lj}} = \sum_{k \in succl} \frac{\delta E}{\delta z_k} \frac{\delta z_k}{\delta z_{lj}}$$

$$\frac{\delta E}{\delta z_{lj}} = \sum_{k \in succl} -\delta_k \frac{\delta z_k}{\delta a_{kj}} \frac{\delta a_{kj}}{\delta z_{lj}}$$

$$\frac{\delta E}{\delta z_{lj}} = \sum_{k \in succl} -\delta_k \theta_{kj} a'_{kj}$$

$$\delta_i = a'_{kj} \sum_{k \in succl} \delta_k \theta_{kj}$$

For each layer there is a matrix, where the columns and rows represent the *theta* between the current layer and the next layer. We have a matrix for each layer in the network.

57.1.4 Ill conditioning

Skips over actual max if not "smooth" enough.

57.2 More than 2 classes

57.2.1 Local Winner Takes All (LWTA) layer

The output for all nodes in a layer is 0 unless it is the greatest.

57.2.2 Maxout layer

Single node, spits out the max of all inputs.

57.3 Regression

57.4 Deep neural networks

57.4.1 More layers allow for more complex function

With additional hidden layers we can map more complex functions.

These allow the effective combination of logic gates.

2 hidden layers can map highly complex functions

With only two hidden layers we can map any function for classification, including discontinuous functions.

57.4.2 Convexity

The error function for neural networks is nearly convex.

57.4.3 Unstable gradient problem

Vanishing gradient problem

Gradients can be become small, and so propagation can be very slow.

Exploding gradient problem

Gradients can become too large and not converge.

ReLU

This addresses the unstable gradient problem.

57.4.4 Curse of dimensionality

57.4.5 Increasing numbers of dimensions in a unit

Topology of layers. Increasing number of units in subsequent layers is like increasing dimension.

We are trying to make data linearly separable. it may be that we need additional dimensions to do this, rather than a series of transformations within the existing number of dimensions.

eg for a circle of data within a circle of data, there is no linear separable line, so no depth without increasing dimensions will split data.

57.5 Optimisation

57.5.1 Input layer normalisation

We normalise the input layer.

This speeds up training, and makes regularisation saner.

57.5.2 Batch normalisation

We can normalise other layers. We take each input, subtract the batch mean divided by the batch standard deviation.

Batch normalisation and covariance shift

Batch normalisation can make networks better adapted for related problems.

Training with batch normalisation

57.6 Alternatives to backpropagation

57.6.1 Greedy pretraining

57.6.2 Cascade-correlation learning architecture

This is a method for both building and training.

We start with a bare bones network. We then add nodes one by one, training and then fixing their values.

57.6.3 Extreme learning machines

This is an alternative to backprobagation for training a feedforward neural network.

We start with random parameters for each layer W_i.

We have:

$\hat{y} = W_2\sigma(W_1 x)$

Etc.

We calculate:

$W_2 = \sigma(W_1 x)^+ Y$

So W_1 is random and not updated.

W_2 is assigned to minimise loss, where W_2 has no activation function.

57.7 Pre-training

57.7.1 Pre-training

Pre-training. eg train on general pictures before specific stuff. means you fit many parameters for detecting edges etc firsts

Learning rate

Reduce the learning rate.

57.7.2 Resetting parameters

Replace last layer (softmax) for new problem.

57.7.3 Freeze layers

Freeze feature learning of early layers.

57.8 Other

57.8.1 Representational sparsity

This is where the values in nodes are often 0, as opposed to just the parameters.

57.8.2 Catastrophic interference

57.8.3 Probabilistic neural networks

Introduction

Input layer

The input is the feature vector.

Pattern layer

The first layer has a node for each variable in the training set.

In each node, the value is the distance from the input to the comparator.

This can be calculated using Gaussian distribution, or another method.

Summation layer

One neuron for each category.

We map from the pattern layer to the summation layer according to the actual label of each training item.

Ie, if a sample is red, it will be fed only to the red neuron.

The values are summed.

Largest value is selected.

Chapter 58

Regularising neural networks

58.1 Regularising neural networks

58.1.1 Feature normalisation

58.1.2 Dropout, and dropout layers

58.1.3 L_2 regularisation (including how to change back-prob algorithm)

58.1.4 Sparse networks

Parameters are set to 0 and not trained.

58.1.5 Parameter sharing

Parameters share the same value and are trained together.

58.1.6 Weight decay

After each update, multiply the parameter by $p < 1$.

58.1.7 The anomoly detection problem

Can change input to get any classification.

58.1.8 Early stopping

58.1.9 Residual blocks

In a node we have:

$a_{ij} = \sigma_{ij}(W_{ij}a_{i-1})$

That is, the value of a node, is the activation on the sum of the weights of the previous layer.

Residual block however look further back that one layer. They include the full data from an older layer (without weights)

$a_{ij} = \sigma_{ij}(W_{ij}a_{i-1} + a_k)$

Chapter 59

Convolutional layers for neural networks

59.1 Convolutional layers

59.1.1 Convolutional layers

Can connect each node in first hidden layer to a subset of the input layer, eg one node for each 5x5 pixels

We also share weights for each of the first layer. Much fewer parameters, and can learn all good stuff

This also uses windows. Instead of max we multiply the window by a matrix elementwise and sum the values

Each matrix can represent some feature, like a curve.

We can use multiply convolution matrices to create multiple output matrices.

Matrices are called kernels. they are trained. start off random

Training convolutional layers

59.1.2 Invariance of convolutional layers (rotation, translation)

59.1.3 Flattening layers

We split the data up everytime we use convolional layers

Flattening layers bring them all back together

Parameters are that for pooling layers (height, width, stride, padding, but also set of convolutions.

59.1.4 Multi-scale convolutions

We use different window sizes in parallel.

59.1.5 Inception modules

59.2 Pooling (max pooling, average pooling, sub-sampling)

59.2.1 Pooling layers

The input is a matrix. We place a number of windows on the input matrix. The max of each window is an input to the next layer.

Means fewer parameters, easier to compute, less chance of overfitting

Parameters: height, width of window, stride (amount shifts by each window)

We can also add padding to the edge of the image so we don't lose data.

Same padding (use 0), valid padding (no padding)

Pooling layer compresses, takes 2x2. Max pooling returns highest activiation

59.2.2 Vector of Locally Aggregated Descriptors (VLAD)

59.3 Other window layers

59.3.1 Capsules

Primary capsule layers

Outputs of convolutions are scalars. however we can also create vectors, if we associate some convolutions with each other

eg if we have 6 convolutions, the output of these can be used to create a 6 dimensional vector for each window.

Normalisation in primary capsule layers (vector squishing)

We can normalise the length of these vectors to between 0 and 1.

The output of this repesents the chance of finding the feature they are looking for, and the orientation

If the vector length is low, feature not found. if high, feature found.

We have orientation from vector, and position from window

Routing capsule layers

We now have a layer of position and orientation of basic shapes (triangles, rectangles etc)

We want to know which more complex thing they are part of.

So the output of this step is again a matrix with position and orientation, but of more complex features

To determine the activation from each basic shape to the next feature we use routing-by-agreement.

This takes each basic shape and works out what it would look like if the complex feature was present.

If a complex feature has two basic shapes, they will both have the same predicted complex shape. Otherwise the relationship is spurious and they will not

If they agree we have a high weight

This process is complex and computationally expensive.

However we don't need pooling layers now

caps net

Does normal conv first, then primary, then secondary.

caps: reconstruction

We have vector space of feature position and orietnation. we can recreate output

Part XIV

Generative neural networks

Chapter 60

Autoencoders and Variational Autoencoders (VAE)

60.1 Autoencoders

60.1.1 Autoencoders

Autoencoders are a type of neural network.

For autoencoders the goal for the output layer is the reconstructed input layer, rather than a classification.

By including sparsity in the neural network we can reduce the dimensions. This splits the network into an encoder and a decoder.

Middle vector is called latent variables.

60.2 Variational autoencoders

60.2.1 Variational Autoencoders (VAE)

Like AE, but force middle vector to have unit gaussian by adding new loss function

Now we can generate new images by sampling for latent normal of unit 1.

Chapter 61

Restricted Boltzmann Machines (RBMs)

61.1 Restricted Boltzmann Machines (RBMs)

61.1.1 Restricted Boltzmann Machines (RMBs)

61.1.2 Deep Belief Networks (DBNs)

61.1.3 Training with Contrastive Divergence (CD)

Chapter 62

Self-organising maps

62.1 Self-organising maps

Chapter 63

Generative neural networks

63.1 Generative adversarial networks

63.1.1 Generative Adversarial Networks (GANs)

Introduction

The GAN generator

Generator: take latent multivar normal as input layer

Output of generator is of same dimension as input iamges

Generative NN: need probability distribution on input layer

Generator typically deconvolutional

Downside of gan. only discriminates between real and fake.

