

Tuesday
19/09/23

Course Outline:

1. Review of Linear models
2. Non-linear regression; non-linear least squares, estimation and asymptotic properties
3. Mixed models; Random effects and fixed effects modeling
4. Simple non-parametric regression: Concepts, models, estimators and asymptotic properties
5. Simple neural network regression: Concepts, models, estimators and asymptotic properties
6. Generalized linear models
7. Use of statistical software

CHAPTER 1:

Review of Linear Models

- A problem that often arises in economics, engineering, finance, medicine among other disciplines is that of investigating the mathematical relationship between 2 or more variables
- The goal is often to model a continuous r.v., γ as a function of 1 or more input variables, say, x_1, x_2, \dots, x_k
- One can express this model as:
$$\gamma = g(x_1, x_2, \dots, x_k) + e$$
where $g: \mathbb{R}^k \rightarrow \mathbb{R}$ and e is a random error term
that satisfies certain conditions
- This is called a Regression Model
- The presence of the error term, e conveys that the relationship between γ and the input variables, through $g(x_1, x_2, \dots, x_k)$ is likely not perfect.
- If it were perfect, this would be a deterministic model.
- The input vars x_1, x_2, \dots, x_k are assumed

to be fixed (non-random) and they are measured without error.

- There are diff. types of regression models:

A non parametric regression model would leave the form of g unspecified essentially regarding the relationship between y and the independent variables x_1, x_2, \dots, x_k to be specified by a fn.

A parametric model would dictate the specific form of g . For example,

$$y = B_0 + B_1 x_1 + B_2 x_2 + \dots + B_k x_k + e$$

$\underbrace{\qquad\qquad\qquad}_{g(x_1, x_2, \dots, x_k)}$

where B_0, B_1, \dots, B_k are unknown regression parameters. This is called a linear regression model.

- The adjective "linear" does not refer to the shape of the fn $g(x_1, x_2, \dots, x_k)$ instead it refers to the manner in which the regression parameters, B_0, B_1, \dots, B_k appear in the fn g .

- With the g fn above,

$$\frac{\partial g(x_1, x_2, \dots, x_k)}{\partial \beta_0} = 1$$

$$\frac{\partial g(x_1, x_2, \dots, x_k)}{\partial \beta_1} = x_1$$

$$\frac{\partial g(x_1, x_2, \dots, x_k)}{\partial \beta_k} = x_k$$

- All of these partial derivatives are free of $\beta_0, \beta_1, \dots, \beta_k$ meaning g is a linear fn of the regressor parameters.
- With this def. in mind, we see that all of the following models are linear in the regression parameters:

i) $Y = \beta_0 + \beta_1 x + \epsilon_{\text{rand}}$

$\underbrace{\beta_0 + \beta_1 x}_{g(x)}$

$$\text{ii) } Y = \beta_0 + \beta_1 x + \beta_2 x^2 + e$$

we are not
 focusing on this

$\underbrace{\qquad\qquad}_{g(x)}$

$$\text{iii) } Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + e$$

$\underbrace{\qquad\qquad}_{g(x_1, x_2)}$

$$\text{iv) } Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + e$$

$\underbrace{\qquad\qquad}_{g(x_1, x_2)}$

- These are all examples of linear regression models. This is true even though only 2 of these models are linear functions of the independent variables (the 1st & the 3rd). I think

- An example of a non-linear model is:

$$Y = \frac{\beta_0}{1 + \beta_1 e^{\beta_2 x}} + e$$

The model is not linear in its parameters, e.g.,

$$\frac{\partial g(x)}{\partial \beta_0} = \frac{1}{1 + \beta_1 e^{\beta_2 x}}$$

which is NOT

free of β_1 and β_2

Simple Linear Regression

- A SLR is of the form:

$$Y = \beta_0 + \beta_1 X + e \text{ if } E(e) = 0, \text{ then}$$

$$E(Y) = E(\beta_0 + \beta_1 X + e) = \beta_0 + \beta_1 \alpha$$

- The regression parameter β_1 quantifies the change in $E(Y)$ brought about by a unit change in X .
- The regression parameter β_0 represents the mean of $Y, E(Y)$ when the input var X is zero.

Estimation and Sampling Distribution:

When we say we fit a model or estimate a model, we mean, we would like to estimate the population level model parameters β_0 and β_1 with the observed data.

A widely accepted method of estimating the pop. parameters β_0 and β_1 is to use least squares,

which says to choose the values of β_0 & β_1 that
 minimize the objective fun $Q(\beta_0, \beta_1)$ where :

$$Q(\beta_0, \beta_1) = \sum_{i=1}^n (\gamma_i - \beta_0 - \beta_1 x_i)^2$$

Taking partial derivatives of $Q(\beta_0, \beta_1)$ we get :

$$\frac{\partial Q(\beta_0, \beta_1)}{\partial \beta_0} = -2 \sum_{i=1}^n (\gamma_i - \beta_0 - \beta_1 x_i) = 0$$

$$\frac{\partial Q(\beta_0, \beta_1)}{\partial \beta_1} = -2 \sum_{i=1}^n (\gamma_i - \beta_0 - \beta_1 x_i) x_i = 0$$

Solving for β_0 and β_1 gives :

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

$$\text{and } \hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(\gamma_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

these are the least square estimators of β_0 & β_1 , respectively.

iid \rightarrow Indpd, identically

Assumptions:

- Let's wish to investigate the sampling properties of $\hat{\beta}_0$ and $\hat{\beta}_1$ as estimators of β_0 & β_1 in the SLRM $y_i = \beta_0 + \beta_1 x_i + e_i$ for $i=1,2,\dots,n$
- To do this, we will assume:
 $e_i \sim \text{iid } N(0, \sigma^2)$
- This means that
 - $E(e_i) = 0$
 - $V(e_i) = \sigma^2$
 - the r.v.s e_i are Indpd
 - the r.v.s y_i are normally distributed

Remark Under the assumption that

$e_i \sim \text{iid } N(0, \sigma^2)$, we see that

$$y_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2)$$

- This means that① $E[y_i] = \beta_0 + \beta_1 x_i$, for $i=1,2,\dots,n$
- ② $V(y_i) = \sigma^2$ that is, the variance is constant
- ③ The R.V.s y_i are Indpd (because they are fns of e_i)
- ④ The r.v.s y_i are normally distributed

Result 1:

The least squares estimators $\hat{\beta}_0$ and $\hat{\beta}_1$ are unbiased estimators of β_0 and β_1 , that is, $E(\hat{\beta}_0) = \beta_0$ and $E(\hat{\beta}_1) = \beta_1$.

Result 2:

The least squares estimators, $\hat{\beta}_0$ and $\hat{\beta}_1$, have the following variances:

$$V(\hat{\beta}_0) = \sigma^2 \left[\frac{\sum_{i=1}^n x_i^2}{n \sum_{i=1}^n (x_i - \bar{x})^2} \right]$$

$$V(\hat{\beta}_1) = \sigma^2 \left[\frac{1}{\sum_{i=1}^n (x_i - \bar{x})^2} \right]$$

$$\text{Cov}(\hat{\beta}_0, \hat{\beta}_1) = \sigma^2 \left[-\frac{\bar{x}}{\sum_{i=1}^n (x_i - \bar{x})^2} \right]$$

Result 3:

The least squares estimators, $\hat{\beta}_0$ and $\hat{\beta}_1$, are normally distributed.

Result 4: (Mean Squared Error)

$$-\hat{\sigma}^2 = \frac{SSE}{n-2}$$

where,

$$SSE = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

- $\hat{\sigma}^2$ is an unbiased estimator of the error variance

STATISTICAL INFERENCE AND HYPOTHESIS TESTING

- In the SLRM, $Y_i = \beta_0 + \beta_1 X_i + e_i$ for $i=1, \dots, n$
where $e_i \sim i.i.d N(0, \sigma^2)$
the pop. level reg. parameters β_0 & β_1 are unknown
and the error var. σ^2
- It is of interest to perform statistical inference
for these parameters, ie, write an interval estimator
or perform a hyp. test.

Tuesday
03-10-23

Inference for β_1

- Under our model assumptions, $\hat{\beta}_1 \sim N(\beta_1, \sigma^2 \left[\frac{1}{\sum_{i=1}^n (x_i - \bar{x})^2} \right])$
- The Confidence Interval estimate of β_1 is:
obtain using the t distribution.

$$\hat{\beta}_1 \pm t_{\alpha/2}(n-2) \sqrt{\hat{\sigma}^2 \left[\frac{1}{\sum_{i=1}^n (x_i - \bar{x})^2} \right]}$$

$$\left\{ \begin{array}{l} \text{point estimate} \\ + C.C \times SE(\text{estimate}) \end{array} \right\}$$

Inference for β_0

$$\hat{\beta}_0 \sim N(\beta_0, \hat{\sigma}^2 \left[\frac{\sum_{i=1}^n x_i^2}{n \sum_{i=1}^n (x_i - \bar{x})^2} \right])$$

- The Confidence Interval estimate is

$$\hat{\beta}_0 \pm t_{\alpha/2}(n-2) \sqrt{\hat{\sigma}^2 \left[\frac{\sum_{i=1}^n x_i^2}{n \sum_{i=1}^n (x_i - \bar{x})^2} \right]}$$

To perform a hypothesis test on the regression coefficients under simple Linear regression, we use the t statistic

i) Hypothesis Test for β_0

Hyp:

$$H_0 : \beta_0 = \beta_{00}$$

$$H_a : \beta_0 \neq \beta_{00}$$

Test Stat: $t = \frac{\hat{\beta}_0 - \beta_{00}}{\sqrt{\hat{\sigma}^2 \left[\frac{\sum_{i=1}^n x_i^2}{n \sum_{i=1}^n (x_i - \bar{x})^2} \right]}}$

Decision: Reject the null hyp if $|t| > t_{\alpha/2}(n-2)$

ii) Hypothesis Test for β_1

Hyp: $H_0 : \beta_1 \leq \beta_{10}$

$$H_a : \beta_1 > \beta_{10}$$

$$\text{Test Stat: } t = \frac{\hat{\beta}_1 - \beta_{10}}{\sqrt{\hat{\sigma}^2 \left[\frac{1}{\sum_{i=1}^n (x_i - \bar{x})^2} \right]}}$$

Example 1

Consider ^{an} ice cream sales data obtained in a certain town where there are 2 variables, i.e., the ice cream sales and average weekend temperature.

