MODULE 3

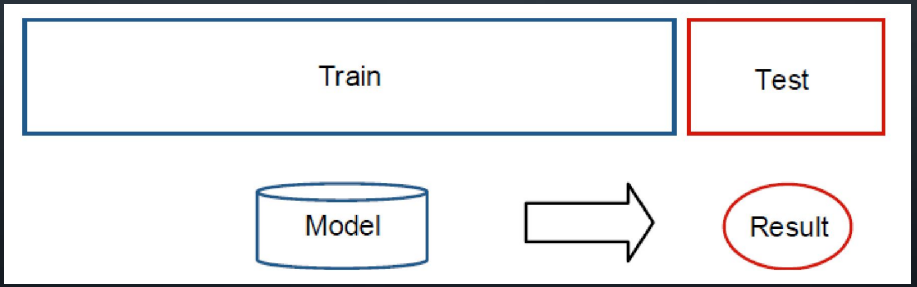
SLIDES ADAPTED (AND EXTENDED) FROM ETHEM ALPAYDIN © THE MIT PRESS, 2004

SYLLABUS

Classification- Cross validation and re-sampling methods- Kfold cross validation, Boot strapping, Measuring classifier performance- Precision, recall, ROC curves. Bayes Theorem, Bayesian classifier, Maximum Likelihood estimation, Density functions, Regression

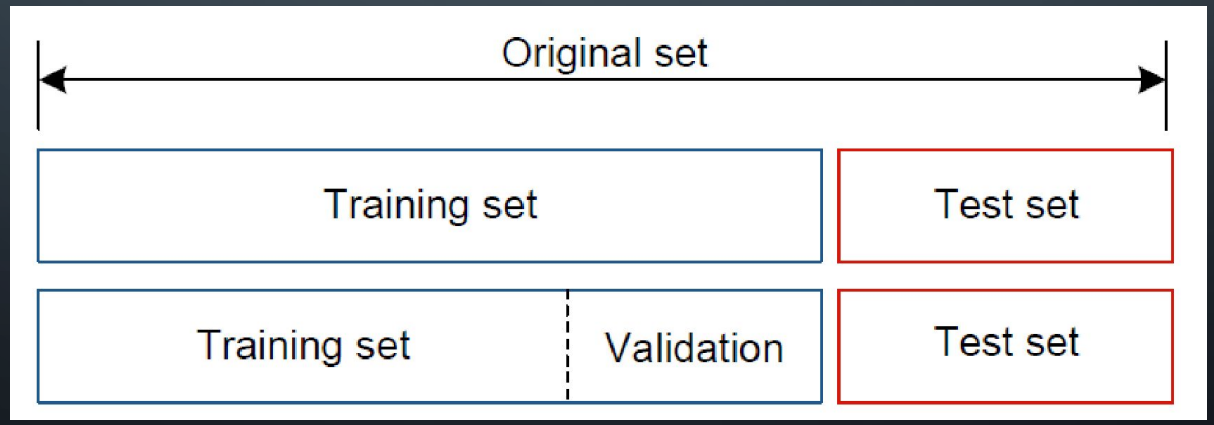
EVALUATION OF CLASSIFIERS

- Several Classification algorithms exist
- More than one classifier may be applicable to a problem
- Assess how good a selected algorithm is
- Datasets in ML are usually split into disjunct sets for training and testing
  - Training set is used to fit and calibrate the model parameters
  - Test set is used to measure the predictive performance on unseen data



VALIDATION SET

- Three-fold split into training, validation and test set
  - Fit model of different complexity to training data
  - Select model based on performance on unseen data from the validation set
  - Measure predictive performance based on the test set



TRAINING ERROR AND TESTING ERROR

- **Training error** results from applying the model to the training data
- **Test error** is the average error when predicting on unseen observations
- Alternative terms refer to **in-sample** and **out-of-sample** performance

METHODS OF EVALUATION

- Need for multiple validation sets:
  - Divide dataset into Training set and validation set
  - Accuracy assessed based on the performance on the validation set
  - Performance measure from single validation set does not give true picture
- Statistical Distribution of errors
  - Multiple classifiers obtained from same algorithm to average over randomness
  - Test the classifiers on multiple datasets & Record sample of validation errors
  - Evaluation is based on statistical distribution of validation errors.

METHODS OF EVALUATION

- No free lunch theorem
  - Conclusion drawn from one analysis is conditioned on the dataset given
  - Classifier is not general, comparison is done on some particular application
  - No ‘best’ learning algorithm
  - Any learning algorithm gives high accuracy for a dataset and low accuracy for another

METHODS OF EVALUATION

- Other factors for comparison:
  - Risks on error generalization
  - Training time and space complexity
  - Testing time and space complexity
  - Interpretability (knowledge extraction)
  - Ease of programmability

RESAMPLING

- Often only **limited data** is available for measuring performance
- Sometimes performance is subject to the (random) split
  - If splitting is repeated randomly, there might be a **high variability** across the results
  - Especially relevant for time-dependent or ordered data
  - Making splits **random** can be of importance here
- Model performance is often inferior the less data is used
- Resampling:
  - Repeatedly draw sub-samples from the given data set
  - Then use these splits to fit and assess the model

CROSS VALIDATION

- Validation :

  Deciding whether the numerical results quantifying hypothesized relationships between variables, are acceptable as descriptions of the data

- Cross- Validation :

  Technique to evaluate predictive models by partitioning the original sample into a training set to train the model, and a test set to evaluate it