The GAN discriminator

Discriminator: use normal CNN for deep

Part XV

Applied supervised machine learning

Chapter 64

Classifying written characters

64.1 Character recognition

64.1.1 MNIST

Chapter 65

Text recognition

65.1 Image recognition

65.1.1 CIFAR-10

65.1.2 ImageNet

Chapter 66

Facial recognition

66.1 Facial recognition

66.1.1 FERET

Chapter 67

Computer vision

67.1 Camera vision

67.1.1 Camera inputs

67.2 Classifying images

67.3 Semantic image segmentation

67.4 Building 3D models

67.4.1 Multi-view CNNs

67.4.2 Volumetric models

67.4.3 Point clouds

67.4.4 Polygon mesh

67.4.5 Generative Query Network (GQN)

67.4.6 Primitive-based CAD

67.4.7 3D ShapeNets

67.4.8 Building 3D models from scans

67.5 LIDAR

67.5.1 LIDAR

67.5.2 Classification with voxels

67.5.3 Absolute risk aversion

67.5.4 Parsing

Mulitple objects in scene, objects have parts segmentation.

Part XVI

Linear regression for inference

Chapter 68

Ordinary Least Squares for inference

68.1 Bias of OLS estimators

68.1.1 Expectation of OLS estimators

Expectation in terms of observables

We have: $\hat{\theta} = (X^T X)^{-1} X^T y$

Lets take the expectation.

$E[\hat{\theta}] = E[(X^T X)^{-1} X^T y]$

Expectation in terms of errors

Lets model y as a function of X. As we place no restrictions on the error terms, this is not as assumption.

$y = X\theta + \epsilon.$

$E[\hat{\theta}] = E[(X^T X)^{-1} X^T (X\theta + \epsilon)]$

$E[\hat{\theta}] = E[(X^T X)^{-1} X^T X\theta] + E[(X^T X)^{-1} X^T \epsilon)]$

$E[\hat{\theta}] = \theta + E[(X^T X)^{-1} X^T \epsilon)]$

$E[\hat{\theta}] = \theta + E[(X^T X)^{-1} X^T] E[\epsilon] + cov[(X^T X)^{-1} X^T, \epsilon]$

The Gauss-Markov: Expected error is 0

$E[\epsilon = 0]$

This means that:

$E[\hat{\theta}] = \theta + cov[(X^T X)^{-1} X^T, \epsilon]$

The Gauss-Markov: Errors and indepedent variables are uncorrelated

If the error terms and X are uncorrelated then $E[\epsilon|X] = 0$ and therefore:

$E[\hat{\theta}] = \theta$

So this is an unbiased estimator, so long as the condition holds.

68.2 Variance of OLS estimators

68.2.1 Variance of OLS estimators

Variance-covariance matrix

We know:

$\hat{\theta} = (X^T X)^{-1} X^T y$

$y = X\theta + \epsilon$

Therefore:

$\hat{\theta} = (X^T X)^{-1} X^T (X\theta + \epsilon)$

$\hat{\theta} = \theta + (X^T X)^{-1} X^T \epsilon$

$\hat{\theta} - \theta = (X^T X)^{-1} X^T \epsilon$

$Var[\hat{\theta}] = E[(\hat{\theta} - \theta)(\hat{\theta} - \theta)^T]$

$Var[\hat{\theta}] = E[(X^T X)^{-1} X^T \epsilon (X^T X^{-1} X^T \epsilon)^T]$

$Var[\hat{\theta}] = E[(X^T X)^{-1} X^T \epsilon\epsilon^T X (X^T X)^{-1}]$

$Var[\hat{\theta}] = (X^T X)^{-1} X^T E[\epsilon\epsilon^T] X (X^T X)^{-1}$

We write:

$\Omega = E[\epsilon\epsilon^T]$

$Var[\hat{\theta}] = (X^T X)^{-1} X^T \Omega X (X^T X)^{-1}$

Depending on how we estimate Ω, we get different variance terms.

Variance under IID

If IID:

$\Omega = I\sigma_\epsilon^2$

$Var[\hat{\theta}] = (X^T X)^{-1} X^T I\sigma_\epsilon^2 X (X^T X)^{-1}$

$Var[\hat{\theta}] = \sigma_\epsilon^2 (X^T X)^{-1}$

68.2.2 Heteroskedasticity-Consistent (HC) standard errors

Variance of OLS estimators

$Var[\hat{\theta}] = (X^T X)^{-1} X^T \Omega X (X^T X)^{-1}$

Robust standard errors for heteroskedasticity

$\Omega_{ij} = \delta_{ij} \epsilon_i \epsilon_j$

These are also known as the Eicker-Huber-White standard errors, or the White correction.

These are also refered to as robust standard errors.

68.3 Properties of the OLS estimator

68.3.1 Maximum Likelihood Estimator (MLE) and OLS equivalence

The OLS estimator

$\hat{\theta}_{OLS} = (X^T X)^{-1} X^T y$

$E[\hat{\theta}_{OLS}] = w$

$Var[\hat{\theta}_{OLS}] = \sigma^2 (X^T X)^{-1}$

The MLE estimator

$y_i = \mathbf{x}_i \theta + \epsilon_i$

$P(y = y_i | x = x_i) = P(\epsilon_i = y_i - \mathbf{x}_i \theta)$

If we assume $\epsilon_i \sim N(0, \sigma_\epsilon^2)$ we have:

$$P(y = y_i|x = x_i) = \frac{1}{\sqrt{2\pi\sigma_\epsilon^2}} e^{-\frac{(y_i - \mathbf{x}_i\theta)^2}{2\sigma_\epsilon^2}}$$

$$L(X, \theta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_\epsilon^2}} e^{-\frac{(y_i - \mathbf{x}_i\theta)^2}{2\sigma_\epsilon^2}}$$

$$l(X, \theta) = \sum_{i=1}^{n} -\frac{1}{2}\ln(2\pi\sigma_\epsilon^2) - \frac{(y_i - \mathbf{x}_i\theta)^2}{2\sigma_\epsilon^2}$$

$$\frac{\delta l}{\delta \theta_j} = \sum_{i=1}^{n} 2x_{ij}\frac{y_i - \mathbf{x}_i\theta}{2\sigma_\epsilon^2}$$

$$\sum_{i=1}^{n} x_{ij}(y_i - \hat{\theta}_{MLE}\mathbf{x}_i) = 0$$

$$X^T(y - X\hat{\theta}_{MLE}) = 0$$

$$X^T y = X^T X\hat{\theta}_{MLE}$$

$$\hat{\theta}_{MLE} = (X^T X)^{-1}X^T y$$

Equivalence

If errors are normally IID then:

$\hat{\theta}_{OLS} = \hat{\theta}_{MLE}$

68.3.2 Gauss-Markov theorem

Mean of errors zero + If the model should only have errors on upside or downside for some reason, OLS will not provide this.

Homoscedastic (all have the same variance) + The results arent biased, but variances etc are

Errors are uncorrelated + (this would mean you should add lagged variables etc)

show bias from each GM violation

OLS is BUE under normally distributed errors

OLS is BLUE for non-normally distribed errors

68.4 Selection

68.4.1 T-test selection

68.4.2 Post-LASSO

68.5 Heteroskedasticity

68.5.1 Checking for heteroskedasticity using the White test

68.5.2 Robust standard errors

68.5.3 Noise

68.5.4 Regression dilution

Noise in y doesn't cause bias.

Noise in x does cause bias.

Need to correct.

68.5.5 Causality

68.5.6 Introduction

Causality v correlation. If just getting correlation, could have bad out of sample performance

Section on causality. Difference between disease causes symptom and symptom causes disease

Linear models can be manipulated to have any variable on the left.

Chapter 69

Testing regression parameter estimates with Z-tests and T-tests

Chapter 70

Multiple hypothesis testing

70.1 Multiple hypothesis testing

70.1.1 P-hacking

Likely to see some significant results from random chance.

70.1.2 Family-Wise Error Rate (FWER)

What is the chance of making at least one false positive result?

Number of tests: m

Number of false positive results: V

$FWER = P(V > 0)$

70.1.3 False Discovery Rate (FDR)

The proportion of false discoveries is:

$Q = \frac{V}{V+S}$

Where: V is the number of false positives

S is the number of true positives

The FRD is $E[Q]$.

