

Machine Learning for Finance

Prepared by Dr. Diaa Salama

FNT633

K Means Clustering
Apriori algorithm

Lecture 7



Machine Learning for Finance

Lecture 7

FNT633

K Means Clustering
Apriori algorithm

Prepared by
Dr. Diao Salama

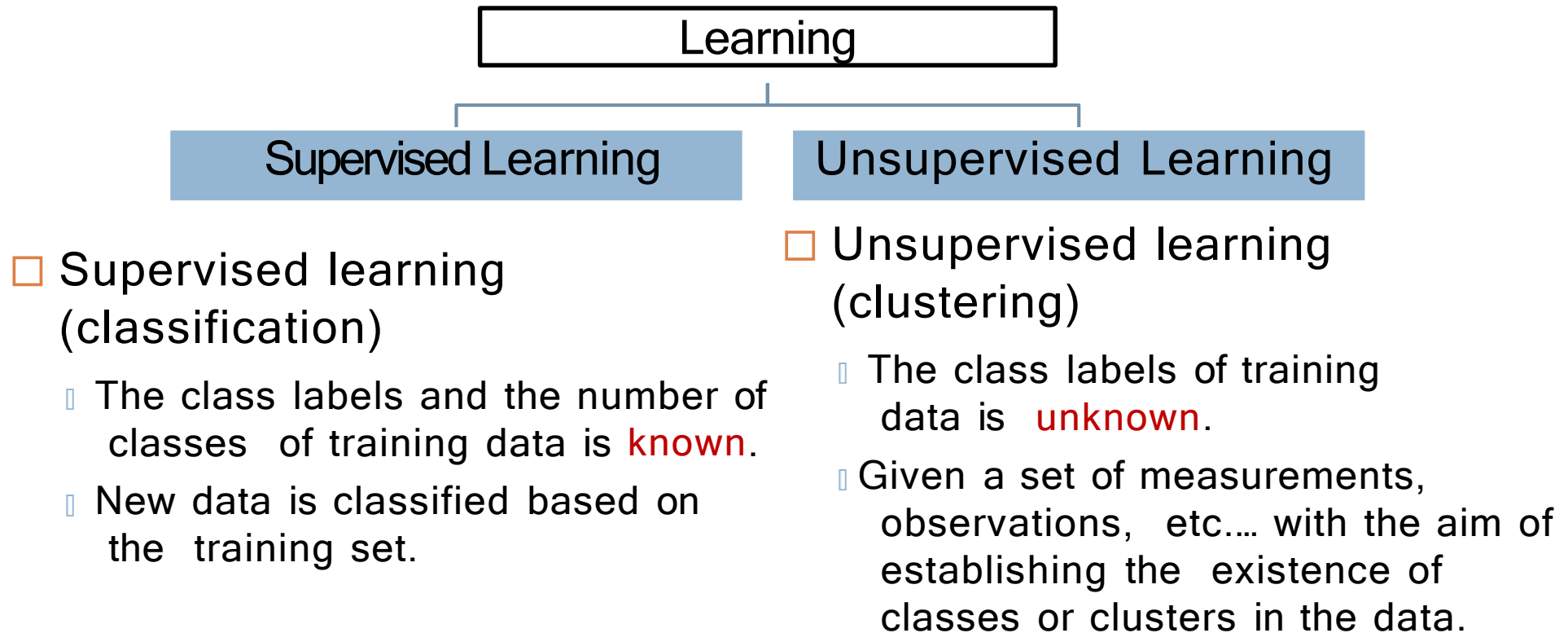
www.midocean.ae

- Clustering : **K means** Clustering
- Association Rules : **Apriori**

Supervised learning vs. unsupervised learning

- **Supervised learning:** discover patterns in the data that relate data attributes with a target (class) attribute.
 - These patterns are then utilized to predict the values of the target attribute in future data instances.
- **Unsupervised learning:** The data have no target attribute.
 - We want to explore the data to find some intrinsic structures in them.

- Supervised learning (classification)
 - Supervision: The training data (observations, measurements, etc.) are accompanied by **labels** indicating the class of the observations
 - New data is classified based on the training set
- Unsupervised learning (clustering)
 - The class labels of training data is unknown
 - Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data



What Kind of Problem do I Need to Solve? How do I Solve it?

The Problem to Solve	The Category of Techniques	Covered in this Course
I want to group items by similarity. I want to find structure (commonalities) in the data	Clustering	K-means clustering
I want to discover relationships between actions or items	Association Rules	Apriori
I want to determine the relationship between the outcome and the input variables	Regression	Linear Regression Logistic Regression
I want to assign (known) labels to objects	Classification	Naïve Bayes Decision Trees
I want to find the structure in a temporal process I want to forecast the behavior of a temporal process	Time Series Analysis	ACF, PACF, ARIMA
I want to analyze my text data	Text Analysis	Regular expressions, Document representation (Bag of Words).

K Means Clustering

- Used for
 - Classification
 - Regression

- K nearest neighbors measured by a distance function.
- These functions are valid only with continuous variables.

Euclidean

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

Manhattan

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

Minkowski

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

- Consider the following data concerning credit default. Age and Loan are two numerical variables (predictors) and Default is the target.

Age	Loan	Default
25	\$40,000	N
35	\$60,000	N
45	\$80,000	N
20	\$20,000	N
35	\$120,000	N
52	\$18,000	N
23	\$95,000	Y
40	\$62,000	Y
60	\$100,000	Y
48	\$220,000	Y
33	\$150,000	Y

- Know we want to classify unknown case ((Age=48 and Loan=\$142,000) using Euclidean distance.)

!!!!

- Using Euclidean distance: E.g. distance between (48, \$142k) and (33, 150k)
- □ $\text{Sqrt}[(48-33)^2 + (142000-150000)^2] = 8000.01$
- So for all the dataset

Age	Loan	Default	Distance	
25	\$40,000	N	102000	
35	\$60,000	N	82000	
45	\$80,000	N	62000	
20	\$20,000	N	122000	
35	\$120,000	N	22000	2
52	\$18,000	N	124000	
23	\$95,000	Y	47000	
40	\$62,000	Y	80000	
60	\$100,000	Y	42000	3
48	\$220,000	Y	78000	
33	\$150,000	Y	8000	1
48	\$142,000	?		

Euclidean Distance

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

- During this lesson the following topics are covered:
- Clustering – Unsupervised learning method
- K-means clustering:
- Use cases
- The algorithm
- Determining the optimum value for K
- Diagnostics to evaluate the effectiveness of the method
- Reasons to Choose (+) and Cautions (-) of the method

Clustering

- Clustering is a technique for finding **similarity groups** in data, called **clusters**.
I.e.,
 - it groups data instances that are similar to (near) each other in one cluster and data instances that are very different (far away) from each other into different clusters.
- Clustering is often called an **unsupervised learning** task as no class values denoting an a priori grouping of the data instances are given, which is the case in supervised learning.
- Due to historical reasons, clustering is often considered synonymous with unsupervised learning.
 - In fact, association rule mining is also unsupervised
- This chapter focuses on clustering.

