ML: Clustering Technique

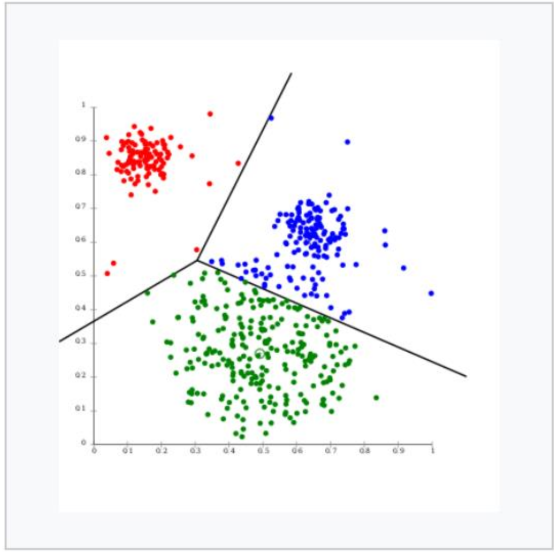
Clustering vs Classification

Supervised Classification	Unsupervised Clustering
• known number of classes	• unknown number of classes
• based on a training set	• no prior knowledge
• used to classify future observations	• used to understand (explore) data

Clustering Algorithm

- **Unsupervised learning**
  - **Iterative process to find best partitioning groups**
    - **Based on data itself**
    - **No class label is supplied for training**
- **Example clustering algorithm**
  - **Centroid-based clustering**
    - **K-mean clustering**
  - **Hierarchical clustering**
  - **Distribution-based clustering**
  - **Etc.**

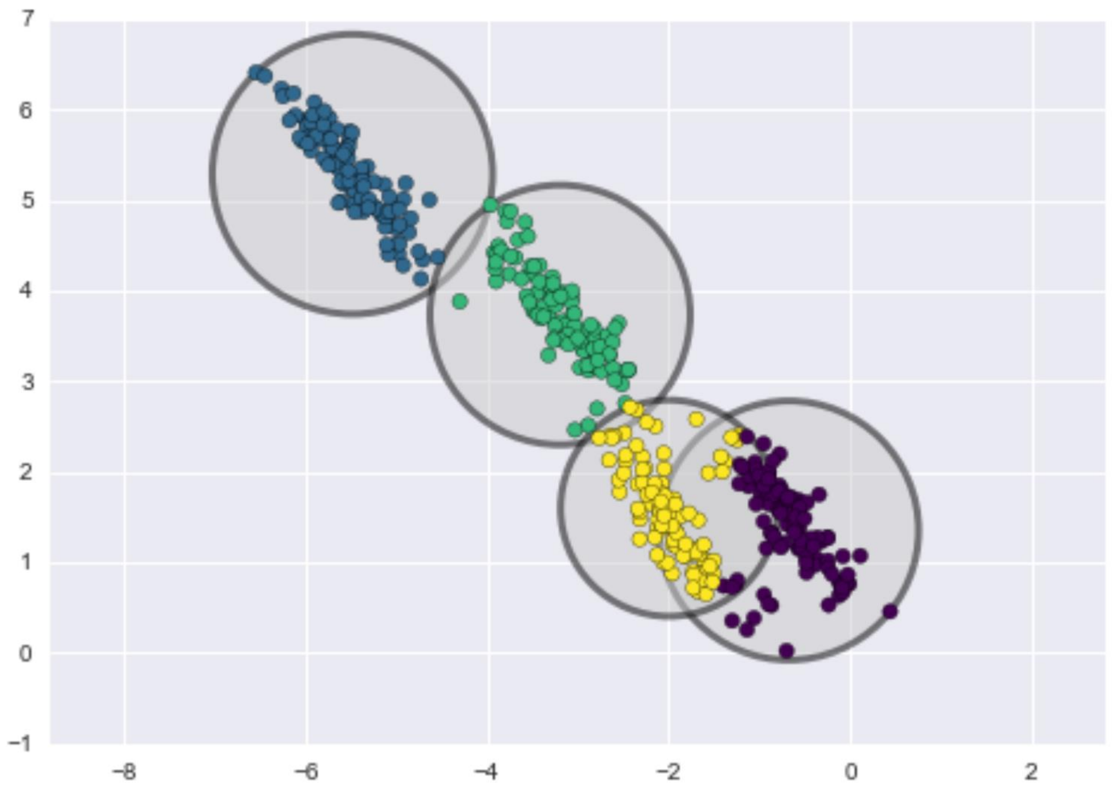
Centroid-based clustering



k-means separates data into Voronoi cells, which assumes equal-sized clusters (not adequate here)

