There are two types of learners in classification as lazy learner and eager learner

(i) Lazy learner :- Lazy learners simply store the Training data and wait until a testing data appear. When it does, classification is conducted based on the most related data in the stored training data.

Compared to eager learners, lazy learners have less training time but more time in predicting.

Ex:- K-nearest neighbour, case-based reasoning

(ii) Eager learners :-

Eager learners construct a classification model based on the given Training data before receiving data for classification. It must be able to commit to a single hypothesis that covers the entire state space. Due to model construction, eager learners take a long time for Train and less time to Predict.

Ex:- Decision Tree, Naive Baysian, Artificial Neural N/w.

K-nearest Neighbour Classifier :→

Nearest-neighbour classifiers are based on learning by analogy i.e. by comparing a given test tuple with training tuples that are similar to it. The Training tuple are described by n attributes. Each tuple represents a Point in an n-dimensional space. In This way, all of The Training tuples are stored in an n-dimensional Pattern space. When given an unknown tuple, a K-nearest-neighbour classifier searches The Pattern space for The K-Training tuples that are closest to the unknown tuple. These K-Training tuples are the K "nearest neighbours" of the unknown tuples

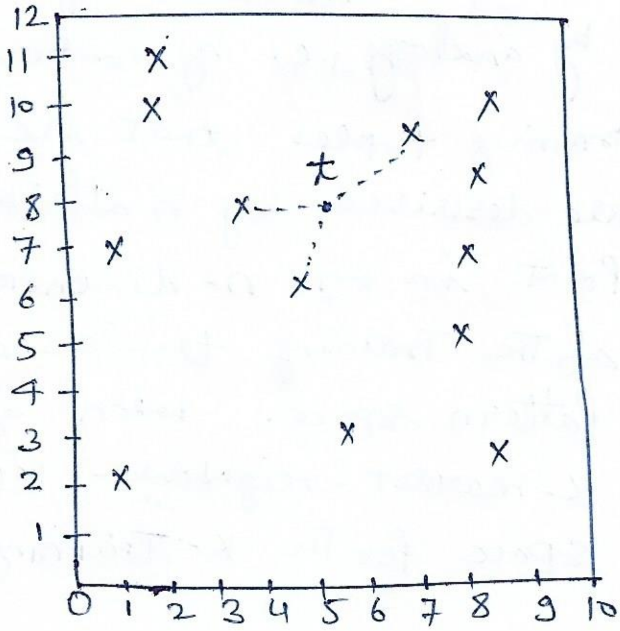
"Closeness" is defined in terms of a distance metric such as Euclidean distance. The euclidean distance b/w two points or tuples, say

$X_1 = (x_{11}, x_{12} --- x_{1n})$ and
$X_2 = (x_{21}, x_{22} --- x_{2n})$, is

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2} \quad -(1)$$

Here for each numeric attribute, we take the difference b/w The corresponding values of that attribute in tuple X_1 and in tuple X_2, square this difference and accumulate it. The square root is taken of the total accumulated distance count.

Figure illustrate the process used by KNN. Here the Points in The Training set are shown and k=3. The Three closest items in The Training set are shown; t will be placed in The class to which most of These are members.



For K-nearest neighbour classification, The unknown tuple is assigned the most common class among its K-nearest neighbours, when K=1, The unknown tuple is assigned to the class of The Training tuple that is closest to it in Pattern space.

Nearest neighbour classifiers can also be used for Prediction, i.e. to return a real-valued prediction for a given unknown tuple. In This case, The classifier returns The average value of The real-valued labels associated with The K nearest neighbours of The unknown tuple.

<u>K-nearest neighbour classifier</u>:→ In Pattern Recognition, The K-nearest neighbors algorithm (or K-NN for short) is a non-Parametric Method used for classification and Regression. In both the cases, The input consists of the K-closest Training examples in The feature space. The output depends on whether K-nn is used for classification or Regression.

In K-nn Classification, The output is a a class membership. An object is classified by a Majority vote of its neighbours, with the object being assigned to the class Most Common among its K-nearest neighbours (K is a Positive integer, typically small). If K=1, then The object is simply assigned to the class of that single nearest neighbour.