70.1.4 The Bonferroni correction

We change the significance level.

reject if $p \leq \frac{\alpha}{m}$

If $m = 1$ this is the standard test.

Chapter 71

Generalised Least Squares

71.1 Generalised Least Squares (GLS)

71.1.1 The Generalised Least Squares (GLS) estimator

Introduction

We make the same assumptions as OLS.

$\mathbf{y} = \mathbf{X}\theta + \boldsymbol{\epsilon}$

We assume:

- $E[\epsilon|\boldsymbol{X}] = 0$
- $Cov[\epsilon|\boldsymbol{X}] = \boldsymbol{\Omega}$

The GLS estimator

GLS estimator is:

$\hat{\theta}_{GLS} = argmin_b (y - Xb)^T \Omega^{-1} (y - Xb)$

$\hat{\theta}_{GLS} = (X^T \Omega^{-1} X)^{-1} X^T \Omega^{-1} y$

This is the vector that minimises the Mahalanobis distance.

This is equivalent to doing OLS on a linearly transformed version of the data.

Identifying Ω

If Ω is known, we can proceed. Generally, however, Ω is not known, and so the GLS estimate in infeasible.

71.2 Feasible Generalised Least Squares (FGLS)

71.2.1 The Feasible Generalised Least Squares (FGLS) estimator

Introduction

We do OLS to get a consistent estimate of Ω, $\hat{\Omega}$.

We then plug this into the GLS estimator.

71.3 Heteroskedasticity

71.3.1 Weighted least squares

71.4 Bias and variance of the GLS estimator

71.4.1 Introduction

you have the same sandwich term as before, so same process, right?

71.5 Linear discriminant analysis

Chapter 72

General Linear Models

72.1 Cross-sectional regression

72.1.1 The cross-sectional model

Hierarchical data

Our standard linear model is:

$y_i = \alpha + X_i\theta + \epsilon_i$

If we had two sets of data we could view these as:

$y_{i,0} = \alpha_0 + X_{i,0}\theta_0 + \epsilon_{i,0}$

$y_{i,1} = \alpha_1 + X_{i,1}\theta_1 + \epsilon_{i,1}$

Here, the data data from 1 does not affect the parameters in 2.

Pooled data

If we think the data generating process is similar between models, then by restricting the freedom of parameters between models we can get more data for each estimate.

For example if we think that all parameters are the same between the models we can estimate:

$y_{i,0} = \alpha + X_{i,0}\theta + \epsilon_{i,0}$

$y_{i,1} = \alpha + X_{i,1}\theta + \epsilon_{i,1}$

Or:

$y_{ij} = \alpha + X_{ij}\theta + \epsilon_{ij}$

Fixed slopes

Intercepts may be different between the groups. In this case we can instead use the model:

$y_{ij} = \alpha + X_{ij}\theta + \xi_j + \epsilon_{ij}$

There are different ways of estimating this model:

- Pooled OLS
- Fixed effects
- Random effects

72.1.2 Unbalanced data

72.2 The pooled OLS estimator

72.2.1 Pooled OLS

Introduction

Our model is:

$y_{ij} = \alpha + X_{ij}\theta + \xi_j + \epsilon_{ij}$

The pooled OLS estimator

72.3 The fixed effects estimator

72.3.1 Within and between transformation

Introduction

We can group the data in two ways, one gets between differences and the other within differences.

In the above example, we could find the effects of schools, or of departments.

$y_{ij} = \alpha + X_{ij}\theta + \epsilon_{ij}$

$(y_{ij} - \bar{y}_j) = (\alpha - \bar{\alpha}) + (X_{ij} - \bar{X}_j)\theta + (\epsilon_{ij} - \bar{\epsilon}_j)$

$(y_{ij} - \bar{y_j}) = (X_{ij} - \bar{X_j})\theta + (\epsilon_{ij} - \bar{\epsilon}_j)$

Or alternatively:

$(y_{ij} - \bar{y_i}) = (X_{ij} - \bar{X_i})\theta + (\epsilon_{ij} - \bar{\epsilon}_i)$

Regardless of the form we choose, we can write this as:

$\ddot{y}_{ij} = \ddot{X}_{ij}\theta + \ddot{\epsilon}_{ij}$

72.3.2 The fixed effects estimator

Recap on the model

Our model is:

$y_{ij} = \alpha + X_{ij}\theta + \xi_j + \epsilon_{ij}$

The fixed effects estimator

With fixed effects we assume that U_{ij} is a constant for each group. That is:

$U_{ij} = \delta_{ij}U_j$

$y_{ij} = \alpha + X_{ij}\theta + \epsilon_{ij} + \delta_{ij}U_j$

We can use this in a regression if the standard assumptions of OLS are met. In particular, that group membership is uncorrelated with the error term.

We add these dummies to X_{ij} and regress:

$y_{ij} = \alpha + X_{ij}\theta + \epsilon_{ij}$

The parameter for the dummy is the fixed effect of group membership.

As we are including membership in the dependent variables, there is no problem if group membership correlates with other independent variables.

Using the within and between transformations

$(y_{ij} - \bar{y_i}) = (X_{ij} - \bar{X_i})\theta + (U_{ij} - \bar{U_i}) + (\epsilon_{ij} - \bar{\epsilon}_i)$

Or:

$\ddot{y}_{ij} = \ddot{X}_{ij}\theta + \ddot{\epsilon}_{ij}$

This this get the same outcome, but is a different computational process.

72.4 The random effects estimator

72.4.1 The random effects estimator

Introduction

Our model is:

$y_{ij} = \alpha + X_{ij}\theta + \xi_j + \epsilon_{ij}$

FGLS recap

The random effects estimator

For fixed effects, we had the requirement that group membership be uncorrelated with the error term, but that it could be correlated with other independent variables.

For random effects models, group membership cannot be correlated with other variables.

We have:

$y_{ij} = \alpha + X_{ij}\theta + \epsilon_{ij} + U_{ij}$

We now model $U_{ij} = \bar{U}_j + \rho_j$.

$y_{ij} = \alpha + X_{ij}\theta + \epsilon_{ij} + \bar{U}_j + \rho_j$

This randomness of the effect implies, for example, that if we ran the survey again we would expect a different effect

Clustering standard error

Estimation

We use GLS.

72.5 Choosing the model form

72.5.1 The Hausman specification test

Introduction

The Hausman specification test allows you to choose between a fixed effects model and a random effects model.

Efficiency

Random effects models are more efficient.

72.6 The mixed effects estimator

72.6.1 The mixed effects estimator

Introduction

72.7 Manipulating data

72.7.1 Disaggregation

Used in polls

72.7.2 Multilevel Regression with Poststratification (Mr P)

Part XVII

Advanced inference

Chapter 73

Analysis of variance (ANOVA)

73.1 Cross-sectional data

73.1.1 Cross-sectional data

73.1.2 Group means and the grand mean

Introduction

73.1.3 Within-group variance and between-group variance

Introduction

73.2 Analysis of variance (ANOVA)

73.2.1 Analysis of variance (ANOVA) table

Chapter 74

Instrumental Variables

74.1 Motivation

74.1.1 Bias of OLS estimator from ommitted variables

74.1.2 Bias of OLS estimator from measurement error

74.2 Parameter estimation for simultaneous equations

74.2.1 Structural and reduced forms

74.2.2 Parameter identification problem with simultaneous equations

Identification terminology

A system is under-identified if there are not enough estimators for all structural parameters.

A system is exactly identified if there are the same number of estimators as structural parameters.

A system is over-identified if there are more estimators than structural parameters.

In general we have in our structural form:

$\sum_i^n \beta_{ij} y_i = \sum_i^m \gamma_{ij} x_i + \epsilon_j$

This is a system with n endogeneous variables and m exogeneous variables.

We can write this in matrix form.

$B\mathbf{y} = \Gamma\mathbf{x} + \epsilon$

We can use this to get:

$\mathbf{y} = B^{-1}\Gamma\mathbf{x} + B^{-1}\epsilon$

We estimate by placing restrictions on Γ.

Strucutral models

If our data generating process is:

$Q = \alpha + \beta P + \epsilon$

We can estimate $\alpha$and β through measuring P and Q.