- **Grouping member according to**
  - **Cluster centroid**
    - **May not be a member of the data set**
    - **Conceptually close to nearest neighbor classification**
- **Example of centroid based Clustering**
  - **K-mean clustering**

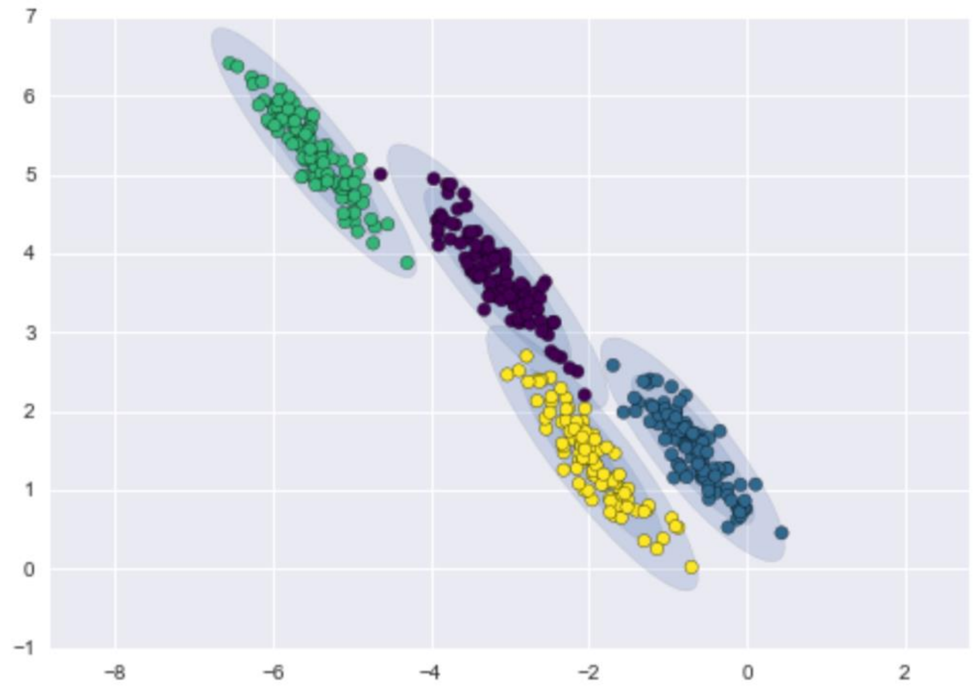
Centroid-based clustering



- **K-mean Limitation**
  - **K-means often doesn’t work**
    - **clusters are not round shaped**
    - **because of distance function**

https://jakevdp.github.io/PythonDataScienceHandbook/05.12-gaussian-mixtures.html

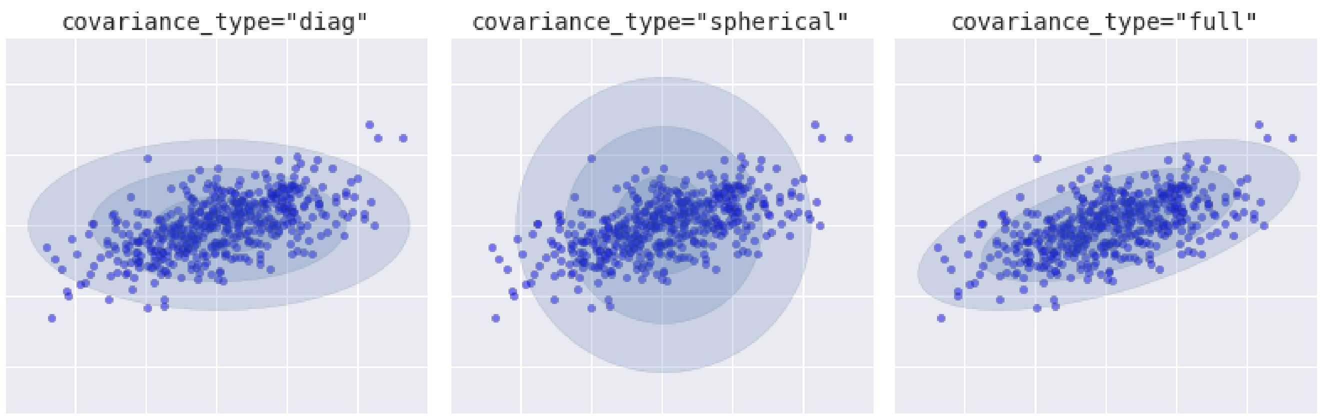
Distribution-based clustering



- **Grouping member according to**
  - **most closely related to probability statistics is based on distribution models**
    - **Gaussian mixture models**