Weekend	1	2	3	4	5	6	7	8
X Temperature	25	16	28	20	22	23	16	18
Y Sales (in 100 shillings)	125	79	140	103	111	115	80	91

Suppose the data is assumed to follow a linear relationship $Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$, $i = 1, 2, \dots, n$. Test the following hypothesis:

- i) $H_0: \beta_1 = 0$ against $H_1: \beta_1 \neq 0$
- ii) $H_0: \beta_1 \leq 0$ against $H_1: \beta_1 > 0$
- iii) $H_0: \beta_1 \geq 0$ against $H_1: \beta_1 < 0$

MULTIPLE LINEAR REGRESSION

- Is an extension of SLM to include multiple indpt variables x_1, x_2, \dots, x_n
- Specifically, we consider models of the form:

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_k x_{ik} + \epsilon_i$$

- There are now $p = k+1$ regression parameters and error variance, σ^2 which are to be estimated based on the observed data

⇒ A matrix representation of the linear RM

Consider the model:

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_k x_{ik} + \epsilon_i \text{ for } i=1, \dots, n$$

Define

$$Y = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix}, X = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1k} \\ 1 & x_{21} & x_{22} & \dots & x_{2k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{nk} \end{bmatrix}$$

$$\beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{pmatrix}, \epsilon = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

- With these definitions, the MLRM can be expressed as follows:

$$Y = X\beta + \epsilon$$

in this representation

- Y is an $n \times 1$ vector of responses
- X is an $n \times p$ matrix of input variables
- β is a vector of unknown regression coefficients
- ϵ is an $n \times 1$ vector of unobserved errors

Sampling Distributions & Estimation of Parameters

- The least squares estimators of the regression coefficients is given as:

$$\hat{\beta} = (\mathbf{x}^T \mathbf{x})^{-1} (\mathbf{x}^T \mathbf{y})$$

- In our matrix formulation, the model is:

$$\mathbf{Y} = \mathbf{X}\beta + \mathbf{e}, \text{ where } \mathbf{e} \sim N_n(0, \sigma^2 \mathbf{I})$$

$$\beta \sim N(\hat{\beta}, \sigma^2 (\mathbf{x}^T \mathbf{x})^{-1})$$

Analysis of Variance Approach to Regression

- The observed variable, \mathbf{Y} will always have some variability associated with it
- This variability can be broken into the variability related to the predictor & the variability unrelated to the predictor
- The analysis of this variability is called ANOVA (analysis of var)
- The ANOVA summary table applies the regression model to account for all the

Variation in \hat{e} dptt vgr.

- For k explanatory variables, the model df are equal to k .
- The error or residual df are equal to $n-k-1$ and the total degrees of freedom are $= n-1$
- The corresponding ANOVA Table is as below:

Sources of variation	df	SS	MS	F
Regression (model)	k	SSR_g $= \sum (y_i - \hat{y}_i)^2$	MSR_g $= \frac{SSR_g}{k}$	$F^* = \frac{MSR_g}{MSE}$
Error (residual)	$n-k-1$	SSR_e $= \sum (y_i - \hat{y}_i)^2$	MSE $= \frac{SSR_e}{n-k-1}$	
Total	$n-1$	TSS		

- The F ratio is the test stat used to decide whether the model as a whole is statistically significant; ie, whether or not the proportion of

variance in Y explained by \bar{e} regression is due to chance

- Under \bar{e} null hypothesis that \bar{e} model has no predictive capability, ie, all regression coefs are zero simultaneously

$$H_0 : \beta_1 = \beta_2 = \beta_3 = \dots = \beta_k = 0$$

$$H_1 : \beta_i \neq 0, i=1, 2, \dots, k$$

R-squared

- Is \bar{e} , regression sum of squares to \bar{e} total sum of squares, ie,

$$\text{SSR}_g = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2$$

$$\frac{\text{SSR}_g}{\text{TSS}}$$

- Since \bar{e} total sum of squares is the sum of the regression and \bar{e} residual sum of squares, R squared can be written as;

$$1 - \frac{SSE}{TSS}$$

$$\frac{SSR_g}{TSS} = \frac{TSS - SSR}{TSS} = 1 - \frac{SSR}{TSS}$$

- Sometimes R^2 is called proportion of variance explained by the model
 - One criticism of R^2 is that it never decreases when a new indept variable is added to a LRM, even if the new var. is completely unrelated to the indept variables
 - The Adjusted R^2 overcomes this limitation in that it punishes the model if it adds irrelevant variables.
- ⇒ It doesn't increase unless the new var. or vars. have additional predictive capability.

$$\text{adj } R^2 = \frac{1 - \frac{SSR}{TSS} \cdot \frac{(n-1)}{n-k-1}}{\frac{TSS}{(n-1)}}$$

- A statistically insignificant variable will increase R^2 but decrease adj R^2
- The test stat for testing the overall significance of the model can be stated in terms of R^2 as follows:

$$F = \frac{SSR_g/k}{SSR/(n-k-1)} = \frac{R^2/k}{(1-R^2)/(n-k-1)}$$

Tues

17/10/23

CHAPTER 2: Non-Linear Regression

Estimation in Non-Linear Models

- The basic goal of regression analysis is to fit a model that best describes the relationship between one or more predictor variables and a response.
- Sometimes linear models are not sufficient to capture real world phenomena and therefore non-linear models are necessary.
- Non-linear models are models in which the derivatives of the mean fxn with respect to the parameters depend on 1 or more parameters.

Example

$$y_i = \frac{e^{B_0 + B_1 x_i}}{1 + e^{B_0 + B_1 x_i}} + e_i$$

$$y_i = \frac{\beta_0 + \beta_1 x_i}{1 + \beta_2 e^{\beta_3 x_i}} + e_i$$

$$y_i = \beta_0 + (0.4 - \beta_0) e^{-\beta_1 (x_i - 5)} + e_i$$

- However, there are some non-linear models called **intrinsically linear** because they can be made linear in parameters by a simple transformation
- 4 common transformations to induce linearity :
 - a) Logarithmic
 - b) Square root transformation
 - c) Inverse transformation
 - d) Square transformation
- It is sometimes useful to use the knowledge of least squares to estimate the parameters of non-linear models.
- This involves, first, transforming the given non-linear data to a linear relationship; following the

data, the parameters are estimated using the linear least squares method. The slns are then transformed back to the original variables.

Example

(i) Exponential model

- Consider a model of the form:

$$y = a e^{bx}$$

- Taking natural log on both sides, we obtain:

$$\ln y = \ln a + bx$$

- Let $z = \ln y$, $a_0 = \ln a$, $\Rightarrow a = e^{a_0}$ & $a_1 = b$

$$\text{then, } z = a_0 + a_1 x$$

- The model becomes $z_i = a_0 + a_1 x_i + \epsilon_i$, $i=1, 2, \dots, n$

- The data z vs x is now a linear model.

- The constants a_0 and a_1 can be found using the eqn for the linear model: $\ln \alpha = a_0 + a_1 z_i$

$$a_1 = \frac{\sum_{i=1}^n \alpha_i z_i - \bar{z} \sum_{i=1}^n \alpha_i}{\sum_{i=1}^n \alpha_i^2 - (\sum_{i=1}^n \alpha_i)^2}$$

$$a_0 = \bar{z} - a_1 \bar{\alpha}$$

- Now since a_0 and a_1 are found, the original constants a and b can be found as follows:

$$b = a_1, \quad a = e^{a_0}$$

(ii) Logarithmic functions

- The form of the log regression model is:

$$y = \beta_0 + \beta_1 \ln \alpha$$

- This is a linear fn between y and $\ln \alpha$ and the usual Least Squares method applies in which y is the response variable & $\ln \alpha$ is the regressor

$$\ln(ab e^x) = \ln a + \ln b e^x \neq \ln(a+b e^x)$$

- The model becomes:
- $y_i = a + b\alpha_i + e_i$
- This is a linear relationship between y and α & the coefficients b (and a) are found as follows:

$$b = \frac{n \sum_{i=1}^n \alpha_i y_i - \sum_{i=1}^n \alpha_i \sum_{i=1}^n y_i}{n \sum_{i=1}^n \alpha_i^2 - (\sum_{i=1}^n \alpha_i)^2}$$

$$a = \bar{y} - b \bar{\alpha}$$

(iii) Power Functions

- Consider a model of the form: $y = a x^b$
- The method of least squares is applied to the power function by 1st linearizing the data as follows:

$$\ln y = \ln a + b \ln x$$

- The resulting eqn shows a linear relation between $\ln y$ and $\ln x$.

- Let $z = \ln y$, $w = \ln x$, $a_0 = \ln a$, $a_1 = b$; we get :

$$Z_i = a_0 + a_1 w_i + e_i$$

- The estimates a_0 & a_1 are obtained as follows:

$$a_1 = \frac{n \sum_{i=1}^n w_i Z_i - \sum_{i=1}^n w_i \sum_{i=1}^n Z_i}{n \sum_{i=1}^n w_i^2 - \left(\sum_{i=1}^n w_i \right)^2}$$

$$a_0 = \bar{Z} - a_1 \bar{w}$$

- Transforming back to the original data ; $b = a_1$ and $a = e^{a_0}$

$$\frac{\partial y}{\partial \theta_1} = -1 (\theta_1 + \theta_2 e^{-x})$$

EXAMPLES $\frac{\partial y}{\partial \theta_2} = -1 (\theta_1 + \theta_2 e^{-x}) e^{-x}$

Linearise the following functions :

i) $y = \frac{1}{(\theta_1 + \theta_2 e^{-x})}$

It is linear in parameter ✓

$$\text{ii) } y = \Theta_1 + \Theta_2 x^{\Theta_2} \quad \text{also } \ln y = \Theta_1 + \Theta_2 \ln x$$

$$\ln y = \ln \Theta_1 + \Theta_2 \ln x$$

$$\text{iii) } y = \Theta_1 \exp(\Theta_2 g(x))$$

$$\ln y = \ln \Theta_1 + \Theta_2 g(x)$$

* Transforming a model to its linear form often provides better inference procedures & confidence intervals but one must be aware of the effect that the transformation has on the dist. of the errors.

Cases in which it is not possible to transform the model to a linear form:

- In this case, the normal eqns are non-linear fxns of the parameters and cannot be written explicitly in terms of the parameters

$$\ln y_i = \ln a + b x_i$$

1) The Exponential Model

- Given $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, the exp. model

$$y_i = a e^{b x_i} + e_i, \quad i=1, 2, \dots, n$$

$e_i \sim \text{iid } N(0, \sigma^2)$

can be fitted to the data as follows:

$$e_i = y_i - a e^{b x_i}$$

- The sum of squared errors is given by:

$$S_r = \sum_{i=1}^n (y_i - a e^{b x_i})^2$$

- To find constants a and b of the exponential model, we can minimize S_r w.r.t a and b or by differentiating w.r.t. a and b and equating the resulting eqns to zero.

