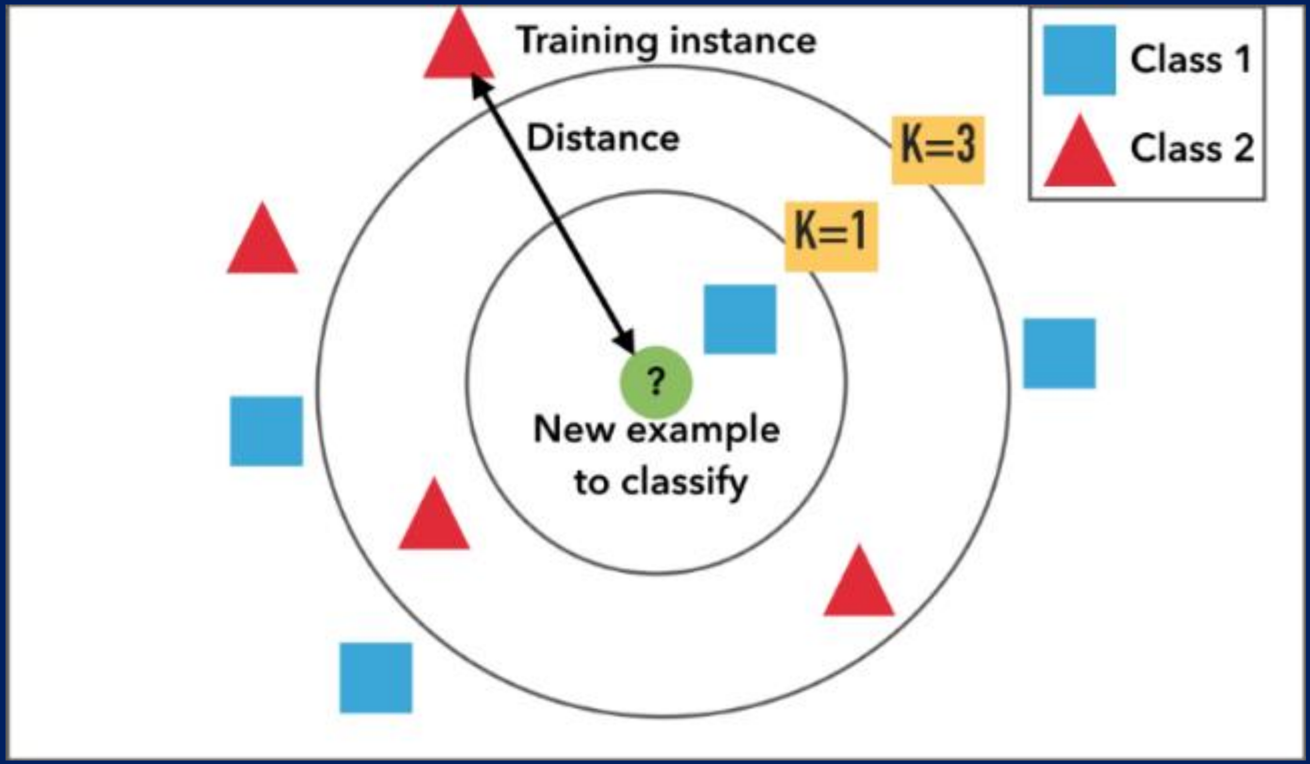
MCSE0007: Machine Learning



k Nearest Neighbors (k-NN)