What is clustering for?

- Let us see some real-life examples
- **Example 1:** groups people of similar sizes together to make “small”, “medium” and “large” T-Shirts.
 - Tailor-made for each person: too expensive
 - One-size-fits-all: does not fit all.
- **Example 2:** In marketing, segment customers according to their similarities
 - To do targeted marketing.

What is clustering for? (cont...)

- **Example 3:** Given a collection of text documents, we want to organize them according to their content similarities,
 - To produce a topic hierarchy
- **In fact, clustering is one of the most utilized data mining techniques.**
 - It has a long history, and used in almost every field, e.g., medicine, psychology, botany, sociology, biology, archeology, marketing, insurance, libraries, etc.
 - In recent years, due to the rapid increase of online documents, text clustering becomes important.

- How do I group these documents by topic?
- How do I group my customers by purchase patterns?
- Sort items into groups by similarity:
 - Items in a cluster are more similar to each other than they are to items in other clusters.
 - Need to detail the properties that characterize “similarity”
 - Or of distance, the "inverse" of similarity
- Not a predictive method; finds similarities, relationships
- Our Example: **K-means Clustering**

What is clustering

- Aspects of clustering

A clustering algorithm

Partitional clustering

Hierarchical clustering

...

A distance (similarity, or dissimilarity) function

Clustering quality

Inter-clusters distance $\Rightarrow$ maximized

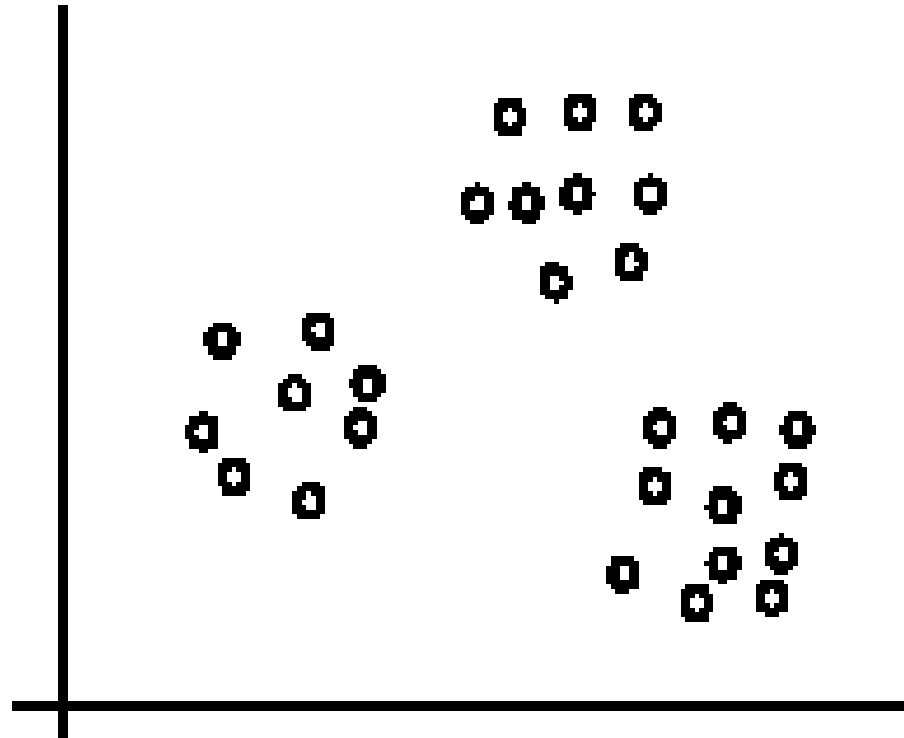
Intra-clusters distance $\Rightarrow$ minimized

The **quality** of a clustering result depends on the algorithm, the distance function, and the application.

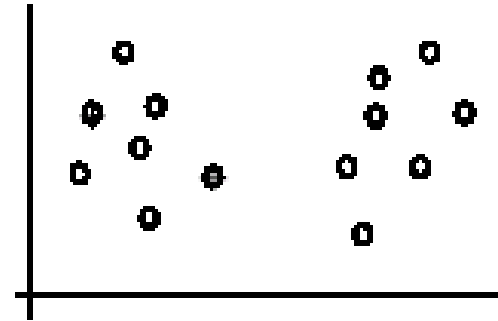
- Given k , the k-means algorithm works as follows:
 1. Randomly choose k data points (**seeds**) to be the initial **centroids**, cluster centers
 2. Assign each data point to the closest **centroid**
 3. Re-compute the **centroids** using the current cluster memberships.
 4. If a convergence criterion is not met, go to **2**).

An illustration

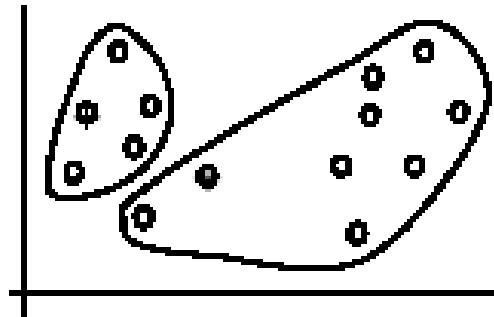
- The data set has three natural groups of data points, i.e., 3 natural clusters.



