Theory of probability

Definition of Probability

- Let A be an event of Ω. Then the frequency ratio of A is

$$f(A) = \frac{n(A)}{n(\Omega)}$$

- If the random experiment Ω is repeated a large number of times under identical or uniform conditions then the frequency ratio of A will be approximately equal to its probability, i.e.,

$$P(A) \cong f(A)$$

- Thus, f(A) can be taken as to be an experimentally measured value of the idealised number P(A).
- Longer is the sequence of repetitions of Ω more accurate is the measured value.
- The definition of probability in terms of frequency interpretation restricts the class of random experiments, i.e., the random experiment must be repeated a large number of times in a uniform condition.

Example-1

- What is the probability that a positive integer selected at random from the set of positive integers not exceeding 100 is divisible by (i) 5, (ii)5 or 3 (iii)5 and 3?

Solution: $\Omega = \{1, 2, \ldots, 100\}$ so, $n(\Omega) = 100$

Let A be an event that no. is divisible by 5, so A = {5,10,…,100}; so n(A)= 20. So, $P(A) = \frac{n(A)}{n(\Omega)} = \frac{20}{100} = 0.2$

Let B be an event that no. is divisible by 3, so B ={3,6,…,99}, so n(B)=33; so $P(B) = \frac{n(B)}{n(\Omega)} = \frac{33}{100} = 0.33$

Example-1

- *Let C be an event that no. is divisible by 5 or 3, C =* A ∪ B, and n(A ∪ B)=47; so $P(C) = \frac{n(C)}{n(\Omega)} = \frac{47}{100} = 0.47$

- *Let D be an event that no. is divisible by 5 and 3, D = A ∩ B, and n(A ∩ B)=6; so* $P(D) = \frac{n(D)}{n(\Omega)} = \frac{6}{100} = 0.06$

Deduction of some important rules

1. For any event A, $0 \le n(A) \le n(\Omega), i.e., 0 \le \frac{n(A)}{n(\Omega)} \le 1$
   In the limit as $n(\Omega) \to \infty$, So, $0 \le P(A) \le 1$
2. For certain event S, $n(S) = n(\Omega)$, so in the limit P(S)=1
3. For impossible event O, n(O)=0. Hence, P(O)=0
4. For an event A, A+ A′=S (certain event)
   So, P(A+ A′)=P(S) => P(A) + P(A′)=1
   => P(A)=1-P(A′)

Addition rule for pairwise mutually exclusive events

For two mutually exclusive events, A and B, n(A ∪ B) = n(A+B)= n(A) + n(B)

So, $\frac{n(A+B)}{n(\Omega)} = \frac{n(A)}{n(\Omega)} + \frac{n(B)}{n(\Omega)}$, therefore, in the limit P(A+B)=P(A)+P(B)

- If A, B, C be pairwise mutually exclusive events, => $A \cap B = \emptyset$, $B \cap C = \emptyset$, and $C \cap A = \emptyset$

  Also => A and $B \cup C$ are mutually exclusive

so we have $P(A \cup (B \cup C)) = P(A) \cup P(B \cup C)$

- In general, if $A_1, A_2, \ldots, A_n$ be *n* pairwise mutually exclusive events then we have the following **addition rule**:

$$P(A_1 + A_2 + \cdots + A_n) = P(A_1) + P(A_2) + \cdots + P(A_n)$$

Conditional Probability

- Let us consider two events A and B of Ω. Let us make the hypothesis that the event A has occurred.
- Let n(A) times event A has occurred out of n(Ω).
- Let, among these n(A) occurrence of A, the event B also occurs (along with A) $n(A \cap B) i.e., n(AB)$ times.
- The ratio n(AB)/n(A) is called the conditional frequency ratio of B on the hypothesis that A has occurred and denoted by f(B/A), i.e., f(B/A)= n(AB)/n(A) =>

$$f(B/A) = \frac{n(AB)/n(\Omega)}{n(A)/n(\Omega)} = \frac{f(AB)}{f(A)}$$

- By Empirical or Statistical definition, $P(B/A) = \lim_{n(\Omega)\to\alpha} f(B/A)$
- We assume that, this limit exist, this limit is called the conditional probability of B on the hypothesis that A has occurred.
- So, as n(Ω) -> ∞, $P(B/A) = \frac{P(AB)}{P(A)}$ provided P(A) ≠ 0

Conditional Probability

- Similarly, $P(A/B) = \frac{P(AB)}{P(B)}$ provided P(B) ≠ 0

- Hence, if P(A), P(B) ≠ 0, we have $P(AB) = P(A)P(B/A) = P(B)P(A/B)$
  This is the Multiplication Rule

- **Addition rule**: $P(A_1 + A_2 + \cdots + A_n) = P(A_1) + P(A_2) + \cdots + P(A_n)$