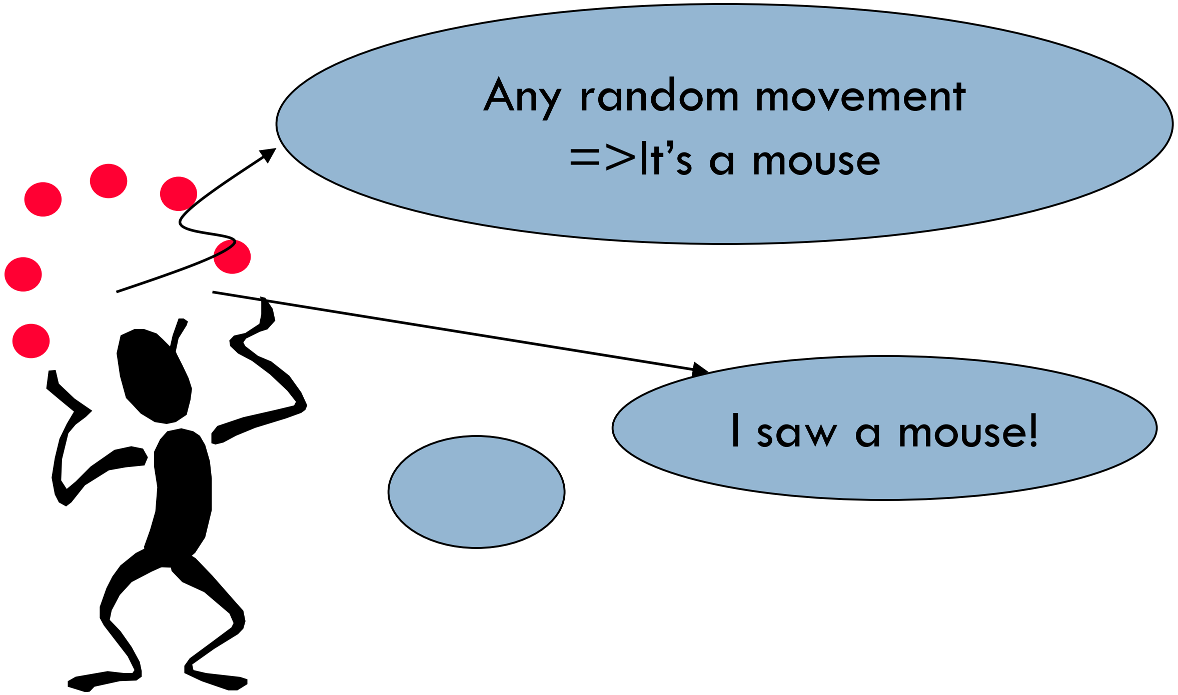
Class Presentations on Machine Learning by Prof. Anand Singh Jalal

Different Learning Methods

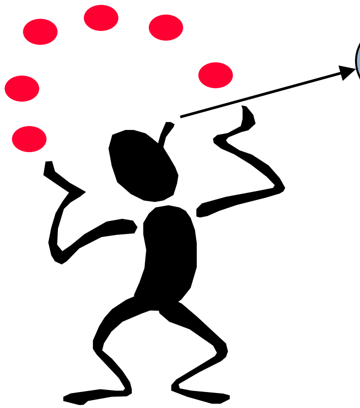
- Eager Learning
  - Explicit description of target function on the whole training set
- Instance-based Learning
  - Learning=storing all training instances
  - Classification=assigning target function to a new instance
  - Referred to as **"Lazy"** learning

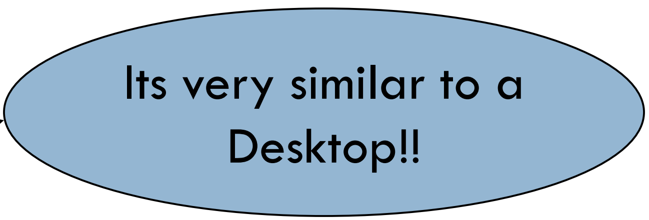
Different Learning Methods …

- Eager Learning



Different Learning Methods …





Instance-Based Learning

Idea:

- Similar examples have similar label.
- Classify new examples like similar training examples.

Algorithm:

- Given some new example **x** for which we need to predict its class **y**
- Find most similar training examples
- Classify ***x*** "like" these most similar examples

Questions:

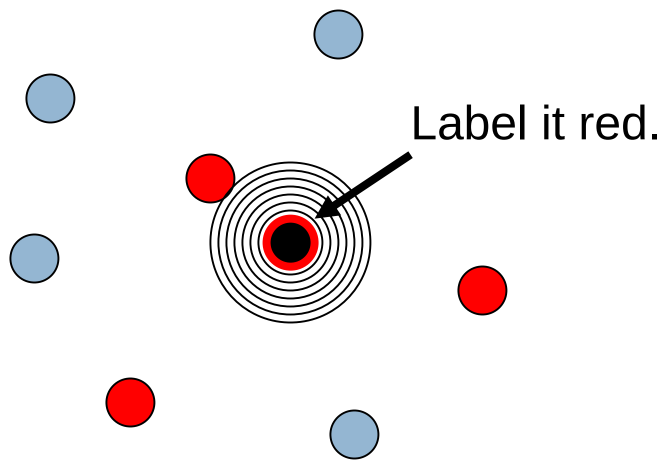
- How to determine similarity?
- How many similar training examples to consider?
- How to resolve inconsistencies among the training examples?