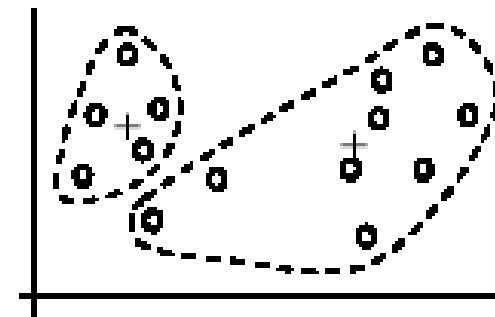
An example



(A). Random selection of k centers

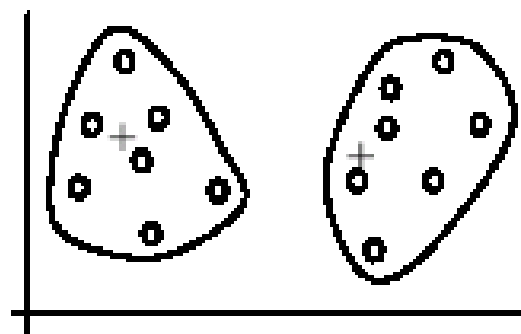


Iteration 1: (B). Cluster assignment

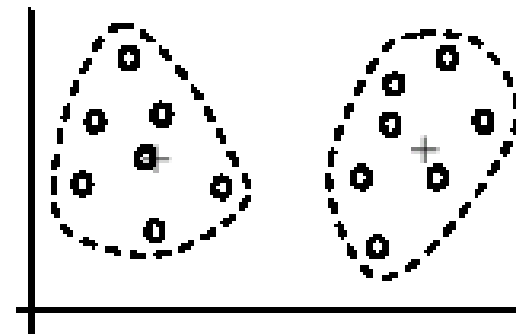


(C). Re-compute centroids

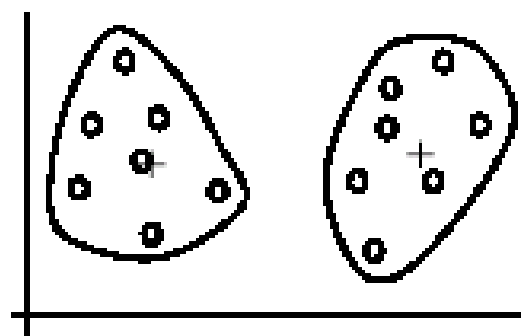
An example (cont ...)



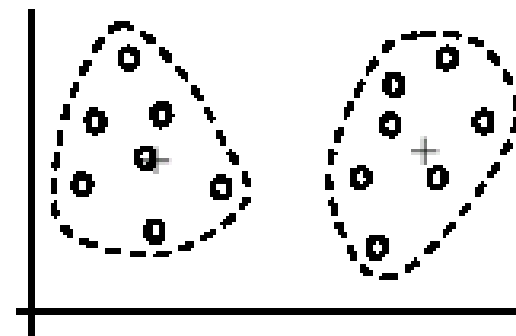
Iteration 2: (D). Cluster assignment



(E). Re-compute centroids



Iteration 3: (F). Cluster assignment



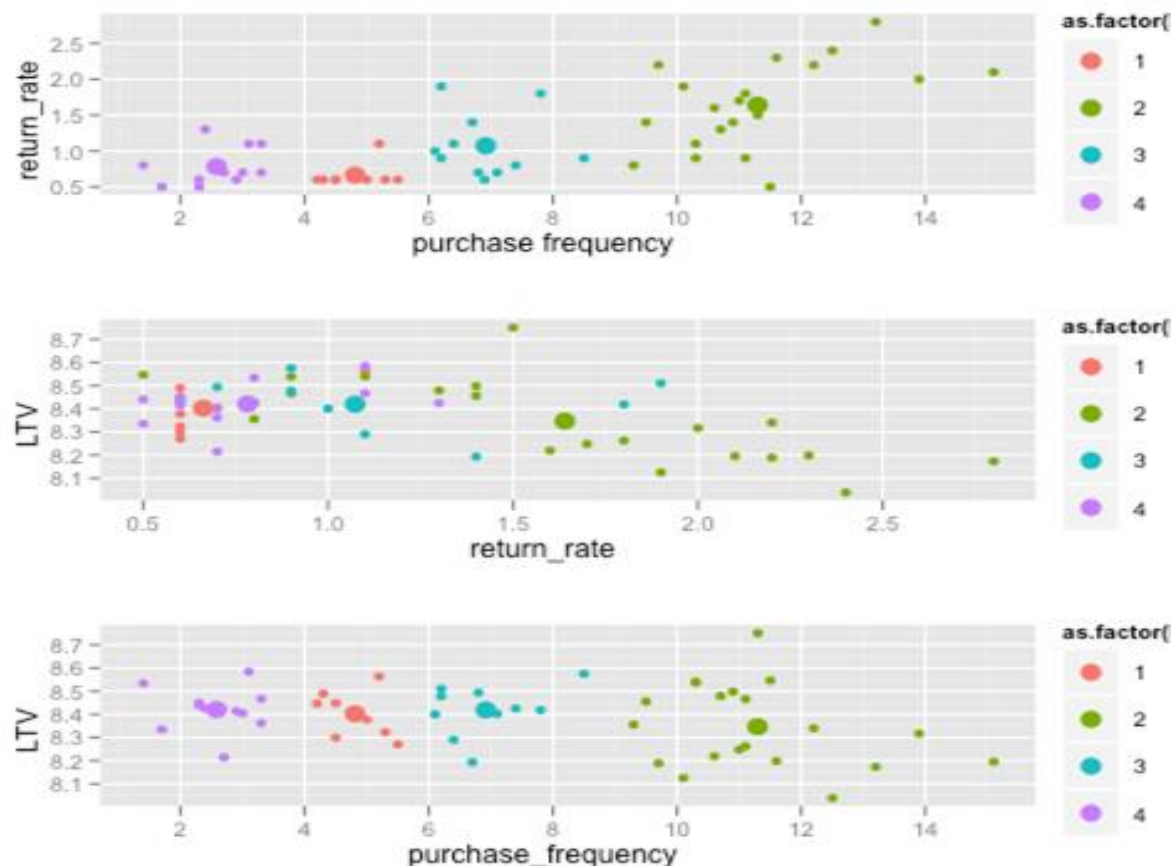
(G). Re-compute centroids

K-Means Clustering - What is it?

- Used for clustering numerical data, usually a set of measurements about objects of interest.
- **Input:** numerical. There must be a distance metric defined over the variable space.
- Euclidian distance
- **Output:** The centers of each discovered cluster, and the assignment of each input datum to a cluster.
- Centroid

- Often an exploratory technique:
 - Discover structure in the data
 - Summarize the properties of each cluster
- Sometimes a prelude to classification:
 - "Discovering the classes"
- Examples
 - The height, weight and average lifespan of animals
 - Household income, yearly purchase amount in dollars, number of household members of customer households