If, however the data generating process involves simulataneous equations, we can have:

$Q = \alpha_1 + \beta_1 P + \epsilon_1$

$Q = \alpha_2 + \beta_2 P + \epsilon_2$

Reduced form

We can reduce this:

$\alpha_1 + \beta_1 P + \epsilon_1 = \alpha_2 + \beta_2 P + \epsilon_2$

$(\alpha_1 - \alpha_2) + (\beta_1 - \beta_2)P + (\epsilon_1 - \epsilon_2) = 0$

$$P = \frac{\alpha_2 - \alpha_1}{\beta_1 - \beta_2} + \frac{\epsilon_2 - \epsilon_1}{\beta_1 - \beta_2}$$

We can rewrite this as:

$P = \pi_1 + \tau_1$

Similarly we can reduce for Q:

$$Q = \frac{\alpha_2\beta_1 - \alpha_1\beta_2}{\beta_1 - \beta_2} + \frac{\beta_1\epsilon_2 - \beta_2\epsilon_1}{\beta_1 - \beta_2}$$

$Q = \pi_2 + \tau_2$

We can't directly estimate structural models

If P is correlated with $epsilon_1$ or ϵ_2 then our estimates for β_1 and β_2 will be biased.

This also affects Q.

From the reduced forms we can see that P will be correlated, due to simultaneity.

The identification problem

We can estimate π_1 and π_2, but this does not allow us to identify any of the structural parameters.

We have 2 estimators, but 4 parameters.

This is the identification problem.

74.3 2 Stage OLS

74.3.1 2 Stage OLS (2SOLS) estimator

Motivation

If x is correlated with the error term the OLS estimate will be biased.

2 Stage OLS - first stage

We have

$y_i = x_i\theta + \epsilon_i$

$x_i = z_i\rho + \mu_i$

We do OLS on the second to get $\hat{\rho}$.

$\hat{\rho} = (Z^T Z)^{-1} Z^T X$

We use this to get predicted values of X.

$\hat{X} = Z\rho = Z(Z^T Z)^{-1} Z^T X = P_Z X$

2 Stage OLS - second stage

We then regress y on the estimated X:

$y_i = \hat{x}_i\theta + \epsilon_i$

Our prediction is then:

$\hat{\theta_{2SOLS}} = (\hat{X^T}\hat{X})^{-1}\hat{X^T}y)$

$\hat{\theta_{2SOLS}} = ((P_ZX)^TP_ZX)^{-1}(P_ZX)^Ty)$

$\hat{\theta_{2SOLS}} = (X^TP_ZX)^{-1}X^TP_Zy)$

If the dimension of Z is the same as X this collapses to:

$\hat{\theta_{2SOLS}} = (Z^TX)^{-1}Z^Ty$

74.3.2 Bias of the 2SOLS estimator

74.3.3 Variance of the 2SOLS estimator

74.4 More

74.4.1 Identification through exogeneous variables

Previously our structural model was:

$Q = \alpha_1 + \beta_1P + \epsilon_1$

$Q = \alpha_2 + \beta_2P + \epsilon_2$

And our reduced form:

$$P = \frac{\alpha_2 - \alpha_1}{\beta_1 - \beta_2} + \frac{\epsilon_2 - \epsilon_1}{\beta_1 - \beta_2}$$

$$Q = \frac{\alpha_2\beta_1 - \alpha_1\beta_2}{\beta_1 - \beta_2} + \frac{\beta_1\epsilon_2 - \beta_2\epsilon_1}{\beta_1 - \beta_2}$$

Or:

$P = \pi_1 + \tau_1$

$Q = \pi_2 + \tau_2$

Adding another variable

This time we add another measured variable, I.

$Q = \alpha_1 + \beta_1P + \theta_1I + \epsilon_1$

$Q = \alpha_2 + \beta_2 P + \theta_2 I + \epsilon_2$

The reduced form is now:

$$P = \frac{\alpha_2 - alpha_1}{\beta_1 - \beta_2} + \frac{\theta_2 - \theta_1}{\beta_1 - \beta_2} I + \frac{\epsilon_2 - \epsilon_1}{\beta_1 - \beta_2}$$

$$Q = \frac{\alpha_2\beta_1 - \alpha_1\beta_2}{\beta_1 - \beta_2} + \frac{\theta_2\beta_1 - \theta_1\beta_2}{\beta_1 - \beta_2} I + \frac{\beta_1\epsilon_2 - \beta_2\epsilon_1}{\beta_1 - \beta_2}$$

Or:

$P = \pi_{11} + \pi_{12} I + \tau_1$

$Q = \pi_{21} + \pi_{22} I + \tau_2$

We can estimate π_1 and π_2 as $\hat{\pi}_1$ and $\hat{\pi}_2$ respectively.

We can now create estimators $\hat{\pi}_{11}$, $\hat{\pi}_{12}$, $\hat{\pi}_{21}$ and $\hat{\pi}_{22}$.

Identification with an exogeneous variable

We now have 4 estimators and 6 parameters, meaning that we still cannot identify the model.

Partial identification

Can we use $\hat{\pi}$ to identify any of the structural parameters?

We know that:

- $\pi_{11} = \frac{\alpha_2 - \alpha_1}{\beta_1 - \beta_2}$
- $\pi_{12} = \frac{\theta_2 - \theta_1}{\beta_1 - \beta_2}$
- $\pi_{21} = \frac{\alpha_2\beta_1 - \alpha_1\beta_2}{\beta_1 - \beta_2}$
- $\pi_{22} = \frac{\theta_2\beta_1 - \theta_1\beta_2}{\beta_1 - \beta_2}$

If the exogenous variable only affects one side of the equation, so $\theta_1 = 0$, we have:

- $\pi_{11} = \frac{\alpha_2 - \alpha_1}{\beta_1 - \beta_2}$
- $\pi_{12} = \frac{\theta_2}{\beta_1 - \beta_2}$
- $\pi_{21} = \frac{\alpha_2\beta_1 - \alpha_1\beta_2}{\beta_1 - \beta_2}$

- $\pi_{22} = \dfrac{\theta_2\beta_1}{\beta_1 - \beta_2}$

So we can see that:

$\hat{\beta}_1 = \dfrac{\hat{\pi}_{22}}{\hat{\pi}_{12}}$

This means we now have:

- $\pi_{11} = \dfrac{\pi_{12}(\alpha_2 - \alpha_1)}{\pi_{22} - \pi_{12}\beta_2}$
- $\pi_{12} = \dfrac{\pi_{12}\theta_2}{\pi_{22} - \pi_{12}\beta_2}$
- $\pi_{21} = \dfrac{\pi_{12}(\alpha_2\beta_1 - \alpha_1\beta_2)}{\pi_{22} - \pi_{12}\beta_2}$
- $\pi_{22} = \dfrac{\pi_{12}\theta_2\beta_1}{\pi_{22} - \pi_{12}\beta_2}$

We can use this to also identify α_1.

Complete identification

If we have independent variables for each of the two equations, we can fully identify the model.

We will have 6 estimators and 6 parameters.

We are estimating:

$Q = \alpha_1 + \beta_1 P + \theta_1 I + \epsilon_1$

$Q = \alpha_2 + \beta_2 P + \theta_2 J + \epsilon_2$

I and J are essentially instrumental variables for the model.

I is an instrumental variable for demand shocks, and J is an instrumental variable for supply shocks.

74.4.2 Power of instruments

74.5 The Instrumental Variable (IV) estimator

74.5.1 Instrumental Variable (IV) estimator

$\hat{\theta_{IV}} = (Z^T X)^{-1} Z^T y$

2SOLS collpases to IV in some circumstances.

74.5.2 Bias of the IV estimator

Equal to actual parameter so long as ϵ uncorrelated with Z.

74.5.3 Variance of the IV estimator

In OLS we had:

$\hat{\theta_{OLS}} = (X^T X)^{-1} X^T y$

$Var[\hat{\theta_{OLS}}] = (X^T X)^{-1} X^T \Omega X (X^T X)^{-1}$

With IV we have

$\hat{\theta_{IV}} = (Z^T X)^{-1} Z^T y$

$Var[\hat{\theta_{IV}}] = (Z^T X)^{-1} Z^T \Omega Z (Z^T X)^{-1}$

We can use weighted least squares for Ω.

74.6 Choosing instrumental variables

74.6.1 Double selection

74.7 Other

74.7.1 Natural experiments

74.7.2 Non-linear models in the first stage

74.7.3 Random Effects Instrumental Variables (REIV)

74.7.4 Fixed Effects Instrumental Variables (FEIV)

74.7.5 SORT

synthetic IV indep on nuisance as alternative to matching.