- **Multiplication Rule:** $P(AB) = P(A)P(B/A) = P(B)P(A/B)$

General Addition Rule

General Addition Rule

- Let us consider two events A and B of Ω. In general, they are not mutually exclusive.
- But the events, A-AB, AB, and B-AB are always pairwise mutually exclusive.
- So, A=(A-AB)+AB, B=(B-AB)+AB, and A+B=(A-AB)+AB+(B-AB)
- By Addition rule for mutually exclusive events,
  P(A)= P(A-AB)+P(AB), P(B)=P(B-AB)+P(AB), and
  P(A+B)=P(A-AB)+P(AB)+P(B-AB)
  = P(A)-P(AB)+P(AB)+P(B)-P(AB) = P(A)+P(B)-P(AB)

i.e., **P(A+B)= P(A)+P(B)-P(AB)**

General Addition Rule

- For three events A, B, and C

$$P(A+B+C) = P(A+(B+C))=P(A)+P(B+C)-P(A(B+C))$$
$$= P(A)+P(B)+P(C)-P(BC)-P(AB+AC)$$
$$= P(A)+ P(B)+P(C)-P(BC)-[P(AB)+P(AC)-P(AB.AC)]$$
$$= P(A)+ P(B)+P(C)-P(BC)-P(AB)-P(AC)+P(AB.AC)$$
$$= \mathbf{P(A)+ P(B)+P(C)-P(AB)-P(BC)-P(CA)+P(ABC)}$$

- Generalising for n events, $A_1, A_2, \ldots, A_n$, we get

$$P(A_1 + A_2 + \cdots + A_n)$$
$$= P(A_1) + P(A_2) + \cdots + P(A_n) - P(A_1A_2) - P(A_1A_3) - \cdots - P(A_{n-1}A_n)$$
$$+ P(A_1A_2A_3) + P(A_1A_2A_4) + \cdots + P(A_{n-2}A_{n-1}A_n) + \cdots + (-1)^{n-1}P(A_1A_2 \ldots A_n)$$

Examples

1. A coin is tossed 3 times in succession. Find the probability of (a) 2 heads (b) 2 consecutive heads

Examples

1. A coin is tossed 3 times in succession. Find the probability of (a) 2 heads (b) 2 consecutive heads

Sol: (a) Here, $n(\Omega) = 2^3$

Let, A is the event that 2 heads occur, then $n(A) = 3_{C_2} = 3$

So, P(A) = $\frac{n(A)}{n(\Omega)}$ = 3/8

(b) Let B be the event that 2 consecutive heads occur, then n(B) = 3-1=2 [as head in first and third positions are not consecutive]

So, P(B)= $\frac{n(B)}{n(\Omega)}$ = 2/8=1/4

Examples

2. Two dice are thrown. Find the probability that the sum of the faces equals or exceeds 10.

Examples

2. Two dice are thrown. Find the probability that the sum of the faces equals or exceeds 10.

Sol: Here, $n(\Omega) = 6^2 = 36$

Let, A, B, and C denote the events 'Sum 10', 'Sum 11', and 'Sum 12' respectively. So, A+B+C is the required event, where A, B, and C are pairwise mutually exclusive.

So, P(A+B+C) = P(A)+P(B)+P(C).

Now, P(A)= 3/36 as (4,6), (5,5), and (6,4) lie in A,

P(B)= 2/36 as (5,6) and (6,5) lie in B, and

P(C)= 1/36 as only (6,6) lies in C

So, **P(A+B+C) =** P(A)+P(B)+P(C)=3/36+2/36+1/36=6/36=**1/6**

Generalisation of Conditional Probability

- Frequency ratio $f(B/A) = \frac{n(AB)/n(\Omega)}{n(A)/n(\Omega)} = \frac{f(AB)}{f(A)}$
- For a long sequence of repetitions of the random experiment under uniform conditions, the conditional frequency ratio f(B/A) is taken to be an approximate value of the conditional probability P(B/A).
- So, the conditional probability of B on the hypothesis that A has occurred is $P(B/A) = \frac{P(AB)}{P(A)}$

- Which gives the multiplication rule: $P(AB) = P(A)P(B/A) = P(B)P(A/B)$
- For three events A, B, C, we have $P(ABC) = P(A)P(B/A)P(C/AB)$

Generalisation of Conditional Probability

- For three events A, B, C, we have $P(ABC) = P(A)P(B/A)P(C/AB)$

Proof: R.H.S= $P(A)P(B/A)P(C/AB) = P(A)\frac{P(AB)}{P(A)}\frac{P(ABC)}{P(AB)} = P(ABC) = L.H.S$

- In general, for n events the **multiplication rule** is:

$$P(A_1A_2 \ldots A_n) = P(A_1)P(A_2/A_1)\,P(A_3/A_1A_2) \ldots . P(A_n/A_1A_2 \ldots A_{n-1})$$

Examples

1. A die is rolled. If the result is either an even face or a multiple of 3, then you win. What is the probability that multiple of 3 occurs on the hypothesis that even face occurs?