- The derivative of S_r w.r.t a :

$$\frac{\partial S_r}{\partial a} = 2 \sum_{i=1}^n (y_i - a e^{b x_i}) (e^{b x_i}) = 0$$

$$\frac{\partial S_r}{\partial b} = 2 \sum_{i=1}^n (y_i - a e^{b x_i}) (-a x_i e^{b x_i}) = 0$$

$$\frac{\partial S_r}{\partial a} = - \sum_{i=1}^n y_i e^{bx_i} + a \sum_{i=1}^n e^{2bx_i} = 0 \quad \text{--- (i)}$$

$$\frac{\partial S_r}{\partial b} = \sum_{i=1}^n y_i x_i e^{bx_i} - a \sum_{i=1}^n x_i e^{2bx_i} = 0 \quad \text{--- (ii)}$$

→ Eqns (i) & (ii) are non-linear in a and b &
therefore not in a closed form to be solved
as was the case of linear regression.

- In general, iterative methods such as:

Gauss-Newton algorithm and Levenberg
Method of steepest descent
Direct search method etc must be used to find values of a and b .

- Growth models are used to describe how

ii) Growth Model

- Growth models are used to describe how sth grows with changes in the regressor variable (often time)
- A good example in this category is the growth of pop with time.

Growth models include:

$$y = \frac{a}{1+be^{-cx}} \quad \text{where } a, b, c \text{ are constants}$$

- At $x=0$, $y = \frac{a}{1+b}$

and

$$\text{as, } x \rightarrow \infty, y = a$$

- The sum of squared residuals is given by:

$$S_r = \sum_{i=1}^n \left(y_i - \frac{a}{1+b e^{-cx_i}} \right)^2$$

- To find the constants a, b and c , we minimize

S_r by differentiating w.r.t $a, b \& c$ and equating the equations to zero.

$$\frac{\partial S_r}{\partial a} = \sum_{i=1}^n \left(\frac{2e^{cx_i} [ae^{cx_i} - y_i(e^{cx_i} + b)]}{(e^{cx_i} + b)^2} \right) = 0$$

$$\frac{\partial S_r}{\partial b} = \sum_{i=1}^n \left(\frac{2ae^{cx_i} [by_i + e^{cx_i}(y_i - a)]}{(e^{cx_i} + b)^3} \right) = 0$$

$$\frac{\partial S_r}{\partial c} = \sum_{i=1}^n \left(\frac{-2ab^2 e^{cx_i} [by_i + e^{cx_i}(y_i - a)]}{(e^{cx_i} + b)^3} \right) = 0$$

- One can use Newton Raphson method to solve the above set of simultaneous non-linear eqns for a, b and c

III) Polynomial Model

- Given n data pnts $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ use the least squares method to regress the data to a polynomial of order m , ie,

$$y_i = a_0 + a_1 x_i + a_2 x_i^2 + \dots + a_m x_i^m$$

- The sum of squares of the residuals is given by:

$$S_r = \sum_{i=1}^n (y_i - a_0 - a_1 x_i - \dots - a_m x_i^m)^2$$

- To find the constants of the polynomial regression model, we differentiate \rightarrow objective fn S_r w.r.t a_i and equate to zero; ie,

$$\frac{\partial S_r}{\partial a_0} = 2 \sum_{i=1}^n (y_i - a_0 - a_1 x_i - \dots - a_m x_i^m) (-1) = 0$$

$$\frac{\partial S_r}{\partial a_1} = 2 \sum_{i=1}^n (y_i - a_0 - a_1 x_i - \dots - a_m x_i^m) (-x_i) = 0$$

$$\frac{\partial S_r}{\partial a_m} = -2 \sum_{i=1}^n (y_i - a_0 - a_1 x_i - a_2 x_i^2 - \dots - a_m x_i^m) (x_i^m)$$

• Setting those eqns in matrix form gives:

$$\begin{bmatrix} n & \left(\sum_{i=1}^n x_i \right) & \left(\sum_{i=1}^n x_i^m \right) & a_0 \\ \left(\sum_{i=1}^n x_i \right) & \left(\sum_{i=1}^n x_i^2 \right) & \left(\sum_{i=1}^n x_i^{m+1} \right) & a_1 \\ \left(\sum_{i=1}^n x_i^m \right) & \left(\sum_{i=1}^n x_i^{m+1} \right) & \left(\sum_{i=1}^n x_i^{2m} \right) & a_m \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \\ \sum_{i=1}^n x_i^m y_i \end{bmatrix}$$

$$\frac{\partial S_r}{\partial a_0} = 2 \sum_{i=1}^n (y_i - a_0 - a_1 x_i - \dots - a_m x_i^m) (-1) = 0$$

$$= \sum_{i=1}^n (y_i - a_0 - a_1 x_i - \dots - a_m x_i^m) = 0$$

$$= \sum_{i=1}^n y_i - n a_0 - a_1 \sum_{i=1}^n x_i - \dots - a_m \sum_{i=1}^n x_i^m = 0$$

$$n a_0 + a_1 \sum_{i=1}^n x_i + \dots + a_m \sum_{i=1}^n x_i^m = \sum_{i=1}^n y_i$$

Example 1 (Exponential Model)

Tuesday
24/10/2023

Fit an exponential model to the following data.

x	y	$z_i = \ln y_i$	$x_i z_i$	x_i^2	$w_i = \ln x_i$	w_i^2	$w_i z_i$
1	1.93	0.6575	0.6575	1	0	0	0.0627
1.1	1.61	0.4762	0.5238	1.21	0.0953	0.00908	0.0454 0.0868
1.2	2.27	0.8198	0.9836	1.44	0.1823	0.03323	0.1494
1.3	3.19	1.1600	1.5080	1.69	0.2624	0.06885	0.3044
1.4	3.19	1.1600	1.6240	1.96	0.3365	0.11323	0.3903
1.5	3.71	1.311	1.9665	2.25	0.4055	0.16443	0.5316
1.6	4.29	1.4563	2.3301	2.56	0.4700	0.22090	0.6845
1.7	4.95	1.5994	2.7190	2.89	0.5306	0.28154	0.8486
1.8	6.07	1.8034	3.2461	3.24	0.5878	0.34551	1.0600
1.9	7.48	2.0122	3.8232	3.61	0.6419	0.41204	1.2916
2.0	8.72	2.1656	4.3312	4.00	0.6931	0.48039	1.5000

		$z_i = \ln y_i$	$\Sigma x_i z_i$	Σx_i^2	$w_i = \frac{1}{\ln x_i}$	w_i^2	$w_i z_i$
2.1	9.34	2.2343	4.6920	4.41	0.7419	0.55042	1.6576
2.2	11.62	2.4527	5.3959	4.84	0.7885	0.6273	1.9340
$\sum x_i = 20.8$	$\sum z_i = 19.3084$	$\sum x_i z_i = 33.8012$	$\sum x_i^2 = 35.1$	$\sum w_i = 3.30135$	$\sum w_i^2 = 10.3985$		

The model is of the form:

$$y_i = b_1 e^{a_1 x_i} + e_i \quad \text{for } i=1, 2, \dots, n$$

S_n

- Applying logs on both sides we get;

$$\ln y_i = \ln b_1 + a_1 x_i$$

- Let $z_i = \ln y_i$, $b = \ln b_1$, $a = a_1$,
The model becomes;

$$z_i = b + a x_i + e_i$$

- The estimates of a and b are obtained using linear least squares as follows:

$$\hat{a} = \frac{n \sum_{i=1}^n x_i z_i - \sum_{i=1}^n x_i \sum_{i=1}^n z_i}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2}$$

and $\hat{b} = \bar{z} - \hat{a} \bar{x}$

$$\Rightarrow \hat{a} = \frac{(13 \times 33.801) - (20.8 \times 19.3084)}{(13 \times 35.1) - (20.8)^2}$$

$$= 1.59756044 \approx 1.598$$

$$\hat{b} = \frac{19.3084}{13} = 1.59756044 \times \frac{20.8}{13}$$

$$= -1.070835$$

If you put $1.598 = -1.07154$

Class answer: -1.07156

- The estimated model is:

$$\hat{z}_i = \hat{b} + \hat{a} x_i$$

$$\hat{z}_i = -1.07156 + 1.5980x_i$$

- Transforming to the original model variables;

$$b = \ln b_1 \Rightarrow b_1 = e^b = e^{-1.07156}$$

$$b_1 = 0.3425$$

$$a_1 = a = 1.5980$$

- The estimated exponential model is:

$$\hat{y}_i = b_1 e^{a_1 x_i}$$

$$\hat{y}_i = 0.3425 e^{1.5980 x_i}$$

Example 2 <Exponential Model>

Fit the power model $y_i = b_2 x_i^{a_2}$

where $e_i \sim N(0, \sigma^2)$ to the data in example 1

Sln

- Applying log on both sides:

$$\ln y_i = \ln b_2 + a_2 \ln x_i$$

- Let $z_i = \ln y_i$, $b = \ln b_2$ and $w_i = \ln x_i$, $a = a_2$
The model becomes:

$$z_i = b + a w_i + e_i$$

- The estimates of a and b are;

$$\hat{a} = \frac{n \sum_{i=1}^n w_i z_i - \sum_{i=1}^n w_i \sum_{i=1}^n z_i}{n \sum_{i=1}^n w_i^2 - (\sum_{i=1}^n w_i)^2}$$

$$\hat{b} = \bar{z} - \hat{a} \bar{w}$$

$$\hat{a} = \frac{(13 \times 10.3985) - (5.7358 \times 19.3084)}{(13 \times 3.30135) - (3.30135)^2}$$

$$= 0.763036177 \approx 0.76304$$

$$\hat{b} = \frac{19.3084}{13} - 0.76304 \times \frac{5.7358}{13} = 1.14860$$

$$b = \ln b_2$$

$$\hat{b}_2 = e^{\hat{b}}$$

$$= e^{1.14860} = 3.15377$$

$$\hat{a} = \hat{a}_2$$

$$= 0.76304$$

The estimated power model is:

$$\hat{y}_i = \hat{b}_2 \hat{x}_i^{\hat{a}_2}$$

$$\hat{y}_i = 3.15377 x_i^{0.76304}$$

Example 3 <Growth Model>

Consider the following growth model

$$y_i = \frac{a\alpha_i}{b+\alpha_i} + e_i \quad \text{where } e_i \sim N(0, \sigma^2)$$