k - Nearest Neighbors

- The simplest, most used instance-based learning algorithm is the *k*-NN algorithm
- The k-nearest neighbors algorithm (k-NN) is a non-parametric, lazy learning method used for classification and regression.
- The output based on the majority vote (for classification) or mean (or median, for regression) of the k-nearest neighbors in the feature space.
- ***k*** is the number of neighbors considered

1-Nearest Neighbor

- One of the simplest of all machine learning classifiers
- **Simple idea:** label a new point the same as the closest known point



k - Nearest Neighbors

- For a given instance **T**, get the top **k** dataset instances that are “nearest” to **T**
  - Select a reasonable distance measure
- Inspect the category of these **k** instances, choose the category **C** that represent the most instances
- Conclude that **T** belongs to category **C**

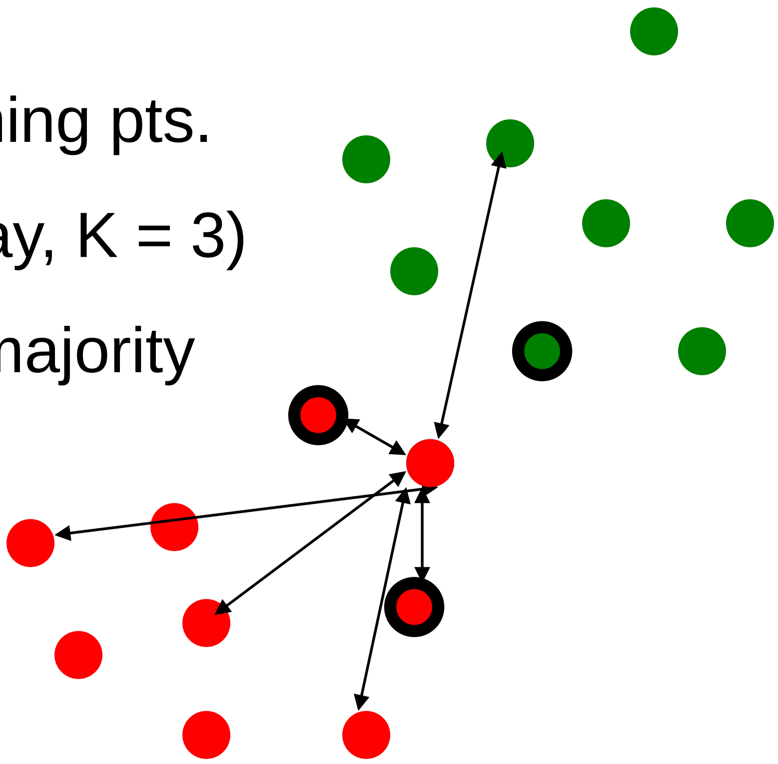
k-NN Classifier Schematic

For a test instance,

1) Calculate distances from training pts.

2) Find k-nearest neighbours (say, K = 3)