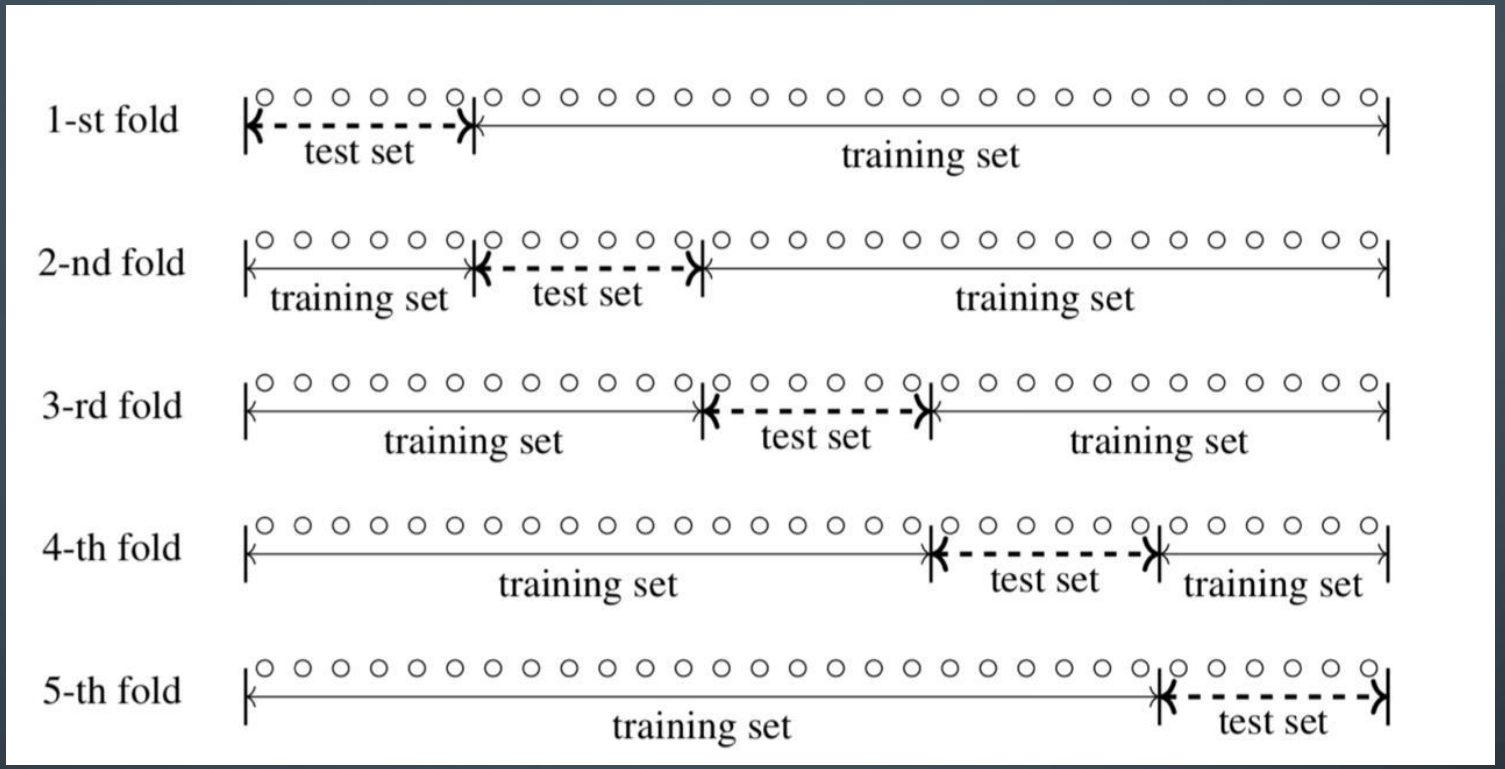
K – FOLD CROSS VALIDATION

- Dataset X is divided randomly into K equal-sized parts, Xi, i =1,….,K
- To generate each pair,
  - keep one of the K parts out as the validation set Vi
  - combine the remaining K − 1 parts to form the training set Ti
  - Doing this K times, each time leaving out another one of the K parts out
  - K pairs (Vi; Ti):

$$
\begin{aligned}
V_1 &= X_1, \quad T_1 = X_2 \cup X_3 \cup \ldots \cup X_K \\
V_2 &= X_2, \quad T_2 = X_1 \cup X_3 \cup \ldots \cup X_K \\
&\ldots \\
V_K &= X_K, \quad T_K = X_1 \cup X_2 \cup \ldots \cup X_{K-1}
\end{aligned}
$$

K – FOLD CROSS VALIDATION

- Two issues:
  - to keep the training set large, we allow small validation sets
  - the training sets overlap considerably, namely, any two training sets share K − 2 parts
- K is typically 10 or 30.
  - As K increases,
    - the percentage of training instances increases and we
    - get more robust estimators, but
    - the validation set becomes smaller.
  - There is the cost of training the classifier K times, which increases as K is increased



5 fold cross validation

- Leave one out cross validation:
  - only one instance used as validation set
  - N pairs for Dataset with N samples
  - Medical diagnosis

5*2 CROSS VALIDATION

- Dataset X is divided into two equal parts
- One half training set & the other half validation set
- Swap the halves
- This is the first fold
- Repeat 4 more times to get ten pairs of training sets and validation sets

$$
\begin{aligned}
T_1 &= X_1^{(1)}, & V_1 &= X_1^{(2)} \\
T_2 &= X_1^{(2)}, & V_2 &= X_1^{(1)} \\
T_3 &= X_2^{(1)}, & V_3 &= X_2^{(2)} \\
T_4 &= X_2^{(2)}, & V_4 &= X_2^{(1)} \\
&\vdots & & \\
T_9 &= X_5^{(1)}, & V_3 &= X_5^{(2)} \\
T_{10} &= X_5^{(2)}, & V_{10} &= X_5^{(1)}
\end{aligned}
$$

MEASURING ERROR

- Consider a binary classification model derived from a two-class dataset
- Let the class labels be c and ¬c
- Let x be a test instance

	Actual label of x is c	Actual label of x is $\neg c$
Predicted label of x is c	True positive	False positive
Predicted label of x is $\neg c$	False negative	True negative

- True positive
  - Let the true class label of x be c
  - If the model predicts the class label of x as c, the classification of x is true positive
- False negative
  - Let the true class label of x be c
  - If the model predicts the class label of x as ¬c, the classification of x is false negative
- True negative
  - Let the true class label of x be ¬c
  - If the model predicts the class label of x as ¬c, the classification of x is true negative
- False positive
  - Let the true class label of x be ¬c
  - If the model predicts the class label of x as c, the classification of x is false positive

CONFUSION MATRIX

- used to describe the performance of a classification model (or “classifier”)
- Used on a set of test data for which the true values are known
- a table that categorizes predictions according to whether they match the actual value

Two-class datasets

- a confusion matrix is a table with two rows and two columns
- It reports the number of false positives, false negatives, true positives, and true negatives

	Actual condition is true	Actual condition is false
Predicted condition is true	True Positive (TP)	False Positive (FP)
Predicted condition is false	False Negative (FN)	True Negative (TN)

Confusion Matrix for Binary Classifier

Multiclass datasets

- Confusion matrices can be constructed for multiclass datasets also

Eg. : a classification system trained to distinguish between cats, dogs and rabbits

- confusion matrix summarizes results of testing the algo for further inspection
- Assuming a sample of 27 animals - 8 cats, 6 dogs, and 13 rabbits

	Actual "cat"	Actual "dog"	Actual "rabbit"
Predicted "cat"	5	2	0
Predicted "dog"	3	3	2
Predicted " rabbit"	0	1	11

PRECISION AND RECALL

- two measures used to assess quality of results produced by a binary classifier

They are formally defined as follows:

- Let a binary classifier classify a collection of test data.
- Let
  - TP = Number of true positives
  - TN = Number of true negatives
  - FP = Number of false positives
  -

The *precision* P is defined as

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

The *recall* R is defined as

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Problem 1

Suppose a computer program for recognizing dogs in photographs identifies eight dogs in a picture containing 12 dogs and some cats. Of the eight dogs identified, five actually are dogs while the rest are cats. Compute the precision and recall of the computer program.

Solution

We have:

$$\begin{aligned} \text{TP} &= 5 \\ \text{FP} &= 3 \\ \text{FN} &= 7 \end{aligned}$$

The *precision* P is

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{5}{5+3} = \frac{5}{8}$$

The *recall* R is

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{5}{5+7} = \frac{5}{12}$$

Problem 2

Let there be 10 balls (6 white and 4 red balls) in a box and let it be required to pick up the red balls from them. Suppose we pick up 7 balls as the red balls of which only 2 are actually red balls. What are the values of precision and recall in picking red ball?