In K-nn Regression, The output is The Property value of The object. This value is The average of the values of its K-nearest neighbours.

Knn is a type of instance based learning or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The K-nn algorithm is among the simplest of all machine learning algorithms.

Both for classification and regression, it can be useful to assign weight to the contributions of the neighbours. So that the nearest neighbours contribute more to the Avg than the More distance ones. A common ~~distance~~ weighing scheme consists in giving each neighbour a weight of 1/d, where d is the distance to the neighbour.

The neighbour are taken from a set of objects for which the class (for Knn classification) or the object property value (for k-nn regression) is known. this can be thought of as the Traing set for the algorithm, Though no explicit Training step is required.

<u>why K-nn is non Parametric?</u> → non Parameteric Means not making any assumptions on the underlying data ~~structure~~ distribution. Non Parametric Methods do not have fixed numbers of Parameters in the model.

Similarly in Knn, model Parameters actually grows with the Training data set. you can imagine each Traing case as a "parameter" in the model.

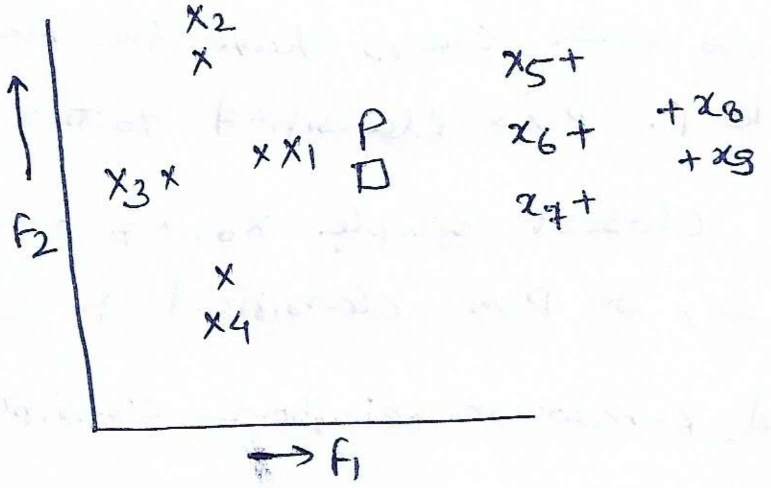
Pros of KNN:-

1. easy to understand
2. No assumption about data
3. Can be applied to both classification and regression
4. works easily on Multi-class problem

Cons of KNN:-

1. Memory intensive / computationally expensive
2. sensitive to scale of data
3. Not work well on rare event (skewed)
4. Struggle when high no. of Independent variables.

Question:- Consider the two class problem shown in figure,



There are four patterns in class 1 marked as 'x' and five patterns in class 2 marked as +. The Test Pattern is P. The closest point to P are X_1, X_6, X_7, X_2 and X_5. If The distance from P to X_1, X_6, X_7, X_2 and X_5 are 0.3, 1.0, 1.1, 1.5 and 1.6. classify the test pattern P using following

1. Nearest neighbour Algorithm
2. K-nearest neighbour (KNN) algorithm with $k=5$
3. Modified K-nearest neighbour (MKNN) classifier.

Sol^n:- (1) Since X_1 is nearest to the pattern as $D(P, X_1) = 0.3$, which is minimum and X_1 belongs to Class 1 so P is classified in class 1.

(2) K-nearest neighbour with $k=5$; in this method; K numbers of samples nearest to P are calculated

$$D(P, X_1) = 0.3 \rightarrow 1$$
$$D(P, X6) = 1.0 \rightarrow 2$$
$$D(P, X7) = 1.1 \rightarrow 2$$
$$D(P, X2) = 1.5 \rightarrow 1$$
$$D(P, X5) = 1.6 \rightarrow 2$$

which so ever class have the maximum closest samples to P. P is classified to that class.

maximum closest sample $X6$, $X7$ and $X5$ belongs to class 2; so P is classified in class 2.