Examples

1. A die is rolled. If the result is either an even face or a multiple of 3, then you win. What is the probability that multiple of 3 occurs on the hypothesis that even face occurs?

Sol: Here, $n(\Omega) = 6$

Let, A and B denote the events 'even face', 'multiple of 3, respectively. So, B/A is the required event.

So, P(B/A) = P(AB)/P(A).

Now, P(A)= 3/6 as (2), (4), and (6) lie in A,

P(B)= 2/6 as (3) and (6) lie in B

P(AB)= 1/6 as only (6) lies in AB

So, **P(B/A) =** P(AB)/P(A)=(1/6)/ (3/6)=**1/3**

Similarly, P(A/B)=P(AB)/P(B)=1/2

Examples

2. Two cards are drawn successively from a pack without replacing the first. If the first card is a spade, find the probability that the second card is also a spade.

Examples

2. Two cards are drawn successively from a pack without replacing the first. If the first card is a spade, find the probability that the second card is also a spade.

Sol1: Let A = first card is a spade, B=second card is a spade.

So, AB=both cards are spades.

$n(\Omega)$ = 52. 51, n(A)=13. 51, n(AB)=13.12

So, P(B/A)=P(AB)/P(A) = n(AB)/n(A)=(13.12) / (13.51) = 4/17

Sol2: When the first card is seen to be a spade, there are 51 cards remain in the pack out of which 12 are spade.

Hence, the probability that the second card is also a spade is 12/51=4/17

Bayes' Theorem

- Theorem: If $A_1, A_2, \ldots, A_n$ be a given set of n pairwise mutually exclusive events, one of which certainly occurs, i.e., $A_i A_j = \emptyset \ (i \neq j; i, j = 1,2, \ldots, n)$ and $A_1 + A_2 + \cdots + A_n = S$

  then for any arbitrary event X,

(i) $P(X) = P(A_1)P(X/A_1) + P(A_2)P(X/A_2) + \cdots + P(A_n)P(X/A_n)$

(ii) Bayes' theorem: if P(X) ≠ 0,

$$P(A_i/X) = \frac{P(A_i)P(X/A_i)}{P(A_1)P(X/A_1) + P(A_2)P(X/A_2) + \cdots + P(A_n)P(X/A_n)} \ (for\ i = 1, 2, \ldots, n)$$

Proof: For any event X, we have X=SX= $(A_1 + A_2 + \cdots + A_n)X = A_1X + A_2X + \cdots + A_nX$

Since, $(A_iX)(A_jX) = A_iA_jX = \emptyset X = \emptyset, for\ i \neq j; i, j = 1,2, \ldots, n$

Therefore, $A_1X, A_2X, \ldots, A_nX$ are pairwise mutually exclusive events, and hence $P(X) = P(A_1X + A_2X + \cdots + A_nX) = P(A_1X) + P(A_2X) + \cdots + P(A_nX)$ [Addition Rule]

Since, $P(A_iX) = P(A_i)P(X/A_i)$ [Multiplication Rule]

Therefore, $P(X) = P(A_1)P(X/A_1) + P(A_2)P(X/A_2) + \cdots + P(A_n)P(X/A_n)$ [(i) is proved]

Bayes' Theorem

We have already proved that $P(X) = P(A_1)P(X/A_1) + P(A_2)P(X/A_2) + \cdots + P(A_n)P(X/A_n)$

Now we have to prove the Bayes' theorem: i.e., if P(X) ≠ 0,

$$P(A_i/X) = \frac{P(A_i)P(X/A_i)}{P(A_1)P(X/A_1) + P(A_2)P(X/A_2) + \cdots + P(A_n)P(X/A_n)} \; (for\ i = 1, 2, \ldots, n)$$

Proof: $P(A_iX) = P(A_i)P(X/A_i)$ [Multiplication Rule]

Also, $P(A_iX) = P(X)P(A_i/X)$ [Multiplication Rule]

Hence, if P(X) ≠ 0,

$$P(A_i/X) = \frac{P(A_iX)}{P(X)} = \frac{P(A_i)P(X/A_i)}{P(X)} = \frac{P(A_i)P(X/A_i)}{P(A_1)P(X/A_1) + P(A_2)P(X/A_2) + \cdots + P(A_n)P(X/A_n)}$$

Thus the Bayes' theorem is proved.

Example on Bayes' Theorem

Example-1: There are three identical urns containing white and black balls. The first urn contains 2 white and 3 black balls, the second urn 3 white and 5 black balls, and the third urn 5 white and 2 black balls. An urn is chosen at random, and a ball is drawn from it. If the ball drawn is white, what is the probability that the second urn is chosen?