Solution

Obviously we have:

$$\begin{aligned} \text{TP} &= 2 \\ \text{FP} &= 7 - 2 = 5 \\ \text{FN} &= 4 - 2 = 2 \end{aligned}$$

The *precision* P is

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{2}{2+5} = \frac{2}{7}$$

The *recall* R is

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{2}{2+2} = \frac{1}{2}$$

OTHER MEASURES OF PERFORMANCE

1. Accuracy $= \dfrac{TP + TN}{TP + TN + FP + FN}$
2. Error rate $= 1 -$ Accuracy
3. Sensitivity $= \dfrac{TP}{TP + FN}$
4. Specificity $= \dfrac{TN}{TN + FP}$
5. F-measure $= \dfrac{2 \times TP}{2 \times TP + FP + FN}$

RECEIVER OPERATING CHARACTERISTIC (ROC)

TPR and FPR

- Let a binary classifier classify a collection of test data.

$$\text{TPR} = \text{True Positive Rate}$$
$$= \frac{\text{TP}}{\text{TP} + \text{FN}}$$
$$= \text{Fraction of positive examples correctly classified}$$
$$= \text{Sensitivity}$$
$$\text{FPR} = \text{False Positive Rate}$$
$$= \frac{\text{FP}}{\text{FP} + \text{TN}}$$
$$= \text{Fraction of negative examples incorrectly classified}$$
$$= 1 - \text{Specificity}$$

- TP = Number of t
- TN = Number of true negatives
- FP = Number of false positives
- FN = Number of false negatives

ROC SPACE

- plot the values of in a plane
  - FPR along the horizontal axis (that is , x-axis)
  - TPR along the vertical axis (that is, y-axis)
- For each classifier, there is a unique point in this plane with coordinates (FPR, TPR)
- The ROC space is the part of the plane whose points correspond to (FPR, TPR)
- Position of (FPR, TPR) gives an indication of the performance of the classifier

Consider some special points in the space:

- The left bottom corner point (0; 0): **Always negative**
  - All positive instances are wrongly predicted
  - All negative instances are correctly predicted
  - It commits no false positive errors.
- The right top corner point (1; 1): **Always positive**
  - All positive instances are correctly predicted
  - All negative instances are wrongly predicted
  - It commits no false negative errors
- The left top corner point (0; 1): **Perfect prediction**
  - It produces no false positives and no false negatives
- Points along the **diagonal**: **Random performance**

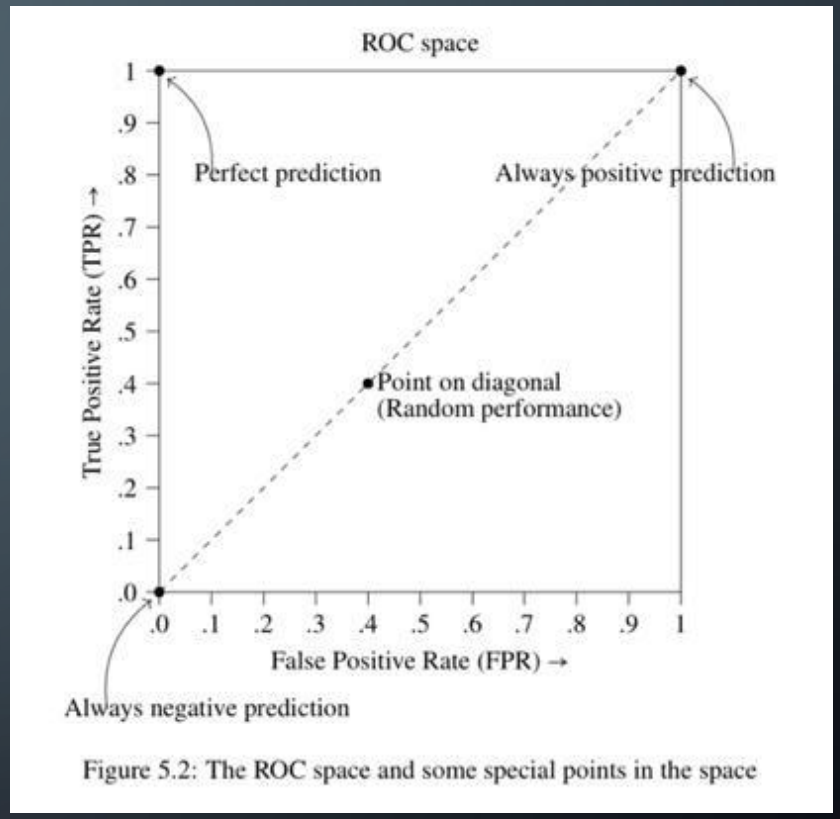


Figure 5.2: The ROC space and some special points in the space

ROC CURVE

- For certain algorithms, the classifier may depend on a parameter
- Different values of the parameter will give different classifiers
- These in turn give different values to TPR and FPR
- The ROC curve is the curve obtained by plotting (TPR , FPR) in the ROC space
- Obtained by assigning all possible values to the parameter in the classifier

- Closer the ROC curve is to the top left corner better the accuracy of the classifier

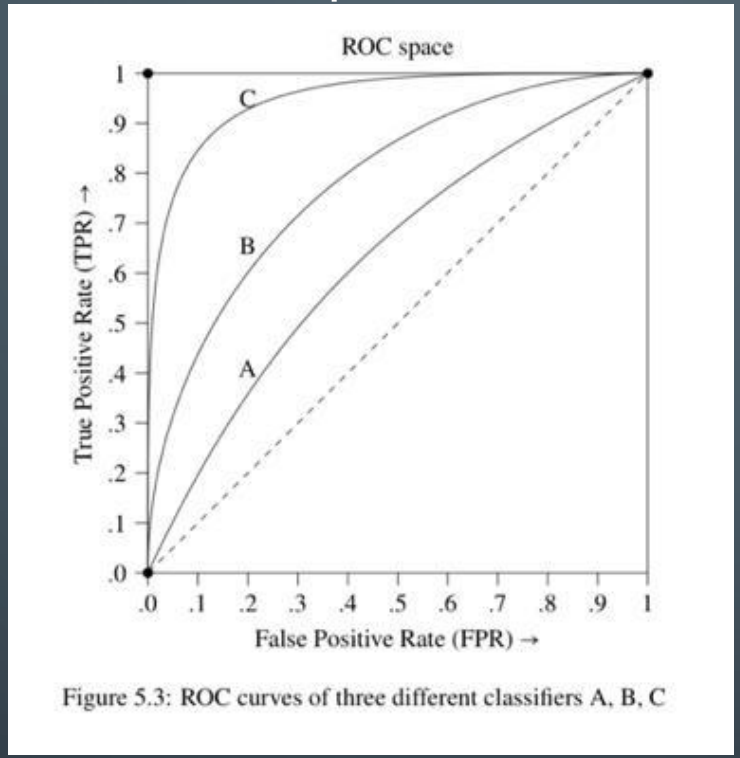


Figure 5.3: ROC curves of three different classifiers A, B, C

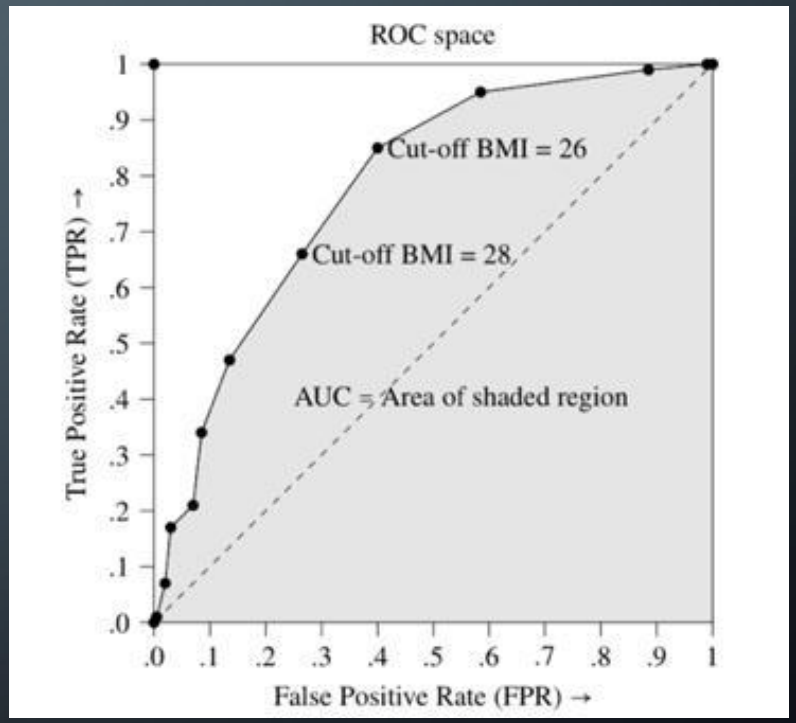
Area under the ROC curve

- The measure of the area under the ROC curve is denoted by the acronym AUC
- Value of AUC is a measure of the performance of a classifier
- For the perfect classifier, AUC = 1.0

