Priyank Lohariwal
00171050105 5

Pattern Recognition
BCSE FINAL YEAR, 1st Sem Examinations
Date: 24-01-2021

1. a) False - A Naïve Bayes classifier always assumes that each feature x_i is conditionally independent of every other feature x_j for $j \neq i$. Hence, the term naive is used.

~~b) False~~

b) ~~False~~ True - Pattern classification's (using decision functions) success depends on two factors :-
i) form of decision functions
ii) ability to determine coefficients of the function.
~~One~~ i) is directly related to the geometrical patterns of the pattern classes under consideration. Once, a form is selected, ~~the~~, we still need to determine coefficients of the function.

c) False - In a typical Syntactic Pattern classification system the patterns are classified using the set of input pattern primitives and the grammar that define the structure of those primitives. Hence, the training phase is not important in general.

d) False - The points which are present ~~present~~ closer to ~~within~~ ~~or~~ ~~the margin~~ of the hyperplane (decision boundaries that help classify the data points) can only be the support vectors. They influence the position and orientation of the hyperplane.

e) True :- The density based clustering algorithms ~~find~~ ~~uses the~~ partitions the ~~a~~ ~~closely dens~~ highly dense regions with that of low dense regions. The high density regions are further clustered and the low dense regions are ~~outlast~~ regarded as outliers or noise.

f) True :- Since, the NB classifier assumes class conditional independence, in real life scenarios, there is seldom a case when this actually happens. Hence, its not valid for most real life scenarios.

For ~~ex~~ :- ~~for~~ classifying a ~~message~~ mail as 'spam' or 'not spam'. In a normal message, we can have 'dear friend' and in a spam message, we can have 'friend dear'. But the NB classifier will classify both of them as equal.

For ex :- The probability of a grass being wet may depend on the probability of raining & ~~or~~ the probability of lawn showers being on. But the NB classifier disregards this dependence which may lead to unexpected results.

g) False :- Syntactic Pattern Recognition attempts to classify patterns based on some primitive patterns and a grammar that defines the structure of these primitives in ~~an output~~ a pattern class.

h) False :- Hierarchical clustering methods help in exploring data at different levels of granularity, not partitional methods.

i) False :- A Hopfield net is mainly used for optimization.

j) False :- In case of patterns being pairwise-separable, the pattern classification scheme having M classes needs to compute $\frac{M(M-1)}{2}$ decision surfaces to perform classification.

k) False :- Data preprocessing is required to ensure accuracy of the data along with completeness, consistency, timeliness, believability and interpretability.

---

2. No. of pattern classes = M

Assuming prototype patterns of these classes be represented as $z_1, z_2, z_3, \ldots, z_M$, let us derive the decision function to be used for classification purposes.

Let D_i be the euclidean distance between an arbitrary pattern vector X and the i^{th} prototype. Then,

$$D_i = \|X - z_i\| = \sqrt{(X - z_i)'(X - z_i)} \quad \text{---(1)}$$

where, $X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$ (A column matrix).

A minimum distance classifier computes the distance from a pattern X of unknown classification to the prototype of each class, and assigns the pattern to the class to which it is closest.

We can also say, X is assigned to class w_i if $D_i < D_j \ \forall \ i \neq j$. Ties are resolved arbitrarily.

$$① \Rightarrow D_i^2 = \|x - z_i\|^2 = (x - z_i)'(x - z_i)$$

$$= x'x - 2x'z_i + z_i'z_i$$

$$= x'x - 2\left(x'z_i - \frac{1}{2} z_i'z_i\right)$$

~~①~~ choosing the minimum D_i^2 is equivalent to ~~choose~~ choosing the minimum D_i as all the distances are positive.

Also, $x'x$ is independent of i.

Hence, choosing minimum D_i^2 is equivalent to choosing the minimum of $-2\left(x'z_i - \frac{1}{2} z_i'z_i\right)$ which is equivalent to choosing the maximum of $\left(x'z_i - \frac{1}{2} z_i'z_i\right)$

Thus, we can define our decision function as,

$$d_i(x) = x'z_i - \frac{1}{2} z_i'z_i \qquad \forall i = 1, \ldots, M \qquad \text{—— (11)}$$

x is assigned to class w_i, if $d_i(x) > d_j(x)\ \forall\ i \neq j$

This is our required decision function.