Solution:

- Let A= the event that the ball is drawn from the first urn, B= the event that the ball is drawn from the second urn, and C= the event that the ball is drawn from the third urn.
- The events A, B, and C are pairwise mutually exclusive, and one of these necessarily occurs.
- P(A) is the probability that 1^{st} urn is chosen, and so on. So, P(A)=P(B)=P(C)=1/3
- Let X=the event that ball drawn is white. So, P(X/A)=2/5, P(X/B)=3/8, P(X/C)=5/7.

- We have to compute P(B/X)

Example-1 Cont...
Compute P(B/X)

- $P(X)=P(A)P(X/A) + P(B)P(X/B)+P(C)P(X/C)$
  $=(1/3)(2/5)+ (1/3)(3/8)+ (1/3)(5/7)$
  $=(1/3)[2/5+3/8+5/7]$
  $=(1/3)(417/280) = 139/280$
- So, By Bayes' theorem:

$P(B/X) = [P(X/B)\ P(B)]/\ P(X)$
$= [(3/8)(1/3)] / (139/280)$
$=35/139$ (Ans.)

Bayesian Classifier

Bayesian Classification: Why?

- A statistical classifier: performs *probabilistic prediction, i.e.,* predicts class membership probabilities
- Foundation: Based on Bayes' Theorem.
- Performance: A simple Bayesian classifier, *naïve Bayesian classifier*, has comparable performance with decision tree and selected neural network classifiers