Use-Case Example – On-line Retailer



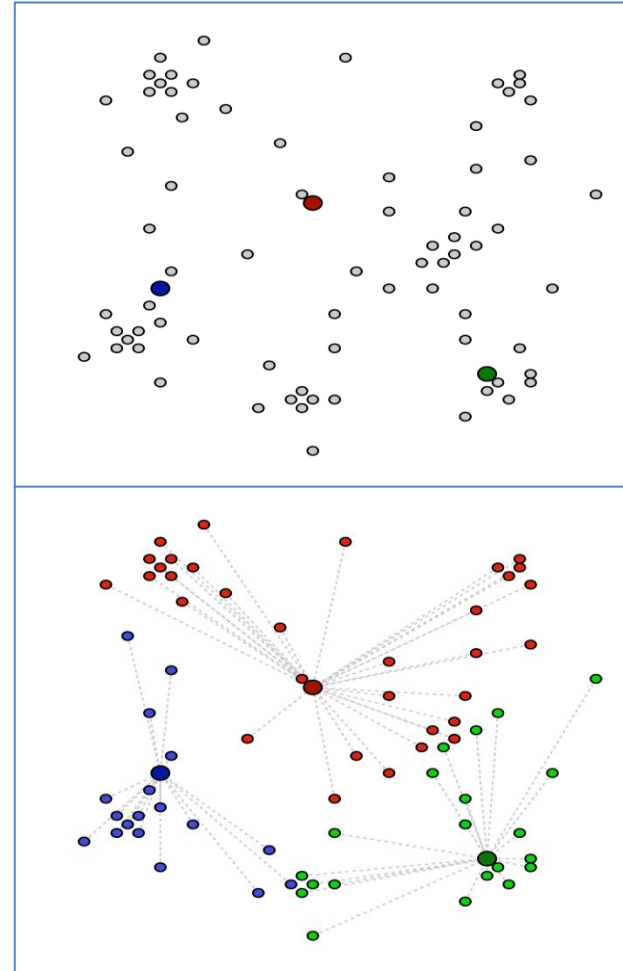
LTV – Lifetime Customer Value

The Algorithm

1- Choose K ; then select K random "centroids"

In our example, $K=3$

2. Assign records to the cluster with the closest centroid



The Algorithm

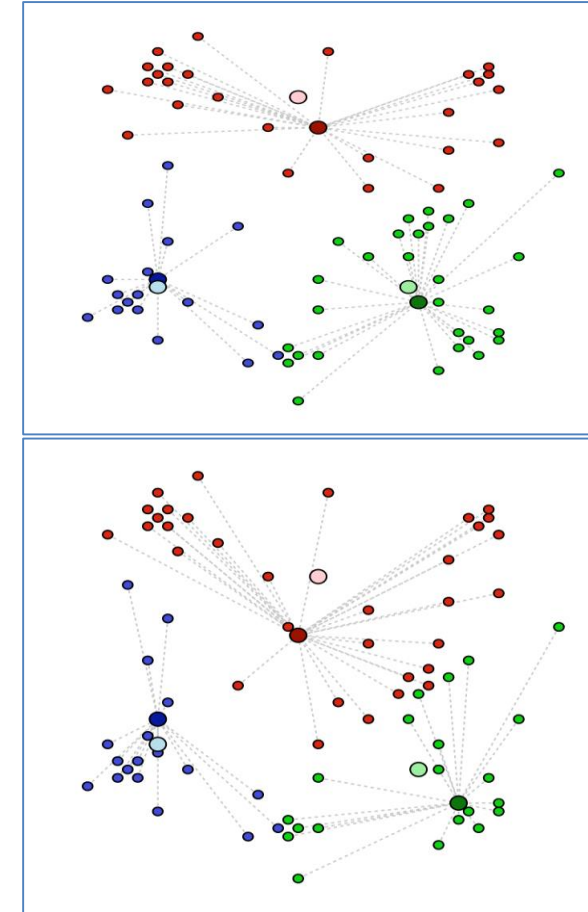
- 3- Recalculate the resulting centroids

Centroid: the mean value of all the records in the cluster

4. Repeat steps 2 & 3 until record assignments no longer change

Model Output:

- The final cluster centers
- The final cluster assignments of the training data



- Heuristic: find the "elbow" of the within-sum-of-squares (wss) plot as a function of K.

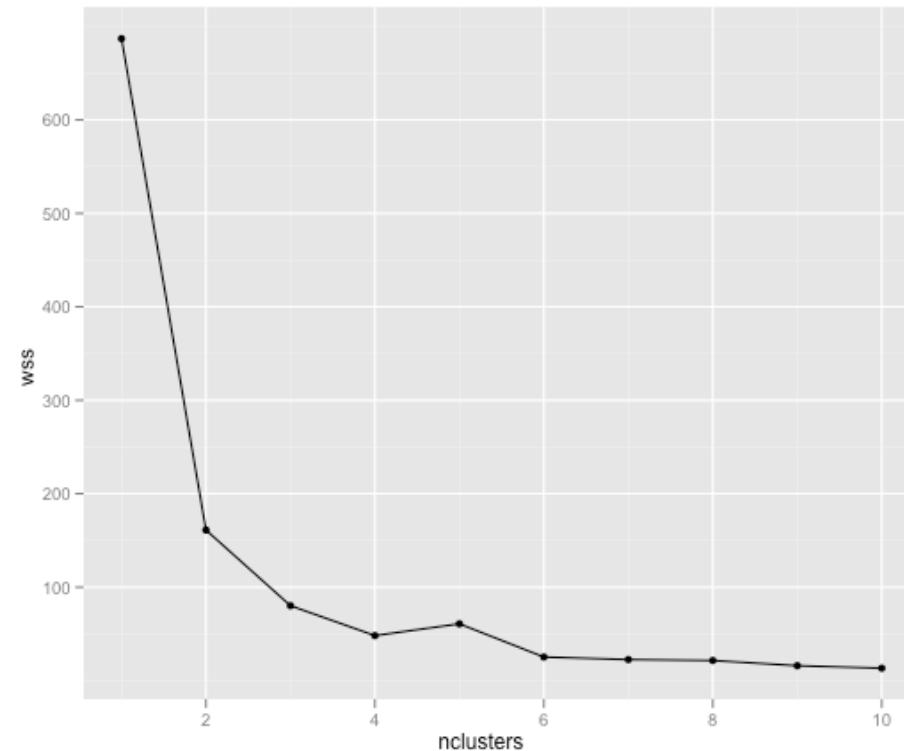
$$WSS = \sum_{i=1}^k \sum_{j=1}^{n_i} |x_{ij} - c_i|^2$$

K: # of clusters

n_i : # points in i^{th} cluster

c_i : centroid of i^{th} cluster

x_{ij} : j^{th} point of i^{th} cluster



Diagnostics – Evaluating the Model

- Do the clusters look separated in at least some of the plots when you do pair-wise plots of the clusters?
 - Pair-wise plots can be used when there are not many variables
- Do you have any clusters with few data points?
 - Try decreasing the value of K
- Are there splits on variables that you would expect, but don't see?
 - Try increasing the value K
- Do any of the centroids seem too close to each other?
 - Try decreasing the value of K

Example 1

Exercise 1. K-means clustering

Use the k-means algorithm and Euclidean distance to cluster the following 8 examples into 3 clusters:

A1=(2,10), A2=(2,5), A3=(8,4), A4=(5,8), A5=(7,5), A6=(6,4), A7=(1,2), A8=(4,9).