(3) Modified K-nearest neighbour classifier:

$$w_j = \begin{cases} \dfrac{d_k - d_j}{d_k - d_i} & \text{if } d_k \neq d_i \\ 1 & \text{if } d_k = d_i \end{cases}$$

$w_1 = 1$ $\quad i=1, j=1$

$$w_6 = \frac{1.6-1}{1.6-0.3} = \frac{0.6}{1.3}$$

X_1, X_6; $i=1, j=6, k=5$ (asked)

$d_i = 0.3$

$d_6 = 1$

for class 1 $\quad 1 + \frac{0.1}{1.3} = 1.077$

$w_7 \Rightarrow \frac{1.6 - 1.1}{1.6 - 0.3} \Rightarrow \frac{0.5}{1.3} = 0.384$

$i = 1,\ j = 7$ $\quad$ for class 2 $\quad \frac{0.6 + 0.5 + 0}{1.3}$

$\Rightarrow \frac{1.1}{1.3} \Rightarrow 0.846$

$w_2 = \frac{1.6 - 1.5}{1.6 - 0.3} = \frac{0.1}{1.3} \Rightarrow i = 1,\ j = 1.5$

$\Rightarrow 0.0769$

$w_5 = \frac{1.6 - 1.6}{1.6 - 0.3}$; where $j = 5$, $w_5 = 0$

Therefore P belongs to class 1

Q. Consider the two class problem class 1 and class 2, test Pattern is P. Five nearest neighbours Patterns to test Pattern P is x_1, x_2, x_3, x_4, and x_5. x_1 and x_4 belong to class 1 and x_2, x_3 and x_5 belong to class 2. Distance from P, $d(x_1, P) = 1$, $d(x_2, P) = 2$, $d(x_3, P) = 2.5$, $d(x_4, P) = 4$ and $d(x_5, P) = 5$.

Classify the test Pattern P using modified K-nearest neighbour algorithm

Sol^n:-

$$w_j = \begin{cases} \dfrac{d_k - d_j}{d_k - d_i} & \text{if } d_k \neq d_i \\ 1 & \text{if } d_k = d_i \end{cases}$$

$$w_1 = 1 \Rightarrow \frac{5-1}{5-1} \Rightarrow 1$$

$$w_2 = \frac{5-2}{5-1} = \frac{3}{4} \Rightarrow 0.75$$

$$w_3 = \frac{5-2.5}{5-1} = \frac{2.5}{4} \Rightarrow 0.625$$

$$w_4 = \frac{5-4}{5-1} = \frac{1}{4} = 0.25$$

$$w_5 = 0$$

for class 1 $\Rightarrow$ $1 + 0.025 = 1.25$

for class 2 $\Rightarrow$ $0.75 + 0.625 + 0 \Rightarrow 1.375$

Therefore, P belongs to class 2

Nearest Neighbour based classifiers:-

One of the simplest decision procedure that can be used for classification is the nearest neighbour (NN) rule. It classify a sample based on the category of its nearest neighbour. The nearest neighbour based classifiers use some or all the patterns available in the training set to classify a test pattern. These classifiers essentially involve finding the similarity b/w the test pattern and every pattern in the training set.

Nearest Neighbour Algorithm:- The nearest neighbour algo. assigns to a test pattern to the class label of its closest neighbour. Let there be n train tuples/patterns, $(x_1, \theta_1), (x_2, \theta_2)$ ---- (x_n, θ_n); where x_i is of dimension d and θ_i is the class label of the i^{th} pattern. If P is the test pattern, then if

$$d(P, x_k) = \min\{d(P, x_i)\}$$

where $i = 1, 2, --- n$. Pattern P is assigned to the class θ_k associated with x_k.

for Ex:- Let the training set consist of the following three dimensional patterns

$X_1 = (0.8, 0.8, 1)$; $x_2 = (1, 1, 1)$, $x_3 = (1.2, 0.8, 1)$

$X_4 = (0.8, 1.2, 1)$; $X_5 = (1.2, 1.2, 1)$, $X_6 = (4.0, 3.0, 2)$;