IV: h3 on non-linear models for first stage

discontinuity

controlled experiments

two sources: missing data and simultaneous

variations in government rollouts, lotteries

IV may only affect subset of individuals

For example IV of draft number for military service. This only is an instrument for conscripts, not volunteers

generally, need to rationalise this and time series. There's stuff there on natural experiments etc

define confounding in IV? or in dependent variables? is different issue to the one of correlation with error?

h3 on Limited Information Maximum Likelihood

h3 on K-class estimation

Contrast loss and Siamese h3? One shot classification

IV: frame around parameter estimation when don't observe some variables. This can mean the direct variable can't be measured, or that some controls can't be measured

which factors to include? All?

page on structural and reduced forms

h3 on simultaneous equations there? Eg $y = c_1 + \theta_1 X + \epsilon_1$ $y = c_2 + \theta_2 X + \rho Z \epsilon_2$

We can turn this into the reduced form: $y = c_3 + \theta_3 Z + \epsilon_3$ $y = c_4 + \theta_4 Z + \epsilon_4$

difference between confounding and correlation with error?

Chapter 75

Missing data and measurement error

75.1 Non-negative matrix factorisation

75.1.1 Non-negative matrix factorisation

75.2 Recommenders

75.2.1 The recommendation problem

We have a collection of users and a collection of products. We want to recommend products to customers.

Once a customer "consumes" something, we get feedback. however for vast majority of items, not consumed. goal: predict feedback score beforehand to make good recommendations.

We have a matrix of how much each customer "likes" things. however this is v sparce.

75.3 Approaches

75.3.1 Content filtering

We have metadata on customers and products. Eg stated preferences, genres, actors etc.

We use this to create recommendations.

75.3.2 Collaborative filtering

We look at the actions of customers. We then recommend things based on customers who are similar.

Doesn't need stated preferences, or metadata on content.

Needs data on customers (behavours, etc)

2 steps:

- Look for similar users
- Use ratings from those users to make predictions for missing items

75.3.3 Cold start

When you start you have no data on views. This is the cold start problem.

75.4 Other

75.4.1 Omitted variable bias

75.5 Measurement error

75.5.1 Missing data

In missing data: missing data mechanism page, missing at random page. Prediction with missing data; inference with missing data. Should be relatively late page

Missing time series data (ARIMA interpolation?) Last Observation Carried Forward (LOCF); Next Observation Carried Backward (NOCB), linear interpolation,

Deleting whole row if missing data (bias if not random)

Interpolation: mean, conditional (in IID, different for time series)

Create multiple imputations, do analysis, then combine results

Multiple imputation

Multi period averages for imputation on time series

Missing Completely At Random (MCAR), different to MAR

Chapter 76

Semi-parametric regression

76.1 The Robinson estimator

76.1.1 Partially linear models

76.1.2 The Robinson estimator

Partialling out

$y_i = \mathbf{x}_i\theta + g(\mathbf{z}_i) + \epsilon_i$

Consider:

$E(y_i|\mathbf{z}_i) = E(\mathbf{x}_i\theta + g(\mathbf{z}_i) + \epsilon_i|\mathbf{z}_i)$

$E(y_i|\mathbf{z}_i) = E(\mathbf{x}_i\theta|\mathbf{z}_i) + E(g(\mathbf{z}_i)|\mathbf{z}_i) + E(\epsilon_i|\mathbf{z}_i)$

$E(y_i|\mathbf{z}_i) = E(\mathbf{x}_i|\mathbf{z}_i)\theta + g(\mathbf{z}_i)$

We can now remove the parametric part:

$y_i - E(y_i|\mathbf{z}_i) = \mathbf{x}_i\theta + g(\mathbf{z}_i) + \epsilon_i - E(\mathbf{x}_i|\mathbf{z}_i)\theta - g(\mathbf{z}_i)$

$y_i - E(y_i|\mathbf{z}_i) = (\mathbf{x}_i - E(\mathbf{x}_i|\mathbf{z}_i))\theta + \epsilon_i$

We define:

- $\bar{y}_i = y_i - E(y_i|\mathbf{z}_i)$
- $\bar{x}_i = \mathbf{x}_i - E(\mathbf{x}_i|\mathbf{z}_i)$

$\bar{y}_i = \bar{x}_i\theta + \epsilon_i$

Estimating $\bar{y}_i$ and $\bar{x}_i$

So we can use OLS if we can estimate.

- $E(y_i|\mathbf{z}_i)$
- $E(\mathbf{x}_i|\mathbf{z}_i)$

We can do this with non-parametric methods.

76.1.3 Bias and variance of the Robinson estimator

robinson: can't have confounded in dummy. but can in real. general result of propensity stuff?

Framing: Partialling out is an alternative to OLS where $n << p$ doesn't hold. alterntive to LASSO etc

$\hat{\theta} \approx N(\theta, V/n)$

$V = (E[\hat{D}^2)^{-1} E[\hat{D}^2 \epsilon^2](E[\hat{D}^2])^{-1}$

These are robust standard errors.

Moments of the Robinson estimator

If IID then

$$Var(\hat{\theta}) = \frac{\sigma_\epsilon^2}{\sum_i (x_i - \hat{X}_i)^2}$$

Otherwise, can use GLM

What are the properties of the estimator?

$$E[\hat{\theta}] = E[\frac{\sum_i (X_i - \hat{X}_i)(y_i - \hat{y}_i)}{\sum_i (x_i - \hat{X}_i)^2}]$$

76.1.4 Non-linear treatment effects in the Robinson estimator

Page on reformulating as non-linear. can do it. show can be estimated using arg min https://arxiv.org/pdf/1712.04912.pdf

76.1.5 DML

in DML. page on orthogonality scores, page on constructing them; page on using them to estimate parameters (GMM)

We have $P(X) = f(\theta, \rho)$ $\hat{\theta} = f(X, n)$ $\theta = g(\rho, X)$

So error is: $\hat{\theta} - \theta = f(X, n) - g(\rho, X)$

Bias is defined as: $Bias(\hat{\theta}, \theta) = E[\hat{\theta} - \theta] = E[\hat{\theta}] - \theta$ $Bias = E[\hat{\theta} - \theta] = E[f(X, n) - g(\rho, X)]$ $Bias = E[\hat{\theta} - \theta] = E[f(X, n)] - g(\rho, X)$

double ML: regression each parametric parameter on ML of other variables. eg: get $e(x|z)$ $e(d|x)$ $d = m(x) + v$ d is correlated with x so bias. v is corrleated with d but not x. use as "iv". Still need estimate for $g(x)$.

for iterative, process is: + estimate $g(x)$ + plug into other and estimate theta + this section should be in sample splitting. rename iterative estimation. separate pages for bias, variance + how does this work?? paper says random forest regression and OLS. intialise θ randomly? + page on bias, variance, efficiency? + page on sample splitting, why?

+ page on goal: x and z orthogonal for split sampling + page on $X = m_0(Z) + \mu$, first stage machine learning, synthetic instrumental variables? h3 on that for multiple variables on interest. regression for each

76.1.6 DML1

Divide into k.

For each do ML on nuicance (how???) use all instances outside of sample

Then do GMM using orthogonality condition to calculate θ. (how??) use instances in sample

Average θ from each class

76.1.7 Last stage Robinson

Separate page for last stage: note we can do OLS, GLS etc with choice of Ω.

76.2 Causal trees

Chapter 77

Homogeneous treatment effects

77.1 Introduction

77.1.1 Treatment data

Recap

With multilevel data with fixed coefficients we have:

$y_{ij} = \mathbf{x}_{ij}\theta + m_j + \epsilon_{ij}$

We can estimate m_j using fixed effects or similar methods.

Treatment data

If the data is grouped by whether an entity was treated then will have:

- y_{i0} - the outcome if the entity was not treated
- y_{i1} - the outcome if the entity was treated

However we only observe y_i and D_i.

$y_i = y_{i0} + D_i(y_{i1} - y_{10})$

77.1.2 Average Treatment Effects (ATE, ATET, ATEUT)

Average Treatment Effect (ATE)

$ATE = E[y_{i1} - y_{i0}]$

Average Treatment Effect on the Treated (ATET)

$ATE = E[y_{i1} - y_{i0}|D_i = 1]$
$ATE = E[y_{i1}|D_i = 1] - E[y_{i0}|D_i = 1]$

Average Treatment Effect on the Untreated (ATEUT)

77.1.3 Conditional Average Treatment Effect (CATE)

$E[y_{i1} - y_{i0}|\mathbf{x}_i]$

77.2 Exogenous treatment

77.2.1 Randomly Controlled Trials (RCTs)

If the model is:

$y_i = D_i\theta + g(X) + \epsilon_i$

And D is randomly assigned, then we can estimate

$y_i = D_i\theta + \epsilon_i$

To get an estimate for θ without collecting data on X.

77.2.2 Calculating CATEs in RCTs with interaction terms

77.2.3 Calculating CATEs in RCTs with subgroup analysis

77.3 Calculating treatment effects without estimating missing data

77.3.1 Regression

We can simply regress outcomes on variables, including treatment.