Linearize and fit the model to the data in Ex.1

Sln

$$\underline{l} = b + \alpha_i \underline{x_i}$$

$$y_i = \frac{a\alpha_i}{b+a\alpha_i} = \frac{a\alpha_i}{a+b\alpha_i}$$

$$\frac{l}{y_i} = \frac{b}{a} \cdot \frac{1}{x_i} + \frac{x_i}{a} = \frac{b}{a} \cdot \frac{1}{x_i} + \frac{l}{a}$$

- $k = \frac{b}{a}$
- Let $z_i = \frac{1}{y_i}$, $k = \frac{b}{a}$, $w_i = \frac{1}{x_i}$, $p = \frac{1}{a}$
The model becomes:

$$z_i = p + kw_i + e_i$$

After getting the values, to transform back:

$$a = \frac{1}{p}, \quad b = ka$$

Example 4 <Polynomial Model>

The data below contains measurements of yield from an experiment done at 5 different temp. levels. The variables are: $y = \text{yield}$, $x = \text{temp.}$

Temp(x)	Yield(y)	x^2	x^3	x^4	$\sum x_i y_i$	$\sum x_i^2 y_i$
50	3.3	2500	125000	6250000	165	8250
50	2.8	2500	125000	6250000	140	7000
50	2.9	2500	125000	6250000	145	7250
70	2.3	4900	343000	24010000	161	11270
70	2.6	4900	343000	24010000	182	12740
70	2.1	4900	343000	24010000	147	10290
80	2.5	6400	512000	40960000	200	16000
80	2.9	6400	512000	40960000	232	18560
80	2.4	6400	512000	40960000	192	15360
90	3.0	8100	729000	65610000	270	24300
90	3.1	8100	729000	65610000	279	25110
90	2.8	8100	729000	65610000	252	22680
100	3.3	10000	1000000	100000000	330	33000
100	3.5	10000	1000000	100000000	350	35000
100	3.0	10000	1000000	100000000	300	30000
	42.5	95,700	8,127,000	710,490,000	3345	276,810

Fit a polynomial of order 2 to the data:

$$y_i = a + bx_i + cx_i^2 + e_i \quad \text{where } e_i \sim N(0, \sigma^2)$$

S
l_n

$$Q = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - a - bx_i - cx_i^2)^2$$

$$\frac{\partial Q}{\partial a} = 2 \sum_{i=1}^n (y_i - a - bx_i - cx_i^2)(-1) = 0$$

∂a

$$= \sum_{i=1}^n (y_i - a - bx_i - cx_i^2) = 0$$

$$= \sum y_i - na - b \sum x_i - c \sum x_i^2 = 0$$

$$\therefore \sum_{i=1}^n y_i = na - b \sum_{i=1}^n x_i - c \sum_{i=1}^n x_i^2 \quad \dots \text{--- (i)}$$

\langle 1st normal eqn \rangle

$$\frac{\partial Q}{\partial b} = 2 \sum_{i=1}^n (y_i - a - bx_i - cx_i^2)(-x_i) = 0$$

$$= \sum (y_i - a - bx_i - cx_i^2)(x_i) = 0$$

$$a \sum_{i=1}^n x_i + b \sum_{i=1}^n x_i^2 + c \sum_{i=1}^n x_i^3 = \sum_{i=1}^n x_i y_i \quad \dots \text{--- (ii)}$$

So for c;

$$a \sum_{i=1}^n x_i^2 + b \sum_{i=1}^n x_i^3 + c \sum_{i=1}^n x_i^4 = \sum_{i=1}^n x_i^2 y_i \quad \text{--- (iii)}$$

which gives the matrix:

$$\begin{bmatrix} n & \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i^3 \\ \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i^3 & \sum_{i=1}^n x_i^4 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \\ \sum_{i=1}^n x_i^2 y_i \end{bmatrix}$$

The 3 eqns are

$$\sum_{i=1}^n y_i = n a + b \sum_{i=1}^n x_i + c \sum_{i=1}^n x_i^2 \quad \text{--- (i)}$$

$$\sum_{i=1}^n x_i y_i = a \sum_{i=1}^n x_i + b \sum_{i=1}^n x_i^2 + c \sum_{i=1}^n x_i^3 \quad \text{--- (ii)}$$

$$\sum_{i=1}^n x_i^2 y_i = a \sum_{i=1}^n x_i^2 + b \sum_{i=1}^n x_i^3 + c \sum_{i=1}^n x_i^4 \quad \text{--- (iii)}$$

$$42.5 = 15a + 1170b + 957000c \quad \text{--- (i)}$$

$$3345 = 1170a + 95700b + 8127000c \quad \text{--- (ii)}$$

$$276,810 = 95700a + 8127000b + 710490000c \quad \text{--- (iii)}$$

For (i) and (ii)

$$3315 = 1170a + 91260b + 7464600c$$

$$3345 = 1170a + 95700b + 8127000c$$

$$30 = 4440b + 662400c$$

$$4440b = 662400c - 30$$

$$b = 1 \quad (662400c - 30)$$

$$4440$$

$$42.5 = 15a + 1170(662400c - 30) + 957000c$$

$$4440$$

$$\Rightarrow 188,700 = 66,600a + 775,008,000c - 35100 + 424,908,000c$$

$$223,800 = 66,600a + 1,199,916,000c$$

$$a = \frac{1}{66,600} (223,800 - 1,199,916,000c)$$

Applying a and b in (iii);

$$276,810 = 95700 (223,800 + 1,199,916,000c)$$

66600

$$30 + 8127000(662400c - 30) + 7104900000c$$

4440

$$(276,810 \times 66600 \times 4440) = 95700 (223,800 \times 4440) +$$

$$(8127000 \times 66600 \times 30) +$$

$$C [(-95700 \times 119991600 \times 4440) + (8127000 \times 662400 \times 66600) \\ + (7104900000 \times 66600 \times 4440)]$$

$$C = -0.000056948$$

$$\approx -5.694768927 \times 10^{-5}$$

14 Nov 23

NON-LINEAR LEAST SQUARES DATA FITTING

- Non-linear least squares is the form of least squares analysis used to fit a set of n observations with a model that is non-linear in p unknown parameters ($n > p$)
- The basis of the method is to approximate the model by a linear one and to refine the parameters by successive iterations
- Consider the model of the form

$$Y = f(x; \beta) + e$$

where $f(\cdot)$ is a known function, x is an $n \times 1$ vector of explanatory variables, β is a $p \times 1$ vector of parameters and e is an error term

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_T \end{bmatrix}$$

and

$$f(x; \beta) = \begin{bmatrix} f(x_1, \beta) \\ f(x_2, \beta) \\ \vdots \\ f(x_T, \beta) \end{bmatrix}$$

So that the equation can be expressed as;

$$y = f(x_1, x_2, \dots, x_r; \beta) + e$$

- The objective is to find a k -dimensional surface that best fits the data (y_t, x_t) , $t = 1, 2, \dots, T$
- Similar to the OLS method, the method of non-linear least squares (NLLS) is used to minimize the following NLLS with respect to β

$$\begin{aligned} Q_T(\beta) &= \frac{1}{2} \left[y - f(x_1, x_2, \dots, x_r; \beta) \right]^T \left[y - f(x_1, x_2, \dots, x_r; \beta) \right] \\ &= \frac{1}{2} \sum_{t=1}^T [y_t - f(x_t, \beta)]^2 \end{aligned}$$

Conditions for non-linear least square minimization

i) First Order Condition

$$\nabla_B Q_T(B) = \frac{-q}{2} \nabla_B f(x_1, x_2, \dots, x_T; B) [y - f(x_1, x_2, \dots, x_T; B)]$$

Where why is there no $\sum_{t=1}^T$ here ??? $f(x_1, \dots, x_T; B)$

$$\nabla_B f(x_1, \dots, x_T; B) = [\nabla_B f(x_1, B), \nabla_B f(x_2, B), \dots, \nabla_B f(x_T, B)]$$

is a $k \times T$ matrix

- A soln to this minimization problem $\hat{B} \in \Theta$ solves the first order condition $\nabla_B Q_T(B) = 0$ and satisfies $\nabla_B^2 Q_T(B)$ is a positive definite

ii) Second Order Condition

2nd derivative exists & is continuous

$f(x, B)$ is twice continuously differentiable in the second argument B such that for a given data $(y_t, x_t), t = 1, 2, \dots, T$, $\nabla_B^2 Q_T(B)$ is a positive definite at some interior pt of Θ

$y_t = f(x_t; \beta) + e$ so $r_t(x; \beta)$ is the residuals(error)

$\nabla_B^2 Q_T(\beta)$ is a square matrix of second order partial derivatives of $Q_T(\beta)$

Remarks

- (i) $\nabla_B Q_T(\beta)$ is the gradient
- (ii) $\nabla_B^2 Q_T(\beta)$ is the Hessian matrix

$\rightarrow r_t(x; \beta) = y_t - f(x_t; \beta)$ then the Jacobian $J(x; \beta)$ is a matrix of all $\nabla_B r_t(x; \beta)$.

\rightarrow The Jacobian is then:

$$J(x; \beta) = \begin{bmatrix} \nabla r_1(x; \beta) \\ \nabla r_2(x; \beta) \\ \vdots \\ \nabla r_T(x; \beta) \end{bmatrix} = \begin{bmatrix} \frac{\partial r_t(x; \beta)}{\partial \beta_j} \end{bmatrix}$$

where $j = 1, 2, \dots, k$

\rightarrow The gradient and the Hessian matrix of the objective function $Q_T(\beta)$ can be expressed in terms of

the Jacobian matrix as follows

$$\nabla_{\beta} Q_T(\beta) = J(x; \beta)^T r_t(x; \beta)$$

$$\nabla_{\beta}^2 Q_T(\beta) = J(x; \beta)^T J(x; \beta)$$

$$+ \sum_{t=1}^T r_t(x; \beta) \nabla^2 r_t(x; \beta)$$

Example

- Consider the following data on the population of antelopes

x_t	1	2	4	5	8
y_t	3	4	6	11	20

$$\text{Let } Y_t = \beta_1 e^{\beta_2 x_t} + e_t$$

- (i) Determine the formula for the gradient and Hessian

Sl.

$$\text{let } r_t(x; \beta) = \beta_1 e^{\beta_2 x_t} - y_t \quad \text{why?? Maybe it's the error term.}$$

The formula for the least squares objective fun:

$$Q_T(\beta) = \frac{1}{2} \sum_{t=1}^T (\beta_1 e^{\beta_2 x_t} - y_t)^2$$

I think b is the partial derivative of a w.r.t β_2
 a is the derivative w.r.t β_1
 $\& b$ is the derivative w.r.t β_2

a) The gradient of the objective fcn is:

$$\nabla_{\beta} Q_T(\beta) = \left[\sum_{t=1}^T (\beta_1 e^{\beta_2 x_t} - y_t) e^{\beta_2 x_t} \right] \left[\sum_{t=1}^T (\beta_1 e^{\beta_2 x_t} - y_t) \beta_1 x_t e^{\beta_2 x_t} \right]$$