3) Assign class label based on majority

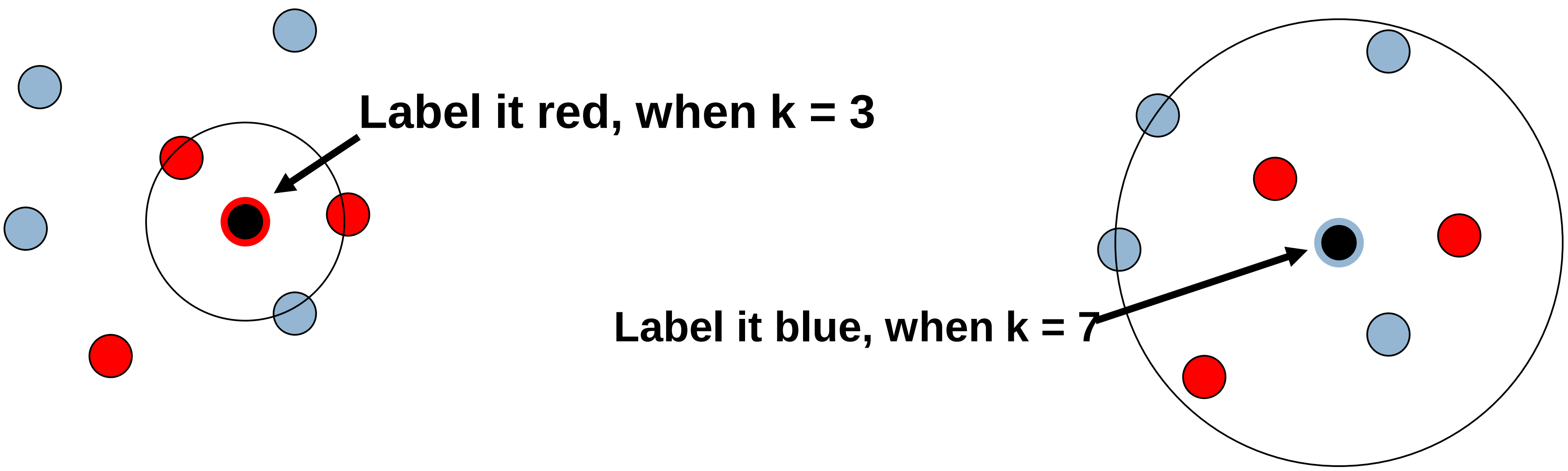


$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}.$$

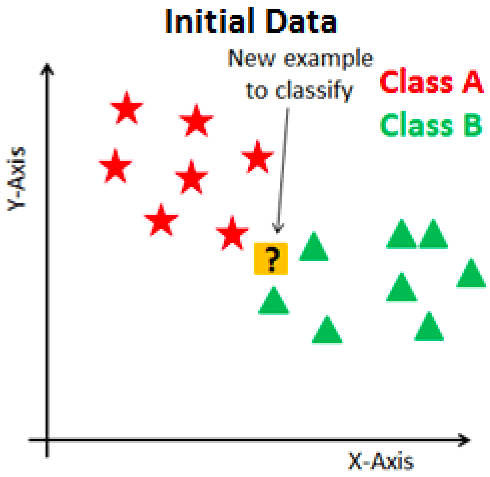
$$v' = \frac{v - min_A}{max_A - min_A},$$

k - Nearest Neighbors …

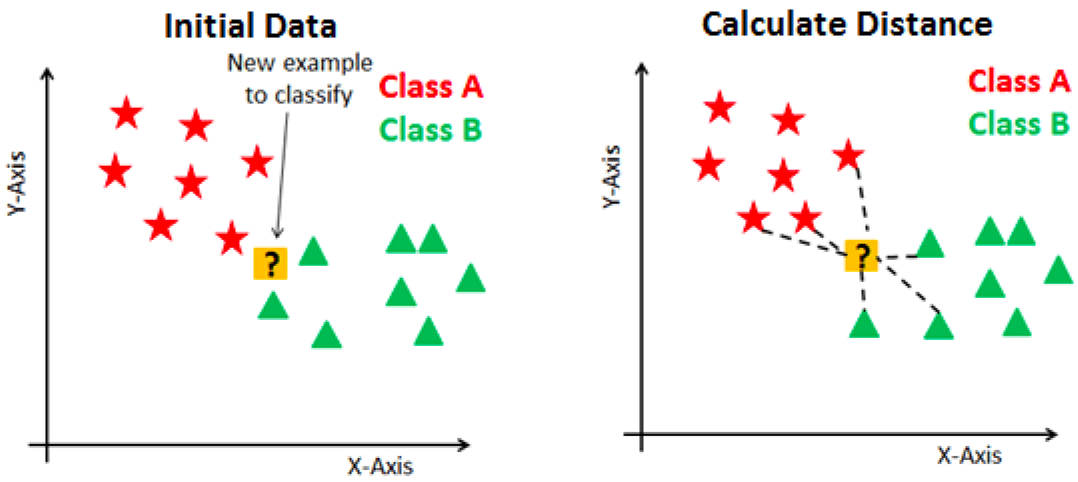
A new point is now assigned the most frequent label of its *k* nearest neighbors