This assumes treatment effects are constant.

This also assumes that outcomes y_{1i} and y_{0i} are independent of D_i, conditional on X.

If we are missing variables in X then we will have biased estimates.

This also assumes the effects of X are linear.

We assume: $E[y_{0i}|\mathbf{x}_i, D_i] = \mathbf{x}_i\theta$.

77.3.2 Instrumental Variables and natural experiments

77.3.3 Regression discontinuity

77.3.4 Synthetic controls

77.4 Calculating treatment effects by estimating missing data

77.4.1 Matching

Matching is similar to regression. We assume that effects are constant, and the effect of treatment on y_{0i} and y_{1i} are independent of treatment, once controlling for X.

Again, this is biased if this is not the case.

We however do not have to assume a linear form for X.

We assume: $E[y_{ji}|\mathbf{x}_i, D_i] = E[y_{ji}|\mathbf{x}_i]$

For each entity, find a near entity which had the opposite treatment.

77.4.2 Propensity score matching

Match on the chance of getting treatment, given covariates.

77.4.3 Matrix completion

$E[y_{i1} - y_{i0}|\mathbf{x}_i]$

77.5 Using semi-parametric

77.6 Other

77.6.1 Estimating ATE using MCMC

77.6.2 Local Average Treatment Effect (LATE)

We have IVs for treatment.

77.6.3 Treatment effects

+ propsentiy score weighting + regression adjustemnt + matching + IV + Regression discontinuity

77.6.4 Meta analysis

big page in advanced analytics? Random effects meta analysis?

meta analysis: fixed effect v random effects model

types of study: + RCT + cohort studies + case-control studies + cross sectional studies

77.6.5 Dose response curve

77.6.6 Sensitivity analysis

77.6.7 Page on Rubin causal model

Chapter 78

Heterogeneous treatment effects

78.1 Heterogenous treatment effects

78.1.1 Introduction

78.1.2 subgroup analysis

78.1.3 interaction terms

78.1.4 efficient policy learning

78.1.5 Het DML

$y = a(z)+db(z)$ Het effects is $b(z)$ We build groups instead of arbitrary function. So we estimate $E[b(z)|G]$

Use part of the data set to estimate

$\hat{y} = \hat{a}(z) + D\hat{b}(z)$

Use $s = \hat{b}(z)$ to stratify. Key point is defining subgroups algorithmically. Less opportunity for hacking

78.1.6 Continuous treatment effects

78.1.7 Intent-to-treat

78.2 (LATE, causal tree (from CART))

78.2.1 Introduction

bart causal is different to causal tree

In stuff now two problems: + non random but constant effect + Random but heterogenous effect

causal trees can find heterogenous treatment effects

Approaches: We have treated and untreated. X and y Estimate y—x for treated, and untreated separately. Then take differnece for a given x to be the estimated treatment effect

2nd approach: have treatment as input diffence is again y—x - y—x treatment minus no treatment

3rd approach: (type of single tree) split not by predictive power, but by treatment effect differnece

4th approach: cross validation at each leaft we note the sample average treatment effect goal is to choose hyper parameters which minimise sum of diffence between these and cross valid data

Once we have the trees from the last one, calculate the effect using test data. nb: separate creating of tree to estimation of treatment effect

78.2.2 Instrumental forests

Estimate LATE

like causal forest, but do IV regression on leaf.

Part XVIII

Estimating time series models

Chapter 79

Estimating Markov chains

79.1 Estimating Markov chains

79.1.1 Estimating the Markov chain stochastic matrix

Introduction

Given a sequence: $x_1, ...x_n$.

The likelihood is:

$L = \prod_{i=2}^{n} p_{x_{i-1},x_i}$

If there are k states we can rewrite this as:

$L = prod_{i=1}^{k} \prod_{j=1}^{k} n_{ij} p_{ij}$

Where p_{ij} is the chance of moving from state i to state j, and n_{ij} is the number of transtions between i and j.

The log likelhood is:

$\ln L = \sum_{i=1}^{k} \sum_{j=1}^{k} n_{ij} \ln p_{ij}$

Constrained optimisation

Not all parameters are free. All probabilities must sum to 1.

$\ln L = \sum_{i=1}^{k} \sum_{j=1}^{k} n_{ij} \ln p_{ij} - sum_{i=1} \lambda_i (\sum_{j=1} p_{ij} - 1)$

This gives us:

$\hat{p}_{ij} = \frac{n_{ij}}{\sum_k n_{ik}}$

79.1.2 Estimating infinite state Markov chains

We can represent the transition matrix as a series of rules to reduce the number of dimensions

$P(x_t|y_{t-1}) = f(x, y)$

can represent states as number, rather than atomic. could be continuous, or even real.

in more complex, can use vectors.

Chapter 80

Estimating Hidden Markov Models (HMMs)

80.1 Estimating Hidden Markov Models (HMMs)

80.1.1 Recap of Hidden Markov Models (HMMs)

We dont see state

Each state produces a visible output. this output is drawn from a distribution for each state.

We observe a sequence of outputs, not states.

80.1.2 Estimating HMMs with the Viterbi algorithm

Assume we know transition matrix. and starting probls

Given we observe sequence of outputs, what were most likely actual paths?

Virbiti returns this

80.1.3 Estimating HMMs with the forward algorithm

Given we have observed outputs, what is the chance of being in a certain state at a certain time?

80.1.4 Estimating HMMs with the forward-backward algorithm

We calculate state x at time t given all obs.

80.1.5 Baum-Welch algorithm

80.1.6 Kalman filters

Chapter 81

Univariate forecasting

81.1 Introduction

81.1.1 Seasonal and non-seasonal trends

We can model the process as:

$y_t = \mu_t + f(t) + \epsilon_t$

81.1.2 Identifying the order of integration using Augmented Dickey-Fuller

The Dickey-Fuller test with deterministic time trend was:

$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \epsilon_t$

The Augmented Dickey-Fuller model adds lags for the differences.

$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \sum_i^p \delta_i \Delta y_{t-i} + \epsilon_t$

81.1.3 Cyclical fluctuations

We can have shocks having effects over time.

This is separate to trends.

81.1.4 Identifying serial correlation using the Durbin-Watson statistic

81.1.5 Introduction to forecasting

We observe a series of observations:

$x_1, x_2, ..., x_t)$

What can we say about x_{t+1}?

If the data was drawn iid then the past data then we would just want to identify moments.

However if the data is not iid, for example because it is increasing in time, then this is not the best way.

Regression formation

We can model

$x_t = \alpha + \epsilon_t$

81.2 Autoregressive model

81.2.1 Autoregressive models (AR)

AR(1)

Our basic model was:

$x_t = \alpha + \epsilon_t$

We add an autoregressive component by adding a lagged observation.

$x_t = \alpha + \beta x_{t-1} + \epsilon_t$

AR(p)

AR(p) has p previous dependent variables.

$x_t = \alpha + \sum_{i=1}^{p} \beta_i x_{t-i}$

Propagation of shocks

A shock bumps up the output variable, which bumps up output variables forever, at a decreasing rate.

81.2.2 Testing for stationarity with Dickey-Fuller (DF) and Augmented Dicky-Fuller (ADF)

Stationarity

Unit roots

Integration order

Dickey-Fuller

The Dickey-Fuller test tests if there is a unit root.

The AR(1) model is:

$y_t = \alpha + \beta y_{t-1} + \epsilon_t$

We can rewrite this as:

$\Delta y_t = \alpha + (\beta - 1) y_{t-1} + \epsilon_t$

We test if $\beta - 1) = 0$.

If the coefficient on the last term is 1 we have a random walk, and the process is non-stationary.

If the last term is < 1 then we have a stationary process.

Variation: Removing the drift

If our model has no intercept it is:

$y_t = \beta y_{t-1} + \epsilon_t$

$\Delta y_t = (\beta - 1) y_{t-1} + \epsilon_t$

Variation: Adding a deterministic trend

If our model has a time trend it is:

$y_t = \alpha\beta y_{t-1} + \gamma t + \epsilon_t$

$\Delta y_t = \alpha + (\beta - 1) y_{t-1} + \gamma t + \epsilon_t$

Augmented Dickey-Fuller

We include more lagged variables.

$y_t = \alpha + \beta t + \sum_i^p \theta_i y_{t-i} + \epsilon_t$

If no unit root, can do normal OLS?