The distance matrix based on the Euclidean distance is given below:

	A1	A2	A3	A4	A5	A6	A7	A8
A1	0	$\sqrt{25}$	$\sqrt{36}$	$\sqrt{13}$	$\sqrt{50}$	$\sqrt{52}$	$\sqrt{65}$	$\sqrt{5}$
A2		0	$\sqrt{37}$	$\sqrt{18}$	$\sqrt{25}$	$\sqrt{17}$	$\sqrt{10}$	$\sqrt{20}$
A3			0	$\sqrt{25}$	$\sqrt{2}$	$\sqrt{2}$	$\sqrt{53}$	$\sqrt{41}$
A4				0	$\sqrt{13}$	$\sqrt{17}$	$\sqrt{52}$	$\sqrt{2}$
A5					0	$\sqrt{2}$	$\sqrt{45}$	$\sqrt{25}$
A6						0	$\sqrt{29}$	$\sqrt{29}$
A7							0	$\sqrt{58}$
A8								0

Suppose that the initial seeds (centers of each cluster) are A1, A4 and A7. Run the k-means algorithm for 1 epoch only. At the end of this epoch show:

- The new clusters (i.e. the examples belonging to each cluster)
- The centers of the new clusters
- Draw a 10 by 10 space with all the 8 points and show the clusters after the first epoch and the new centroids.
- How many more iterations are needed to converge? Draw the result for each epoch.

Example 1 continue

Solution:

a)

$d(a,b)$ denotes the Euclidean distance between a and b . It is obtained directly from the distance matrix or calculated as follows: $d(a,b) = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}$

seed1=A1=(2,10), seed2=A4=(5,8), seed3=A7=(1,2)

epoch1 – start:

A1:

$d(A1, \text{seed1}) = 0$ as A1 is seed1

$d(A1, \text{seed2}) = \sqrt{13} > 0$

$d(A1, \text{seed3}) = \sqrt{65} > 0$

→ A1 ∈ cluster1

A3:

$d(A3, \text{seed1}) = \sqrt{36} = 6$

$d(A3, \text{seed2}) = \sqrt{25} = 5$ ← smaller

$d(A3, \text{seed3}) = \sqrt{53} = 7.28$

→ A3 ∈ cluster2

A5:

$d(A5, \text{seed1}) = \sqrt{50} = 7.07$

A2:

$d(A2, \text{seed1}) = \sqrt{25} = 5$

$d(A2, \text{seed2}) = \sqrt{18} = 4.24$

$d(A2, \text{seed3}) = \sqrt{10} = 3.16$ ← smaller

→ A2 ∈ cluster3

A4:

$d(A4, \text{seed1}) = \sqrt{13}$

$d(A4, \text{seed2}) = 0$ as A4 is seed2

$d(A4, \text{seed3}) = \sqrt{52} > 0$

→ A4 ∈ cluster2

A6:

$d(A6, \text{seed1}) = \sqrt{52} = 7.21$

Example continue

$$d(A5, \text{seed2}) = \sqrt{13} = 3.60 \leftarrow \text{smaller}$$

$$d(A5, \text{seed3}) = \sqrt{45} = 6.70$$

→ A5 ∈ cluster2

A7:

$$d(A7, \text{seed1}) = \sqrt{65} > 0$$

$$d(A7, \text{seed2}) = \sqrt{52} > 0$$

$$d(A7, \text{seed3}) = 0 \text{ as } A7 \text{ is seed3}$$

→ A7 ∈ cluster3

end of epoch1

new clusters: 1: {A1}, 2: {A3, A4, A5, A6, A8}, 3: {A2, A7}

b) centers of the new clusters:

$$C1 = (2, 10), C2 = ((8+5+7+6+4)/5, (4+8+5+4+9)/5) = (6, 6), C3 = ((2+1)/2, (5+2)/2) = (1.5, 3.5)$$

$$d(A6, \text{seed2}) = \sqrt{17} = 4.12 \leftarrow \text{smaller}$$

$$d(A6, \text{seed3}) = \sqrt{29} = 5.38$$

→ A6 ∈ cluster2

A8:

$$d(A8, \text{seed1}) = \sqrt{5}$$

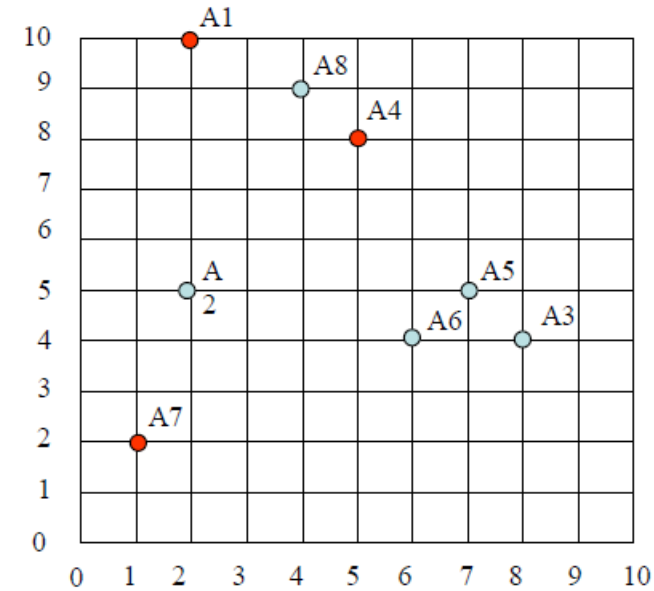
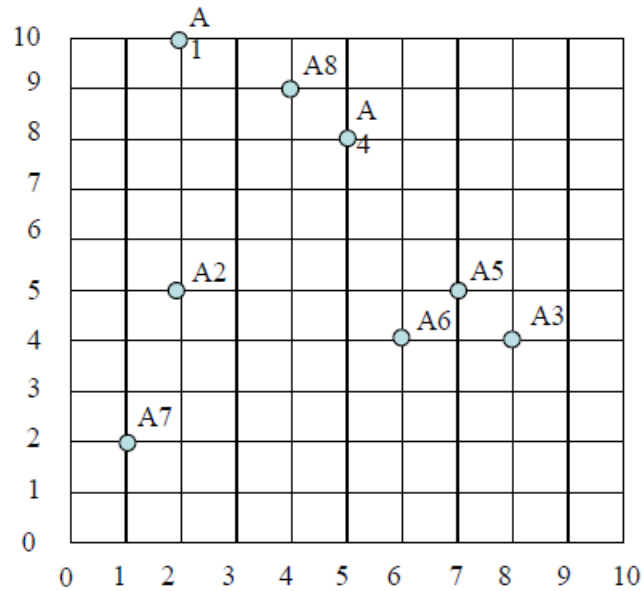
$$d(A8, \text{seed2}) = \sqrt{2} \leftarrow \text{smaller}$$

$$d(A8, \text{seed3}) = \sqrt{58}$$

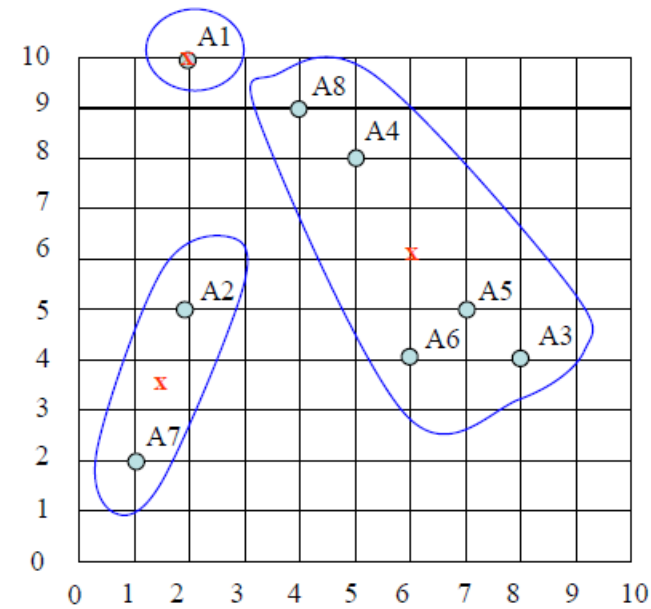
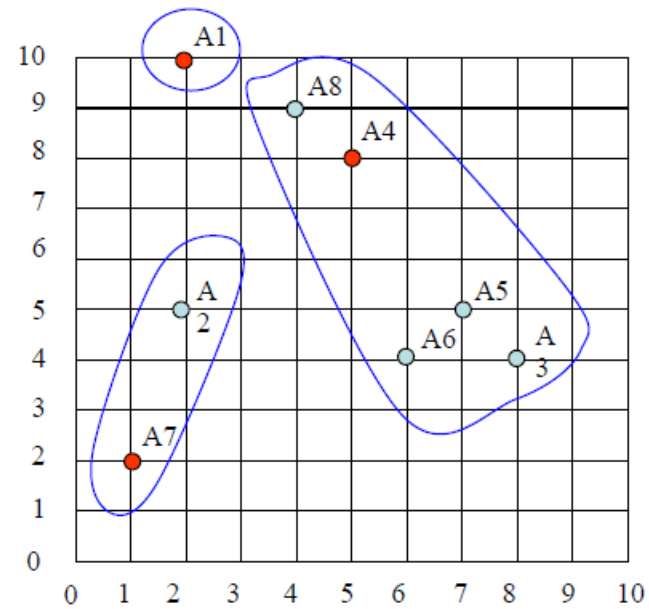
→ A8 ∈ cluster2

Example 1 continue

c)



Example 1 continue



Example 2 continue

d)

We would need two more epochs. After the 2nd epoch the results would be:

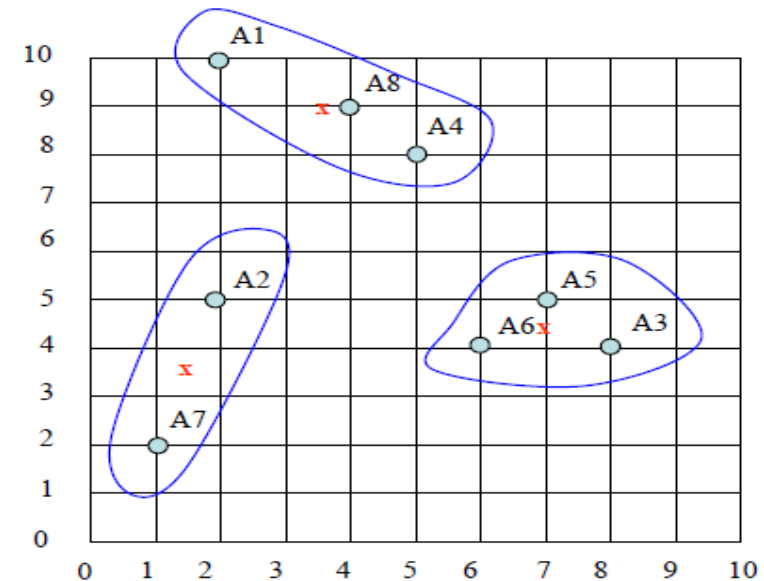
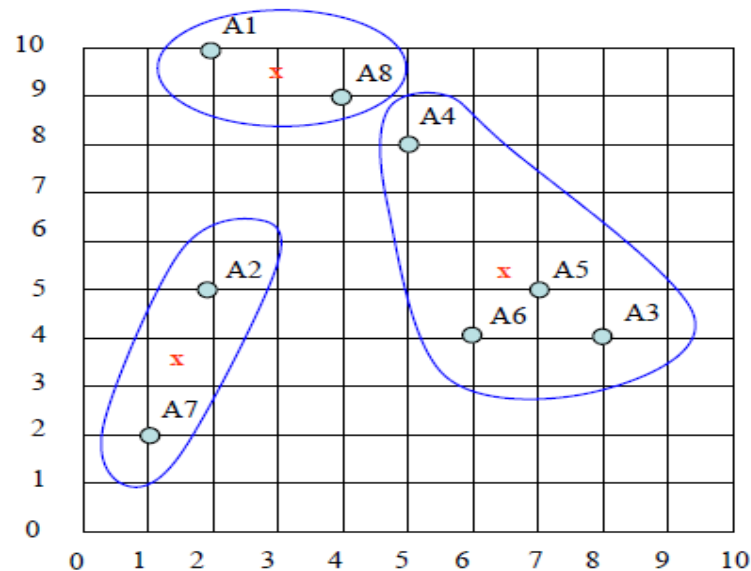
1: {A1, A8}, 2: {A3, A4, A5, A6}, 3: {A2, A7}

with centers $C1=(3, 9.5)$, $C2=(6.5, 5.25)$ and $C3=(1.5, 3.5)$.

After the 3rd epoch, the results would be:

1: {A1, A4, A8}, 2: {A3, A5, A6}, 3: {A2, A7}

with centers $C1=(3.66, 9)$, $C2=(7, 4.33)$ and $C3=(1.5, 3.5)$.



K-Means Clustering - Reasons to Choose (+) and Cautions (-)

Reasons to Choose (+)	Cautions (-)
Easy to implement	Doesn't handle categorical variables
Easy to assign new data to existing clusters Which is the nearest cluster center?	Sensitive to initialization (first guess)
Concise output Coordinates the K cluster centers	Variables should all be measured on similar or compatible scales Not scale-invariant!
	K (the number of clusters) must be known or decided a priori Wrong guess: possibly poor results
	Tends to produce "round" equi-sized clusters. Not always desirable

Check Your Knowledge

- Why do we consider K-means clustering as a unsupervised machine learning algorithm?
- How do you use “pair-wise” plots to evaluate the effectiveness of the clustering?
- Detail the four steps in the K-means clustering algorithm.
- How do we use WSS to pick the value of K?
- What is the most common measure of distance used with K-means clustering algorithms?
- The attributes of a data set are “purchase decision (Yes/No), Gender (M/F), income group (<10K, 10-50K, >50K). Can you use K-means to cluster this data set?