$X_7 = (3.8, 2.8, 2)$; $X_8 = (4.2, 2.8, 2)$; $X_9 = (3.8, 3.2, 2)$

$X_{10} = (4.2, 3.2, 2)$; $X_{11} = (4.4, 2.8, 2)$; $X_{12} = (4.4, 3.2, 2)$

$X_{13} = (3.2, 0.4, 3)$; $X_{14} = (3.2, 0.7, 3)$; $X_{15} = (3.8, 0.5, 3)$

$X_{16} = (3.5, 1.0, 3)$; $X_{17} = (4.0, 1.0, 3)$; $X_{18} = (4.0, 0.7, 3)$

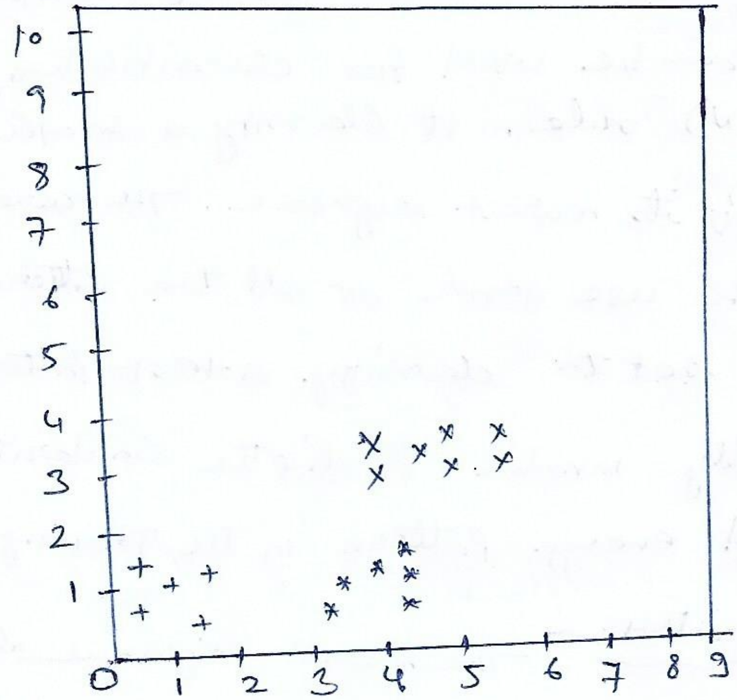


fig:- Example data set

for each pattern, the first two numbers in the triplets gives the first and second features, and third number gives the class label of the pattern.

This can be seen plotted in fig. Here + corresponds to class 1, "x" corresponds to class 2 and * corresponds to class 3.

Now, if there is a test pattern $P = (3.0, 2.0)$, it is necessary to find the distance from P to all training patterns.

let us distance b/w x and P be the euclidean distance $d(x, P) = \sqrt{(x(1)-P(1))^2 + (x(2)-P(2))^2}$

The distance from a point P to every point in the set can be computed using the above formula

for $P(3.0, 2.0)$, The distance to X_1 is

$$d(X_1, P) = \sqrt{(0.8-3.0)^2 + (0.8-2.0)^2}$$

$$= \sqrt{(-2.2)^2 + (-1.2)^2} = 2.51$$

$$x_{16} = \sqrt{(3.5-3)^2+(1-2)^2}$$

$$\Rightarrow \sqrt{0.25+1}$$

$$\Rightarrow \sqrt{1.25} \Rightarrow 1.12$$

$$x_{17} = \sqrt{(4-3)^2+(1-2)^2}$$

$$\Rightarrow \sqrt{1+1} \Rightarrow \sqrt{2} \Rightarrow 1.41$$

We find, after calculating the distance from all the training points to P, i.e closest neighbour of P is X_{16}, which has a distance of 1.12 from P and belongs to class 3.

Hence P is classified as belonging to class 3.

Variants of The NN Algorithm :→

K-nearest neighbours (KNN) algorithm :- In This algo, Instead of finding Just one nearest neighbour as in The NN Algo, K-neighbours are found. The Majority class of These K-nearest neighbours is the class label assigned to the new Pattern. The value chosen for K is crucial. With The right value of K, The classification accuracy will be better than that got by using the nearest neighbour algorithm.