81.2.3 Autoregressive Conditional Heteroskedasticity (ARCH)

Variance of the AR(1) model

The standard AR(1) model is:

$y_t = \alpha + \beta y_{t-1} + \epsilon_t$

The variance is:

$Var(y_t) = Var(\alpha + \beta y_{t-1} + \epsilon_t)$

$Var(y_t)(1 - \beta^2) = Var(\epsilon_t)$

Assuming the errors are IID we have:

$$Var(y_t)) = \frac{\sigma^2}{1 - \beta^2}$$

This is independent of historic observations, which may not be desirable.

Conditional variance

Consider the alternative formulation:

$y_t = \epsilon_t f(y_{t-1})$

This allows for conditional heteroskedasticity.

81.3 Moving average models

81.3.1 Moving Average models (MA)

We add previous error terms as input variables

MA(q) has q previous error terms in the model

Unlike AR models, the effects of any shocks wear off after q terms.

This is harder to fit the OLS, the error terms themselves are not observed.

81.4 Autoregressive Moving Average models

81.4.1 Autoregressive Moving Average models (ARMA)

We include both AR and MA

Estimted using Box-Jenkins

81.4.2 Autoregressive Integrated Moving Average models (ARIMA)

Uses differences to remove non statiority

Also estiamted with box-jenkins

81.4.3 Seasonal ARIMA

81.5 Forecasting

81.5.1 Monte carlo simulations

81.5.2 N-step ahead

81.5.3 Consensus forecasting

Chapter 82

Multivariate forecasting

82.1 Introduction to multiple time series

82.1.1 Testing for cointegration with Johansen

82.2 Vector Autoregression (VAR)

82.2.1 Vector Autoregression (VAR)

We consider a vector of observables, not just one

Autoregressive (AR) model for a vector.

VAR(p) looks p back.

The AR(p) model is:

$y_t = \alpha + \sum_{i=1}^{p} \beta y_{t-i} + \epsilon_t$

VAR(p) generalises this to where y_t is a vector. We define VAR(p) as:

y_t

$y_t = c + \sum_{i=1}^{p} A_i y_{t-i} + \epsilon_t$

82.2.2 VAR impulse response

82.2.3 Bayesian VAR

82.3 Structural models

82.3.1 Autoregressive Distributed Lag (ARDL) model

Include lagged y and lagged x (and current x)

If the processes are stationary, then we can use OLS. THIS IS A BROADER POINT! INTRO??

82.4 ARMAX

82.4.1 ARMAX

82.4.2 Error Correction Model

Static model

Like PAM we start with static estimator.

The ECM

The ECM does a regression with first differences, and includes lagged error terms.

We start with a basic first-difference model.

$\Delta y_t = \Delta x_t$

We could also expand this to include laggs for both x and y. Here we don't.

We know that long term $y_t = \theta x_t$. We use the error from this in a first difference model.

$\Delta y_t = \alpha \Delta x_t + \beta(y_{t-1} - \theta x_{t-1})$

Page on identifying error terms

Also, page on Vector Error Correction Model (VECM)

82.4.3 Partial Adjustment Model

Estimating a static model

We start by estimating a static model.

$y_t = \alpha + \theta x_t + \gamma_t$

Equilibrium

We then use this form an equilibrium for y_t, y_t^*.

$y_t^* = \hat{\alpha} + \hat{\theta} x_t$

The process depends on the difference from this equilibrium.

$y_t - y_{t-1} = \beta(y_t^* - y_{t-1}) + \epsilon_t$

$y_t - y_{t-1} = \beta(\hat{\alpha} + \hat{\theta} x_t - y_{t-1}) + \epsilon_t$

$y_t = \beta\hat{\alpha} + \beta\hat{\theta} x_t + (1-\beta) y_{t-1} + \epsilon_t$

$y_t = \alpha y_{t-1} + (1-\beta)(y_t^* - y_{t-1}) + \epsilon$

The higher β, the slower the adjustment.

If stationary, can we can use OLS.

Chapter 83

Inference with time series

83.1 OLS on time series data

83.1.1 Bias of static models and spurious correlations

Static models

Static models are of the form:

$y_t = \alpha + \beta x_t + \epsilon_t$

These have no lagged variables or difference operators.

Bias of static models

83.1.2 Heteroskedasticity and Autocorrelation (HAC) adjusted standard errors

83.2 Time series

83.2.1 Taking differences

What we use should depend on I(1), I(0) etc from ADF

if we're missing time invariant data, we can do first differences and this isn't a problem if we do diff in diff this removes trends?

page on first difference estiamtion? OLS on first differences. No other lags page on first difference ESTIMATOR

83.2.2 Discontinuity

Create a dummy for before/after a date.

83.3 Panel data

83.3.1 Difference-in-difference

Consider the grouped linear model:

$y_{ij} = \mu + \tau_i + X_j\theta + \epsilon_{ij}$

By taking differences with another observation in the same group we remove the average terms.

$y_{ij} - y_{ik} = (\mu + \tau_i + X_j\theta + \epsilon_{ij}) - (\mu + \tau_i + X_k\theta + \epsilon_{ik})$

$y_{ij} - y_{ik} = (X_j\theta - X_k) + (\epsilon_{ij} - \epsilon_{ik})$

diff in diff: control group and treated group. page on leakiness? are control affected too? Assumption: in absense of treatment, price would have evolved like control

83.3.2 Controlled experiments

83.3.3 Natural experiments

83.3.4 Structural breaks

Testing for structural breaks with the Chow test.

83.3.5 Dynamic or lagged independent variables

Static panel data: No lags of independent variables. Dynamic panel data: Lags of independent variables.

OLS is consistent for static panel data, not for dynamic This results in Nickell's bias for dynamic panel data

Dynamic panel data: y_{t-1} is a regressor Panel data estimation: LSDV. Least squares dummy variable estimator arnello bond

Chapter 84

Natural Language Processing (NLP)

84.1 Other

84.1.1 Probabilistic language models

Introduction

Probabilistic language models can predict future words given a history of words.

This can be used for predictive text. For example if a user types "Did you call your" we may want to estimate the probability that the next word is "child".

We can state this problem:

$P(child|did\ you\ call\ your)$

By definition this is:

$$P(child|did\ you\ call\ your) = \frac{P(did\ you\ call\ your\ child)}{P(did\ you\ call\ your)}$$

We can estimate each of these:

$$P(did\ you\ call\ your\ child) = \frac{|did\ you\ call\ your\ child|}{|5\ word\ sentences|}$$

$$P(did\ you\ call\ your) = \frac{|did\ you\ call\ your|}{|4\ word\ sentences|}$$

Data requirements

This needs a large corpus, which may not be practical.

Additionallly, the words must be indexed, and not simply stored as a bag of words.

Decomposition

We can decompose the probabilities using the chain rule.

$P(did\ you\ call\ your\ child) = P(did)P(you|did)...P(child|did\ you\ call\ your)$

$P(w_1, ..., w_k) = \prod_k p(w_k|w_1, ..., w_{k-1})$

N-grams

We can simplify the decomposition using the Markov assumption:

$P(w_k|w_1, ..., w_{k-1}) = P(w_k|w_{k-1})$

This is a 1-gram.

We can do this for n words back. This is an n-gram.

Smoothing

We can use smoothing to address small corpuses.

$$P(did\ you\ call\ your\ child) = \frac{|did\ you\ call\ your\ child| + 1}{|5\ word\ sentences| + V}$$

$$P(did\ you\ call\ your) = \frac{|did\ you\ call\ your| + 1}{|4\ word\ sentences| + V}$$

For some value V.

Perplexity

We can compare probabilistic language models using perplexity.

We can then choose the model with the lowest perplexity.

$$perplexity(w_1, w_2, ..., w_n) = P(w_1, w_2, ..., w_n)^{-\frac{1}{n}}$$

We can expand this:

$perplexity(w_1, w_2, ..., w_n) = \prod_i P(w_i|w_1, ..., w_{i-1})^{-\frac{1}{n}}$

Depending on which n-gram we use we can then simplify this.

84.1.2 Word2vec

84.1.3 Latent Semantic Analysis

84.2 Machine translation

84.2.1 Machine translation

84.3 Speech

84.3.1 Text-to-speech

84.3.2 Speech-to-text

Chapter 85

Recurrent neural networks

85.1 Simple recurrent neural networks

85.1.1 Simple Recurrent Neural Networks (RNNs)

Introduction

Recurrent Neural Networks (RNN) are an alternative to feedforward networks.

These have loops.

Motivation

We have inputs which are not independent. For example speech input, where each input is a the recording for a length of time.

Unrolling RNNs

The activation unit takes the input, and an outcome from the previous activation unit. It then performs its activation function.

This allows information to be kept across time.

However this degrades, and relevant information was from much earlier, it will be lost.

