CSE-403

Assignment
on
“KNN Algorithm”

Submitted to :

Zinia Sultana

Lecturer, CSE Dept, MIST

Submitted by:

Fahmida Yasmin Rifat (201714022)

Muhammad Zubair Hasan (201714041)

K-Nearest Neighbor's Algorithm (*k*-NN)

Introduction:

In pattern recognition, the ***k*-nearest neighbor's algorithm** (***k*-NN**) is a non-parametric method proposed by Thomas Cover used for classification and regression. In both 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:

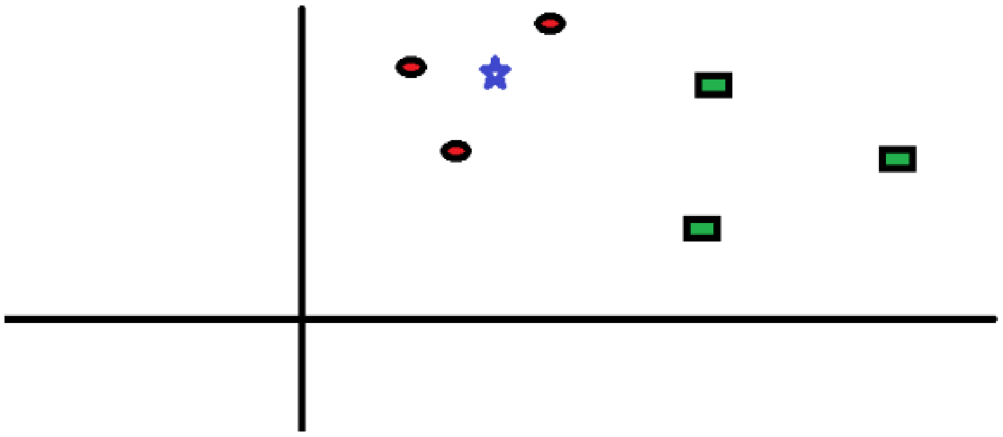
The following two properties would define KNN well −

- **Lazy learning algorithm −** KNN is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification.
- **Non-parametric learning algorithm −** KNN is also a non-parametric learning algorithm because it doesn't assume anything about the underlying data.

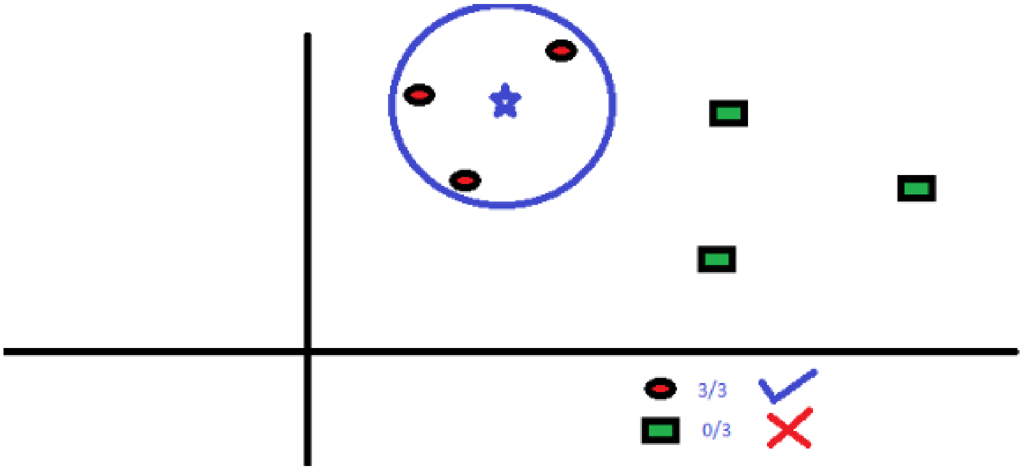
Nearest-neighbor classifiers are based on learning by analogy, that is, by comparing a given test tuple with training tuples that are similar to it. The training tuples are described by n attributes. When k = 1, the unknown tuple is assigned the class of the training tuple that is closest to it in pattern space.

How does the KNN algorithm work?