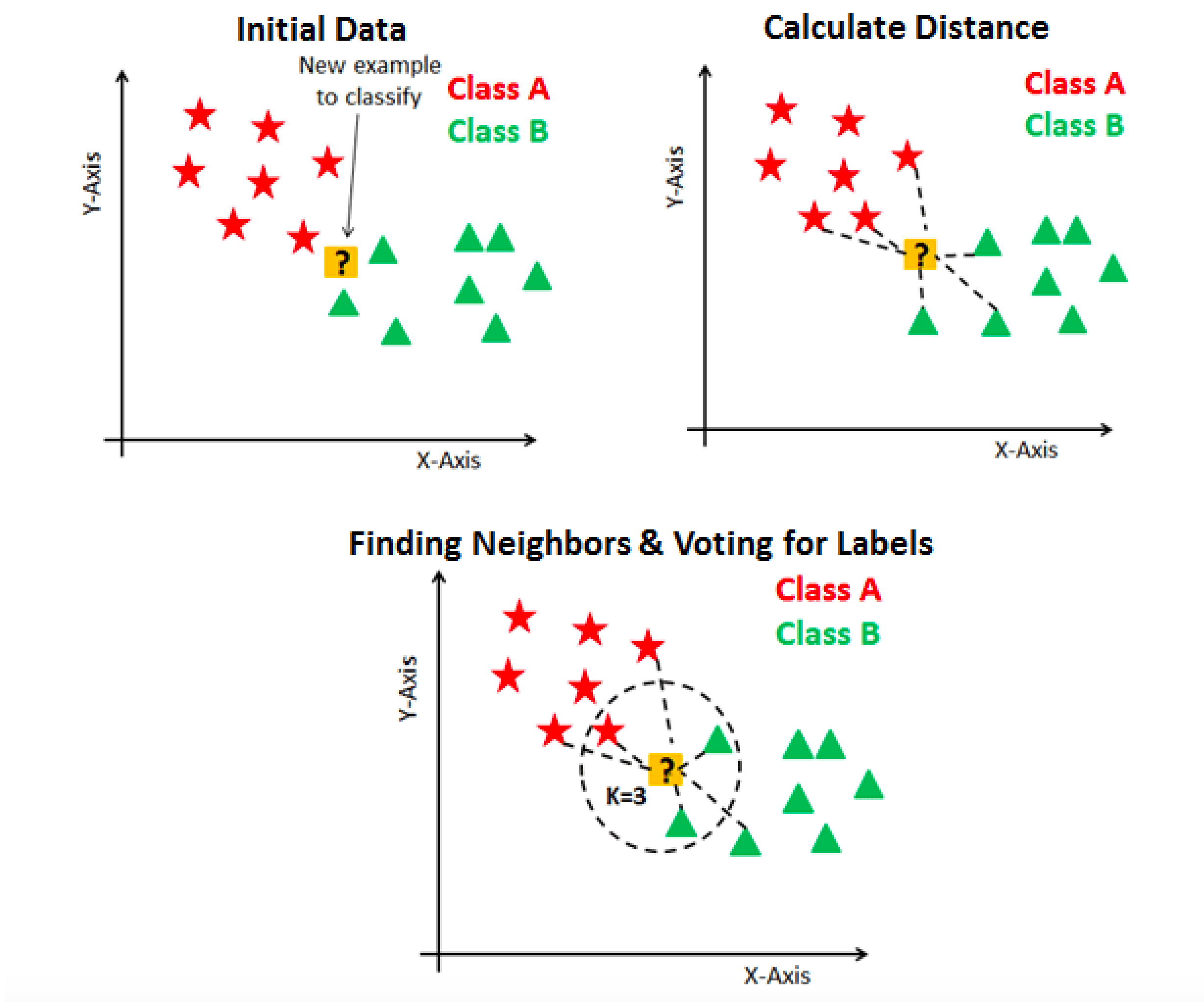
k - Nearest Neighbors …



k - Nearest Neighbors …



k - Nearest Neighbors …



Feature Space

- $\left\{ < \vec{x}^{(1)}, f(\vec{x}^{(1)}) >, < \vec{x}^{(2)} f(\vec{x}^{(2)}) >, \ldots, < \vec{x}^{(n)}, f(\vec{x}^{(n)}) > \right\}$

$$\vec{x} = \begin{cases} x_1 \\ x_2 \\ .. \\ .. \\ x_d \end{cases} \in \Re^d \qquad \left\| \vec{x} - \vec{y} \right\| = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$$

k - Nearest Neighbors Algorithm

- For each training instance $t=(x, f(x))$
  - Add t to the set *Tr_instances*
- Given a query instance q to be classified
  - Let $x_1, \ldots, x_k$ be the k training instances in *Tr_instances* nearest to q
  - Return

$$\hat{f}(q) = \arg\max_{v \in V} \sum_{i=1}^{k} \delta(v, f(x_i))$$

- Where V is the finite set of target class values, and $\delta(a,b)=1$ if $a=b$, and 0 otherwise (Kronecker function)
- Intuitively, the k-NN algorithm assigns to each new query instance the majority class among its k nearest neighbors

K-NN: Example

Customer	Age	Loan	Default
John	25	40000	N
Smith	35	60000	N
Alex	45	80000	N
Jade	20	20000	N
Kate	35	120000	N
Mark	52	18000	N
Anil	23	95000	Y
Pat	40	62000	Y
George	60	100000	Y
Jim	48	220000	Y
Jack	33	150000	Y
Andrew	48	142000	?