85.1.2 Backpropagation Through Time (BPTT)

We can do backpropagation on the unrolled network, backpropagating over time.

85.2 Long Short-Term Memory (LSTM)

85.2.1 Long Short-Term Mmemory (LSTM)

Introduction

These are a more complex RNN architecture.

Cell state

Each cell has as an as input the cell state from the previous cell C_{t-1}
The LSTM cell updates the cell state to C_t and pushes it to the next cell.

Other inputs to the cell

We have x_t, the input of the cell, and h_{t-1}, the output of the previous cell.

Cell output and the output gate

We run an activation function on the cell state C_t to get a candidate output.
We multiply this by the outcome of the output gate to get the actual result.

The input gate

We create a candidate change to the state.
We multiply this by the input gate value, and add it to the state.

The forget gate

This is a multiplication factor. What % of the state should be removed?

85.3 Variants

85.3.1 Peephole LTSM

85.3.2 Gated Recurrent Units (GRUs)

85.4 Forecasting with recurrant neural networks

85.4.1 Introduction

85.5 Other

85.5.1 Attention and Neural Turing Machines

Chapter 86

Recurrent Neural Network (RNN) encoders and decoders

86.1 Recurrent Neural Network (RNN) encoders and decoders

86.1.1 Recurrent Neural Network (RNN) encoders

Final output is the encoding.

The end of the sequence is identified through an End-Of-Sequence token.

seq2seq

86.1.2 Recurrent Neural Network (RNN) decoders

Introduction

We take the encoded vector and pass this through to the decoder. This spits out decoded output.

As we output a word, the word (and previous words) are sent as inputs to the following RNN cells.

Encoding the outputs

As we create outputs, we can pass this as an encoded vector in the target language.

Chapter 87

Applied neural networks

87.1 Text

Chapter 88

Audio recognition

88.1 Audio

Part XIX

Communication

Chapter 89

Hashes

89.1 Data integrity checks

89.1.1 Hash functions

Hash functions (take input and return fixed length output) (h=hash(m))

Data integrity checks

Needs to be very different for small changes. so typo has different hash for example. corrput data needs to be noticed.

Checksums

if two files are the same then hashes the same

Introduction

Want following properties for a hash function

Deterministic, so the same hash is always created.

Quick to compute hash

Cannot generate input from hash, except for brute forcing inputs

Small changes to document should cause large charges to hash, such that the two hashes appear uncorrelated

Can't find multiple documents with the same hash, practically.

Can be used to verify files, check passwords.

So possible vulernabilities are:

Given hash, find message (Pre-image resistence)

Given input, find another input with the same hash (second pre-image resistance)

Collission resistance (find two inputs with same hash)

We want to prevent accidental changes to file, and deliberate changes to file. Vulerabilites are more importnat for latter.

89.2 Adversaries

89.2.1 Brute force attacks

89.2.2 Pre-image attacks

Given hash value h, can we find message m?

89.2.3 Defence from pre-image attacks

89.2.4 Second pre-image attacks

Given m_1, can we find m_2 with same hash?

Defence from second pre-image attacks

89.2.5 Hash collision

Can i find any two matching messages?

Hash collision attacks

I can get someone to vouch for one of the messages, and then claim they vouched the other.

Hash collision defence

89.3 Passwords

89.3.1 Plaintext databases

89.3.2 Hashed passwords

89.3.3 Rainbow tables

89.3.4 Dictionary attacks

89.3.5 Salting

It is possible to brute force hashes, especially for smaller inputs such as short passwords.

If password hashes for a hashing algorithm were brute forced, then passwords could easily be recovered from another hash table.

To prevent this a salt can be added to the document.

If a password is "apple", then instead the salt "xyz" could be added to create "applexyz". This prevents the previous cracking of "apple" to be used.

The salt would then be stored in plaintext alongside the password hash.

89.4 Examples of hash functions

89.4.1 SHA

89.4.2 Sort

89.4.3 Data integrity checks

Hash functions (take input and return fixed length output) ($h = hash(m)$) Needs to be very different for small changes. so typo has different hash for example. corrput data needs to be noticed. Checksums (if two files are the same then hashes the same)

89.4.4 Adversaries

Brute force attacks

Pre-image attacks (h3 on attack, h3 on defence: Given hash value h, can we find message m?) Second pre-image attacks (h3 on attack, h3 on defence: given m_1, can we find m_2 with same hash?) Hash collisions (h3 on attack, h3 on defence: can i find any two matching messages?)

89.4.5 Passwords

Plaintext databases

Hashed passwords

Rainbow tables

Salting

89.4.6 Hash functions

+ SHA

Chapter 90

Classical encryption

90.1 Introduction

90.1.1 Plaintext and ciphertext

90.1.2 ROT13

Rotate 13. It is its own inverse.

90.1.3 Atbash

Revese the alphabet. It is its own inverse.

90.2 Verifying decryptions

90.2.1 Corpus

verifying solutions when spaces are omitted. can rate fitness using corpus information on popularity

90.3 Caesar

90.3.1 Caesar ciphers

Shift along in alphabet by c.

90.3.2 Affine cipher

page on affine cipher too. like caesar but rather than +c, mx+c

90.3.3 Breaking

For Caesar, only 26 possible keys, can just brute force.

For Affine, can also brute force.

90.4 Monoalphabetic substitution

90.4.1 Monoalphabetic substitution ciphers and keys

(key plus algorithm encrpyts and decrypts)

90.4.2 Breaking monoalphabetic substitution ciphers with frequency analysis

(need to identify algorithm and needs to identify key)

finding substitution cyphers

Search space is larger, $26! = 4 * 10^26$. need alternative to brute force.

Letter popularity. Compare against popularity for corpus. Monogram (ie letters); ngrams(ie n letter in a row frequency); common words.

Single letter words are I or A. More generally. corpus smaller for fewer letters

Can test substitution cypher by matching each word against a corpus

90.5 Polyalphabetic substitution

90.5.1 Polyalphabetic ciphers

Multiple substitution

Vigenere

Rotor machines

The Enigma machine

90.5.2 Breaking polyalphabetic ciphers with the Kasiski examination

90.6 Other

90.6.1 Codebooks

((eg sdrgdr is code for "meet at x on y")

90.6.2 Transposition ciphers

90.6.3 Book cipher

Eg use Bible.

90.6.4 One-time pads

Chapter 91

Modern symmetric encryption

91.1 Methods

91.1.1 Block ciphers

91.1.2 Stream ciphers

91.1.3 Motivation

Increased computer power. How to be secure?

kerckhoff's principle. choose cipher such that secure even if everything but key is known

91.2 Symmetric encryption

91.2.1 Symmetric

We have a document we want to be able to transfer on an insecure medium.

We use a key to encrypt the file, and a key to decrypt the file.

With symmetric encryption these are the same key.

91.3 Options for algorithms

91.3.1 Integer factorisation

Option for algorithm.

91.3.2 Elliptical-curve cryptography

Chapter 92

Modern asymmetric encryption

92.1 Asymmetric encryption

92.1.1 Public keys

92.1.2 RSA

92.1.3 Message signing

92.1.4 Pretty Good Privacy (PGP)

92.1.5 Using public keys to facilitate symmetric encryption

92.1.6 Elliptical-curve cryptography

92.1.7 Asymmetric encryption

Here we use different keys to encrypt and decrypt the file.

Consider two users who wish to send a message securely.

One option would be to use symmetric encrpytion. They would have to meet and share this key securely, however, as transferring it over an insecure network would mean it could be copied.

With public key encryption each user has a public and a private key. The private key is kept secure locally, while the public key can be broadcasted.

In order to encrypt the file, the recipient's public key is used, while both the private and public key are needed to decrypt the file.

As a result anyone can encrypt a file to send to the user, but only the user can read what is sent.

Public-key encryption can be used to facilitate symmetric encryption. If only one party has a public key then the other user can send a symmetric key securely using the public key.

Using this, asymmetric encrpyiton is only used at the start.

This is how HTTPS operates, where the website has a public key, but the client does not.

Each user still needs to trust that the public key is accurate. This could be done by hosting the public key on a secure location.

RSA is an algorithm used for public-key encrpytion, including for HTTPS handshakes and PGP.

92.1.8 Sort

**** Asymmetric Pages for: + Public keys + RSA + Message signing + PGP + Public keys to facilitate symmetric encryption

Chapter 93

Signal processing

93.1 Introduction

93.1.1 Quantisation

93.1.2 Sample rate

93.1.3 Discrete Fourier Transform

93.1.4 Down sampling

93.1.5 Fast Fourier Transform

93.1.6 Noisy networks