Bayesian Classification

$$P(B|A) = \frac{P(A|B).P(B)}{P(A)}$$ Bayes' Theorem

P(B|A) = Posterior Probability

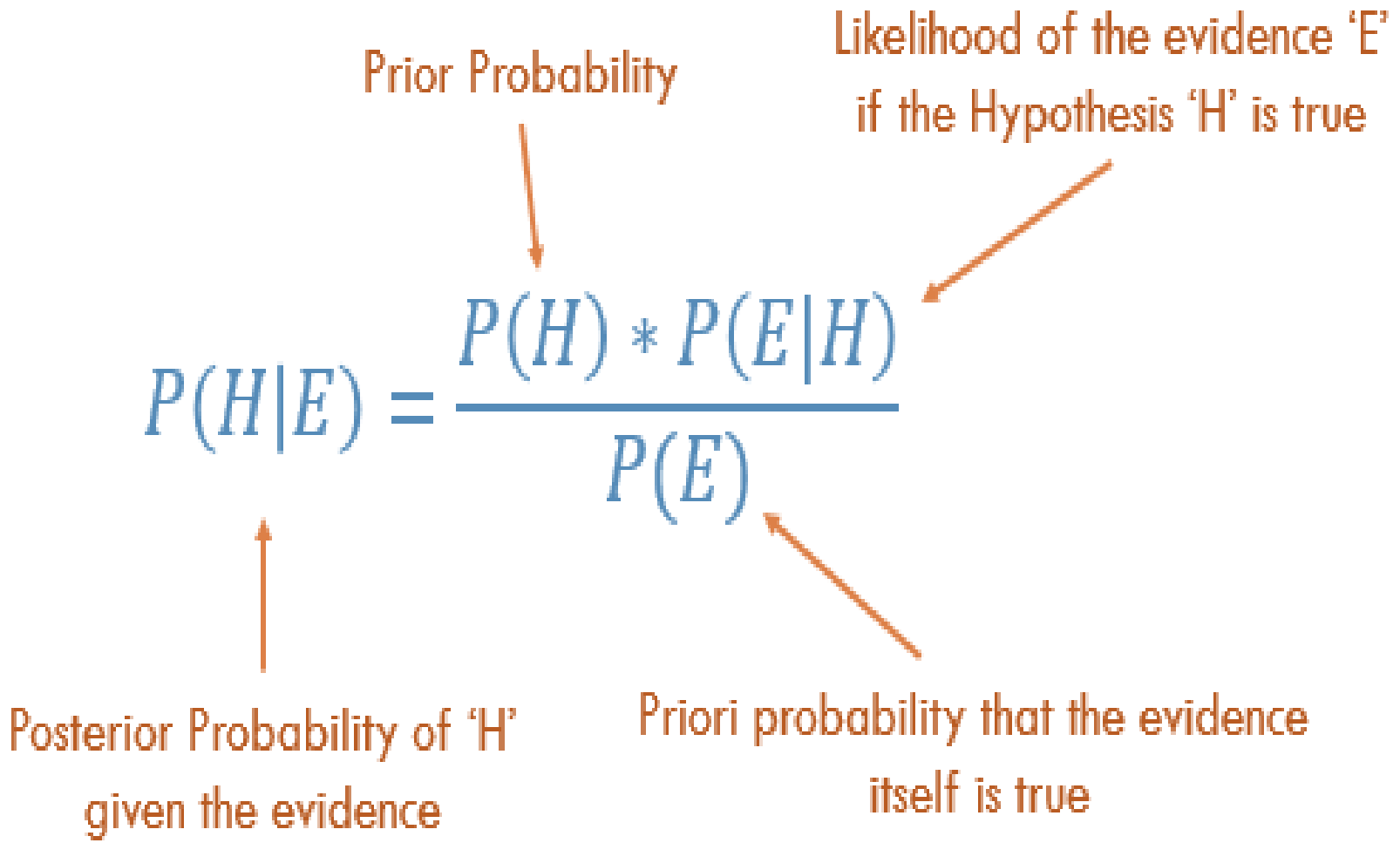
P(A|B) = Likelihood

P(B) = Prior Probability

-> Posterior refers to probability obtained *after* evidence (data) has been taken into consideration.

-> Prior refers to probability *before* evidence (data) has been taken into consideration.

Bayesian Classification



Bayes' Theorem: Basics

- Bayes' Theorem: $$P(H \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid H)P(H)}{P(\mathbf{X})} = P(\mathbf{X} \mid H) \times P(H) / P(\mathbf{X})$$
  - Let **X** be a data sample: class label is unknown
  - Let H be a *hypothesis* that X belongs to class C
  - Classification is to determine P(H|**X**), (i.e., *posteriori probability):* the probability that the hypothesis holds given the observed data sample **X**
  - P(H) (*prior probability*): the initial probability
    - E.g., **X** will buy computer, regardless of age, income, student, credit_rating.
  - P(**X**): probability that sample data is observed
  - P(**X**|H) : the probability of observing the sample **X**, given that the hypothesis holds
    - E.g., Given that **X** will buy computer, the prob. that X is 31..40, medium income

Prediction Based on Bayes' Theorem

- Given training data **X**, *posteriori probability of a hypothesis* H, P(H|**X**), follows the Bayes' theorem

$$P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})} = P(\mathbf{X}|H) \times P(H) / P(\mathbf{X})$$

- Predicts **X** belongs to C_i iff the probability $P(C_i|\mathbf{X})$ is the highest among all the $P(C_k|X)$ for all the *k* classes
- Practical difficulty: It requires initial knowledge of many probabilities, involving significant computational cost

Classification Is to Derive the Maximum Posteriori

- Let D be a training set of tuples and their associated class labels, and each tuple is represented by an n-D attribute vector $\mathbf{X} = (x_1, x_2, \ldots, x_n)$
- Suppose there are *m* classes $C_1, C_2, \ldots, C_m$.
- Classification is to derive the maximum posteriori, i.e., the maximal $P(C_i|\mathbf{X})$
- This can be derived from Bayes' theorem

$$P(C_i|\mathbf{X}) = \frac{P(\mathbf{X}|C_i)P(C_i)}{P(\mathbf{X})}$$

- Since P(X) is constant for all classes, so only

needs to be maximized $P(C_i|\mathbf{X}) = P(\mathbf{X}|C_i)P(C_i)$

Naïve Bayes Classifier

- A simplified assumption: attributes are conditionally independent (i.e., no dependence relation between attributes):

$$P(\mathbf{X}|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times ... \times P(x_n|C_i)$$

- This greatly reduces the computation cost: Only counts the class distribution
- If A_k is categorical, $P(x_k|C_i)$ is the # of tuples in C_i having value x_k for A_k divided by $|C_{i,D}|$ (# of tuples of C_i in D)
- If A_k is continous-valued, $P(x_k|C_i)$ is usually computed $g(x_k, \mu_{C_i}, \sigma_{C_i})$ based on Gaussian distribution with a mean μ and standard deviation σ , where

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

Naïve Bayes Classifier: Training Dataset

Class:
C1:buys_computer = ‘yes’
C2:buys_computer = ‘no’

Data to be classified:
X = (age <=30,
Income = medium,
Student = yes
Credit_rating = Fair)

age	income	student	credit_rating	comp
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

An Example

age	income	student	credit_rating	com
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

- $P(C_i)$: P(buys_computer = "yes") = 9/14 = 0.643
  P(buys_computer = "no") = 5/14= 0.357
- Compute $P(X|C_i)$ for each class
  P(age = "<=30" | buys_computer = "yes") = 2/9 = 0.222
  P(age = "<= 30" | buys_computer = "no") = 3/5 = 0.6
  P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444
  P(income = "medium" | buys_computer = "no") = 2/5 = 0.4
  P(student = "yes" | buys_computer = "yes) = 6/9 = 0.667
  P(student = "yes" | buys_computer = "no") = 1/5 = 0.2
  P(credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667
  P(credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4
- **X = (age <= 30 , income = medium, student = yes, credit_rating = fair)**
- **$P(X|C_i)$:** P(X|buys_computer = "yes") = 0.222 x 0.444 x 0.667 x 0.667 = 0.044
  P(X|buys_computer = "no") = 0.6 x 0.4 x 0.2 x 0.4 = 0.019
- **$P(X|C_i)*P(C_i)$:** P(X|buys_computer = "yes") * P(buys_computer = "yes") = 0.044 x 0.643 = 0.028
  P(X|buys_computer = "no") * P(buys_computer = "no") = 0.019 x 0.357 = 0.007
- **Therefore, X belongs to class ("buys_computer = yes")**

Avoiding the Zero-Probability Problem

- Naïve Bayesian prediction requires each conditional prob. be **non-zero**. Otherwise, the predicted prob. will be zero

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i)$$

- Ex. Suppose a dataset with 1000 tuples, income=low (0), income= medium (990), and income = high (10)
- Use **Laplacian correction** (or Laplacian estimator)
  - *Adding 1 to each case*

    Prob(income = low) = 1/1003

    Prob(income = medium) = 991/1003

    Prob(income = high) = 11/1003
  - The "corrected" prob. estimates are close to their "uncorrected" counterparts

Naïve Bayes Classifier: Comments

- Advantages
  - Easy to implement
  - Good results obtained in most of the cases
- Disadvantages
  - Assumption: class conditional independence, therefore loss of accuracy
  - Practically, dependencies exist among variables
    - E.g., hospitals: patients=> Profile: age, family history, etc.

      Symptoms: fever, cough etc., Disease: lung cancer, diabetes, etc.
    - Dependencies among these cannot be modeled by Naïve Bayes Classifier
- How to deal with these dependencies? Bayesian Belief Networks

kNN Algorithm

kNN Classifier

- k-nearest neighbours (kNN) algorithm as a type of supervised algorithm which can be used for both classification as well as regression predictive problems.
- There are three categories of learning algorithms:

  i) **Lazy learning algorithm:** kNN is a lazy learning algorithm because it does not have a specialized training phase or model and uses all the data for training while classification

  ii) **Non-parametric learning algorithm:** kNN is also a non-parametric learning algorithm because it does not assume about the distribution of the underlying data (as opposed to other algorithms such as Gaussian Mixture Model (GMM), which assume a Gaussian distribution about the data

  iii) **Eager learning algorithm:** Eager learners, when given a set of training tuples, will construct a generalization model before receiving new (e.g., test) tuples to classify.

kNN Classifier

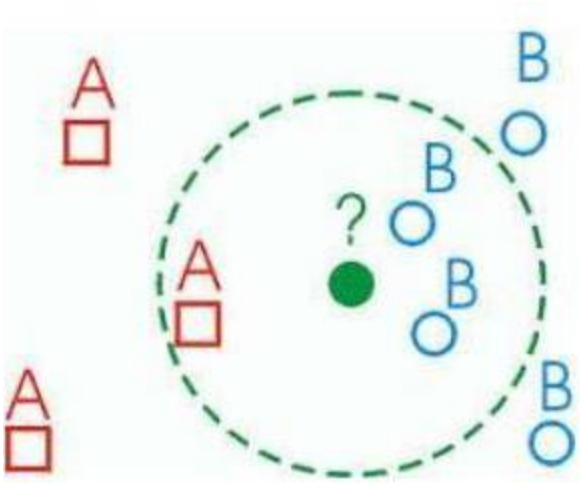
- The kNN algorithm begins with a training dataset made up of examples that are classified into several categories.
- Assume that we have a test dataset containing unlabeled examples that otherwise have the same features as the training data.
- For each example (i.e., record) in the test dataset, kNN identifies k examples in the training data that are the "nearest" in similarity, where k is an integer specified in advance.
- The unlabeled test instance is assigned the class of the majority of the k nearest neighbors

KNN: Classification Approach

- Locating the unlabeled instance's nearest neighbors requires a distance function, or a formula that measures the similarity between two instances.
- There are many different ways to calculate distance. Traditionally, the kNN algorithm uses Euclidean distance.
- The K-NN algorithm works by finding the K nearest neighbors to a given data point based on a distance metric, such as Euclidean distance.
- The class or value of the data point is then determined by the majority vote (for classification) or average (for regression) of the K neighbors.

KNN: Classification Approach

- Classified by "**MAJORITY VOTES**" for its neighbor classes
  - Assigned to the most common class amongst its *K*-nearest neighbors



KNN: Pseudocode

- Step 1: Determine parameter K = number of nearest neighbors
- Step 2: Calculate the distance between the query-instance and all the training examples.
- Step 3: Sort the distance and determine nearest neighbors based on the k-th minimum distance.
- Step 4:Gather the category Y of the nearest neighbors.
- Step 5: Use simple majority of the category of nearest neighbors as the prediction value of the query instance.

Advantages of kNN

- kNN algorithm is a versatile and widely used machine learning algorithm that is primarily used for its simplicity and ease of implementation.
- It does not require any assumptions about the underlying data distribution.
- It can also handle both numerical and categorical data, making it a flexible choice for various types of datasets in classification and regression tasks.
- It is a non-parametric method that makes predictions based on the similarity of data points in a given dataset.

Advantages of kNN

- K-NN is less sensitive to outliers compared to other algorithms.

- **Few Hyperparameters** – The only parameters which are required in the training of a KNN algorithm are the value of k and the choice of the distance metric which we would like to choose from our evaluation metric.

Disadvantages of the KNN Algorithm

- **Does not scale** – It is a lazy Algorithm. The main significance of this term is that this takes lots of computing power as well as data storage. This makes this algorithm both time-consuming and resource exhausting.
- **Curse of Dimensionality** – The KNN algorithm is affected by the curse of dimensionality which implies the algorithm faces a hard time classifying the data points properly when the dimensionality is too high.
- The curse of dimensionality can be particularly problematic in several ways:

(i) Distance Metrics Become Less Informative: In high-dimensional spaces, the concept of "distance" becomes less meaningful because the distances between all pairs of points tend to become more similar. This reduces the effectiveness of KNN, which relies on distance metrics to determine the nearest neighbors.

(ii) **Sparsity of Data**: As the number of dimensions increases, the volume of the space grows exponentially. This means that data points become sparse, making it harder for KNN to find enough neighbors that are truly representative of the underlying distribution of the data.

Disadvantages of the KNN Algorithm

(iii) Increased Computational Complexity: The computational cost of calculating distances between points increases with dimensionality. This can make KNN inefficient in practice when dealing with very high-dimensional data.

(iv) Overfitting: In high-dimensional spaces, KNN can become overly sensitive to noise in the data, leading to overfitting. This happens because with many dimensions, there's a higher likelihood that some of the features will not be relevant, but they can still influence the nearest neighbor calculations.

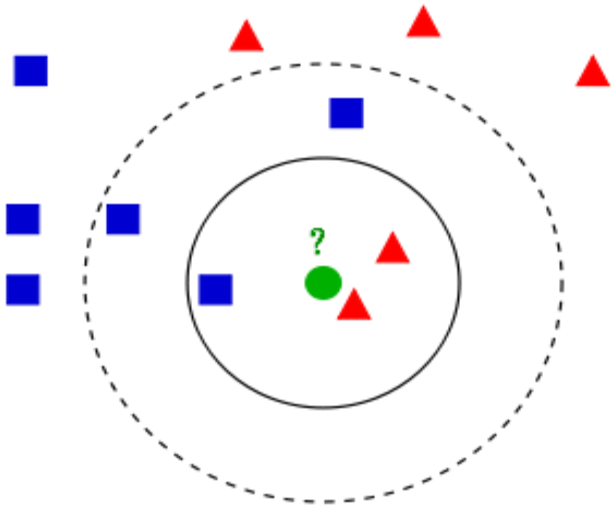
- To mitigate these issues, various techniques can be used:

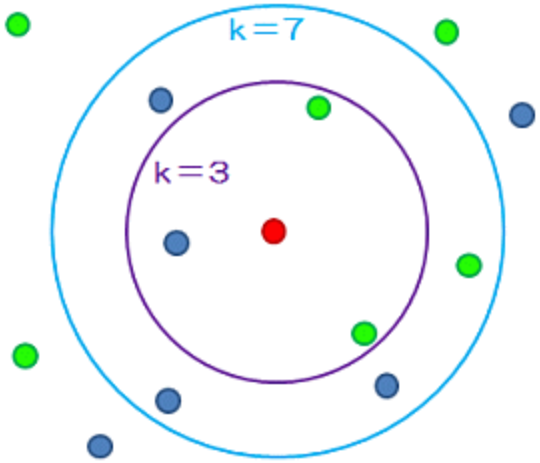
  (i) Dimensionality Reduction

  (ii) Feature Selection

- By addressing the curse of dimensionality through these methods, you can make KNN more effective even in high-dimensional settings.

Variation In kNN





How to Choose the value of k?