$d_i(x)$ is a linear decision function

we have, ~~let~~ $z_i = \begin{bmatrix} z_{i1} \\ z_{i2} \\ \vdots \\ z_{in} \end{bmatrix}$

Let ~~w~~ w_i be another vector for a given class i, where

$$w_{ij} = z_{ij} \quad [j = 1, 2, \ldots, n]$$

and $w_{i(n+1)} = -\frac{1}{2} z_i'z_i$

Then, we can write (II) as

$$d_i(x) = w_i' x_i \quad , i = 1, 2, \ldots, M \qquad \text{— (III)}$$

where, $w_i = \begin{bmatrix} w_{i1} \\ w_{i2} \\ \vdots \\ w_{i,n+1} \end{bmatrix}$ & $x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ 1 \end{bmatrix}$

Clearly, expression (III) is a linear function.

---

3. **Bayes Theorem :-**

Assumptions :-
- Let X be a data sample
- Let H be a hypothesis that X ∈ C (where is a class)

Thus, the probability that the hypothesis H holds given the observed data sample X can be derived using Bayes theorem as follows :-

$$P(H/X) = \frac{P(X/H) \cdot P(H)}{P(X)}$$

where,
- P(H) is the initial probability of the hypothesis
- P(X) is the evidence i.e, the probability of the sample data being observed.
- P(X/H) is the likelihood, the probability of observing the sample X, given the hypothesis H holds.

Naïve Bayes classifier :- Analytical Expression

Assumptions :

- D : Set of data, $D \in R^n$
- X : A tuple in D represented by $(x_1, x_2, \ldots, x_n)$
- $C_1, C_2, \ldots, C_k$: Output k classes

In this classifier, we find the probability of an input data X given a class C_i for all $i = 1, \ldots, k$.

i.e, we find,

$$P(C_i/X) \quad \forall\ i = 1, 2, \ldots, k$$

Using Bayes' theorem,

$$P(C_i/X) = \frac{P(C_i) * P(X/C_i)}{P(X)}$$

Here, P(X) does not depend on i or C_i and all the values of the features (x_i's) are constant across all the probabilities ($P(C_i/X), i = 1, \ldots, k$). So, we eliminate that part.

$$P(C_i/X) \propto P(C_i) * P(X/C_i)$$

$$= P(C_i) * P(x_1, x_2, \ldots, x_n / C_i)$$

where $X = (x_1, x_2, \ldots x_n)$ —(1)

Priyank Lohariwal 0017105010055

Using the chain rule for repeated applications of the definition of conditional probability.

~~(1) $\propto$ $P(C_i)$~~

~~$p(x_1, x_2, \ldots, x_n / C_i) \propto$~~

~~$= p(x_1/x)$~~

$p(x_1, x_2, \ldots, x_n / C_i)$

$= p(x_1 / x_2, x_3, \ldots, x_n, C_i) \times p(x_2 / x_3, x_4 \ldots x_n, C_i) \times p(x_3 / x_4, x_5, \ldots x_n, C_i)$

$\times \ldots \times p(x_{n-1} / x_n, C_i) \times p(x_n / C_i)$ —(ii)

In Naive Bayes classifier, we assume that each feature of the feature vector x is conditionally independent from every other features i.e,

x_i is independent of x_j $\forall$ i, j, $i \neq j$, given C_i.

$\Rightarrow p(x_i / x_{i+1}, \ldots, x_n, C_k) = p(x_i / C_k)$ —(iii)

Putting the value of (ii) in (i),

$p(C_i / x) \propto p(x_1 / x_2, x_3, \ldots, x_n, C_i)\, p(x_2 / x_3, x_4 \ldots, x_n, C_i) \ldots$

$\ldots p(x_{n-1} / x_n, C_i)\, p(x_n / C_i)\, p(C_i)$

Using (iii),

$\Rightarrow p(C_i / x) \propto p(x_1 / C_i)\, p(x_2 / C_i) \ldots p(x_n / C_i)\, p(C_i)$

$= p(C_i) \prod_{j=1}^{n} p(x_j / C_i)$

This model is combined with a decision rule.

- The NB classifier picks the hypothesis that is most probable ($P(C_i/x) > P(C_j/x)\ \forall j, j \neq i$). This is known as the maximum a posteriori or MAP decision rule.

- Thus, the classifier is a function that assigns a class label $\hat{y} = C_k$ for some k as follows:-

$$\hat{y} = \underset{k \in \{1,\dots k\}}{\operatorname{argmax}} P(C_k) \prod_{i=1}^{n} P(x_i/C_k)$$

Calculation of $P(x_i/C_k)$ based on the type of input feature

Assumption:-
Let A_i be the i^{th} feature of a given data sample X.