https://jakevdp.github.io/PythonDataScienceHandbook/05.12-gaussian-mixtures.html

Distribution-based clustering

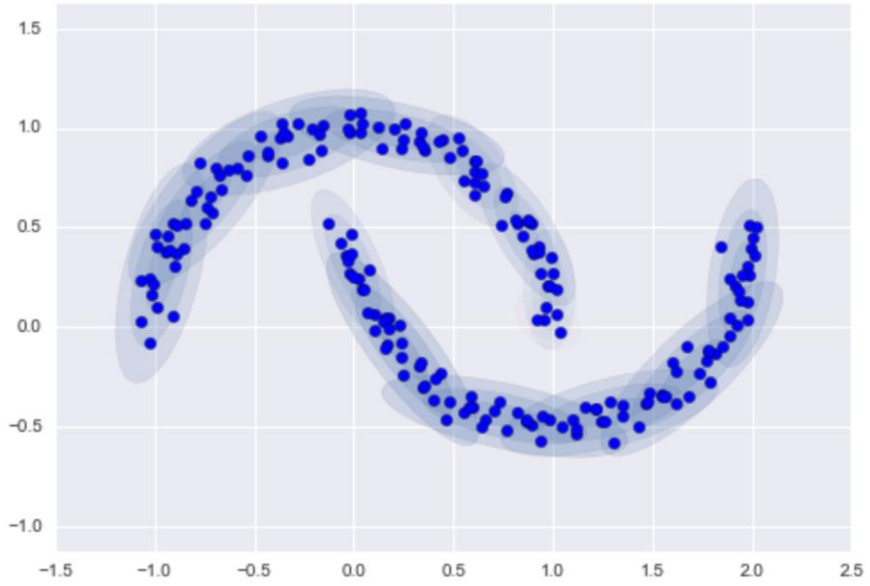


- **Gaussian mixture models**
  - **covariance_type**
    - **controls the degrees of freedom in the shape of each cluster**
    - **Types:**
      - **spherical**
        - **all dimensions are equal**
        - **similar characteristics to that of k-means**
      - **diag**
        - **ellipse constrained to align with the axes**
      - **full**
        - **an ellipse with arbitrary orientation**

https://jakevdp.github.io/PythonDataScienceHandbook/05.12-gaussian-mixtures.html

Distribution-based clustering

```
gmm16 = GMM(n_components=16, covariance_type='full', random_state=0)
plot_gmm(gmm16, Xmoon, label=False)
```

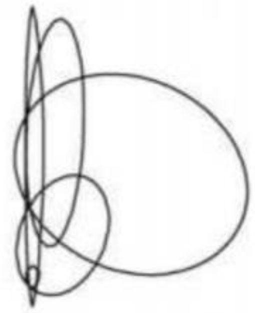


- **GMM: #components**
  - **small # components**
    - **Support**
      - **Simple sphere cluster shapes**