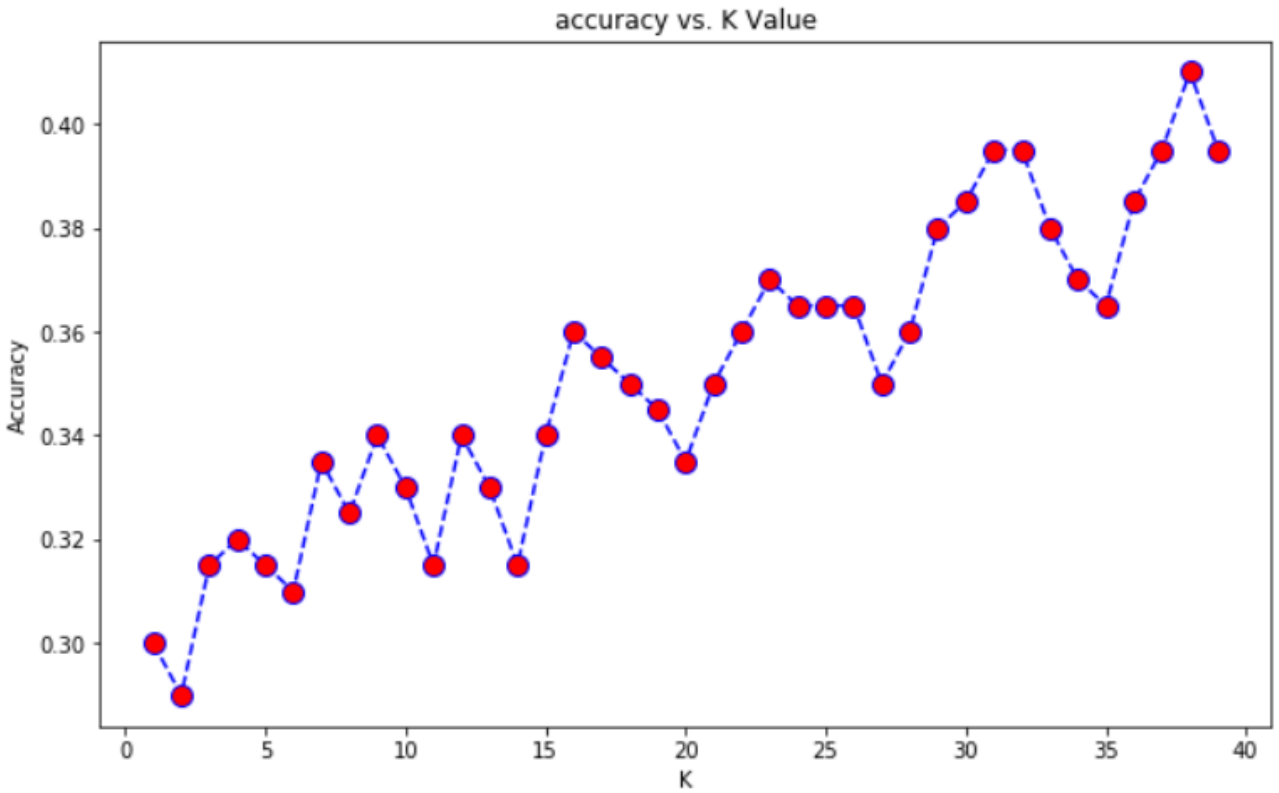
- The value of k is very crucial in the KNN algorithm to define the number of neighbors in the algorithm.
- The value of k in the k-nearest neighbors (k-NN) algorithm should be chosen based on the input data. If the input data has more outliers or noise, a higher value of k would be better.
- It is recommended to choose an odd value for k to avoid ties in classification.

Value of k

- The **small k value isn't suitable** for classification.
- As a rule of thumb, setting k to the square root of the number of training samples can lead to better result. If k becomes 10 after the square root, we can choose either k=9 or k=11 just to make sure that k is odd.
- Use an error plot or accuracy plot to find the most favorable k value.
- kNN performs well with multi-label classes, but you must be aware of the outliers.

How to Choose the value of k?

```
Maximum accuracy:- 0.41 at K = 37
```



• We got the accuracy of **0.41 at K=37**. As we got the minimum error at k=37, so we will get **better efficiency** at that K value.

Is Naïve Bayes a lazy learner?

- The Naive Bayes algorithm is not a *lazy* learner. It is an eager learner. It is different from the nearest neighbor algorithm.
- A real learning takes place for Naive Bayes. The parameters that are learned in Naive Bayes are the *prior probabilities* of different classes, as well as the *likelihood* of different features for each class.
- In the test phase, these learned parameters are used to estimate the probability of each class for the given sample.
- In other words, in Naive Bayes, for each sample in the test set, the parameters determined during training are used to estimate the probability of that sample belonging to different classes.

Is Naïve Bayes a lazy learner?

- For example, $P(c|x) \propto P(c)\ P(x_1|c)\ P(x_2|c)\ \ldots\ p(x_n|c)$, where c is a class and x is a test sample.
- All quantities P(c) and $P(x_i|c)$ are parameters which are determined during training and are used during testing.
- This is similar to NN, but the kind of learning and the kind of applying the learned model is different.

THANK YOU