Let's take a simple case to understand this algorithm. Following is a spread of red circles (RC) and green squares (GS) :



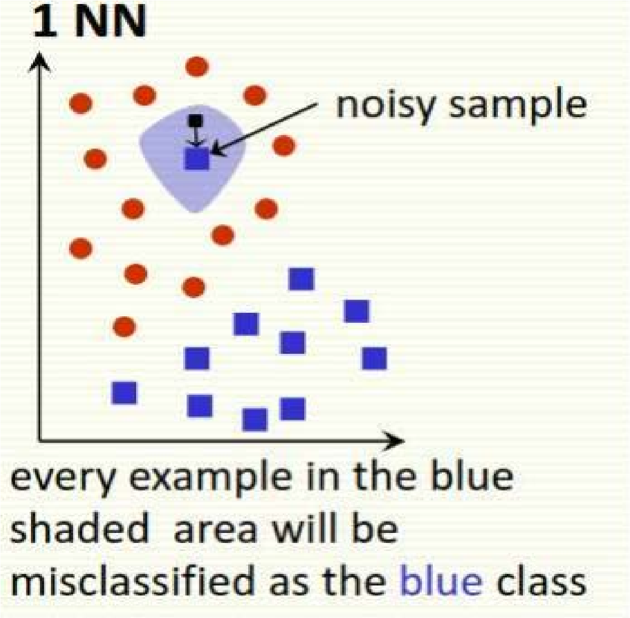
You intend to find out the class of the blue star (BS). BS can either be RC or GS and nothing else. The “K” is KNN algorithm is the nearest neighbor we wish to take the vote from. Let’s say K = 3. Hence, we will now make a circle with BS as the center just as big as to enclose only three data points on the plane. Refer to the following diagram for more details:

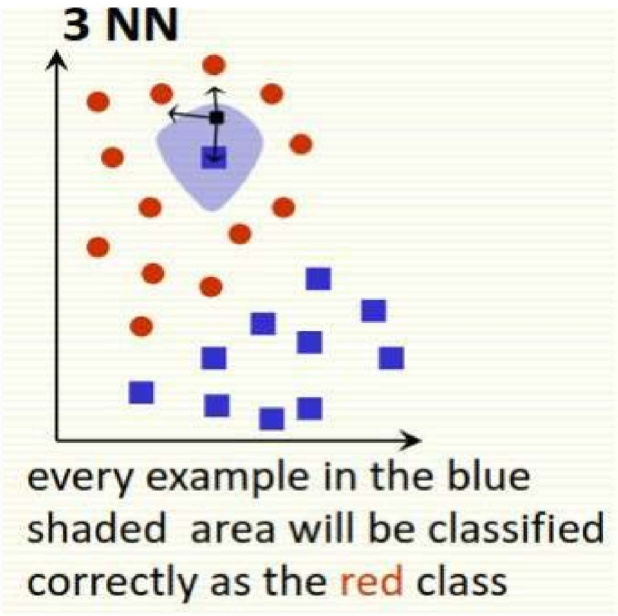


The three closest points to BS is all RC. Hence, with a good confidence level, we can say that the BS should belong to the class RC. Here, the choice became very obvious as all three votes from the closest neighbor went to RC. The choice of the parameter K is very crucial in this algorithm. Next, we will understand what are the factors to be considered to conclude the best K.

How to choose K?

If infinite number of samples available, the larger is k, the better is classification. k = 1 is often used for efficiency, but sensitive to “noise”


1 NN
noisy sample
every example in the blue shaded area will be misclassified as the blue class


3 NN
every example in the blue shaded area will be classified correctly as the red class

Larger k gives smoother boundaries, better for generalization, but only if locality is preserved. Locality is not preserved if end up looking at samples too far away, not from the same class. Interesting relation to find k for large sample data: k <= sqrt(n) where n is number of examples and k is an odd number. K is always assumed odd because if k is even it won’t be possible to break the tie.

Distance Measures:

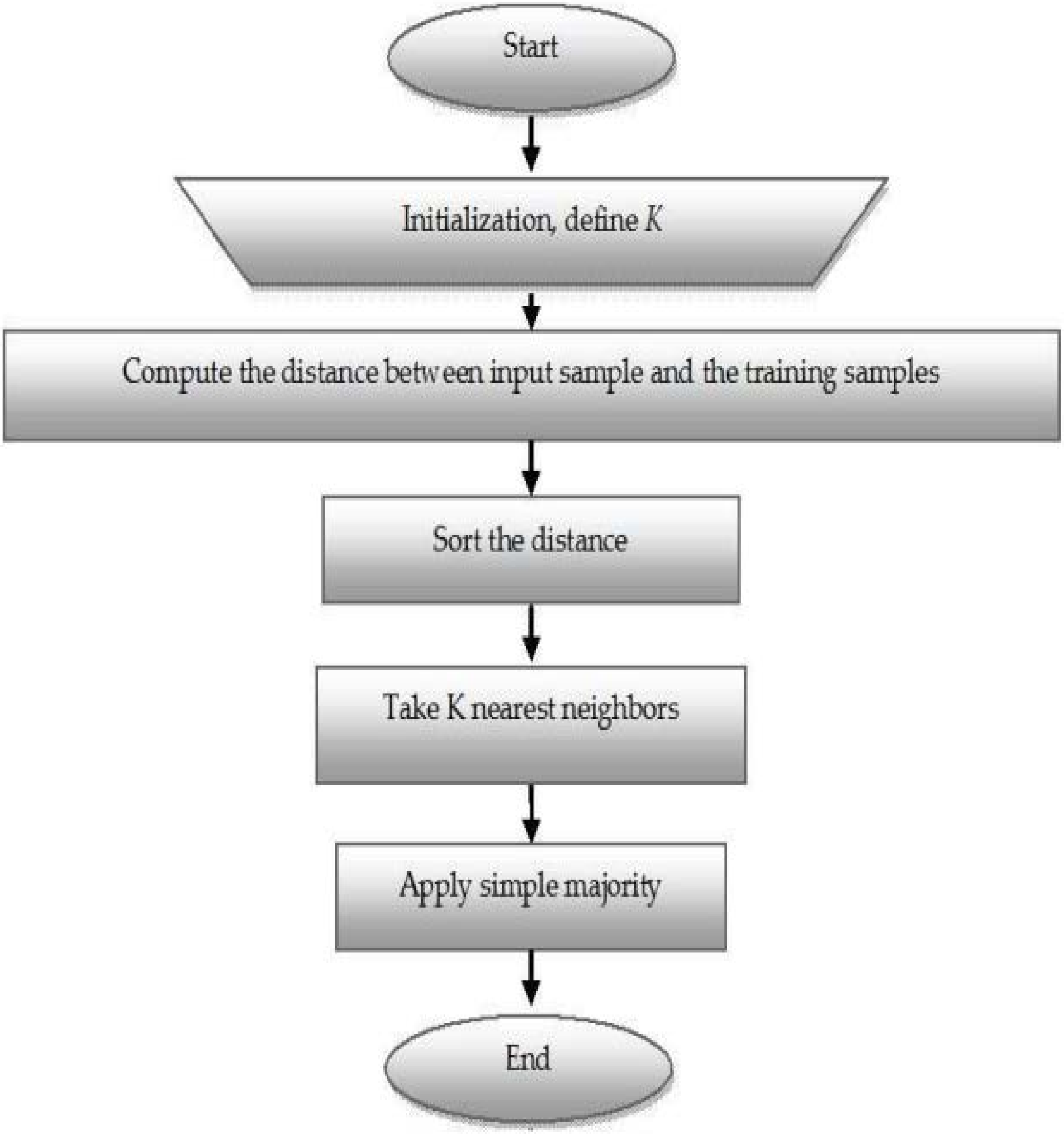
$Euclidean\ distance : (x, y) = \sqrt{}\ \sum(xi - yi)^{2}$

$Squared\ Euclidean\ distance : d\ (x, y) = \sum\ (xi - yi)^{2}$

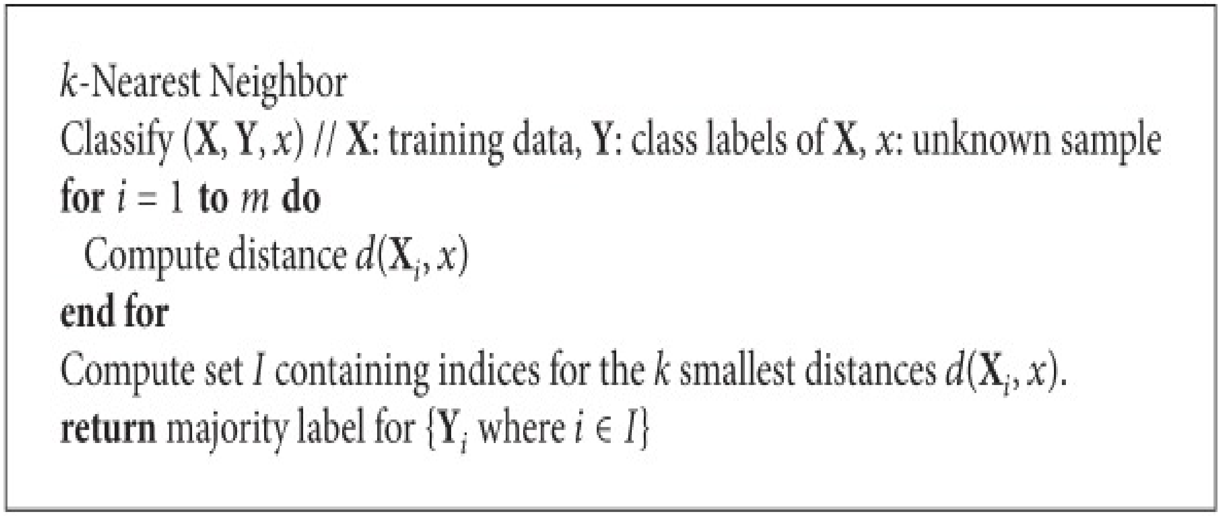
$Manhattan\ distance : d\ (x, y) = \sum|(xi - yi)|$

We use Euclidean Distance as it treats each feature as equally important.

KNN Classifier Algorithm:


Start
Initialization, define K
Compute the distance betw een input sample and the training samples
Sort the distance
Take K nearest neighbors
Apply simple majority
End

Pseudocode:



```
k-Nearest Neighbor
Classify ($\mathbf{X}, \mathbf{Y}, x$) // $\mathbf{X}$: training data, $\mathbf{Y}$: class labels of $\mathbf{X}$, $x$: unknown sample
for $i = 1$ to $m$ do
  Compute distance $d(\mathbf{X}_i, x)$
end for
Compute set $I$ containing indices for the $k$ smallest distances $d(\mathbf{X}_i, x)$.
return majority label for $\{\mathbf{Y}_i$ where $i \in I\}$
```

Complexity:

Basic KNN algorithm stores all examples. Suppose we have n examples each of dimension d. So, O(d) is needed to compute distance to one examples . And O(nd) is to computed distances to all examples . And we need O(nk) time to find k closest examples .So the total time is O(nk+nd) which is very expensive for a large number of samples . But we need a large number of samples for KNN to work well.

Simulation:

We have data from the questionnaires survey and objective testing with two attributes (acid durability and strength) to classify whether a special paper tissue is good or not. Here are four training samples:

X1 = Acid Durability (seconds)	X2 = Strength (kg/square meter)	Y = Classification
7	7	Bad
7	4	Bad
3	4	Good
1	4	Good

Now the factory produces a new paper tissue that passes the laboratory test with X1 = 3 and X2 = 7. Guess the classification of this new tissue.

Step 1: Initialize and Define k. Let's say, k = 3 (Always choose k as an odd number if the number of attributes is even to avoid a tie in the class prediction)

Step 2: Compute the distance between input sample and training sample - Co-ordinate of the input sample is (3,7). - Instead of calculating the Euclidean distance, we calculate the Squared Euclidean distance.

X1 = Acid Durability (seconds)	X2 = Strength (kg/square meter)	Squared Euclidean distance
7	7	$(7-3)^2 + (7-7)^2 = 16$
7	4	$(7-3)^2 + (4-7)^2 = 25$
3	4	$(3-3)^2 + (4-7)^2 = 09$
1	4	$(1-3)^2 + (4-7)^2 = 13$

Step 3: Sort the distance and determine the nearest neighbors based of the Kth minimum distance:

X1 = Acid Durability (seconds)	X2 = Strength (kg/square meter)	Squared Euclidean distance	Rank minimum distance	Is it included in 3-Nearest Neighbor?
7	7	16	3	Yes
7	4	25	4	No
3	4	09	1	Yes
1	4	13	2	Yes

Step 4: Take 3-Nearest Neighbors: Gather the category Y of the nearest neighbors.

X1 = Acid Durability (seconds)	X2 = Strength (kg/square meter)	Squared Euclidean distance	Rank minimum distance	Is it included in 3-Nearest Neighbor?	Y = Category of the nearest neighbor
7	7	16	3	Yes	Bad
7	4	25	4	No	-
3	4	09	1	Yes	Good
1	4	13	2	Yes	Good

Step 5: Apply simple majority. Use simple majority of the category of the nearest neighbors as the prediction value of the query instance. We have 2 “good” and 1 “bad”. Thus we conclude that the new paper tissue that passes the laboratory test with X1 = 3 and X2 = 7 is included in the “good” category.

Advantages of KNN classifier:

- Can be applied to the data from any distribution for example, data does not have to be separable with a linear boundary
- Very simple and intuitive
- Good classification if the number of samples is large enough.

Disadvantages of KNN classifier:

- Choosing k may be tricky
- Test stage is computationally expensive
- No training stage, all the work is done during the test stage

Applications of KNN Classifier:

- Used in classification
- Used to get missing values
- Used in gene expression
- Used in protein-protein prediction
- Used to get 3D structure of protein
- Used to measure document similarity