EXAMPLE

Cut-off value of BMI	Breast cancer		Normal persons		TPR	FPR
	TP	FN	FP	TN		
18	100	0	200	0	1.00	1.000
20	100	0	198	2	1.00	0.990
22	99	1	177	23	0.99	0.885
24	95	5	117	83	0.95	0.585
26	85	15	80	120	0.85	0.400
28	66	34	53	147	0.66	0.265
30	47	53	27	173	0.47	0.135
32	34	66	17	183	0.34	0.085
34	21	79	14	186	0.21	0.070
36	17	83	6	194	0.17	0.030
38	7	93	4	196	0.07	0.020
40	1	99	1	199	0.01	0.005

Table 5.3: Data on breast cancer for various values of BMI



CONDITIONAL PROBABILITY

- Probability of the occurrence of an event A given that an event B has already occurred
- Conditional probability of A given B
- Denoted by P(A|B)
- It is given as:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad \text{if} \quad P(B) \neq 0$$

INDEPENDENT EVENTS

- Two events A and B are said to be independent if

$$P(A \cap B) = P(A)P(B)$$

- Three events A, B, C are said to be pairwise independent if

$$P(B \cap C) = P(B)P(C)$$
$$P(C \cap A) = P(C)P(A)$$
$$P(A \cap B) = P(A)P(B)$$

- Three events A, B, C are said to be mutually independent if

$$P(B \cap C) = P(B)P(C)$$
$$P(C \cap A) = P(C)P(A)$$
$$P(A \cap B) = P(A)P(B)$$
$$P(A \cap B \cap C) = P(A)P(B)P(C)$$

- In general a family of k events $A_1, A_2, \ldots, A_k$ is said to be mutually independent if for any subfamily consisting of $A_{i1}, \ldots\ldots, A_{im}$ we have

$$P(A_{i_1} \cap \ldots \cap A_{i_m}) = P(A_{i_1}) \ldots P(A_{i_m})$$

BAYES THEOREM

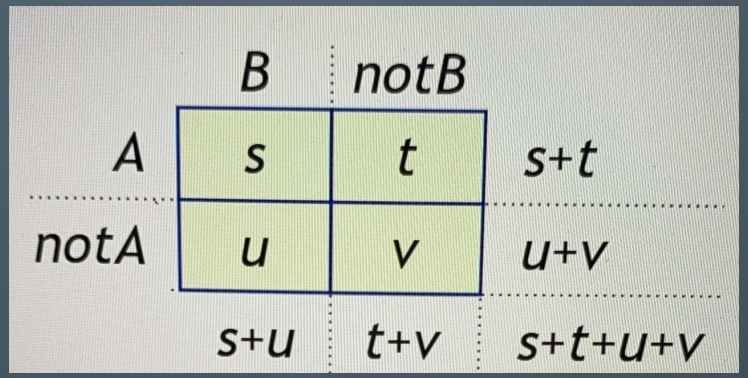
- Let A and B any two events in a random experiment. If P(A) ≠ 0, then

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}$$

  - it helps us to "invert" conditional probabilities
  - express the conditional probability P(A|B) in terms of P(B|A)

- Terminology:
  - A is called the proposition
  - B is called the evidence
  - P(A) is called the prior probability of proposition
  - P(B) is called the prior probability of evidence
  - P(A|B) is called the posterior probability of A given B
  - P(B|A) is called the likelihood of B given A

DERIVATION:



Now let us look at **probabilities**. So we take some ratios:

- the overall probability of "A" is $P(A) = \frac{s+t}{s+t+u+v}$
- the probability of "B given A" is $P(B|A) = \frac{s}{s+t}$

And then multiply them together like this:

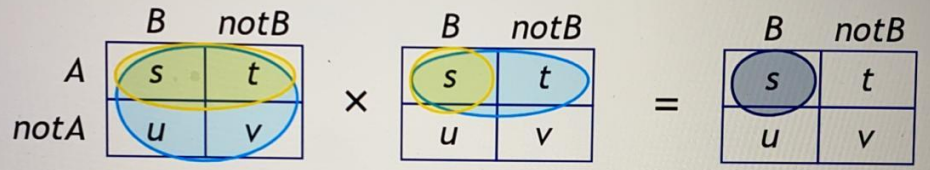
$$P(A) \times P(B|A) = P(A)\,P(B|A)$$

$$\frac{s+t}{s+t+u+v} \times \frac{s}{s+t} = \frac{s}{s+t+u+v}$$

And then multiply them together like this:

$$P(A) \times P(B|A) = P(A)\,P(B|A)$$

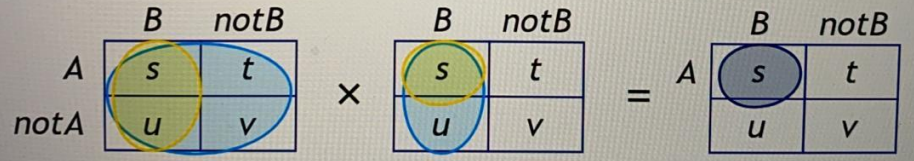
$$\frac{s+t}{s+t+u+v} \times \frac{s}{s+t} = \frac{s}{s+t+u+v}$$



Now let us do that again but use **P(B)** and **P(A|B)**:

$$P(B) \times P(A|B) = P(B)\,P(A|B)$$

$$\frac{s+u}{s+t+u+v} \times \frac{s}{s+u} = \frac{s}{s+t+u+v}$$



$$P(B)\,P(A|B) = P(A)\,P(B|A)$$

$$P(A|B) = \frac{P(A)\,P(B|A)}{P(B)}$$

GENERALISATION

- Let sample space be divided into disjoint events $B_1, B_2, \ldots, B_n$
- Let A be any event

$$P(B_k|A) = \frac{P(A|B_k)P(B_k)}{\sum_{i=1}^{n} P(A|B_i)P(B_i)}$$

- Example problems in notes (page 63)

NAIVE BAYES ALGORITHM

ASSUMPTIONS

1. All the **features** are **independent** and are unrelated to each other
2. The data has class-conditional independence
    - **events** are **independent** so long as they are conditioned on the same class value

- Assumptions are in general, true in many real world problems
- Hence called 'naïve'

BASIC IDEA

GIVEN

- Training data set consisting of N examples having n features
- Features be named as (F_1 , … ,F_n)
- feature vector is of the form (f_1, f_2,……, f_n)
- set of class labels be {c_1, c_2, …. , c_p}
- A class label is associated with each example
- Suppose we have test instance X
  - X = (x_1, x_2,……, x_n)

AIM:

To determine the most appropriate class label that should be assigned to the test instance

SOLUTION:

- we compute the following conditional probabilities

  $P(c_1 \mid X), P(c_2 \mid X), ….., P(c_p \mid X)$

- choose the maximum among them

- The direct computation of the probabilities are difficult
- Bayes' theorem can be applied to obtain a simpler method
- Using Bayes' theorem:

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

- The data has class-conditional independence
- Hence the events "$x_1|c_k$", "$x_2|c_k$", …. , "$x_n|c_k$"are independent

$$\begin{aligned} P(X|c_k) &= P((x_1, x_2, \ldots, x_n)|c_k) \\ &= P(x_1|c_k)P(x_2|c_k)\cdots P(x_n|c_k) \end{aligned}$$

- Using the previous equation:

$$P(c_k|X) = \frac{P(x_1|c_k)P(x_2|c_k)\cdots P(x_n|c_k)P(c_k)}{P(X)}.$$

Since the denominator $P(X)$ is independent of the class labels, we have

$$P(c_k|X) \propto P(x_1|c_k)P(x_2|c_k)\cdots P(x_n|c_k)P(c_k).$$

So it is enough to find the maximum among the following values:

$$P(x_1|c_k)P(x_2|c_k)\cdots P(x_n|c_k)P(c_k), \quad k = 1, \ldots, p.$$

The various probabilities in the above expression are computed as follows:

$$P(c_k) = \frac{\text{No. of examples with class label } c_k}{\text{Total number of examples}}$$

$$P(x_i \mid c_k) = \frac{\text{No. of examples with } j^{th} \text{ feature equal to } x_i \text{ and class label } c_k}{\text{No. of examples with class label ck}}$$

ALGORITHM: NAIVE BAYES

GIVEN

- Training data set consisting of N examples having n features
- Features be named as (F_1 , … ,F_n)
- Feature vector is of the form (f_1, f_2,……, f_n)
- Set of class labels be {c_1, c_2, …. , c_p}
- A class label is associated with each example
- Suppose we have test instance X

  X = (x_1, x_2,……, x_n)

ALGORITHM STEPS:

Step 1. Compute the probabilities $P(c_k)$ for $k = 1, \ldots, p$.

Step 2. Form a table showing the conditional probabilities

$$P(f_1|c_k), \quad P(f_2|c_k), \quad \ldots \quad , P(f_n|c_k)$$

for all values of $f_1, f_2, \ldots, f_n$ and for $k = 1, \ldots, p$.

Step 3. Compute the products

$$q_k = P(x_1|c_k)P(x_2|c_k)\cdots P(x_n|c_k)P(c_k)$$

for $k = 1, \ldots, p$.

Step 4. Find j such $q_j = \max\{q_1, q_2, \ldots, q_p\}$.

Step 5. Assign the class label c_j to the test instance X.

- Steps 1 and 2 constitute the learning phase of the algorithm.
- The remaining steps constitute the testing phase.
- For testing:
  - only the table of probabilities is required
  - the original data set is not required
- Example problem in notes (pg 65-67)

USING NUMERIC FEATURES WITH NAIVE BAYES ALGORITHM

- Naive Bayes algo can be applied only if features are categorical
  - The various probabilities are computed using the various frequencies
  - Frequencies can be counted only if each feature has a limited set of values
- If a feature is numeric, it has to be discretized before applying the algorithm
- Effected by putting the numeric values into categories known as bins
- Also known as binning
- Ideal when there are large amounts of data

Ways to discretize a numeric feature:

1. Find natural categories or cut points in the distribution of values
   - use these cut points to create the bins

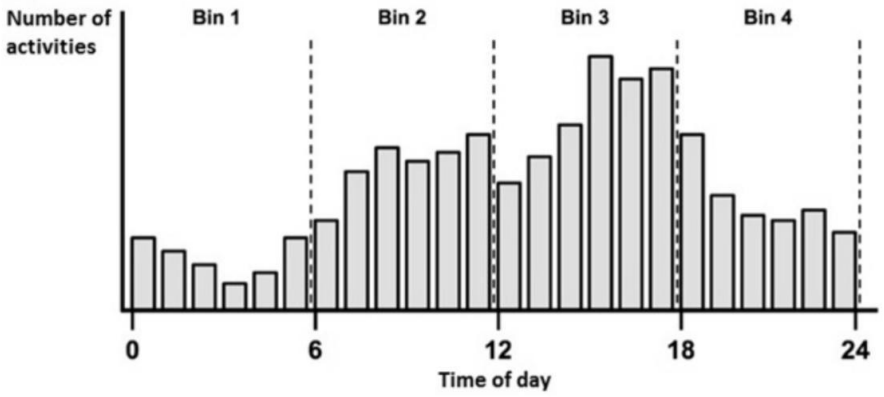


Figure 6.3: Discretization of numeric data: Example

2. If there are no obvious cut points, we may discretize the feature using quantiles,quartiles,deciles..

MAXIMUM LIKELIHOOD ESTIMATION (ML ESTIMATION)

- There is need to know whether the sample is **truly random**
- Computed probabilities are good approximations to true probabilities
- Underlying distribution has a particular form - binomial, Poisson or normal
- These forms are defined by probability functions or probability density functions.
  - Certain **Parameters** define these functions
  - Parameters to be estimated to test whether a given data follow some particular distribution
- Maximum likelihood estimation :

particular method to estimate the parameters of a probability distribution

MAXIMUM LIKELIHOOD ESTIMATION (MLE)

- Method of estimating the parameters of a statistical model, given observations
- Attempts to find the parameter values that maximize the likelihood function, given the observations
- The resulting estimate is called a maximum likelihood estimate
- It is also abbreviated as MLE

MLE METHOD

- random sample X = $\{x_1, \ldots, x_n\}$ taken from a probability distribution
- Probability mass function or probability density function

$$p(x|\theta)$$

Where: x denotes a value of the random variable

Theta denotes the set of parameters that appear in the function

- The likelihood that sample X is a function of theta is defined as:

$$l(\theta) = p(x_1|\theta)p(x_2|\theta)\ldots p(x_n|\theta)$$

- Log of the likelihood function is taken for computational convenience:

$$\begin{aligned} L(\theta) &= \log l(\theta) \\ &= \log p(x_1|\theta) + \log p(x_2|\theta) + \cdots + \log p(x_n|\theta) \end{aligned}$$

SPECIAL CASES

- Bernoulli Density
- Multinomial Density
- Gaussian (normal) Density

REGRESSION

- Predicting the value of a numeric variable based on observed values of the variable
  - Output variable(y): may be a number, such as an integer or a floating point value
  - Input variables(x): may be discrete or real-valued
  - Suppose we are required to estimate the price of a car -
    - aged 25 years
    - distance 53240 KM
    - weight 1200 pounds
  - General Approach: Find some mathematical relation between x and y

    $y = f(x,\theta)$

  - Optimize the parameters in the set such that the approximation error is minimized
  - The estimates of the values of the dependent variable y are as close as possible to the

Price (US$)	Age (years)	Distance (KM)	Weight (pounds)
13500	23	46986	1165
13750	23	72937	1165
13950	24	41711	1165
14950	26	48000	1165
13750	30	38500	1170
12950	32	61000	1170
16900	27	94612	1245
18600	30	75889	1245
21500	27	19700	1185
12950	23	71138	1105

REGRESSION

- The model may be $y = f(x, \theta)$

$$\text{Price} = a0 + a1 \times (\text{Age}) + a2 \times (\text{Distance}) + a3 \times (\text{Weight})$$

where x = (Age, Distance, Weight) denotes the set of input variables

θ = (a0, a1, a2, a3) denotes the set of parameters of the model

Different regression models:

- Simple linear regression: There is only one continuous independent variable x

$$y = a + bx$$

- Multivariate linear regression: There are more than one independent variable, say $x_1 .. x_n$

$$y = a_0 + a_1x_1 + \ldots + a_nx_n$$

- Polynomial regression: There is only one continuous independent variable x and the assumed model is

$$y = a_0 + a_1x + \ldots + a_nx^n$$

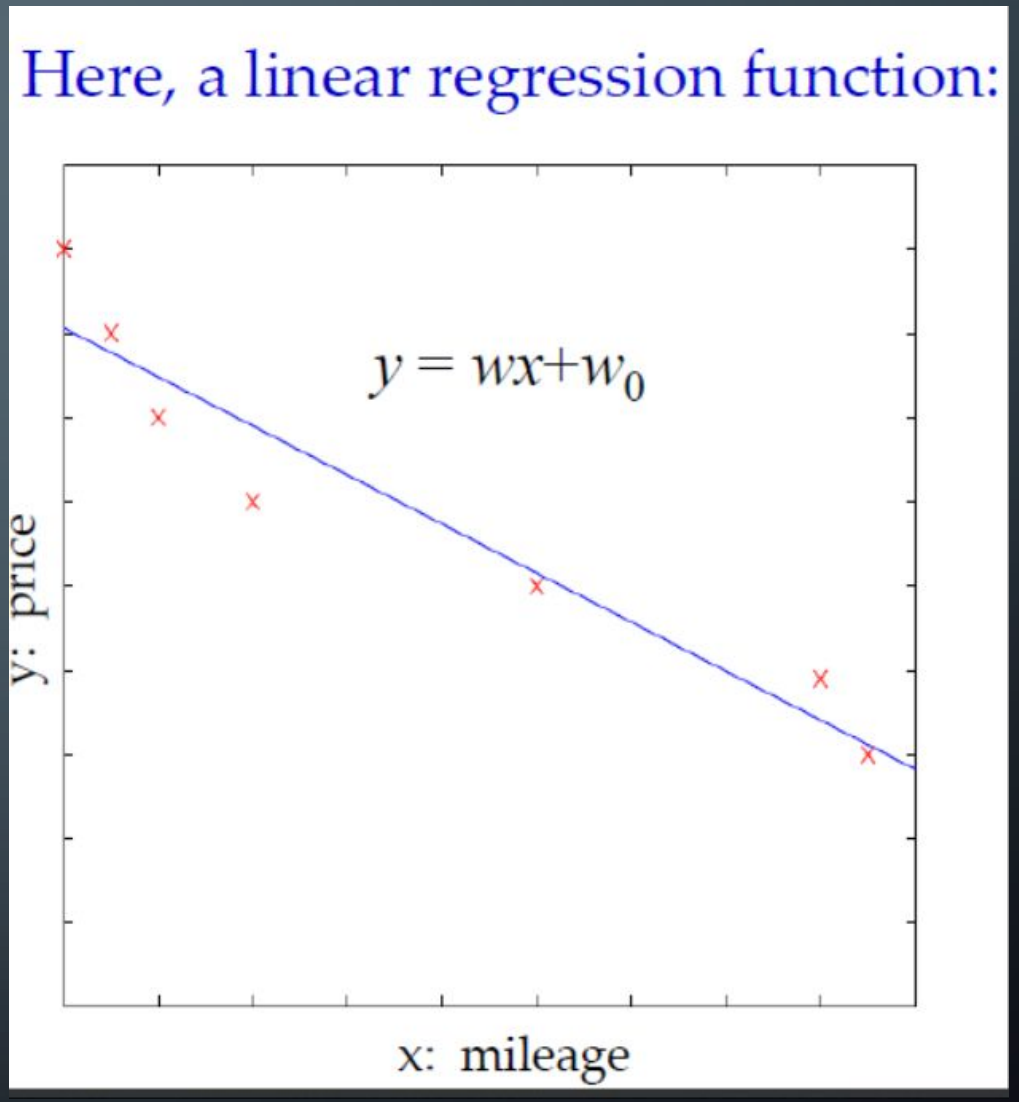
- Logistic regression: The dependent variable is binary.

REGRESSION

- E.g. Predict price of a used car
- (Input) x : car attributes

  (output) y : Price
- Task: Lean the mapping from input to output
  - G() model
  - θ parameters that minimize the error in the approximation

$$y=g(x|\theta)$$



CRITERION FOR MINIMIZING ERROR

- In regression
  - numeric output y, called the dependent variable
  - input x, called the independent variable
  - y is a function of x: $y = f(x) + \epsilon$
  - f(x) is unknown
  - approximate it by some estimator g(x, θ) containing a set of parameters θ
- Assume that the random error follows normal distribution with mean 0
- Values of θ which maximizes the likelihood function are the values of θ that minimizes the following sum of squares:

$$E(\theta) = (y_1 - g(x_1, \theta))^2 + \cdots + (y_n - g(x_n, \theta))^2$$

- Known as the ordinary least squares method

x	x_1	x_2	$\cdots$	x_n
y	y_1	y_2	$\cdots$	y_n

Table 7.1: Data set for simple linear regression

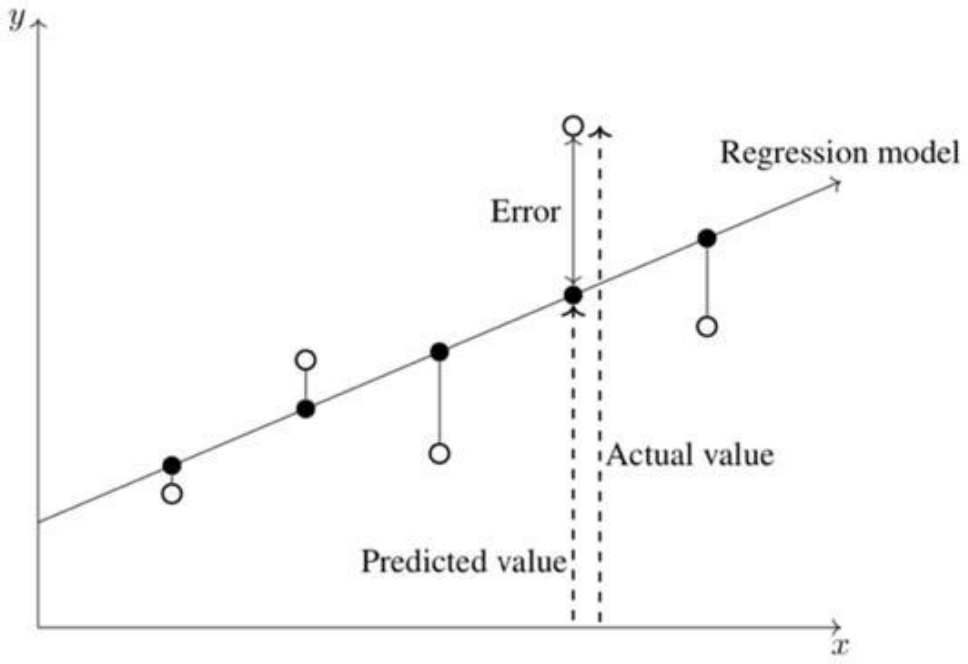


Figure 7.1: Errors in observed values

SIMPLE LINEAR REGRESSION

- x be the independent predictor variable
- y the dependent variable
- Assume that we have a set of observed values of x and y
- Simple linear regression model defines the relationship between x and y

$$y = \alpha + \beta x$$

- To determine the optimal estimates Ordinary Least Squares (OLS) is used

OLS METHOD

- Minimize the sum of the squared errors
- Values of y-intercept and slope are chosen accordingly
  - vertical distance between the predicted y-value and the actual y-value
  - sum of the squares of all the differences
- Then the sum of squares of errors is given by

$$E = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$
$$= \sum_{i=1}^{n} [y_i - (\alpha + \beta x_i)]^2$$