We need to predict Andrew default status by using Euclidean distance

K-NN: Example …

Calculate Euclidean distance for all the data points.

Customer	Age	Loan	Default	Euclidean distance
John	25	40000	N	1,02,000.00
Smith	35	60000	N	82,000.00
Alex	45	80000	N	62,000.00
Jade	20	20000	N	1,22,000.00
Kate	35	120000	N	22,000.00
Mark	52	18000	N	1,24,000.00
Anil	23	95000	Y	47,000.01
Pat	40	62000	Y	80,000.00
George	60	100000	Y	42,000.00
Jim	48	220000	Y	78,000.00
Jack	33	150000	Y	8,000.01
Andrew	**48**	**142000**	**?**	

First Step calculate the Euclidean distance dist(d) = Sq.rt $(x_1-y_1)^2 + (x_2-y_2)^2$
= Sq.rt$(48-25)^2 + (142000 - 40000)^2$
dist (d_1) = 1,02,000.

We need to calcuate the distance for all the datapoints

K-NN: Example …

Customer	Age	Loan	Default	Euclidean distance	Minimum Euclidean Distance
John	25	40000	N	1,02,000.00	
Smith	35	60000	N	82,000.00	
Alex	**45**	**80000**	**N**	**62,000.00**	**5**
Jade	20	20000	N	1,22,000.00	
Kate	**35**	**120000**	**N**	**22,000.00**	**2**
Mark	52	18000	N	1,24,000.00	
Anil	**23**	**95000**	**Y**	**47,000.01**	**4**
Pat	40	62000	Y	80,000.00	
George	**60**	**100000**	**Y**	**42,000.00**	**3**
Jim	48	220000	Y	78,000.00	
Jack	**33**	**150000**	**Y**	**8,000.01**	**1**
Andrew	48	142000	?		

Let assume K = 5

Find minimum euclidean distance and rank in order (ascending)

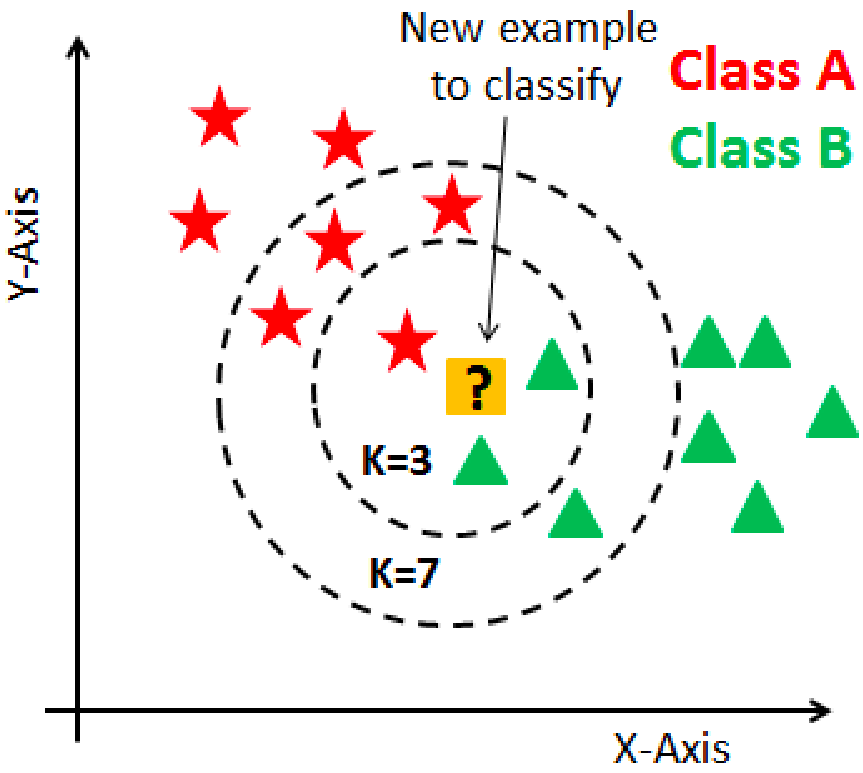
In this case, 5 minimum euclidean distance. With k=5, there are two Default = N and three Default = Y out of five closest neighbors.