K-means Clustering - Summary

- During this lesson the following topics were covered:
- Clustering – Unsupervised learning method
- What is K-means clustering
- Use cases with K-means clustering
- The K-means clustering algorithm
- Determining the optimum value for K
- Diagnostics to evaluate the effectiveness of K-means clustering
- Reasons to Choose (+) and Cautions (-) of K-means clustering

Association Rules :

Apriori algorithm

Association Rules

- During this lesson the following topics are covered:
 - Association Rules mining
 - Apriori Algorithm
 - Prominent use cases of Association Rules
 - Support and Confidence parameters
 - Lift and Leverage
 - Diagnostics to evaluate the effectiveness of rules generated
 - Reasons to Choose (+) and Cautions (-) of the Apriori algorithm

- Which of my products tend to be purchased together?
- What do other people like this person tend to like/buy/watch?
- Discover "interesting" relationships among variables in a large database
 - Rules of the form "When X observed, Y also observed"
 - The definition of "interesting" varies with the algorithm used for discovery
- Not a predictive method; finds similarities, relationships

Week

08

'Basket data'



- A very common type of data; often also called transaction data. Next slide shows example transaction database, where each record represents a transaction between (usually) a customer and a shop. Each record in a supermarket's transaction DB, for example, corresponds to a basket of specific items.

Week

08

ID apples, beer, cheese, dates, eggs, fish, glue, honey, ice-cream

1	1	1		1			1	1	
2			1	1	1				
3		1	1			1			
4		1				1			1
5					1		1		
6						1			1
7	1			1				1	
8						1			1
9			1		1				
10		1					1		
11					1		1		
12	1								
13			1			1			
14			1			1			
15								1	1
16				1					
17	1					1			
18	1	1	1	1				1	
19	1	1		1			1	1	
20					1				

- example transaction DB has 20 records of supermarket transactions, from a supermarket that only sells 9 things
- One month in a large supermarket with five stores spread around a reasonably sized city might easily yield a DB of 20,000,000 baskets, each containing a set of products from a pool of around 1,000

Association Rules - Apriori

- Specifically designed for mining over transactions in databases
- **Used over itemsets:** sets of discrete variables that are linked:
 - Retail items that are purchased together
 - A set of tasks done in one day
 - A set of links clicked on by one user in a single session
- **Our Example: Apriori**

Apriori Algorithm - What is it? Support

- Earliest of the association rule algorithms
- Frequent itemset: a set of items L that appears together "often enough":
 - Formally: meets a *minimum support* criterion
 - **Support**: the % of transactions that contain L
- Apriori Property: Any subset of a frequent itemset is also frequent
 - It has at least the support of its superset

Apriori Algorithm (Continued) Confidence

- Iteratively grow the frequent itemsets from size 1 to size K (or until we run out of support).
 - Apriori property tells us how to prune the search space
- Frequent itemsets are used to find rules $X \rightarrow Y$ with a minimum **confidence**:
 - **Confidence**: The % of transactions that contain X, which also contain Y
- **Output**: The set of all rules $X \rightarrow Y$ with minimum support and confidence

$$\text{Lift}(X \rightarrow Y) = \frac{\text{Support}(X \wedge Y)}{\text{Support}(X) * \text{Support}(Y)}$$

$$\begin{aligned} \text{Leverage}(X \rightarrow Y) = & \text{Support}(X \wedge Y) \\ & - \text{Support}(X) * \text{Support}(Y) \end{aligned}$$

Association Rules Implementations

- Market Basket Analysis
 - People who buy milk also buy cookies 60% of the time.
- Recommender Systems
 - "People who bought what you bought also purchased....".
- Discovering web usage patterns
 - People who land on page X click on link Y 76% of the time.

Use Case Example: Credit Records

Credit ID	Attributes
1	credit_good, female_married, job_skilled, home_owner, ...
2	credit_bad, male_single, job_unskilled, renter, ...

Minimum Support:
50%

Frequent Itemset	Support
credit_good	70%
male_single	55%
job_skilled	63%
home_owner	71%
home_owner, credit_good	53%

The itemset {home_owner, credit_good} has minimum support.

The possible rules are

credit_good -> home_owner

and

home_owner -> credit_good

A Sketch of the Algorithm

- If L_k is the set of frequent k -itemsets:
 - Generate the candidate set C_{k+1} by joining L_k to itself
 - Prune out the $(k+1)$ -itemsets that don't have minimum support
Now we have L_{k+1}
- We know this catches all the frequent $(k+1)$ -itemsets by the apriori property
 - a $(k+1)$ -itemset can't be frequent if any of its subsets aren't frequent
- Continue until we reach $k_{\max}$, or run out of support
- From the union of all the L_k , find all the rules with minimum confidence

Step 1: 1-itemsets (L1)

- let min_support = 0.5
- 1000 credit records
- Scan the database
- Prune

Frequent Itemset	Count
credit_good	700
credit_bad	300
male_single	550
male_mar_or_wid	92
female	310
job_skilled	631
job_unskilled	200
home_owner	710
renter	179

Step 2: 2-itemsets (L2)

- Join L1 to itself
- Scan the database to get the counts
- Prune

Frequent Itemset	Count
credit_good, male_single	402
credit_good, job_skilled	544
credit_good, home_owner	527
male_single, job_skilled	340
male_single, home_owner	408
job_skilled, home_owner	452

Step 3: 3-itemsets

- We have run out of support.
- Candidate rules come from L2:
 - credit_good -> job_skilled
 - job_skilled -> credit_good
 - credit_good -> home_owner
 - home_owner -> credit_good

Frequent Itemset	Count
credit_good, job_skilled, home_owner	428

Finally: Find Confidence Rules

Rule	Set	Cnt	Set	Cnt	Confidence
IF credit_good THEN job_skilled	credit_good	700	credit_good AND job_skilled	544	$544/700=77\%$
IF credit_good THEN home_owner	credit_good	700	credit_good AND home_owner	527	$527/700=75\%$
IF job_skilled THEN credit_good	job_skilled	631	job_skilled AND credit_good	544	$544/631=86\%$
IF home_owner THEN credit_good	home_owner	710	home_owner AND credit_good	527	$527/710=74\%$

If we want confidence > 80%:
IF job_skilled THEN credit_good

Example 2

Q.1) For the following given Transaction Data-set, Generate Rules using Apriori Algorithm.