- Find the values α of β to minimize E
- Values of a and b for which E is minimum is obtained the following equations

$$\sum_{i=1}^{n} y_i = na + b\sum_{i=1}^{n} x_i$$

$$\sum_{i=1}^{n} x_i y_i = a\sum_{i=1}^{n} x_i + b\sum_{i=1}^{n} x_i^2$$

Formulas to find a and b

Recall that the means of x and y are given by

$$\bar{x} = \frac{1}{n}\sum x_i$$
$$\bar{y} = \frac{1}{n}\sum y_i$$

and also that the variance of x is given by

$$\text{Var}\,(x) = \frac{1}{n-1}\sum (x_i - \bar{x}_i)^2.$$

The *covariance of x and y*, denoted by $\text{Cov}(x, y)$ is defined as

$$\text{Cov}(x, y) = \frac{1}{n-1}\sum (x_i - \bar{x})(y_i - \bar{y})$$

It can be shown that the values of a and b can be computed using the following formulas:

$$b = \frac{\text{Cov}\,(x, y)}{\text{Var}\,(x)}$$
$$a = \bar{y} - b\bar{x}$$

- Example Problem:

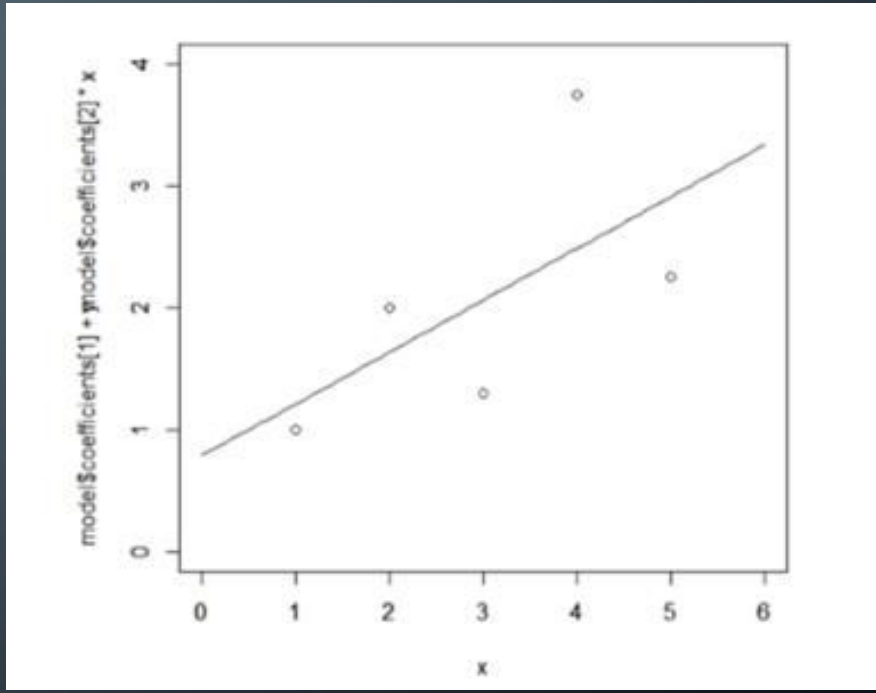
x	1.0	2.0	3.0	4.0	5.0
y	1.00	2.00	1.30	3.75	2.25

In the usual notations of simple linear regression, we have

$$
\begin{aligned}
n &= 5 \\
\bar{x} &= \frac{1}{5}(1.0 + 2.0 + 3.0 + 4.0 + 5.0) \\
&= 3.0 \\
\bar{y} &= \frac{1}{5}(1.00 + 2.00 + 1.30 + 3.75 + 2.25) \\
&= 2.06 \\
\text{Cov}\,(x, y) &= \frac{1}{4}[(1.0 - 3.0)(1.00 - 2.06) + \cdots + (5.0 - 3.0)(2.25 - 2.06)] \\
&= 1.0625 \\
\text{Var}\,(x) &= \frac{1}{4}[(1.0 - 3.0)^2 + \cdots + (5.0 - 3.0)^2] \\
&= 2.5 \\
b &= \frac{1.0625}{2.5} \\
&= 0.425 \\
a &= 2.06 - 0.425 \times 3.0 \\
&= 0.785
\end{aligned}
$$

Therefore, the linear regression model for the data is

$$y = 0.785 + 0.425x.$$



OTHER METHODS

- Different variants of the least squares method have been developed
- Examples:
  - Weighted least squares method

$$E = \sum_{i=1}^{n} w_i [y_i - (a + bx_i)]^2,$$

  - generalised least squares method
  - partial least squares method
  - total least squares method

POLYNOMIAL REGRESSION

- x be the independent predictor variable and y the dependent variable

A polynomial regression model defines the relationship between x and y by an equation in the following form:

$$y = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \cdots + \alpha_k x^k.$$

To determine the optimal values of the parameters $\alpha_0, \alpha_1, \ldots, \alpha_k$ the method of ordinary least squares is used. The values of the parameters are those values which minimizes the sum of squares:

$$E = \sum_{i=1}^{n} [y_i - (\alpha_0 + \alpha_1 x_i + \alpha_2 x_i^2 + \cdots + \alpha_k x_i^k)]^2.$$

The optimal values of the parameters are obtained by solving the following system of equations:

$$\frac{\partial E}{\partial \alpha_i} = 0, \quad i = 0, 1, \ldots, k. \tag{7.2}$$

Let the values of values of the parameters which minimizes E be

$$\alpha_i = a_i, \quad i = 0, 1, 2, \ldots, n. \tag{7.3}$$

Simplifying Eq. (7.2) and using Eq. (7.3), we can see that the values of a_i can be obtained by solving the the following system of $(k+1)$ linear equations:

$$\begin{aligned}
\sum y_i &= \alpha_0 n + \alpha_1 (\sum x_i) + \cdots + \alpha_k (\sum x_i^k) \\
\sum y_i x_i &= \alpha_0 (\sum x_i) + \alpha_1 (\sum x_i^2) + \cdots + \alpha_k (\sum x_i^{k+1}) \\
\sum y_i x_i^2 &= \alpha_0 (\sum x_i^2) + \alpha_1 (\sum x_i^3) + \cdots + \alpha_k (\sum x_i^{k+2}) \\
&\vdots \\
\sum y_i x_i^k &= \alpha_0 (\sum x_i^k) + \alpha_1 (\sum x_i^{k+1}) + \cdots + \alpha_k (\sum x_i^{2k})
\end{aligned}$$

Solving this system of linear equations we get the optimal values for the parameters.

