innovate achieve lead

BITS Pilani

Machine Learning (IS ZC464) Session 4 : Classification Algorithms

Classification Algorithms

- K-Nearest Neighbors
- Naïve Bayes’ classifier
- Decision tree classifier

K-Nearest Neighbors

- Training instances are stored but no generalization is done before an instance for testing arrives.
- No hypothesis using training input output pairs is constructed.
- A new instance is examined with respect to all training instances (observations) and is given its target value.

Lazy Learning

- These methods are sometimes referred to as "lazy" learning methods because they delay processing until a new instance must be classified.
- The target function is not estimated once for the entire input (instance) space.
- The target label or class of the new instance is computed locally or differently when it comes.

What does it mean by lazy learning?

- The instance based methods keep using the training data from its memory without learning any knowledge from the data.
- Every new instance requires the entire training data to be processed due to which classification is delayed and is called lazy learning.

Instance based methods

- K-Nearest Neighbor Learning
- Locally Weighted Regression

k-nearest Neighbor Learning

- This algorithm assumes all instances correspond to points in the n-dimensional space $\Re$.
- The nearest neighbors are defined in terms of the Euclidean distance.
- Let an arbitrary instance ‘x’ be described by the feature vector $(a_1(x), a_2(x), \ldots, a_n(x))$ where $a_r(x)$ is the value of the r^{th} attribute of instance x.

Distance measure

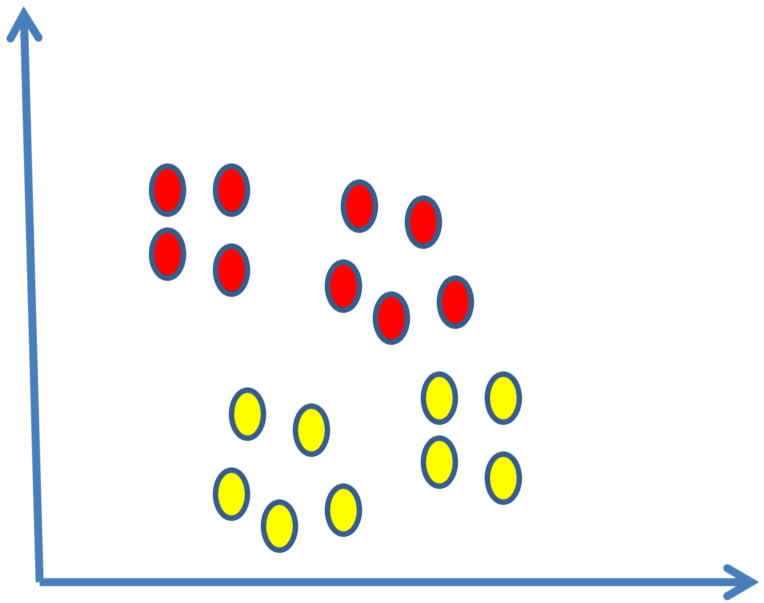
- The distance between two instances x_i and x_j is defined by $d(x_i, x_j)$ where

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2}$$

- The target function in nearest neighbor learning may be either discrete valued or real valued.

K-Nearest Neighbor Algorithm

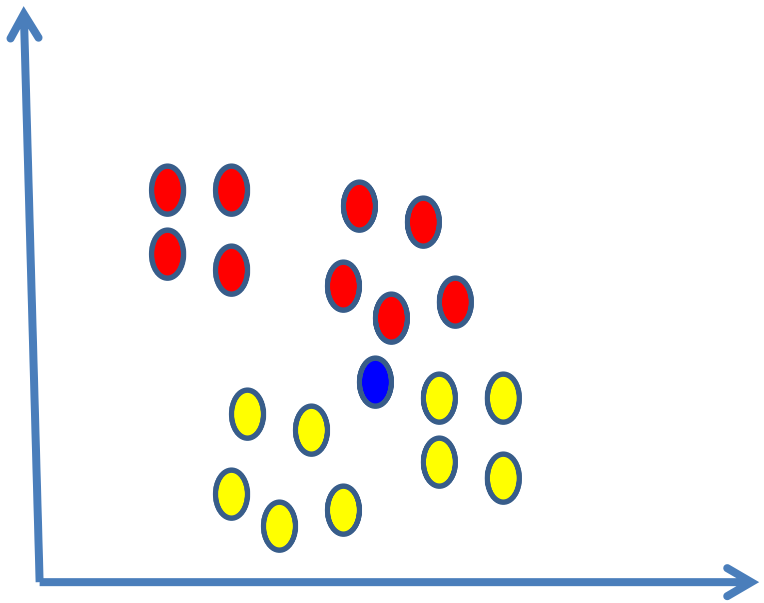
- Consider the following two dimensional points belonging to two different classes represented as two colors