Case 1:- A_i is categorical

$$P(x_i/C_k) = \frac{\text{no. of tuples in } C_k \text{ having } x_i \text{ in } A_i}{\text{no. of tuples of } C_k \text{ in } D\ (|C_{k,D}|)}$$

where, D is the total data set.

Case 2:- A_i is continous

Generally, gaussian distribution is used to calculate $P(x_i/C_k)$

$$P(x_i/C_k) = g(x_i, \mu_{C_k}, \sigma_{C_k})$$

where, μ_{C_k} = mean.

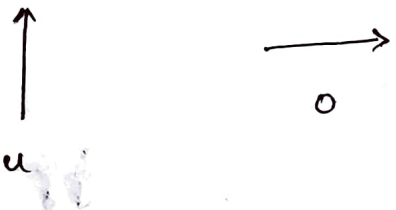
σ_{C_k} = standard deviation

$$g(x_i, \mu_{C_k}, \sigma_{C_k}) = \frac{1}{\sqrt{2\pi}\sigma_{C_k}^2} e^{\frac{-(x_i - \mu_{C_k})^2}{2\sigma_{C_k}^2}}$$

5.

Pattern Data

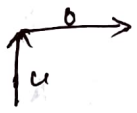
Primitives required :-



Operations required :-

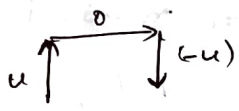
→ + :- concatenation. (head to tail)

Ex :- (u + o)



→ − : head-tail reversal Ex :-  (−u)

→ * :- head-head and tail-tail attachment

Ex :-  (u + o + (−u))

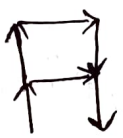
 (o)

(u + o + (−u)) * o

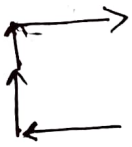
→ X :- ~~head-to~~ tail-tail attachment only.

Ex :-  (u × o)

Representing pattern data using primitives,



A = u + ((u + o + −u) * o) + −u



C = −o + u + u + o

Priyank Lohariwal 00171050105S

P = u + ((u + 0 + -u) * 0)

F = u + (0 x u) + 0

~~In order to~~ Grammar

Let us define the following :-

i) $V_T = \{ u, 0, +, -, *, x, |, (,) \}$

where, V_T is the set of terminal symbols.
(| => or operator), (): opening & closing parantheses)

ii) $V_N = \{ S, A, C, P, F \}$

where, V_N is the set of non-terminal symbols.

~~iii) Production rules, P~~

iii) Start symbol, $S \in V_N$

iv) P (Production rules)

$= \{ S \rightarrow A | C | P | F$

$A \rightarrow u + ((u + 0 + (-u)) * 0) + (-u)$

$C \rightarrow (-0) + u + u + 0$

$P \rightarrow u + ((u + 0 + (-u)) * 0)$

$F \rightarrow u + (0 \times u) + 0 \}$

Hence, we have, $G = \{V_T, V_N, S, P\}$ as the grammar to ~~solve~~ recognise the given patterns using syntactic pattern recognition model.

6. Pattern Recognition.

Use:- The act of Pattern recognition can be divided into ~~two~~ two broad categories:-

i) Recognizing concrete items.

Ex:- Pictures, signatures, waveforms etc.

ii) Recognizing abstract items.

ex:- A conversation, etc.

There are various conventional methods for recognition purposes. Some of them are:- feature extraction, classification, clustering etc.

All of these approaches are based on direct computation through Machines which are math-related techniques.

We can also use application of ~~biologocia~~ biological concepts to electronic machines. This concept ~~to~~ lead to the development of neural networks.

Artificial Neural networks (ANN)

An artificial Neural network is a paralleled distributed information processing structure in the form of a directed graph.

Basically, it consists of massive simple processing units (perceptrons) with a high degree of interconnection between each layer of units. The processing units work cooperatively with each other and achieve massive parallel ~~The~~ distributed processing. The design and function of neural networks & simulate some functionality of biological brains and neural systems.

Priyank Loharivwal 00171050105

Pattern recognition can be done using both conventional computers and neural networks.

A comparison can be done between the two as follows:-

~~However, & neural networks are above~~

→ Neural networks use many simple processors as opposed to general computers ~~who~~ which use few complex processors.

→ Neural nets use fewer processing steps.

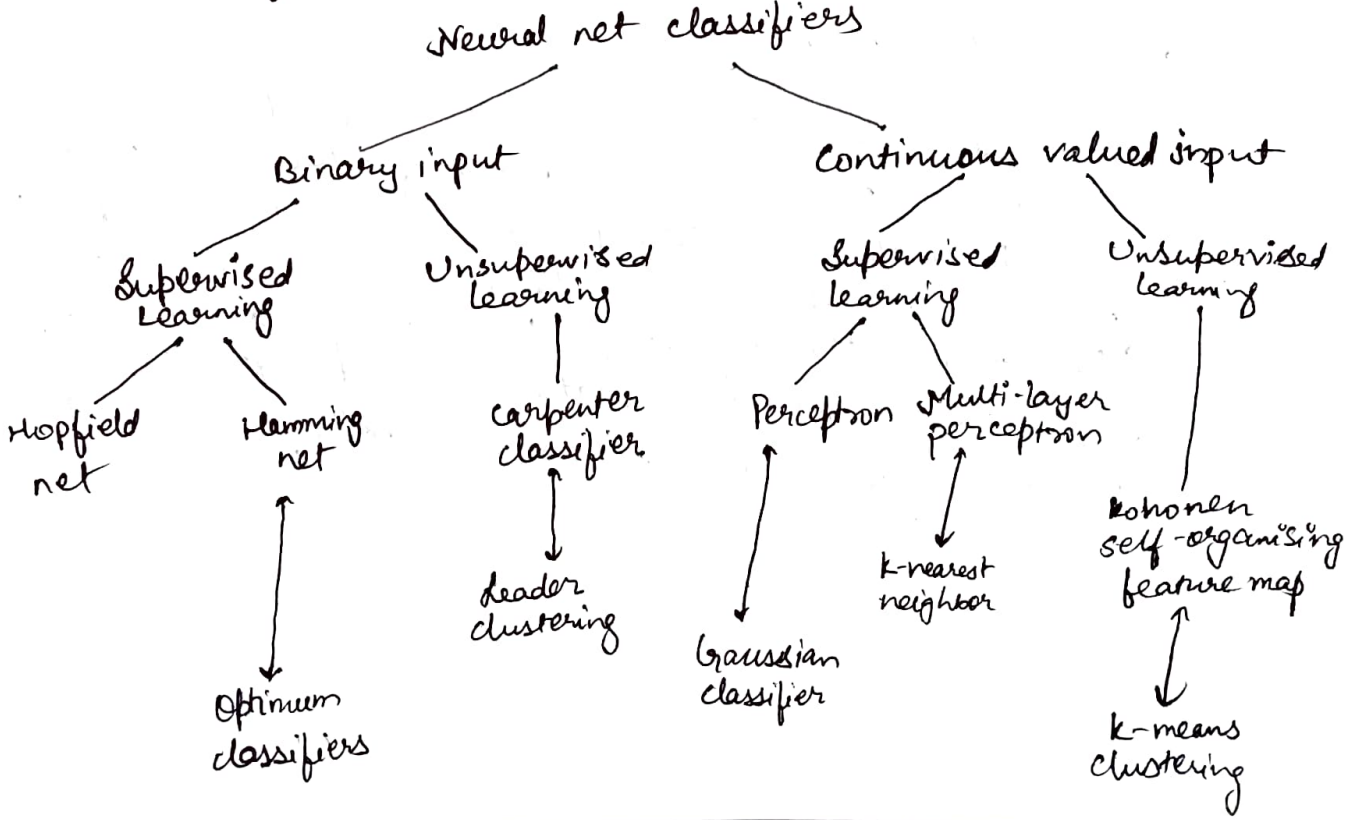
→ Neural nets use the concept of distributed processing making them faster

→ Neural nets are trained by example helping them to achieve far better results even for unknown data.

→ Neural nets are tolerable to noisy patterns.

Due to the ~~outstanding~~ adaptive-learning, self organisation and fault tolerance capabilities of neural nets, ANNs are used for various pattern recognition applications.

A simple taxonomy of six neural nets that can be used as classifiers is shown below:-



We can see the importance of ANN's in Pattern Recognition by looking at the diversity of applications that ANN's have in Pattern Recognition problems.

Algorithm	Type	Usage
Hopfield	recursive	optimization
Multi-layered perceptron	feedforward	classification
Kohonen	Self-organising	data coding
Temporal differences	predictive	forecasting

To conclude, neural networks have the following advantages solidifying their importance in the field of Pattern Recognition :-

- Can work with incomplete data once trained
- Fault tolerance (Robust to noise)
- Distributed & parallel processing
- Can learn non-linear and complex relationships also.
- Generalizability (process unknown relationships also after appropriate learning phase)
- Trained by example
- Adaptive learning.

Date: 24-01-2021