The linear system of equations to find a_i's, has a compact matrix representation. We write:

$$D = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^k \\ 1 & x_2 & x_2^2 & \cdots & x_2^k \\ \vdots & & & & \\ 1 & x_n & x_n^2 & \cdots & x_n^k \end{bmatrix}, \quad \vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \vec{a} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix}$$

Then we have

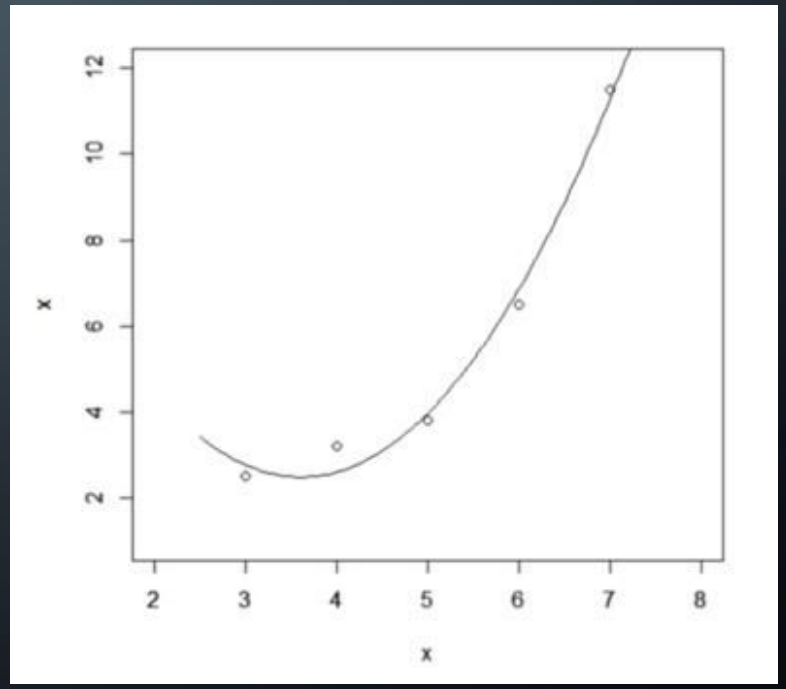
$$\vec{a} = (D^T D)^{-1} D^T \vec{y},$$

where the superscript T denotes the transpose of the matrix.

Example :

Find a quadratic regression model for the following data:

x	3	4	5	6	7
y	2.5	3.2	3.8	6.5	11.5



x	3	4	5	6	7
y	2.5	3.2	3.8	6.5	11.5

Solution

Let the quadratic regression model be

$$y = \alpha_0 + \alpha_1 x + \alpha_2 x^2.$$

The values of α_0, α_1 and α_2 which minimises the sum of squares of errors are a_0, a_1 and a_2 which satisfy the following system of equations:

$$\begin{aligned}
\sum y_i &= na_0 + a_1\left(\sum x_i\right) + a_2\left(\sum x_i^2\right) \\
\sum y_i x_i &= a_0\left(\sum x_i\right) + a_1\left(\sum x_i^2\right) + a_2\left(\sum x_i^3\right) \\
\sum y_i x_i^2 &= a_0\left(\sum x_i^2\right) + a_1\left(\sum x_i^3\right) + a_2\left(\sum x_i^4\right)
\end{aligned}$$

Using the given data we have

$$\begin{aligned}
27.5 &= 5a_0 + 25a_1 + 135a_2 \\
158.8 &= 25a_0 + 135a_1 + 775a_2 \\
966.2 &= 135a_0 + 775a_1 + 4659a_2
\end{aligned}$$

Solving this system of equations we get

$$\begin{aligned}
a_0 &= \ \ 12.4285714 \\
a_1 &= -5.5128571 \\
a_2 &= \ \ 0.7642857
\end{aligned}$$

The required quadratic polynomial model is

$$y = 12.4285714 - 5.5128571x + 0.7642857x^2.$$

MULTIPLE LINEAR REGRESSION

- N independent variables $x_1, x_2, \ldots, x_N$
- the dependent variable is y
- n observed values of these variables

Variables (features)	Values (examples)			
	Example 1	Example 2	⋯	Example n
x_1	x_{11}	x_{12}	⋯	x_{1n}
x_2	x_{21}	x_{22}	⋯	x_{2n}
⋯				
x_N	x_{N1}	x_{N2}	⋯	x_{Nn}
y (outcomes)	y_1	y_2	⋯	y_n

- Relationship between the N independent variables and the dependent variable is given by

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_N x_N$$

- Ordinary Least Sqares method is used here as well for optimal estimates of $\beta_0, \beta_1, \ldots, \beta_N$
- Thus we have :

$$X = \begin{bmatrix} 1 & x_{11} & x_{21} & \cdots & x_{N1} \\ 1 & x_{12} & x_{22} & \cdots & x_{N2} \\ \vdots & & & & \\ 1 & x_{1n} & x_{2n} & \cdots & x_{Nn} \end{bmatrix}, \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad B = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_N \end{bmatrix}$$

Then it can be shown that the regression coefficients are given by

$$B = (X^T X)^{-1} X^T Y$$

Example

Fit a multiple linear regression model to the following data:

x_1	1	1	2	0
x_2	1	2	2	1
y	3.25	6.5	3.5	5.0

Table 7.4: Example data for multi-linear regression

Solution

In this problem, there are two independent variables andfour sets of values of the variables. Thus, in the notations used above, we have $n = 2$ and $N = 4$. The multiple linear regression model for this problem has the form

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2.$$

The computations are shown below.

$$X = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 2 \\ 1 & 0 & 1 \end{bmatrix}, \quad Y = \begin{bmatrix} 3.25 \\ 6.5 \\ 3.5 \\ 5.0 \end{bmatrix}, \quad B = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}$$

$$X^T X = \begin{bmatrix} 4 & 4 & 6 \\ 4 & 6 & 7 \\ 6 & 7 & 10 \end{bmatrix}$$

$$(X^T X)^{-1} = \begin{bmatrix} \frac{11}{4} & \frac{1}{2} & -2 \\ \frac{1}{2} & 1 & -1 \\ -2 & -1 & 2 \end{bmatrix}$$

$$B = (X^T X)^{-1} X^T Y$$

$$= \begin{bmatrix} 2.0625 \\ -2.3750 \\ 3.2500 \end{bmatrix}$$

The required model is

$$y = 2.0625 - 2.3750x_1 + 3.2500x_2.$$

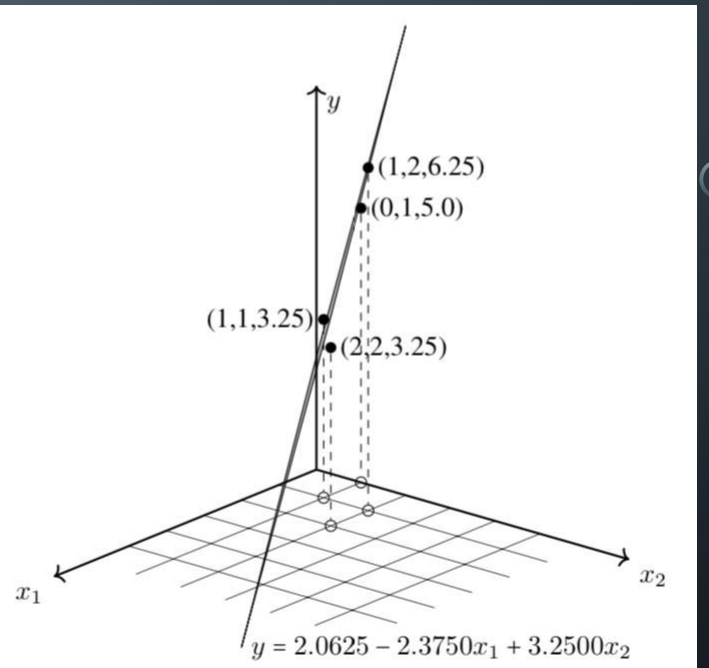


Figure 7.4: The regression plane for the data in Table 7.4

THANK YOU!