Natural Language Processing

Assist. Prof. Dr. Tuğba YILDIZ

İSTANBUL BİLGİ UNIVERSITY
Department of Computer Engineering

April 19, 2019

Machine Learning Algorithms

- studies how to automatically learn to make accurate **predictions** based on past observations
- make **description**
- Three different learning styles in machine learning algorithms:
  1. Supervised Learning
  2. Unsupervised Learning
  3. Semi-supervised Learning

Machine Learning Algorithms

- Three different learning styles in machine learning algorithms:
  1. Supervised Learning
     - input data is called training data and has a known label or result such as spam/not-spam etc.
     - a model is prepared through a training process in which it is required to make predictions and is corrected when those predictions are wrong.
     - the training process continues until the model achieves a desired level of accuracy on the training data.
     - studies how to automatically learn to make accurate predictions based on past observations
- Decision trees, linear regression, naive bayes, knn

Machine Learning Algorithms for Classification

- text categorization (e.g., spam filtering)
- fraud detection
- optical character recognition
- machine vision (e.g., face detection)
- natural-language processing (e.g., spoken language understanding)
- market segmentation (e.g.: predict if customer will respond to promotion)
- bioinformatics (e.g., classify proteins according to their function)

Machine Learning Algorithms

- Three different learning styles in machine learning algorithms:
  1. Unsupervised Learning
     - Input data is not labeled and does not have a known result.
     - A model is prepared by deducing structures present in the input data.
     - This may be to extract general rules. It may be through a mathematical process to systematically reduce redundancy, or it may be to organize data by similarity.
- Example: K-means, Apriori

Decision Tree

- The goal is to create a model that predicts the value of a target variable based on several input variables.
- constructs a model of decisions made based on actual values of attributes in the data.

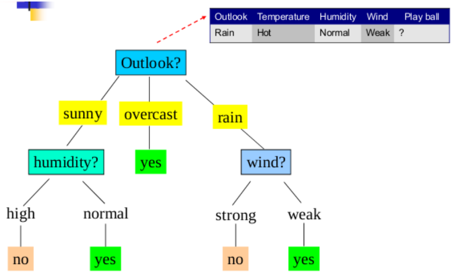
Decision Tree

- Decision Tree- Example

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Decision Tree

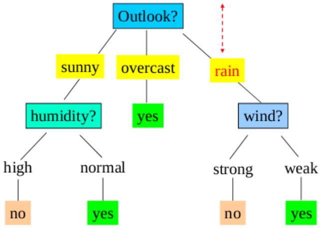
- Decision Tree- Example



Decision Tree

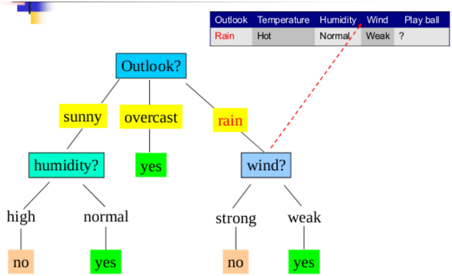
- Decision Tree- Example

Outlook	Temperature	Humidity	Wind	Play ball
Rain	Hot	Normal	Weak	?



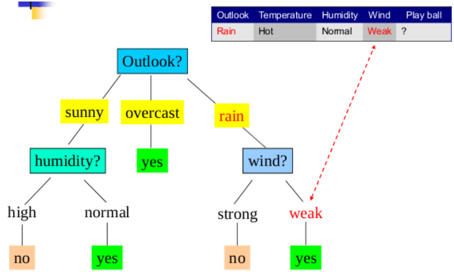
Decision Tree

- Decision Tree- Example



Decision Tree

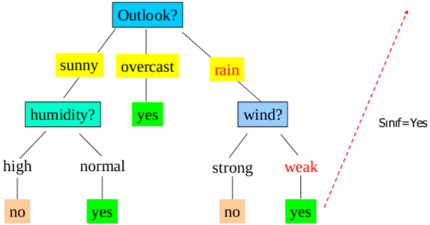
- Decision Tree- Example



Decision Tree

- Decision Tree- Example

Outlook	Temperature	Humidity	Wind	Play ball
Rain	Hot	Normal	Weak	?

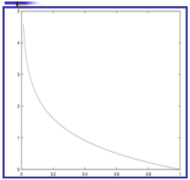


Decision Tree

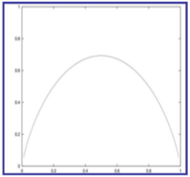
- ID3, C4.5
- to choose the most informative feature. How?
  - information gain
  - gini index

Decision Tree

- Information Gain
- Entropy
- $H(P_1, P_2, ..., P_s) = -\sum_{i=1}^{s} p_i log(p_i)$



log (p)



H(p,1-p)

- examples are in same class=0
- examples are distrubuted equally= 1
- examples are distrubted randomly 0<entropy<1

Decision Tree

- Entropy
- $H(p_1, p_2, ..., p_s) = -\sum_{i=1}^{s} p_i log(p_i)$
- In S, we have 14 examples: C0=9, C1=5
- examples are distrubuted equally= 1
- examples are distrubted randomly 0<entropy<1
- H(p1,p2)= - (9/14) Log2 (9/14) - (5/14) Log2 (5/14) = 0.940

Decision Tree

- Information gain of Attribute A in S

$$Gain(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|Sv|}{|S|} Entropy(S_v)$$

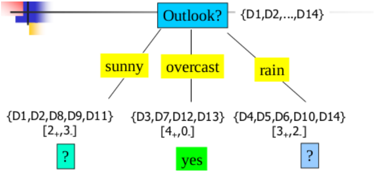
Decision Tree

- s1=9(yes), s2=5(no)
- Entropy(S) = - (9/14) Log 2 (9/14) - (5/14) Log 2 (5/14) = 0.940
- wind: weak=8, strong=6
- weak: no=2, yes=6
- strong: no=3, yes=3
- Entropy(S_{weak}) = - (6/8)*log2(6/8) - (2/8)*log2(2/8) = 0.811
- Entropy(S_{strong}) = - (3/6)*log2(3/6) - (3/6)*log2(3/6) = 1.00
- Entropy wind (S) = (8/14)*0.811 + (6/14)*1.00
- Gain(wind)=0.940 - (8/14)*0.811 - (6/14)*1.00

Gain(Outlook) = 0.246
Gain(Humidity) = 0.151
Gain(wind)=0.048
Gain(Temperature) = 0.029

Decision Tree

- Decision Tree



Decision Tree

- Decision Tree- Example
- S_{sunny} = D1,D2,D8,D9,D11, Entropy(Ssunny)=0.970
- humidity : high=3, normal=2
- high: no=3, yes=0
- normal: no=0, yes=2
- Entropy(S_{high}) = 0
- Entropy(S_{normal}) = 0
- Gain(S_{sunny} ,Humidity)=0.970-(3/5)0.0- (3/5)0.0=0.970

Naive Bayes Algorithm

- Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature
- It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors.
- Naive Bayes model is easy to build and particularly useful for very large data sets.
- Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Naive Bayes Algorithm

- Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c).
- Look at the equation below:

$$\underbrace{P(c \mid x)}_{\text{Posterior Probability}} = \frac{\overbrace{P(x \mid c)}^{\text{Likelihood}}\,\overbrace{P(c)}^{\text{Class Prior Probability}}}{\underbrace{P(x)}_{\text{Predictor Prior Probability}}}$$

$$P(c \mid \mathbf{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

- P(c|x) is the posterior probability of class (c, target) given predictor (x, attributes).
- P(c) is the prior probability of class.
- P(x|c) is the likelihood which is the probability of predictor given class.
- P(x) is the prior probability of predictor.

Naive Bayes Algorithm

- Naive Bayes Algorithm - example

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Naive Bayes Algorithm

- Naive Bayes Algorithm - example
  1. Convert the data set into a frequency table
  2. Create Likelihood table by finding the probabilities
  3. Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

Naive Bayes Algorithm

- Naive Bayes Algorithm - example
  1. Convert the data set into a frequency table

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Naive Bayes Algorithm

- Naive Bayes Algorithm - example
  1. Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table

Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table

Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

Naive Bayes Algorithm

- Naive Bayes Algorithm - example
- Problem: Players will play if weather is sunny. Is this statement is correct?

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

- P(Yes | Sunny) = (P(Sunny | Yes) * P(Yes)) / P (Sunny)
- P (Sunny | Yes) = 3/9 = 0.33
- P(Sunny) = 5/14 = 0.36
- P(Yes)= 9/14 = 0.64
- P (Yes | Sunny) = 0.33 * 0.64 / 0.36 = 0.60, which has higher probability.

Naive Bayes Algorithm

- Naive Bayes Algorithm - example

Outlook			Temperature			Humidity			Windy			Play	
	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								

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Naive Bayes Algorithm

- Naive Bayes Algorithm - example

Outlook			Temperature			Humidity			Windy			Play	
	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								

- Yeni veri

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?

$$P(C_i|X) = P(X|C_i)\times P(C_i) = \prod_{k=1}^{n} P(x_k|C_i)\times P(C_i)$$

İki Sınıf için olasılık:

P("yes"|X) = 2/9 × 3/9 × 3/9 × 3/9 × 9/14 = 0.0053

P("no"|X) = 3/5 × 1/5 × 4/5 × 3/5 × 5/14 = 0.0206

Normalize edilmiş olasılıklar:

P("yes") = 0.0053 / (0.0053 + 0.0206) = 0.205

P("no") = 0.0206 / (0.0053 + 0.0206) = 0.795

K-Nearest Neighbor Algorithm

- K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions).
- the task of classifying a new object among a number of known examples

K-Nearest Neighbor Algorithm

- Distance Functions:

Distance functions

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

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

Minkowski $$\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$$

K-Nearest Neighbor Algorithm

- K-NN Algorithm - Example:
  1. Determine parameter K=number of nearest neighbors
  2. Calculate the distance between the query-intance and all the training samples
  3. Sort the distance and determine nearest neighbors based on the K-th mininmum distance
  4. Gather the category Y of the nearest neighbors
  5. Use simple majority of the category of nearest neighbors as the prediction value of the query instance

K-Nearest Neighbor Algorithm

- K-NN Algorithm - Example:

X1=acid	X2=strength	Y=Classification
7	7	Bad
7	4	Bad
3	4	Good
1	4	Good

K-Nearest Neighbor Algorithm

- K-NN Algorithm - Example:

X1=acid	X2=strength	Y=Classification
7	7	Bad
7	4	Bad
3	4	Good
1	4	Good
3	7	?

K-Nearest Neighbor Algorithm

- K-NN Algorithm - Example:
- Determine parameter K=number of nearest neighbors (suppose K=3)

X1=acid	X2=strength	Y=Classification
7	7	Bad
7	4	Bad
3	4	Good
1	4	Good
3	7	?

K-Nearest Neighbor Algorithm

- K-NN Algorithm - Example:
- Calculate the distance between the query-intance and all the training samples

X1=acid	X2=strength	Distance(3,7)
7	7	$\sqrt{(7-3)^2+(7-7)^2}$=16
7	4	$\sqrt{(7-3)^2+(4-7)^2}$=25
3	4	$\sqrt{(3-3)^2+(4-7)^2}$=9
1	4	$\sqrt{(1-3)^2+(4-7)^2}$=13

K-Nearest Neighbor Algorithm

- K-NN Algorithm - Example:
- Sort the distance and determine nearest neighbors based on the K-th mininmum distance

X1=acid	X2=strength	Distance(3,7)	Rank	is in 3-NI
7	7	$\sqrt{(7-3)^2+(7-7)^2}$=16	3	Yes
7	4	$\sqrt{(7-3)^2+(4-7)^2}$=25	4	No
3	4	$\sqrt{(3-3)^2+(4-7)^2}$=9	1	Yes
1	4	$\sqrt{(1-3)^2+(4-7)^2}$=13	2	Yes

K-Nearest Neighbor Algorithm

- K-NN Algorithm - Example:
- Gather the category Y of the nearest neighbors

X1	X2	Distance(3,7)	Rank	is in 3-NN?	Y
7	7	$\sqrt{(7-3)^2+(7-7)^2}$=16	3	Yes	Bad
7	4	$\sqrt{(7-3)^2+(4-7)^2}$=25	4	No	-
3	4	$\sqrt{(3-3)^2+(4-7)^2}$=9	1	Yes	Good
1	4	$\sqrt{(1-3)^2+(4-7)^2}$=13	2	Yes	Good

K-Nearest Neighbor Algorithm

- K-NN Algorithm - Example:
- Use simple majority of the category of nearest neighbors as the prediction value of the query instance
- We have two Good, one Bad. So **Class = Good**

Clustering

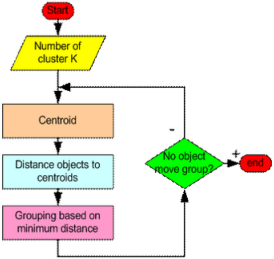
- Clustering is the process of partitioning a group of data points into a small number of clusters.
- is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters)

K-Means Algorithm

- Clustering is the process of partitioning a group of data points into a small number of clusters.
- The Lloyd's algorithm, mostly known as k-means algorithm, is used to solve the k-means clustering problem
- In the begining, we determine number of cluster K
- And algorithm works as follows:
  1. Determine the centroid coordinate
  2. Determine the distance of each object to the centroids
  3. Group the object based on minimum distance

K-Means Algorithm

- K-Means Algorithm Process



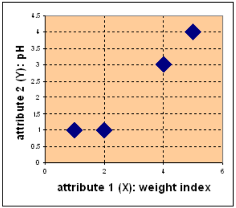
K-Means Algorithm

- K-Means Algorithm - Example:
- $K = 2$

Object	X	Y
Medicine A	1	1
Medicine B	2	1
Medicine C	4	3
Medicine D	5	4

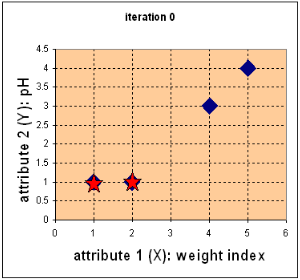
K-Means Algorithm

- K-Means Algorithm Process



K-Means Algorithm

- 1. Initial value of centroids: Suppose we use medicine A and B as first centroids.
- Let C1 and C2 denote the coordinate of the centroid, then C1=(1,1) and C2=(2,1)



K-Means Algorithm

- 2. Objects-Centroids distance : we calculate the distance between cluster centroid to each object.
- Let us use Euclidean distance, then we have distance matric at iteration 0

$$\begin{array}{cccc} A & B & C & D \\ \end{array}$$
$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \begin{array}{c} X \\ Y \end{array}$$

$$\mathbf{D}^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \begin{array}{ll} \mathbf{c}_1 = (1,1) & group-1 \\ \mathbf{c}_2 = (2,1) & group-2 \end{array}$$

- Exm: A=(1,1)
- C1=(1,1) is $\sqrt{(1-1)^2+(1-1)^2} = 0$
- C2=(2,1) is $\sqrt{(1-2)^2+(1-1)^2} = 1$
- Exm: C=(4,3)
- C1=(1,1) is $\sqrt{(4-1)^2+(3-1)^2} = 3.61$
- C2=(2,1) is $\sqrt{(4-2)^2+(3-1)^2} = 2.83$

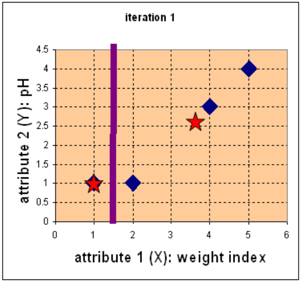
K-Means Algorithm

- 3. Objects-Clustering: We assign each object based on the minimum distance.
- A is assigned to Group 1, B to Group 2, C is Group2 and D is Group 2
- Let us use Euclidean distance, then we have distance matric at iteration 0

$$\mathbf{G}^0 = \begin{array}{c} \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{array}\right] \\ \begin{array}{cccc} A & B & C & D \end{array} \end{array} \begin{array}{l} group-1 \\ group-2 \\ \end{array}$$

K-Means Algorithm

- 4. Iteration-1, determine centroids: Knowing the members of each group
- we compute the new centroid of each group based on these new membership
- C1 = (1,1)
- C2 = ((2+4+5)/3, (1+3+4)/3) = (11/3 , 8/3)



K-Means Algorithm

- 5. Iteration-1, Objects-Centroids distance : The next step is to compute the distance of all abjects to the new centroids.

$$
\begin{array}{ll}
\mathbf{D}^1 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix} & \begin{array}{ll} \mathbf{c}_1 = (1,1) & group-1 \\ \mathbf{c}_2 = (\frac{11}{3}, \frac{8}{3}) & group-2 \end{array} \\
\begin{array}{cccc} \quad A & \quad B & \quad C & \quad D \end{array} & \\
\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} & \begin{array}{l} X \\ Y \end{array}
\end{array}
$$

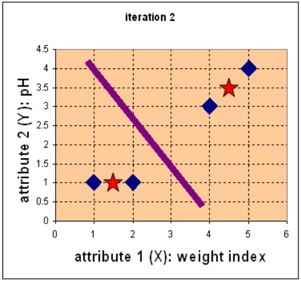
K-Means Algorithm

- 6. Iteration-1, Objects clustering: Similar to step 3, we assign each object based on the minimum distance.

$$\mathbf{G}^1 = \begin{array}{c} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{array}{l} group-1 \\ group-2 \end{array} \\ \begin{array}{cccc} A & B & C & D \end{array} \end{array}$$

K-Means Algorithm

- 7. Iteration-2, determine centroids : Repeat step 4
- C1 = ((1+2)/2, (1+1)/2)= (3/2,1)
- C2 = ((4+5)/2, (3+4)/2) = (9/2, 7/2)



K-Means Algorithm

- 8. Iteration-2, Objects-Centroids distance : Repeat 2

$$\mathbf{D}^2 = \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{bmatrix} \quad \begin{array}{ll} \mathbf{c}_1 = (1\tfrac{1}{2}, 1) & group-1 \\ \mathbf{c}_2 = (4\tfrac{1}{2}, 3\tfrac{1}{2}) & group-2 \end{array}$$

$$\begin{array}{cccc} A & B & C & D \end{array}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \begin{array}{l} X \\ Y \end{array}$$

K-Means Algorithm

- 9. Iteration-2, Objects clustering: Similar to step 3, we assign each object based on the minimum distance.

$$\mathbf{G}^2 = \begin{array}{c} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{array}{l} group-1 \\ group-2 \end{array} \\ \begin{array}{cccc} A & B & C & D \end{array} \end{array}$$

K-Means Algorithm

- We obtain result that $G^2 = G^1$
- K-means reached its stability and no more iteration is needed
- We get final grouping

$$\begin{array}{cc} \mathbf{G}^2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} & \begin{array}{l} group-1 \\ group-2 \end{array} \\ \quad\;\; \begin{array}{cccc} A & B & C & D \end{array} & \end{array}$$

Object	X	Y	Group
Medicine A	1	1	1
Medicine B	2	1	1
Medicine C	4	3	2
Medicine D	5	4	2

References

- Speech and Language Processing (3rd ed. draft) by D. Jurafsky & J. H. Martin (web.stanford.edu)
- http:people.revoledu.comkarditutorialKNNKNN_Numerical-example.html
- https://www.analyticsvidhya.com/blog/2015/09/naive-bayes-explained/
- www.cs.princeton.edu/šchapire/.../picasso-minicourse.pdf
- http://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/