EX:- 2: - In This example shown in fig; if K is taken to be 5, the five nearest neighbours of P are X_{16}, X_7, X_{14}, X_6 and X_{17}. The Majority class of these five Patterns is Class 3 (X_{16}, X_{14}, X_{17})

Example 3:— If P is the Pattern $(4.2, 1.8)$, its nearest neighbour is X_{17} and it would be classified as belonging to class 3, if the nearest neighbour algorithm is used, if the 5 nearest neighbours are taken. It can be seen that they are X_{17} and X_{16}, both belonging to class 3 and X_8, X_7 and X_{11} belong to class 2. Following the Majority class rule, the Pattern would be classified as belonging to class 2.

Modified K-nearest neighbour (MKNN) algorithm :→

This Algo. is similar to the KNN algorithm; In as much as it takes the K-nearest neighbours into consideration. The only difference is that these K-nearest neighbours are weighted according to their distance from the test Point. It is also called the distance weighted K-nearest neighbour algorithm.

Each of the neighbours is associated with the weighted w which is defined as

$$w_j = \begin{cases} \dfrac{d_k - d_j}{d_k - d_1} & \text{if } d_k \neq d_1 \\ 1 & \text{if } d_k = d_1 \end{cases}$$

where $j = 1, 2 \ldots K$; The value of w_j varies from a maximum of 1 for the nearest neighbour down to a minimum of zero for the most distinct having computed the weights w_j the MKNN algorithm assigns the test Pattern P to that class for which the weights of the Representatives among the K-nearest neighbours sums to the greatest value.

Instead of using The simple majority rule, it can be observed that MkNN employs a weighted majority rule. This would mean that outlier patterns have lesser effect on classification.

Example:— Consider $P = (3.0, 2.0)$ in figure for The five nearest points, The distance from pair

$d(P, X16) = 1.12$ $i=1$

$d(P, X7) = 1.13$

$d(P, X14) = 1.32$

$d(P, X6) = 1.41$

$d(P, X17) = 1.41$

The value of w will be

$w16 = 1$ $\frac{1.41 - 1.12}{1.41 - 1.12} = 1$

$$w7 = \frac{(1.41 - 1.13)}{(1.41 - 1.12)} = 0.97$$

$$w17 = \frac{(1.41 - 1.32)}{(1.41 - 1.12)} = 0.31$$

$w6 = 0$

$w17 = 0$

Summing up for each class, class 1 sums to 0, class 2 which $x7$ and $x6$ belong sums to 0.97 and class 3 to which X16, X14 and X17 belong sums to 1.33. Therefore, The point P belongs to class 3.

It is possible that knn and MKNN algorithms assign the same pattern a different class label. This is illustrated in the following example:

Example:- In figure, when $P = (4.2, 1.8)$, the five nearest patterns are $X_{17}, X_8, X_{11}, X_{16}$ and X_7. The distance from P to these patterns are

$$\text{Class 2}\begin{cases} d(P, X_{17}) = 0.83 \\ d(P, X_8) = 1.0 \\ d(P, X_{11}) = 1.02 \end{cases} \qquad \begin{rcases} d(P, X_{16}) = 1.06 \\ d(P, X_7) = 1.08 \end{rcases}\text{class 3}$$

The value of w will be

$$w_{17} = 1$$

$$w_8 = \frac{(1.08 - 1.0)}{(1.08 - 0.83)} = 0.32$$

$$w_{11} = \frac{(1.08 - 1.02)}{(1.08 - 0.83)} = 0.24$$

$$w_{16} = \frac{(1.08 - 1.06)}{(1.08 - 0.83)} = 0.08$$

$$w_7 = 0$$

Summing up for each class, class 1 sums to 0, class 2 to which X_8, X_{11} and X_7 belong sums to 0.56 and class 3 to which X_{17} and X_{16} belong sums to 1.08 and therefore, P is classified as belonging to class 3.

Note that the same pattern is classified as belonging to class when we used to k-nearest neighbours algorithms with $K=5$: