K- nearest neighbour classifier

Prerequisite-

- Liner Algebra and cartesian plane
- Evaluation Metrics for classification accuracy.

Objectives(Lerner will able to understand and explain)

- What is KNN classifier
- Different types of distance measures
- How to choose K value
- Pros and cons of KNN classifier

KNN Algorithm-

KNN algorithm also known as K-Nearest Neighbors Algorithm is used to solve the both problems of classification as well as regression. This working principle of algorithm is mainly based on **feature similarity** in both classification and regression problem. KNN classifier is different from other probabilistic classifiers where the model comprises a learning step of computing probabilities from a training sample and use them for future prediction of a test sample. In probability based model once the model is trained the training sample could be thrown away and classification is done using the computed probabilities.

In KNN there is no learning step involved instead the dataset is stored in memory and is used to classify the test query on the fly. KNN is also known as Lazy learner as it does not create a model using training set in advance. It is one of the simplest methods of classification. In KNN, the term `k’ is a parameter which refers to the number of nearest neighbours. The classification procedure for a query point q works in two steps as:

1. Find the K neighbours in the dataset which are closet to q based on the similarity measure.
2. Use these K neighbours to determine the class of q using majority voting.

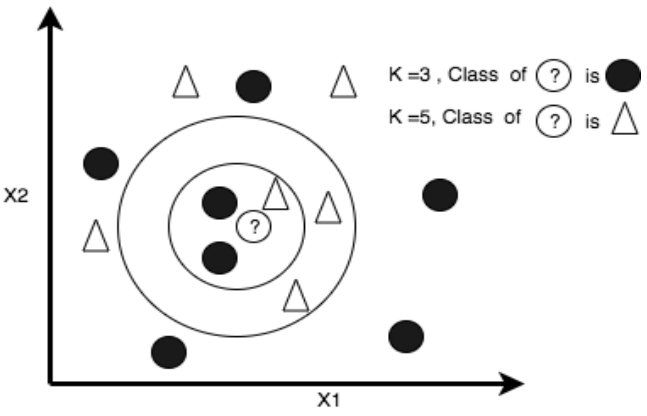


Figure: Illustration of KNN classifier

Distance Measure-

KNN classifier needs to compute similarities or distances of test query from each sample point in training dataset. Several methods are used to compute the distances and the choice completely depends of the types of features in the dataset. The popular distance measurements are as follows:

Euclidean Distance-

It is the most commonly used distance metrics and defined as the square root of the sum of squared differences between the two points. Let the two points are P(x_1, x_2) and Q (y_1, y_2) the Euclidean distance is given by:

$$PQ_{Euclidean} = \sqrt{(x_1 - y_1\,)^2 + \ (x_2 - y_2)^2}$$

In general

$$PQ_{Euclidean} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Manhattan Distance-

It is also known as city block distance or absolute distance. The distance measure is inspired with the structure of Manhattan city where the distance between two points is measured through city road grids. The distance is defined as the sum of absolute differences between two points coordinates.

$$PQ_{Manhattan} = |x_1 - y_1| + |x_2 - y_2|$$

or

$$PQ_{Manhattan} = \sum_{i=1}^{n}|x_i - y_i|$$

Chebyshev Distance-

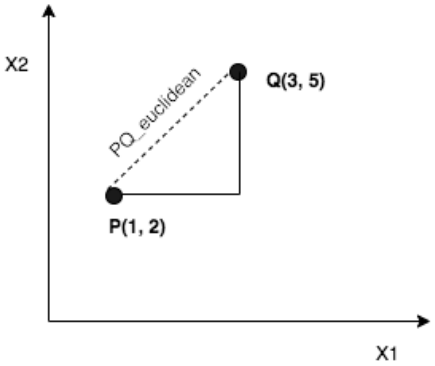
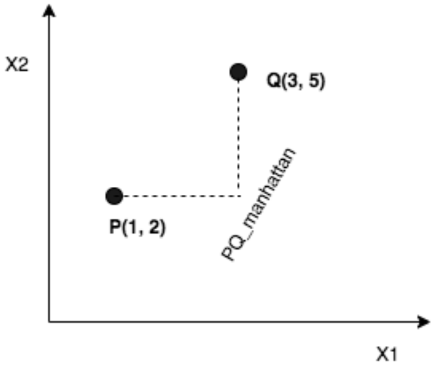
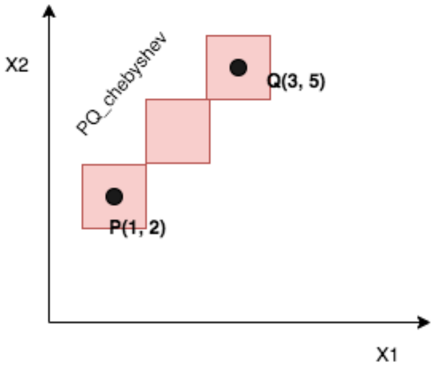
This distance is also known as Maximum value distance or chessboard distance. The distance is based on absolute magnitude between the coordinates of pair of two points. This distance is equally used with the quantitative and ordinal variable.

$$PQ_{Chebyshev} = Max(|x_1 - y_1|, |x_2 - y_2|)$$

or

$$PQ_{Chebyshev} = Max(\,|x_i - y_i|)$$

Example: Let P (1, 2) and Q (3, 5)

$PQ_{Euclidean} = \sqrt{(1-3\,)^2 + \,(2-5)^2} = \sqrt{13}$	
$PQ_{Manhattan} = \|1-3\| + \|2-5\| = 5$	
$PQ_{Chebyshev} = Max(\|1-3\|, \|2-5\|) = 3$	

Minkowski Distance-

Minkowski distance is one of the generalized distance measure, which means that by manipulating the formula different distances measures can be obtained. Above stated distance measures are the special case of Minkowski distance.

$$PQ_{Minkowski} = (\sum_{i}^{n}(x_i - y_i)^p)^{\frac{1}{p}}$$

When p =1 Minkowski has become Manhattan distance.

When p =2 Minkowski has become Euclidean distance.

When p = ∞ Minkowski has become Chebyshev distance.

Mahalanobis Distance-

This distance measure is used to calculate the distance between two points in multivariate space. The idea is to calculate the distance of a point P from any distribution D in terms of standard deviation and mean of distribution D. The main advantage of mahalanobis distance is that it includes the covariance of distribution to measure the similarity between two points. The distance equation is given by:

$$PQ_{mahalaobis} = \sqrt{(P - Q)^T S^{-1}(P - Q)}$$

Where P and Q are two random vectors of same distribution and S is covariance matrix.

NOTE: Most widely used distance measure is Euclidean distance, But all the distances have their respective purpose and importance. One cannot say or claim that only one particular distance measure is always accurate.

How to choose value of K-

The choice of k value in KNN classifier is critical. A small value of K implies a higher influence of noise over result whereas a large value makes it computationally expensive.

Some heuristics suggest to choose a K value = sqrt(N)/2 where N is the size of training dataset. Apart this an **odd** value (3, 5, 7,..) of K helps to avoid tie between predicted classes.

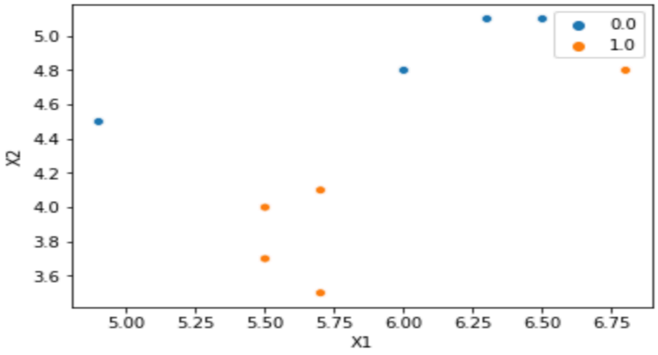
Other Variants-

1. **Radius Neighbour Classifier-** Radius Neighbour classifier implements learning by computing the number of neighbours within a fixed radius **R** for each training point. Radius Neighbour classifier is a good choice in case when sampling of data is not uniform. However, if the dataset has many attributes and is sparse this method becomes ineffective due the curse of dimensionality.

2. **KD Tree Nearest Neighbour-** This method use KD tree approach to implementing classifier. The method helps to reduce overall computation time for KNN classifier and becomes effective when there is large number of samples present in training set with few dimensions.
3. **KNN Regression-** The general principle of KNN regression is very much similar to KNN classifier except the target is continuous real value instead discrete and is predicted by calculating the average of neighbour values.

Example- Consider training data is given as shown in the table and we have asked to predict label of a query point q (5.6, 3.8). Let us use Euclidean distance to measure the similarity between query and training sample points with K = 3.

X1	X2	Label
6.3	5.1	0
6.0	4.8	0
6.8	4.8	1
5.7	3.7	1
5.7	4.1	1
5.5	4.0	1
6.5	5.1	0
5.5	3.7	1
4.9	4.5	0



Step 1- Compute the distance between query and sample points

$$d_1 = \sqrt{(5.6 - 6.3)^2 + (3.8 - 5.1)^2} = 1.476$$

$$d_2 = \sqrt{(5.6 - 6.0)^2 + (3.8 - 4.8)^2} = 1.077$$

$$d_3 = \sqrt{(5.6 - 6.8)^2 + (3.8 - 4.8)^2} = 1.562$$

$$d_4 = \sqrt{(5.6 - 5.7)^2 + (3.8 - 3.7)^2} = 0.141$$

$$d_5 = \sqrt{(5.6 - 5.7)^2 + (3.8 - 4.1)^2} = 0.316$$

$$d_6 = \sqrt{(5.6 - 5.5)^2 + (3.8 - 4.0)^2} = 0.224$$

$$d_7 = \sqrt{(5.6 - 6.5)^2 + (3.8 - 5.1)^2} = 1.581$$

$$d_8 = \sqrt{(5.6 - 5.5)^2 + (3.8 - 3.7)^2} = 0.141$$

$$d_9 = \sqrt{(5.6 - 4.9)^2 + (3.8 - 4.5)^2} = 0.990$$

Step 2- Select top K =3 Neighbours based on similarity

The top 3 neighbours are: P4, P8 and P6 with $d_4 = 0.141, d_8 = 0.141 \; and \; d_6 = 0.224$

Step 3- Determine the class of q by majority voting.

Point	X1	X2	Label
Point 4	5.7	3.7	1
Point 8	5.5	3.7	1
Point 6	5.5	4.0	1
Query q	**5.6**	**3.8**	**Predicted Class- 1**

ONE MORE TRY

Select top K =5 Neighbours based on similarity

The top 5 neighbours are: P4, P8, P6, P5 and P9 with $d_4 = 0.141, d_8 = 0.141, d_6 = 0.224, \; d_5 = 0.316 \; and \; d_9 = 0.990$

Point	X1	X2	Label
Point 4	5.7	3.7	1
Point 8	5.5	3.7	1
Point 6	5.5	4.0	1
Point 5	5.7	4.1	1
Point 9	4.9	4.5	0
Query q	**5.6**	**3.8**	**Predicted Class- 1**

Pros and Cons-

Pros- K- nearest neighbour algorithm is very simple to implement and the algorithm is robust when the error ratio is small. It also does not make any assumption about the distribution of classes and can work with multiple classes simultaneously.

Cons- It calculates distance for every new point so become computationally expensive(Lazy Learner). The method is not effective when distribution overlaps with each other. Fixing an optimal value of K is the challenge in KNN method.

******