- The points are representing training instances by the corresponding feature vectors.
- The feature vector is an x,y coordinate pair for each point

K-Nearest Neighbor Algorithm

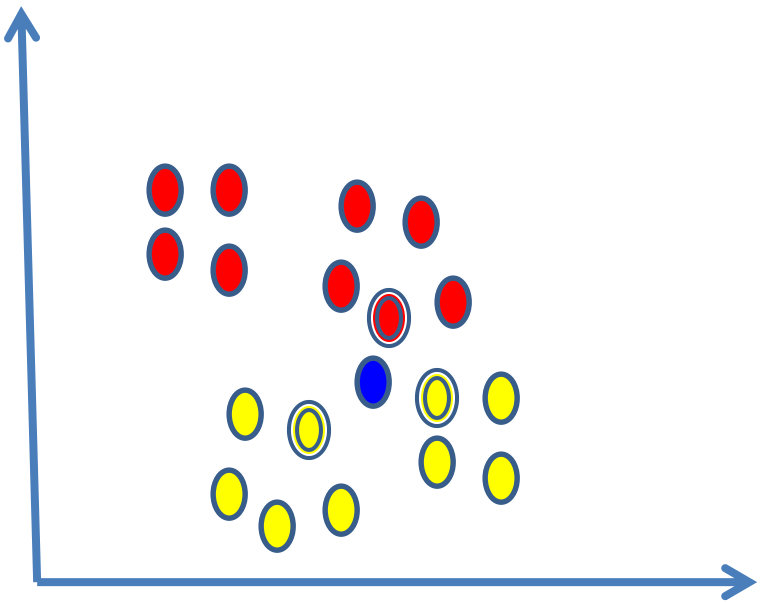
- Let a new testing instance arrives (shown as blue dot) and is plotted on the same coordinate plane



- The k-nearest neighbor algorithm requires ‘k’ to be defines as a value.
- For example if k=3, we compute the distances of the new instance from all training data and mark three closest points

K-Nearest Neighbor Algorithm

The three closest neighbors are marked as double bordered circles with respective colors as class labels



As two yellow colored instances are close to the new testing instance (blue) out of the three closest neighbors, we assign the class (with yellow color as label) to the testing instance

Classification as mapping

- Define $f: \Re^n \rightarrow V$
- Where $V = \{v_1, v_2, ..v_s\}$ set of class labels
- And $\Re^n$ is the n-dimensional space

- A value $f(x_q)$ represents the class label to which the testing sample x_q belongs to

Which class should be assigned to the testing sample x_q?

- Define a function δ as follows

  $\delta(a,b) = 1$ if $a=b$

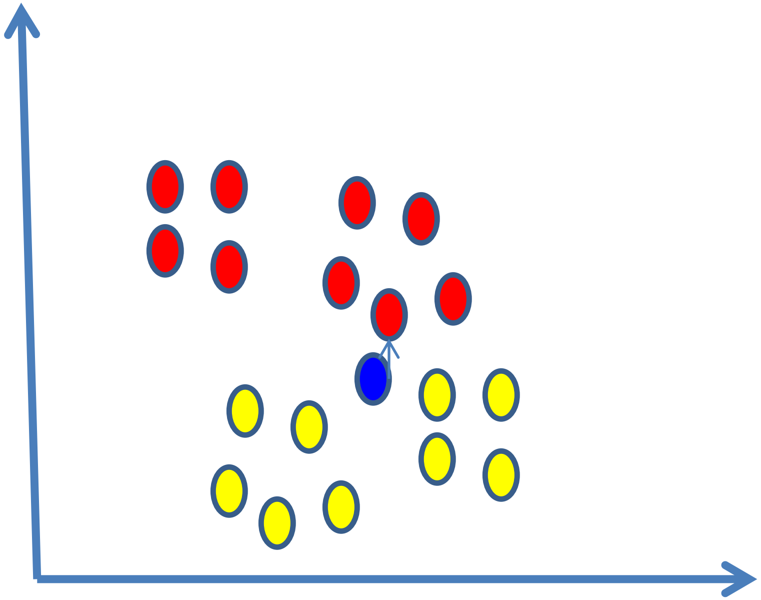
  0 otherwise

- $\delta(v,f(x_i))$ gets a value 1 if the class of training data x_i is v. This value contributes in the classification
- The class of the test instance x_q is given by

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

1-Nearest Neighbor Learning

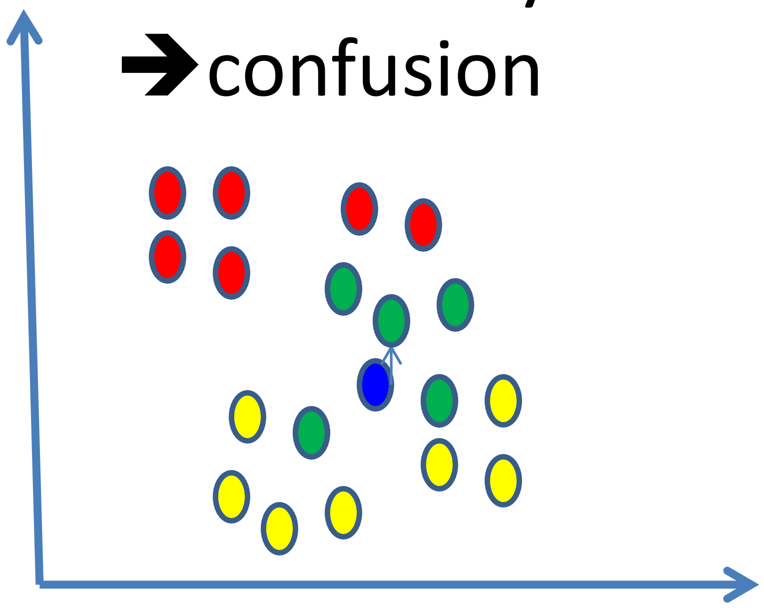
- This method selects value of k as 1 and assigns the target class to the testing instance of the training instance which is nearest to it.



Of the samples around the testing sample, the nearest one is seen and its corresponding class is assigned to the testing sample

Classification accuracy may change if k changes

- 1-NN assigns the class label 1 (red)
- 3-NN assigns the class label 2 (yellow)
- 4-NN has 2 yellow and 2 red nearest samples
  ➔confusion



- 5-NN assigns class label 1 (red)

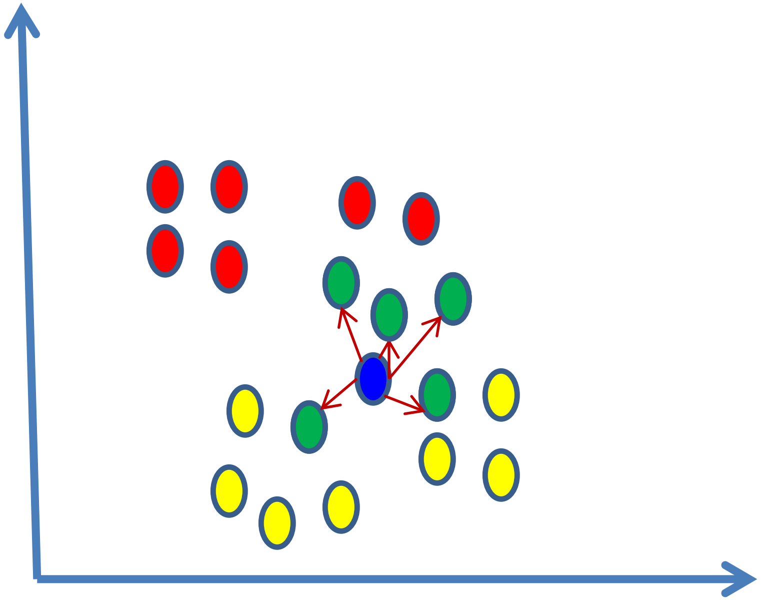
Distance Weighted Nearest Neighbor Algorithm

- The contribution of the k-neighbors is weighed according to their respective distance from the given test data point.
- More weight is given to the training data of the k nearest points being the closest while the points at father distance are given less weight.

Weight

- The weight w_i of the i^{th} training point at distance $d(x_i, x_q)$ is inverse of square of the value of $d(x_i, x_q)$.

$$w_i = \frac{1}{d(x_i, x_j)^2}$$



Which class should be assigned based on weights?

- Define a function δ as follows

$$\delta(a,b) = 1 \text{ if } a=b$$
$$0 \text{ otherwise}$$

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

Example

- 5-NN
- Let the distances of the training instances of class 1 be 4, 6 and 2 and those of the class 2 be 2 and 3 from the test instance
- Compute the class label for test instance
- **Compute for class 1:** $1/(4)^2$,$1/(6)^2$, $1/(2)^2$ which are 0.0625, 0.0277, 0.25 sum of which is 0.3402
- **Compute for class 2:** $1/(2)^2$,$1/(3)^2$ which are 0.25 and 0.1111 respectively, sum of which is 0.3611
- **Class 2 is assigned to the test sample.**

Naïve Bayes' Classifier

- This classifier uses the posterior probabilities computed using Bayes' theorem
- Bayes theorem

$$P(Y|X_1, \ldots, X_n) = \frac{P(X_1, \ldots, X_n|Y)P(Y)}{P(X_1, \ldots, X_n)}$$

- Where

$$P(X_1, \ldots, X_n|Y) = \prod_{i=1}^{n} P(X_i|Y)$$

An example: Training with 14 observations

Training data: The weather data, with counts and probabilities

	Outlook (X1)		temperature (X2)			humidity (X3)			Windy (X4)			Play (Y)	
	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	2	3	hot	2	2	high	3	4	false	6	2	9	5
overcast	4	0	mild	4	2	normal	6	1	true	3	3		
rainy	3	2	cool	3	1								
sunny	2/9	3/5	hot	2/9	2/5	high	3/9	4/5	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	mild	4/9	2/5	normal	6/9	1/5	true	3/9	3/5		
rainy	3/9	2/5	cool	3/9	1/5								

Testing data

outlook	temperature	humidity	windy	play
sunny	cool	high	true	?

Highlight testing variables for Y = yes

Training data: The weather data, with counts and probabilities

	Outlook (X1)		temperature (X2)			humidity (X3)			Windy (X4)			Play (Y)	
	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	2	3	hot	2	2	high	3	4	false	6	2	9	5
overcast	4	0	mild	4	2	normal	6	1	true	3	3		
rainy	3	2	cool	3	1								
sunny	2/9	3/5	hot	2/9	2/5	high	3/9	4/5	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	mild	4/9	2/5	normal	6/9	1/5	true	3/9	3/5		
rainy	3/9	2/5	cool	3/9	1/5								

Testing data

outlook	temperature	humidity	windy	play
sunny	cool	high	true	?

Testing for Y=yes

$$P(Y|X_1,\ldots,X_n) = \frac{P(X_1,\ldots,X_n|Y)P(Y)}{P(X_1,\ldots,X_n)}$$

$$P(X_1,\ldots,X_n|Y) = \prod_{i=1}^{n} P(X_i|Y)$$

For Y = yes (plays)   P(X1=sunny, X2=cool, X3 =high, X4 = true | Y=yes) P(Y=yes)

$$= \frac{2}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{9}{14} = 0.0053$$

Highlight testing variables for Y = no

Training data: The weather data, with counts and probabilities

	Outlook (X1)		temperature (X2)			humidity (X3)			Windy (X4)			Play (Y)	
	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	2	3	hot	2	2	high	3	4	false	6	2	9	5
overcast	4	0	mild	4	2	normal	6	1	true	3	3		
rainy	3	2	cool	3	1								
sunny	2/9	3/5	hot	2/9	2/5	high	3/9	4/5	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	mild	4/9	2/5	normal	6/9	1/5	true	3/9	3/5		
rainy	3/9	2/5	cool	3/9	1/5								

Testing data

outlook	temperature	humidity	windy	play
sunny	cool	high	true	?

Testing for Y=no

$$P(Y|X_1,\ldots,X_n) = \frac{P(X_1,\ldots,X_n|Y)P(Y)}{P(X_1,\ldots,X_n)}$$

$$P(X_1,\ldots,X_n|Y) = \prod_{i=1}^{n} P(X_i|Y)$$

For Y = no P(X1=sunny, X2=cool, X3 =high, X4 = true | Y=no) P(Y=no)

$$= \frac{3}{5} \times \frac{1}{5} \times \frac{4}{5} \times \frac{3}{5} \times \frac{5}{14} = 0.0206$$

Prediction

Testing data				
outlook	temperature	humidity	windy	play
sunny	cool	high	true	?

- P(Y=yes| X1=sunny, X2=cool, X3 =high, X4 = true) = 0.0053/α
- P(Y=no| X1=sunny, X2=cool, X3 =high, X4 = true) = 0.0206/α

- Where α = P(X1, X2, X3, X4) and is constant for both
- Since 0.0206/α is greater than 0.0053/α, therefore the predicted class is **no for the given test data**