We can say Andrew default stauts is 'Y' (Yes)

With K=5, there are two Default=N and three Default=Y out of five closest neighbors. We can say default status for Andrew is 'Y' based on the major similarity of 3 points out of 5.

k - Nearest Neighbors …

What should be the value of K? How do we choose K?



Different K could have different results.

How to determine the good value for k?

- Determined experimentally
- Start with k=1 and use a test set to validate the error rate of the classifier
- Repeat with k=k+2
- Choose the value of k for which the error rate is minimum

- Note: Try and keep the value of k odd in order to avoid confusion between two classes of data

k - Nearest Neighbors …

- In nearest-neighbor learning the target function may be either discrete-valued or real valued
- Learning a discrete valued function

- $f : \Re^d \rightarrow V$, V is the finite set $\{v_1, \ldots\ldots, v_n\}$

- For discrete-valued, the *k*-NN returns the most common value among the k training examples nearest to x_q.

Continuous-Valued Target Functions

- k-NN approximating continous-valued target functions
- Calculate the mean value of the *k* nearest training examples rather than calculate their most common value

$$f : \Re^d \rightarrow \Re$$

$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} f(x_i)}{k}$$

Distance Weighted

- Refinement to kNN is to weight the contribution of each *k* neighbor according to the distance to the query point x_q
  - ✓ Greater weight to closer neighbors
  - ✓ For discrete target functions

$$\hat{f}(x_q) \leftarrow \underset{v \in V}{\arg\max} \sum_{i=1}^{k} w_i \delta(v, f(x_i))$$

$$w_i = \begin{cases} \dfrac{1}{d(x_q, x_i)^2} & if \quad x_q \neq x_i \\ 1 & else \end{cases}$$

Distance Weighted …

For real valued functions

$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} w_i f(x_i)}{\sum_{i=1}^{k} w_i}$$

$$w_i = \begin{cases} \dfrac{1}{d(x_q, x_i)^2} & if \quad x_q \neq x_i \\ 1 & else \end{cases}$$

The k-NN Algorithm

1. Load the data
2. Initialize K to your chosen number of neighbors
3. For each example in the data
    3.1 Calculate the distance between the query example and the current example from the data.
    3.2 Add the distance and the index of the example to an ordered collection
4. Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances
5. Pick the first K entries from the sorted collection
6. Get the labels of the selected K entries
7. If regression, return the mean of the K labels
8. If classification, return the mode of the K labels

k - Nearest Neighbors …

Pros:

- No assumptions about data distribution, useful in real world application
- Simple algorithm to explain and understand
- It can use for both classification and regression

Cons:

- Computationally expensive, because the algorithm stores all of the training data
- High memory requirement, again, it stores all of the training data
- Prediction stage might be slow (with big N)

Summary

- KNN stores the entire training dataset which it uses as its representation.
- KNN does not learn any model.
- KNN makes predictions just-in-time by calculating the similarity between an input sample and each training instance.
- There are many distance measures to choose from to match the structure of your input data.
- That it is a good idea to rescale your data, such as using normalization, when using KNN.

Assignment Questions

1. What is “K” in KNN algorithm?
2. How do we decide the value of "K" in KNN algorithm?
3. Why is the odd value of “K” preferable in KNN algorithm?
4. What is the difference between Euclidean Distance and Manhattan distance? What is the formula of Euclidean distance and Manhattan distance?
5. Why is KNN algorithm called Lazy Learner?
6. Why should we not use KNN algorithm for large datasets?
7. What are the advantages and disadvantages of KNN algorithm?

k-NN algorithm does more computation on test time rather than train time.

A) TRUE

B) FALSE

Ans: True

The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples.

In the testing phase, a test point is classified by assigning the label which are most frequent among the k training samples nearest to that query point – hence higher computation.

Q. A 1-NN classifier has higher variance than a 3-NN classifier. True/False

Ans: True

Q. As we decrease the value of K to 1, our predictions become less stable. True/False

Ans: True

Which of the following distance metric can not be used in k-NN?

A) Manhattan

B) Minkowski

C) Tanimoto

D) Jaccard

E) Mahalanobis

F) All can be used

Solution: F

All of these distance metric can be used as a distance metric for k-NN.

Which of the following option is true about k-NN algorithm?

A) It can be used for classification

B) It can be used for regression

C) It can be used in both classification and regression

Solution: C

We can also use k-NN for regression problems. In this case the prediction can be based on the mean or the median of the k-most similar instances.

Which of the following statement is true about k-NN algorithm?

- k-NN performs much better if all of the data have the same scale
- k-NN works well with a small number of input variables (p), but struggles when the number of inputs is very large
- k-NN makes no assumptions about the functional form of the problem being solved

A) 1 and 2
B) 1 and 3
C) Only 1
D) All of the above

Solution: D

The above mentioned statements are assumptions of kNN algorithm

Which of the following machine learning algorithm can be used for imputing missing values of both categorical and continuous variables?

A) K-NN
B) Linear Regression
C) Logistic Regression

Solution: A

k-NN algorithm can be used for imputing missing value of both categorical and continuous variables.

Which of the following will be Euclidean Distance between the two data point A(1,3) and B(2,3)?

A) 1
B) 2
C) 4
D) 8

Solution: A

sqrt((1-2)^2 + (3-3)^2) = sqrt(1^2 + 0^2) = 1

Which of the following will be Manhattan Distance between the two data point A(1,3) and B(2,3)?

A) 1

B) 2

C) 4

D) 8

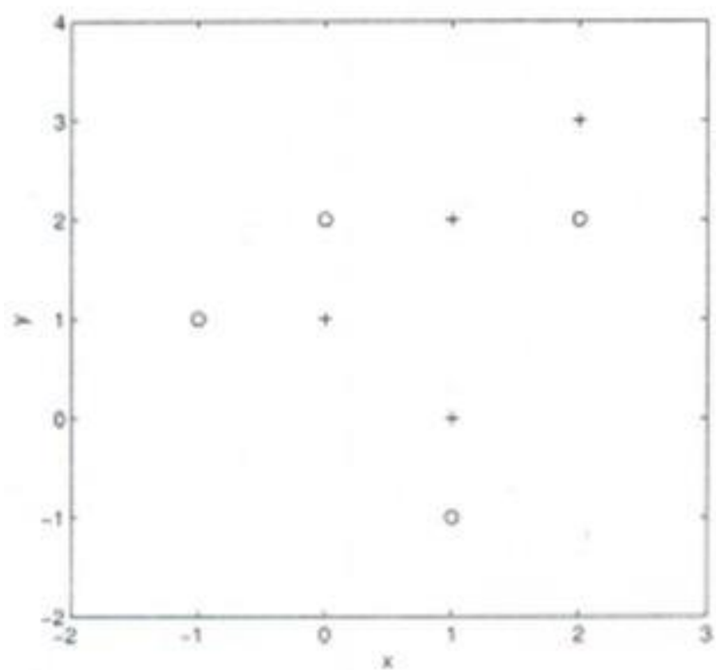
Solution: A

sqrt(mod((1-2)) + mod((3-3))) = sqrt(1 + 0) = 1

Suppose, you have given the following data where x and y are the 2 input variables and Class is the dependent variable.

x	y	Class
-1	1	$-$
0	1	$+$
0	2	$-$
1	-1	$-$
1	0	$+$
1	2	$+$
2	2	$-$
2	3	$+$

Below is a scatter plot which shows the above data in 2D space.



Suppose, you want to predict the class of new data point x=1 and y=1 using eucludian distance in 3-NN. In which class this data point belong to?

A) + Class

B) – Class

C) Can't say

D) None of these

Solution: A

All three nearest point are of +class so this point will be classified as +class.

In the previous question, you are now want use 7-NN instead of 3-KNN which of the following x=1 and y=1 will belong to?

A) + Class

B) – Class

C) Can't say

Solution: B

Now this point will be classified as – class because there are 4 – class and 3 +class point are in nearest circle.

- Suppose you have given height, weight and T-shirt size of some customers. We need to predict the T-shirt size of a new customer with height =161 cm and weight=62 Kg

H (cm)	168	158	158	160	160	163	163	160	163	165	165	163	168
W (kg)	58	59	63	59	60	60	61	64	64	61	62	65	62
Size	M	M	M	M	M	M	M	L	L	L	L	L	L

Assume K=5

Any Questions ?