Consider the values as Support=50% and Confidence=75%

Transaction ID	Items Purchased
1	Bread, Cheese, Egg, Juice
2	Bread, Cheese, Juice
3	Bread, Milk, Yogurt
4	Bread, Juice, Milk
5	Cheese, Juice, Milk

Example 2

Given Support=50% and Confidence=75%

Step 1) Find Frequent Item Set and their support

Item	Frequency	Support (in %)
Bread	4	$4/5=80\%$
Cheese	3	$3/5=60\%$
Egg	1	$1/5=20\%$
Juice	4	$4/5=80\%$
Milk	3	$3/5=60\%$
Yogurt	1	$1/5=20\%$

Support (item) = Frequency of item/Number of transactions

Example 2

Step 2) Remove all the items whose support is below given minimum support.

Item	Frequency	Support (in %)
Bread	4	$4/5=80\%$
Cheese	3	$3/5=60\%$
Juice	4	$4/5=80\%$
Milk	3	$3/5=60\%$

Example 2

Step 3) Now form the two items candidate set and write their frequencies.

Items Pair	Frequency	Support (in %)
Bread, Cheese	2	$2/5=40\%$
Bread, Juice	3	$3/5=60\%$
Bread, Milk	2	$2/5=40\%$
Cheese, Juice	3	$3/5=60\%$
Cheese, Milk	1	$1/5=20\%$
Juice, Milk	2	$2/5=40\%$

Example 2

Step 4) Remove all the items whose support is below given minimum support.

Items Pair	Frequency	Support (in %)
Bread, Juice	3	$3/5=60\%$
Cheese, Juice	3	$3/5=60\%$

Step 5) Generate rules

For Rules we consider item pairs:

- a) (Bread, Juice)
Bread->Juice and Juice->Bread
- b) (Cheese, Juice)
Cheese->Juice and Juice->Cheese

Confidence (A->B) = support (AUB)/support (A)

Example 2

Therefore,

$$1. \text{ Confidence (Bread} \rightarrow \text{Juice)} = \text{support (Bread U Juice)} / \text{support (Bread)} \\ = 3/5 * 5/4 = 3/4 = 75\%$$

$$2. \text{ Confidence (Juice} \rightarrow \text{Bread)} = \text{support (Juice U Bread)} / \text{support (Juice)}$$

$$= 3/5 * 5/4 = 3/4 = 75\%$$

$$3. \text{ Confidence (Cheese} \rightarrow \text{Juice)} = \text{support (Cheese U Juice)} / \text{support (Cheese)}$$

$$= 3/5 * 5/3 = 1 = 100\%$$

$$4. \text{ Confidence (Juice} \rightarrow \text{Cheese)} = \text{support (Juice U Cheese)} / \text{support (Juice)}$$

$$= 3/5 * 5/4 = 3/4 = 75\%$$

All the above rules are good because the confidence of each rule is greater than or equal to the minimum confidence given in the problem.

Example 2

Q.2) For the following given transaction data set, generate rules using Apriori Algorithm. Consider the values as Support=22% and Confidence= 70%

Transaction ID	Items Purchased
1	I1,I2,I5
2	I2,I4
3	I2,I3
4	I1,I2,I4
5	I1,I3
6	I2,I3
7	I1,I3
8	I1,I2,I3,I5
9	I1,I2,I3

Example 2

Given minimum support=22% and confidence=70%

Step 1) Find Frequent Item Set and their support

Item	Frequency	Support (in %)
I1	6	$6/9=66\%$
I2	7	$7/9=80\%$
I3	6	$6/9=66\%$
I4	2	$2/9=22.2\%$
I5	2	$2/9=22.2\%$

Support (item) = Frequency of item/Number of transactions

Example 2

Step 2) Remove all the items whose support is below given minimum support and form the two items candidate set and write their frequencies.

Item	Frequency	Support (in %)
I1,I2	4	$4/9=44.4\%$
I1,I3	4	$4/9=44.4\%$
I1,I4	1	$1/9=11.1\%$
I1,I5	2	$2/9=22.2\%$
I2,I3	4	$4/9=44.4\%$
I2,I4	2	$2/9=22.2\%$
I2,I5	2	$2/9=22.2\%$
I3,I4	0	$0/9=0\%$
I3,I5	1	$1/9=11.1\%$
I4,I5	0	$0/9=0\%$

Step 3) Remove all the items whose support is below given minimum support and form the three items candidate set and write their frequencies

Items Pair	Frequency	Support (in %)
I1,I2,I3	2	$2/9=22.2\%$
I1,I2,I5	2	$2/9=22.2\%$

Example 2

Step 4) Generate rules and find the confidence of each rule. If the confidence is below 70% then discard that rule. Keep only the rules whose confidence is equal to or above 70%.

Confidence (A->B) = support (AUB)/support (A)

For the item set (I1,I2,I3) following rules are possible:

Rule	Confidence
(I1,I2)-> I3	$2/9 * 9/4 = 2/4 = 50\%$
(I1,I3)-> I2	$2/9 * 9/4 = 2/4 = 50\%$
(I2,I3)-> I1	$2/9 * 9/4 = 2/4 = 50\%$
I1-> (I2,I3)	$2/9 * 9/6 = 2/6 = 33.33\%$
I2-> (I1,I3)	$2/9 * 9/7 = 2/7 = 28.57\%$
I3->(I1,I2)	$2/9 * 9/6 = 2/6 = 33.33\%$

Example 2

For the item set (I1,I2,I5) following rules are possible:

Rule	Confidence
(I1,I2)-> I5	$2/9 * 9/4 = 2/4 = 50\%$
(I1,I5)-> I2	$2/9 * 9/2 = 2/2 = 100\%$

(I2,I5)-> I1	$2/9 * 9/2 = 2/2 = 100\%$
I1-> (I2,I5)	$2/9 * 9/6 = 2/6 = 33.33\%$
I2-> (I1,I5)	$2/9 * 9/7 = 2/7 = 28.57\%$
I5->(I1,I2)	$2/9 * 9/2 = 2/2 = 100\%$

From above rules only following rules are valid (because it's confidence $\geq 75\%$):

(I1,I5)-> I2

(I2,I5)-> I1

I5->(I1,I2)

- Do the rules make sense?
 - What does the domain expert say?
- Make a "test set" from hold-out data:
 - Enter some market baskets with a few items missing (selected at random). Can the rules predict the missing items?



Reasons to Choose (+)	Cautions (-)
Easy to implement	Requires many database scans
Uses a clever observation to prune the search space • Apriori property	Exponential time complexity
Easy to parallelize	

Association Rules - Summary

- During this lesson the following topics were covered:
 - Association Rules mining
 - Apriori Algorithm
 - Prominent use cases of Association Rules
 - Support and Confidence parameters
 - Lift and Leverage
 - Diagnostics to evaluate the effectiveness of rules generated
 - Reasons to Choose (+) and Cautions (-) of the Apriori algorithm



Thank You