This can be written in terms of a Jacobian as follows:

$$= J(x; \beta)^T r_t(x; \beta)$$

$$= \begin{bmatrix} e^{\beta_2 x_1} & e^{\beta_2 x_2} & \dots & e^{\beta_2 x_T} \\ \beta_1 x_1 e^{\beta_2 x_1} & \beta_1 x_2 e^{\beta_2 x_2} & \dots & \beta_1 x_T e^{\beta_2 x_T} \end{bmatrix} \begin{bmatrix} \beta_1 e^{\beta_2 x_1} - y_1 \\ \beta_1 e^{\beta_2 x_2} - y_2 \\ \vdots \\ \beta_1 e^{\beta_2 x_T} - y_T \end{bmatrix}$$

b) Hessian

$$\nabla_{\beta}^2 Q_T(\beta) = J(x; \beta)^T J(x; \beta) + \sum_{t=1}^T r_t(x; \beta) \nabla^2 r_t(x; \beta)$$

$$= \begin{bmatrix} e^{\beta_2 x_1} & e^{\beta_2 x_2} & \dots & e^{\beta_2 x_T} \\ \beta_1 x_1 e^{\beta_2 x_1} & \beta_1 x_2 e^{\beta_2 x_2} & \dots & \beta_1 x_T e^{\beta_2 x_T} \end{bmatrix} \begin{bmatrix} e^{\beta_2 x_1} & \beta_1 x_1 e^{\beta_2 x_1} \\ e^{\beta_2 x_2} & \beta_1 x_2 e^{\beta_2 x_2} \\ \vdots & \vdots \\ e^{\beta_2 x_T} & \beta_1 x_T e^{\beta_2 x_T} \end{bmatrix}$$

$$\begin{bmatrix} \partial B_1 \partial B_1 & \partial B_1 \partial B_2 \\ \partial B_2 \partial B_1 & \partial B_2 \partial B_2 \end{bmatrix}$$

$$+ \sum_{t=1}^T \left(\beta_1 e^{\beta_2 x_t} - y_t \right) \begin{bmatrix} 0 & x_t e^{\beta_2 x_t} \\ x_t e^{\beta_2 x_t} & \beta_1 x_t^2 e^{\beta_2 x_t} \end{bmatrix}$$

(ii) Compute the Hessian and the Jacobian matrix for the Antelope Data using $\beta_1 = 2$ and $\beta_2 = 1$

$$\text{Gradient / Jacobian} = \begin{bmatrix} e^{1(1)} & e^2 & e^4 & e^5 & e^8 \\ 2(1)e^{1(1)} & 2(2)e^2 & 2(4)e^4 & 2(5)e^5 & 2(8)e^8 \end{bmatrix} \begin{bmatrix} 2e^1 - 3 \\ 2e^2 - 4 \\ 2e^4 - 6 \\ 2e^5 - 11 \\ 2e^8 - 20 \end{bmatrix}$$

$$a = 6.623266712 + 79.64007567 + 5634.327074 + 42420.38684 + 17712601.88$$

$$b = 13.24653342 + 318.56030277 + 45074.61659 + 424203.8684 + 283401630.1 = 283,871,240.4 \quad \text{Ans} = \begin{bmatrix} 17,760,742.86 \\ 283,871,240.46 \end{bmatrix}$$

$$\text{Hessian} = \begin{bmatrix} e^1 & e^2 & e^4 & e^5 & e^8 \\ 2e^1 & 4e^2 & 8e^4 & 10e^5 & 16e^8 \end{bmatrix} \begin{bmatrix} e^1 & 2e^1 \\ e^2 & 4e^2 \\ e^4 & 8e^4 \\ e^5 & 10e^5 \\ e^8 & 16e^8 \end{bmatrix} + (2e^1 - 3) \begin{bmatrix} 0 & e^1 \\ e^1 & 2e^1 \end{bmatrix}$$

$$+ (2e^3 - 4) \begin{bmatrix} 0 & 2e^2 \\ 2e^2 & 8e^2 \end{bmatrix} + (2e^4 - 6) \begin{bmatrix} 0 & 4e^4 \\ 4e^4 & 8e^4 \end{bmatrix}$$

$$+ (2e^5 - 11) \begin{bmatrix} 0 & 5e^5 \\ 5e^5 & 50e^5 \end{bmatrix} + (2e^8 - 20) \begin{bmatrix} 0 & 8e^8 \\ 8e^8 & 128e^8 \end{bmatrix}$$

$$= \begin{bmatrix} 8,911,179.931 & 142,422,113.800 \\ 142,422,113.800 & 2,277,238,624 \end{bmatrix} + \begin{bmatrix} 0 & 6.62327 \\ 6.62327 & 13.24653 \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 159,28015 \\ 159,28015 & 637,12061 \end{bmatrix} + \begin{bmatrix} 0 & 22,537,3083 \\ 22,537,3083 & 45,074,61659 \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 212,101,9342 \\ 212,101,9342 & 2,121,019,342 \end{bmatrix} + \begin{bmatrix} 0 & 141,700,8151 \\ 141,700,8151 & 2,267,213,04 \end{bmatrix}$$

$$= \begin{bmatrix} 8,911,179.931 & 284,357,734 \\ 284,357,734 & 2,271,657,023 \end{bmatrix}$$

Exercise

Consider the non-linear model; $Y_t \approx \beta_0 e^{\beta_2 X_t} + \beta_3 + \beta_4 X_t$

Define $Q_T(\beta) = \frac{1}{n} (Y_t - \beta_0 e^{\beta_2 X_t} - \beta_3 - \beta_4 X_t)^2$

$r_t(x, \beta) = \beta_0 e^{\beta_2 X_t} - \beta_3 - \beta_4 X_t$. Derive the formula of the hessian & Gradient.

Non-linear Optimization

- When a soln with 1st order condition of the NLS minimization problem cannot be obtained analytically, the NLS estimators can be computed using numerical techniques
- This is done using an iterative algorithm that starts from an initial value of the argument in that fxn and repeatedly calculates the next available value according to a particular rule until an optimum is reached approximately.

The GAUSS NEWTON Method

- The method is based on a linear approximation to the components of the objective fxn $Q_T(\beta)$ in the neighborhood of β for some small h

→ Taylor Series

$$Q_T(\beta + h) = L(h) = Q_T(\beta) + h \underbrace{\nabla_{\beta} Q_T(\beta)}_{\text{gradient}} + \frac{h^2}{2} \underbrace{\nabla_{\beta}^2 Q_T(\beta)}_{\text{Jacobian}} + \dots$$

$$= Q_T(\beta) + h J^T(\beta) r(\beta) + \frac{1}{2} h J^T(\beta) J(\beta) h + \sum r(\beta) \nabla^2 r(\beta) + \dots$$

↓ zero
↓ summation of errors = 0

$$Q_T(\beta_{th}) \approx Q_T(\beta) + h J^T(\beta) r(\beta) + \frac{1}{2} h J^T(\beta) J(\beta) h + \dots$$

Note

The Gauss Newton step h minimizes $L(h)$

$$\text{i.e., } h = \underset{h}{\operatorname{argmin}} \{ L(h) \}$$

- Clearly, $L'(h) = J^T(\beta) r(\beta) + J^T(\beta) J(\beta) h$
[gradient of $L(h)$]

$Q_T(\beta_{th}) \approx L(h) \Rightarrow L'(h)$ gives you the gradient
and

$$L''(h) = J^T(\beta) J(\beta) \quad [\text{Hessian of } L(h)]$$

Remarks

- (i) $L''(h)$ is indpt of h
- (ii) $J(\beta)$ is full rank \rightarrow its columns are linearly indptt

- From (i) and (ii) above, $L(h)$ has a unique minimizer which can be found by solving for h

$$h = -[J^T(\beta) J(\beta)]^{-1} J^T(\beta) r(\beta)$$

- To satisfy the first condition for $Q_T(\beta)$ then

$$\beta^{(i+1)} = \beta^{(i)} + \alpha h ; i = 1, 2, \dots$$

where α is found by line search and $\beta^{(0)}$ is initialized.

Note

Classical Gauss Newton method, where $\alpha = 1$ in all steps, the method can be shown to guarantee convergence provided that:

- (a) $Q_T(\beta)^{(i+1)} \leq Q_T(\beta)^{(i)}$ is bounded
- (b) The Jacobian has full rank

Example

Apply the Gauss Newton Method (GNM) to the data

X_t	1	2	4	5	8
Y_t	3.2939	4.2699	7.1749	9.3008	20.259

and $Y_t \approx \beta_0 e^{\beta_1 X_t}$ and $\beta^{(0)} = \begin{bmatrix} 2.50 \\ 0.25 \end{bmatrix}$

$$\frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \Rightarrow \begin{bmatrix} a & b \\ c & d \end{bmatrix} \text{ inverse of } \begin{bmatrix} B_1 & B_2 \\ 0.25 & 0.25 \end{bmatrix}$$

↓
with $\beta_1 = 2.5$ & $\beta_2 = 0.25$

Sh $r_t = \beta_1 e^{\beta_2 x_t} - y_t$ and;

$$J(\beta)^T = \begin{bmatrix} 1.2840 & 3.2101 \\ 1.2487 & 8.2436 \\ 2.7183 & 27.1828 \\ 3.4903 & 43.6293 \\ 7.3891 & 147.7811 \end{bmatrix}$$

$$= \begin{bmatrix} -0.0838 \\ -0.1481 \\ -0.3792 \\ -0.5749 \\ -1.7864 \end{bmatrix}$$

so that; $J^T(\beta) r(\beta) = \begin{bmatrix} -16.5888 \\ -300.8722 \end{bmatrix}$

and $J^T(\beta) J(\beta) = \begin{bmatrix} 78.5367 & 1335.8479 \\ 1335.8479 & 24559.9419 \end{bmatrix}$

Recall that $h = -[J^T(\beta) J(\beta)]^{-1} J^T(\beta) r(\beta)$

Then $h = \begin{pmatrix} 0.0381 \\ 0.0102 \end{pmatrix}$

and $\beta^{(1)} = \beta^{(0)} + \alpha h$

$$= \begin{pmatrix} 2.50 \\ 0.25 \end{pmatrix} + 1 \begin{pmatrix} 0.0381 \\ 0.0102 \end{pmatrix} = \begin{pmatrix} 2.5381 \\ 0.2602 \end{pmatrix}$$

The iterations go on until there is a global shift.
The complete iteration is:

$$\hat{\beta} = \begin{bmatrix} 2.5411 \\ 0.2595 \end{bmatrix}$$

CHAPTER 3: Non-Parametric Regression

→ Non-parametric reg. differs from parametric reg. in that the shape of the final relationship btwn the response (depd) and the explanatory (indep) variables are not pre-determined but can be adjusted to capture unusual or unexpected features of the data when the relationship btwn the response and the explanatory variables is known. If the relationship is unknown and non-linear, non-parametric models should be used. A general non-parametric reg. model may be defined as: $Y_i = m(x_i) + e_i, i=1, 2, \dots, n$

$x_i, y_i \in \mathbb{R}$ and e_1, e_2, \dots, e_n are iid
 $E(e_i) = 0$ and $\sigma^2 = \text{Var}(e_i) < \infty$

The problem is to estimate the fcn $m(x_i)$ from the so $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$. There are diff. techniques; ie

(i) k - Nearest Neighbors (KNN)

- The idea of nearest neighbors smoother in regression problems is the following:

for each pnt x_0 , take k Nearest neighbors & estimate the value $m(x_0)$ by averaging the values of these neighbors.

- A simple estimate of $m(x_0)$ at any pnt x_0 is the mean of the k pnts closest to x_0 in euclidean distance, that is,

$$\hat{m}(x) = \text{Ave} (y_i \mid x_i \in N_k(x))$$

- The KNN smoother is defined as:

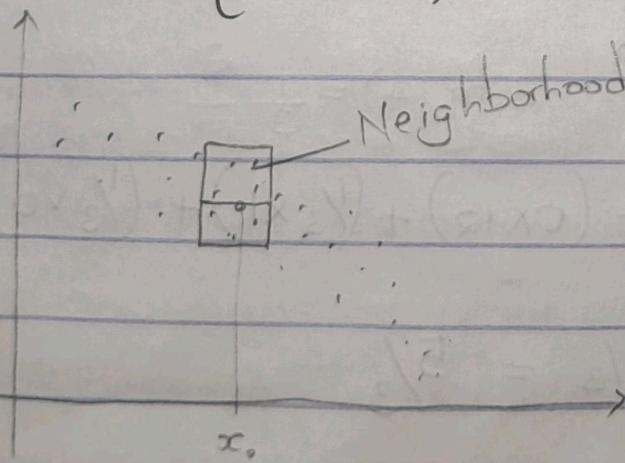
$$\hat{m}_k(x) = \frac{1}{h} \sum_{i=1}^n w_{k_i}(x) y_i$$

where $\{w_{k_i}(x)\}_{i=1}^n$ is a weight sequence defined through a set of indices.

$$J_x = \{ i, x_i \text{ is one of the } k\text{-nearest neighbors to } x \}$$

- The KNN weight sequence defined is then constructed using J_x instead of indices of neighboring observations as follows:

$$w_{k_i}(x) = \begin{cases} \frac{1}{k} & ; \text{ if } i \in J_x \\ 0 & , \text{ otherwise} \end{cases}$$



Example

Let $\{(x_i, y_i)\}_{i=1}^5$ be

$$\{(1, 5), (7, 12), (3, 1), (2, 0), (5, 4)\}$$

$$x = 1, 7, 3, 2, 5 \quad y = 5, 12, 1, 0, 4$$

(i) Compute the KNN estimate for $x=4$ and $k=3$

Sln

$$J_x = J_4 = (2, 3, 5)$$

$$w_{ki}(x) = \frac{1}{k}$$

$$\therefore w_{31}(4) = 0$$

$$w_{34}(4) = \frac{1}{3}$$

$$w_{32}(4) = 0$$

$$w_{35}(4) = \frac{1}{3}$$

$$w_{33}(4) = \frac{1}{3}$$

$$\hat{m}_3(4) = \frac{1}{3} \sum_{i=1}^5 w_{3i}(4) y_i$$

$$\hat{m}_3(4) = (0 \times 5) + (0 \times 12) + \left(\frac{1}{3} \times 1\right) + \left(\frac{1}{3} \times 0\right) + \left(\frac{1}{3} \times 4\right)$$

$$= \frac{1}{3} + \frac{4}{3} = \frac{5}{3}$$

$$\hat{m}_k(x) = \frac{1}{n} \sum_{i=1}^n w_{ki}(x) y_i$$

$$\frac{5}{4} + \frac{1}{4} + \frac{1}{4} = \frac{10}{4}$$

(ii) If $k=4$, find $\hat{m}_k(x)$ for $x=4$

KNN Classifiers:

- KNN algorithm is a robust & intuitive ML method employed to tackle both classification and regression models.
- By capitalizing on the concept of similarity, KNN predicts the label or value of a new data pt considering its k closest neighbors in the training data set.

Distance Metrics used in the KNN algorithms:

As we know, the KNN algorithm helps to identify the nearest pt or the grp for a query pt but to determine the closest grp or nearest pts for a query pt, we need some metric.

i) Euclidean distance:

$$d(x, x_i) = \sqrt{\sum_{j=1}^d (x_j - x_{ij})^2}$$

ii) Manhattan Distance

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

iii) Minkowski Distance

$$d(x, y) = \left[\sum_{i=1}^n (x_i - y_i)^p \right]^{1/p}$$

CHAPTER 4: NEURAL NETWORKS

- A neural networks is an artificial system inspired by biological neural networks.
- ANN (Artificial neural network) acquires a large collection of units that are interconnected in some patterns to allow communication among the units.
- These units also called nodes or neurons are simple processors which operate in parallel.
- Every neuron is connected with other neurons through a connection link.
- Each connection link is associated with a weight.

that has info. about the input neuron \times solve a particular problem

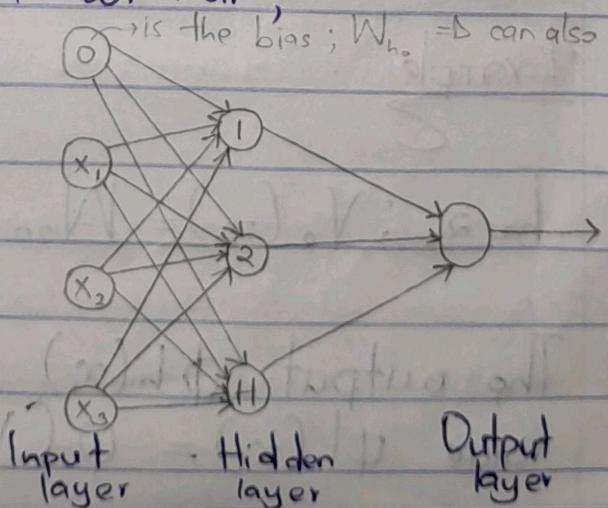
- This is the most useful info for neurons, because the weight $\#$ excites or inhibits the signal that is being communicated.

- Each neuron has an internal state which is called an activation signal
- Output signals which are produced after combining the input signal and an activation rule may be sent to other units
- Mathematically, a neural network is a non linear network transforming real input variables, x_1, x_2, \dots, x_p into 1 or several output variables y_1, y_2, \dots, y_q using several intermediate steps
- Graphically, a neural network is represented as a directed graph
- We defined an ANN with an:

input layer

hidden layer &

output layer



activation fxn - turn inputs to outputs

- The above input-output map has p input nodes, one layer of H hidden nodes and an activation fxn $\psi(x)$
- The inputs of hidden layer nodes are connected weights W_{hj} for $h \in \{1, 2, \dots, H\}$ and $j \in \{0, 1, 2, \dots, p\}$ where W_{h0} is the bias for the h^{th} hidden node.
- The hidden and output layers are connected weights α_h for $h \in \{1, 2, \dots, H\}$
- Considering an input vector X given by $x = \{x_1, x_2, \dots, x_p\} \in \mathbb{R}^p$. The input $V_h(x)$ to the h^{th} hidden node is the value

$$V_h(x) = W_{h0} + \sum W_{hj} x_j$$

Example

$$h=2 ; V_2(x) = W_{20} + \sum W_{2j} x_j$$

The output $\phi_h(x)$ of the h^{th} hidden node is value $\phi_h(x) = \psi(V_h(x))$

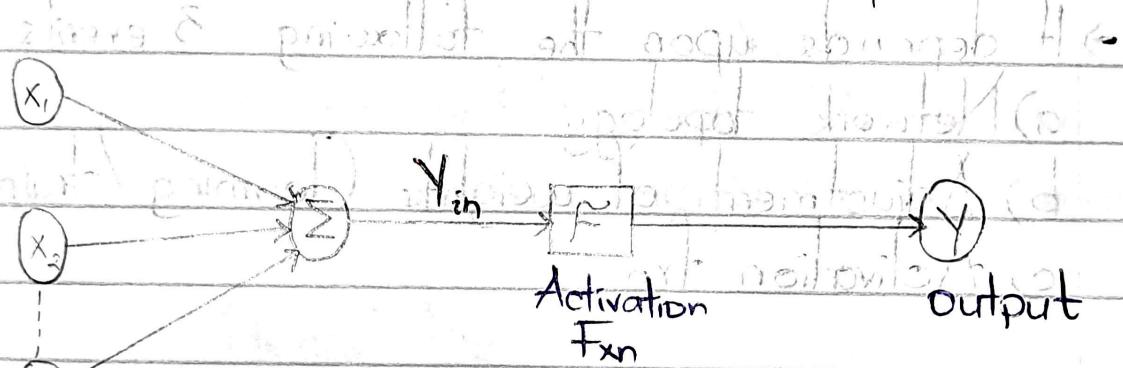
- The net input to the output node is the value

$$z(x) = \alpha_0 + \sum \alpha_n \phi_n(x)$$

Finally, the output $z(x)$ of the net is the value

$$z(x) = \psi(z(x))$$

→ The following diagram represents a perceptron:



→ For the above general model of ANN, the net input can be calculated as:

$$Y_{in} = w_1 x_1 + w_2 x_2 + \dots + w_p x_p$$

i.e., Net input = $Y_{in} = \sum_{i=1}^p x_i w_i$

The output can be calculated applying the activation fxn on the net input

$$Y = F(y_{in})$$

Processing of ANN

→ It depends upon the following 3 events

- Network topology
- Adjustment of weights (learning / training)
- Activation fxn

a) NETWORK TOPOLOGY

Refers to the arrangement of a network along with nodes and connecting lines

- According to network topology, ANN can be classified into the following;

i) Feed Forward Neural Network (FNN)

- Is one of the 2 broad types of artificial neural network characterized by direction of the flow of info btwn its layers

- Its flow is unidirectional: meaning that the flow of information in model flows only in one direction ie forward, from the input nodes through the hidden nodes (if any) and to the output nodes without
- FNN is further divided into

Neural

i) Single Layer Feed Forward Network

ii) Multilayer feed Forward Neural Network

iii) Feed Neural Network

• Also called recurrent neural network / Interactive NN
are networks in which info flow is bi-directional

b) ADJUSTMENT OF WEIGHTS / LEARNING

• Learning in ANN is the method of modifying/adjusting weights of connections b/w the neurons of a specified network.

• learning in ANN can be classified into 3 categories

i) Supervised

ii) Reinforcement learning

iii) Unsupervised

Supervised learning is a process of providing input data as well as correct output data to the ML model. The aim of sup. learning algo is to find mapping fcn to map the input variable X with the output variable y , i.e., it uses labelled data e.g., regression and classification problem.

Unsupervised learning is a type of ML problem in which training data consist of a set of input vectors but no corresponding target values (unlabelled data) & learn on itself without any supervision.

Reinforcement learning is a type of learning used to reinforce or strengthen the network over some critical info. The learning process is similar to supervised, however, we may have some very little info. During the training of the network under reinforcement learning, the net receives some feedback from the environment. This makes it somewhat similar to sup. learning; however, the feedback received obtained here is evaluative & not instructive, meaning there is no

teacher as in supervised learning. After receiving the feedback, the network performs adjustments of the weight to get better critical info in the future

c) ACTIVATION Fxn

- An activation fcn transforms the input to get the output. Examples:

- i) The linear activation fcn
- ii) The sigmoid fcn

$$f(x) = \frac{1}{1 + \exp(-x)}$$
 (Binary Sigmoid fcn)

$$f(x) = \frac{-2}{1 + \exp(-x)}$$
 (Bipolar sigmoid fcn)

Disadvantages of Neural Networks

1. Neural networks are a black box.
2. They take a long time to train.
3. They require lots of data.
4. They are computationally expensive.

Advantages

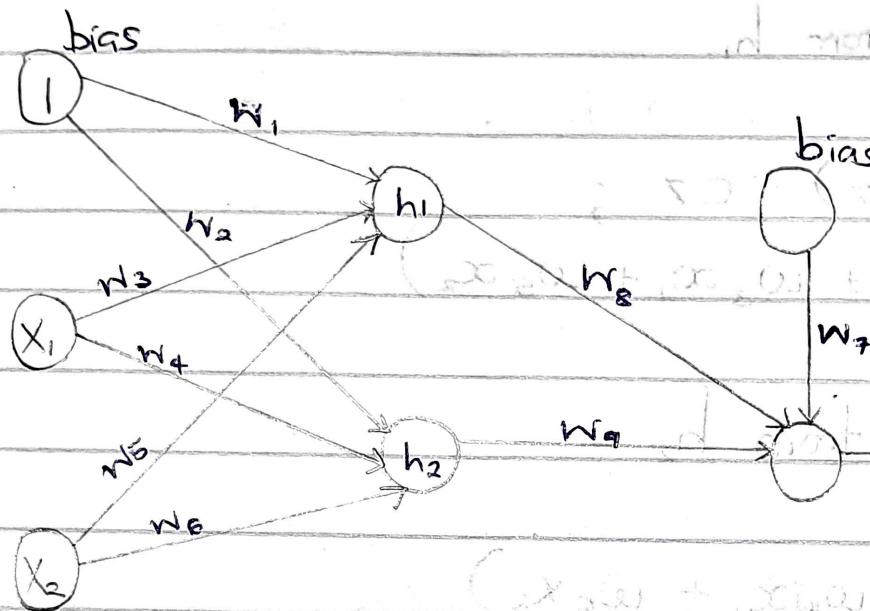
1. They can handle unorganized data.
2. They can improve accuracy.
3. They can increase flexibility.
4. They can lead to faster workflows.

Example - 19)

Consider a neural network for a binary classifier which has 1 hidden layer as shown in the figure. We use a linear activation fn:

$h(z) = cz$ at hidden units and a sigmoid fn $g(x) = \frac{1}{1 + e^{-x}}$ at the output unit to

learn the fn $P(y=1 | \alpha, w)$ where $\alpha = (\alpha_1, \alpha_2)$ and $w = (w_1, w_2, \dots, w_q)$



What is the output $P(y=1|x, \omega)$ from the above neural network?

S_n

→ Input at h_1 :

$$w_1 + w_3 x_1 + w_5 x_2$$

→ Input at h_2 :

$$w_2 + w_4 x_1 + w_6 x_2$$

⇒ Output from h_1 ,

Since $h(z) = cz$;

⇒ $c(\omega_1 + \omega_3x_1 + \omega_5x_2)$

⇒ Output from h_2

$c(\omega_2 + \omega_4x_1 + \omega_6x_2)$

⇒ Input to the output node (net output)

$\omega_7 + \omega_8c(\omega_1 + \omega_3x_1 + \omega_5x_2) + \omega_9c(\omega_2 + \omega_4x_1 + \omega_6x_2)$

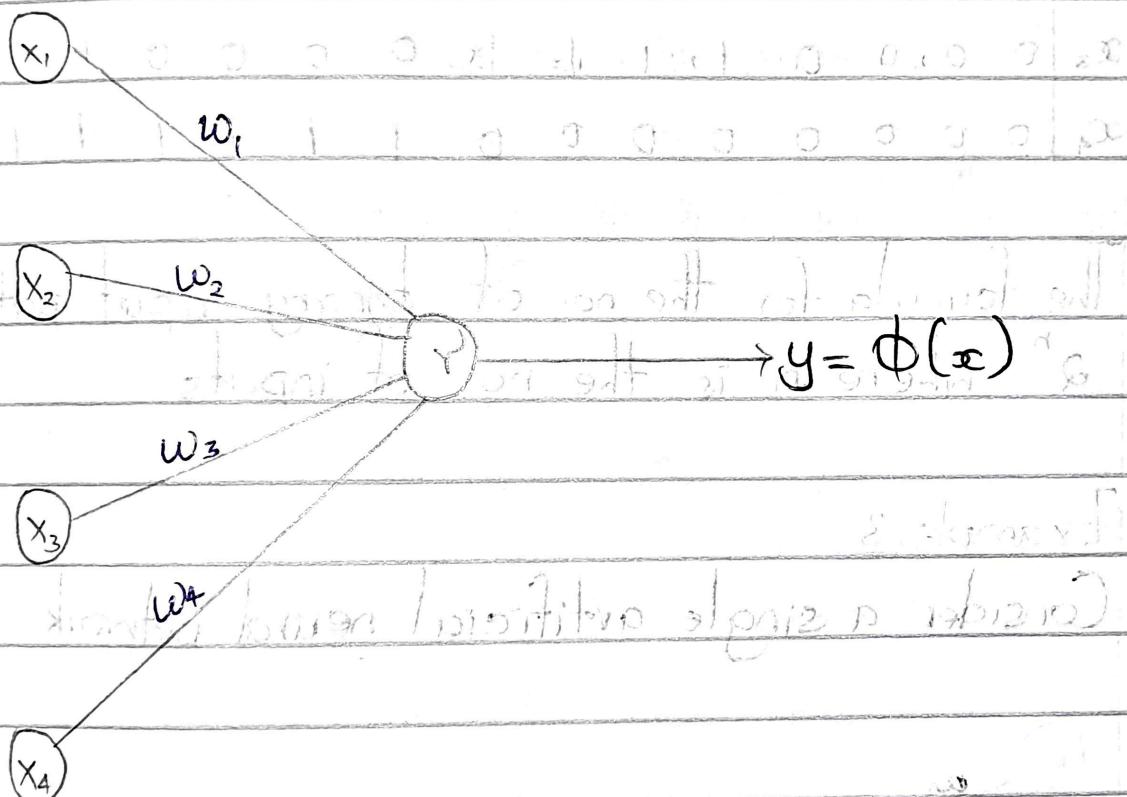
⇒ Output of the net (Output node)

Since $g(x) = \frac{1}{1+e^{-x}}$

= $\frac{1}{1+e^{-(\omega_7 + \omega_8c(\omega_1 + \omega_3x_1 + \omega_5x_2) + \omega_9c(\omega_2 + \omega_4x_1 + \omega_6x_2))}}$

Example 2

Below is a diagram of a single artificial neuron



The node has 4 inputs $x = (x_1, x_2, x_3, x_4)$ that receive only binary signals either zero or 1

- i) How many diff. input patterns can this node receive

Ans a 4x binary signal would have

combinations of 0's and 1's $2^4 = 16$

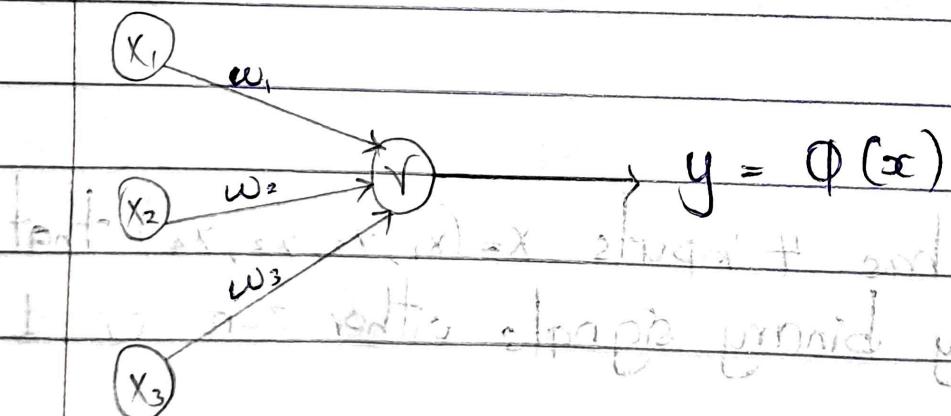
so there are 16 different input patterns

x_1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
x_2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x_3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x_4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

The formula for the no. of binary input patterns is 2^n where n is the no. of inputs

Example 3

Consider a single artificial neural network



The node has 3 inputs: $x = (x_1, x_2, x_3)$ that receive only binary inputs either zero or 1.

Suppose that the weights corresponding to the 3 inputs are: $w_1 = 2$; $w_2 = -4$ & $w_3 = 2$

The activation fn of e unit is given by the step fn

$$f_m(r) = (\omega_1 x_1) + (\omega_2 x_2) + (\omega_3 x_3) = r$$

$$\Phi(r) = \begin{cases} 1, & \text{if } r \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

Calc:

The output value y of the unit for each of the following input patterns

Pattern	P_1	P_2	P_3	P_4
x_1	1	0	1	1
x_2	0	1	0	1
x_3	0	1	1	1

S_h

To find the output value y for each pattern:

1. Calculate the weighted sum:

$$r = \sum_i w_i x_i = \omega_1 x_1 + \omega_2 x_2 + \omega_3 x_3$$

2. Apply the activation fn:

Ques: i) P₁, P₂, P₃ & P₄ are four points in the first quadrant such that

$$V = (2x_1) + (-4x_0) + (1x_0) = 2 \text{ volt}$$

$$\therefore V = 2 - 3x_0 \text{ volt}$$

Since $x_0 > 0$; $V = \phi(x) = 1$

$\therefore P_2, P_3 \& P_4$

do not lie on the same line. If we connect them, they will form a triangle.

∴ The potential difference between P₁ & P₂ is

$V_{P_1 P_2} = V - V_{P_1 P_3} = 2 - 1 = 1 \text{ volt}$

$V_{P_1 P_2} = V - V_{P_1 P_4} = 2 - 1 = 1 \text{ volt}$

$V_{P_1 P_2} = V - V_{P_1 P_3} = 2 - 1 = 1 \text{ volt}$



∴ The potential difference between P₁ & P₂ is 1 volt.

The potential difference between P₁ & P₃ is 1 volt.

The potential difference between P₁ & P₄ is 1 volt.

Example 4

Logical Operators (ie, NOT, AND, OR, NAND, XOR etc) are the building blocks of any computational device

Logical Fns return only 2 possible values, true or false, based on the true or false values of their arguments

P		Q		AND		OR		NOT		NAND		NOR		XOR	
T	F	F	T	T	F	F	T	F	T	F	T	F	T	F	T
F	T	F	T	F	T	T	F	T	F	T	F	T	F	T	F
F	F	F	F	F	T	T	F	T	F	T	F	T	F	T	F

- i) The AND Operation : The output is true if both input values are true. Else, the output will be false
- ii) NAND : The negation of ^{AND} operation gives the output for NAND
- iii) OR : The statement states that if any of the 2 input values are true, the output result is true always.

iv) NOR : The NOR operation gives an output opposite to OR operation. It means the statement which is true for OR is false for NOR.

v) XOR : This operation states that the input values should be exactly true or exactly false.

Eg; If we denote truth by 1 and false by 0, then the logical fxn AND can be represented by the following table

	P_1	P_2	P_3	$\neg P_3$
x_1	0	0	1	
x_2	0	1	0	1
$x_1 \text{ AND } x_2$	0	0	0	1

The fxn can be represented or implemented by a single unit with 2 inputs.

$$y = \phi(v)$$

$v = x_1 w_1 + x_2 w_2$

If the weights $w_1 = 1$ & $w_2 = 1$ and the activation function is

$$\phi(v) = \begin{cases} 1, & \text{if } v \geq 2 \\ 0, & \text{otherwise} \end{cases}$$

Note:

The threshold level is 2

Qstn:

Test how the Neural AND fn works.

Sln:

$$P_1 = V = (1 \times 0) + (1 \times 0) = 0$$

$$(0 < 2) \therefore y = \phi(0) = 0$$

$$P_2 = Y = (1 \times 1) + (1 \times 0) = 1$$

$$(1 < 2) \therefore y = \phi(1) = 0$$

$$P_3 = Y = (1 \times 0) + (1 \times 1) = 1$$

$$(1 < 2) \therefore y = \phi(1) = 0$$

$$P_4 = V = (1 \times 1) + (1 \times 1) = 2$$

$$(2 = 2) \therefore y = \phi(2) = 1$$

Qsn

Suggest how to change either the weights or the threshold level of this single unit in order to implement the logical OR fn.

(That is, true when at least one of the arguments is true)

Tuesday

21/11/23

KNN from a classification Perspective

$$S = (s_1x_1) + (s_2x_2) \dots Y = f$$

$$S = (s_i)\phi = Y \dots (s > 0)$$

- The KNN algorithm compares a new data entry to the values in a given dataset with different categories of classes. based on its closeness or similarity in a given range k of neighbors, the algorithm assigns the new data to a class or category in the data set.
- The steps are as follows:

$S = (s_1x_1) + (s_2x_2) \dots Y = f$

Step 1: Assign a value k ($s = k$)

Step 2: Calculate the distance btwn the new data

and all other entries. Arrange them in ascending order.

Step 3: Find the k-nearest neighbors to the new entry based on the calculated distances

Step 4: Assign the new data entry to the majority class in the nearest neighbors

Example

x	y	$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$	Distance in Ascending Order
Brightness	Saturation	Class	Distance
40	20	Red	25.00
50	50	Blue	33.54
60	90	Blue	68.01
10	25	Red	14.14
70	70	Blue	61.03
60	10	Red	47.17
25	80	Blue	45.28

Consider a new data entry:

Brightness Saturation Class

20 35 ?

and classify it into either Red or Blue using
 $k=5$ and using Euclidean distance

$$d(x,y) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

$$\sqrt{(20-40)^2 + (35-20)^2} =$$

$$\sqrt{(20-50)^2 + (35-50)^2} =$$

$$\sqrt{(20-60)^2 + (35-90)^2} = 85$$

$$\sqrt{(20-10)^2 + (35-25)^2} = 30$$

$$\sqrt{(20-70)^2 + (35-70)^2} = 71$$

$$\sqrt{(20-60)^2 + (35-10)^2} =$$

$$\sqrt{(20-25)^2 + (35-80)^2} =$$

Step 2: Arrange distances in Ascending order
(Done in table)

Since we chose $k=5$, we will only consider the first 5 rows

The majority within the 5 nearest neighbors, the new entry is Red

∴ We will classify the new entry as Red

i) Kernel Smoothing

- The idea of local linear regression

A kernel smoother is a statistical technique used to estimate a real valued fn as a weighted average of neighboring observed data.

Rather than give all pnts in the neighborhood equal weights, we can assign weights that die off smoothly with distance from the target pnt

The basic process is very simple. We proceed thru the data pnt by pnt for each data pnt, we generate a new value that is some fxn of the original value at that pnt & the surrounding data pts

The kernel is a weighting fxn & it gives more importance to closer observations without ignoring observations that are further away

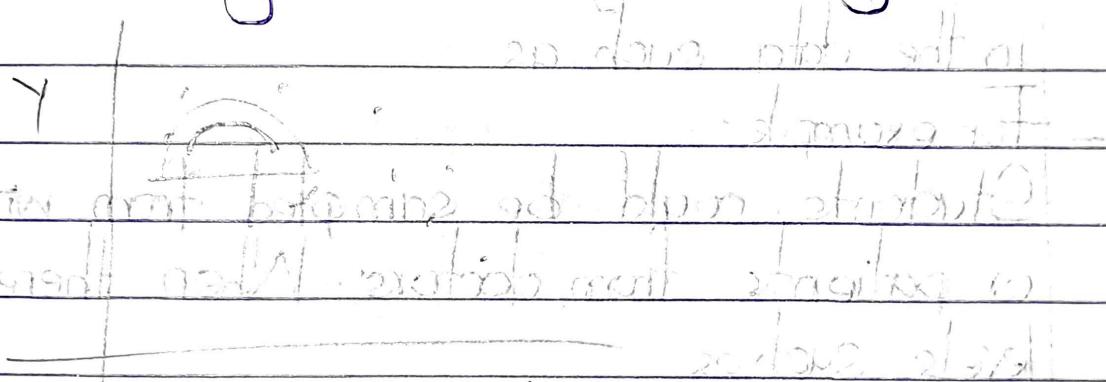
iii)

LOCAL LINEAR REGRESSION

The idea of local linear regression is to fit a straight line (a local linear regression line) to the pts in the neighborhood defined by the kernel fxn.

iv) LOCAL POLYNOMIAL REGRESSION

- Instead of fitting locally linear fns, one can fit polynomial fns eg; local quadratic fns to the neighborhoods defined by the kernel



Let $\pi_i \rightarrow$ performance of i student
Hierarchical

Nesting (Students are nested in classrooms)

Classroom \rightarrow School \rightarrow District

More than one level of nesting

more than one level of nesting

CHAPTER 5:

Linear Mixed Models

- They are an extension of simple linear models to allow for both random and fixed effects and are particularly used when there is non-independence in the data such as
- For example;

Students could be sampled from within classrooms or patients from doctors. When there are multiple levels such as

i) Patients seen by the same doctor

The variability in the outcome can be thought of as being either within group or between groups.

Fixed and Random Effects;

The core of Mixed Models is that they incorporate both fixed & Random Effects.

A fixed effect

→ Is a parameter that does not vary

Random effects

→ Are parameters that are themselves random variables

$z_u \rightarrow$ takes care of the structural variability

Example: bacon weight ($n=22$), multivariate

Consider a linear mixed model of the form

$$Y = X\beta + ZU + \Sigma$$

where:

- Y is an $n \times 1$ vector of n observed records
- β is a $p \times 1$ column vector of fixed effects regression coefficients
- U is a $q \times 1$ vector of q levels of random effects
- Σ is an $n \times 1$ vector of random residual or error terms
- X is a known design matrix of order $n \times p$ which relates the records in Y to the fixed effects in β .
- Z is a known design matrix of order $n \times q$ which relates the records in Y to the random effects in U

different treatments w.r.t to our dependent variable

Assumptions (for a linear mixed model)

1. The dpt var is assumed to be linearly related to the fixed factors, random factors & covariates.
2. The fixed effects model the mean of the dpt variable
3. The random effects model the covariance structure of the dpt variable
- + The dpt var is assumed to come from a normal distribution.

Expectation and Variance

$$E[Y] = E[X\beta] + E[Zu] + E[\Sigma]$$

$$= X\beta + 0 + 0 \quad \text{and } \Sigma = 0$$

$$= X\beta$$

The variance covariance structure of Y is:

$$\text{V} \begin{pmatrix} u \\ \Sigma \end{pmatrix} = \begin{bmatrix} G & 0 \\ 0 & R \end{bmatrix} \quad \text{u & } \Sigma \text{ are indpt}$$

where G is a dispersion matrix for the random effects other than errors & R is the dispersion matrix for error terms for which both are general square matrices assumed to be non-singular & positive definite with elements that are assumed known.

$$\text{Note } V(Y) = \sqrt{(X_B + ZU + \Sigma)}$$

$$= \sqrt{(X_B)} + \sqrt{(ZU)} + \sqrt{(\Sigma)}$$

$$= \sqrt{(ZU)} + \sqrt{(\Sigma)}$$

$$= Z^T G Z + R$$

In linear mixed models, we typically assume $u \sim N(0, G)$; $e \sim N(0, R)$, $\text{Cov}(u, \Sigma) = 0$

R code: LMER 4 } package (moja ya iyi)
 LME 4