https://jakevdp.github.io/PythonDataScienceHandbook/05.12-gaussian-mixtures.html

Distribution-based clustering


dot plot of two structure feature

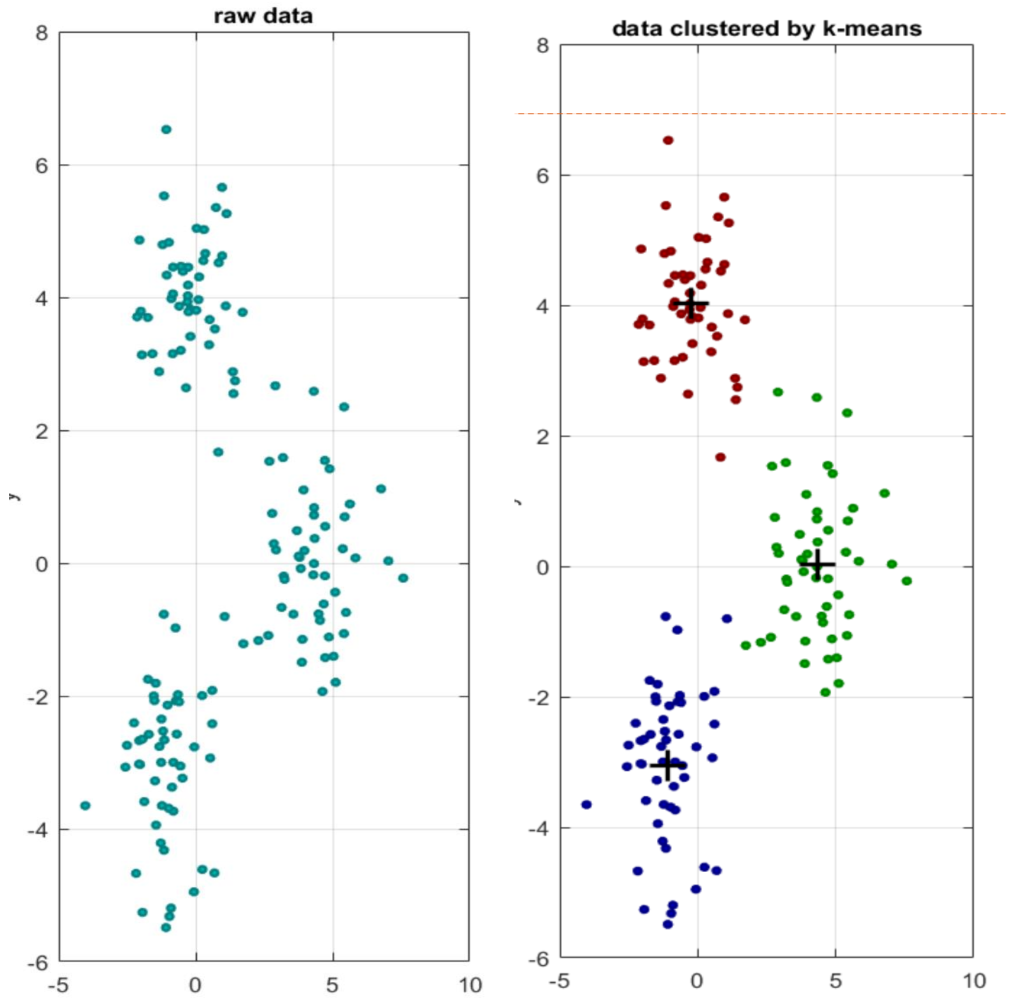

GMM full covariance

- **GMM: Limitation**
  - **Higher Training complexity**
    - **especially**
      - **When distribution shape is not**
        - **Close to sphere or ellipse**

https://jakevdp.github.io/PythonDataScienceHandbook/05.12-gaussian-mixtures.html

K-mean Clustering

Centroid based clustering



Clustering Algorithm:
K-mean clustering

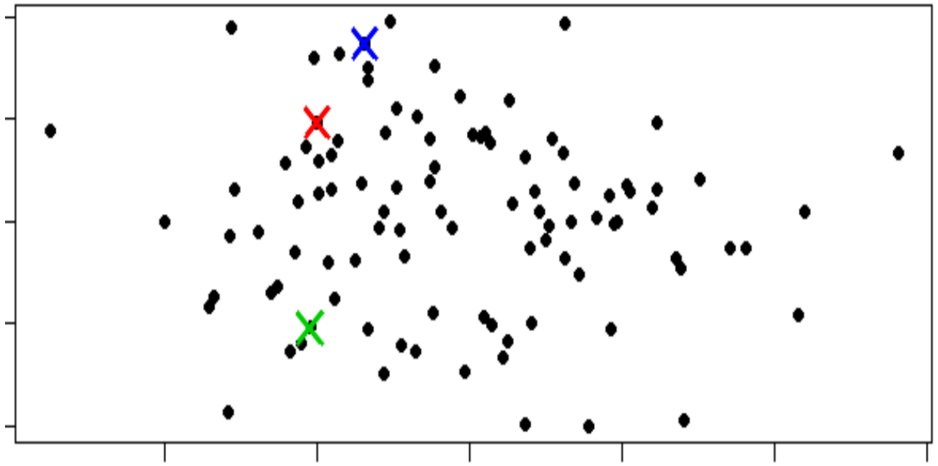
- **Partition based on nearest to group center**

- **Algorithm:**
  - 1: Initial number of groups or regions = k
  - 2: Initial center of each group
  - 3: Perform clustering
    - For all samples
      - Compare distance of each sample to center of each group
      - Assign sample to group i which is closest to that sample
    - End
  - 4: Update center of each group
  - 5: repeat (3) and (4) until feature center of each group changes less than a defined threshold

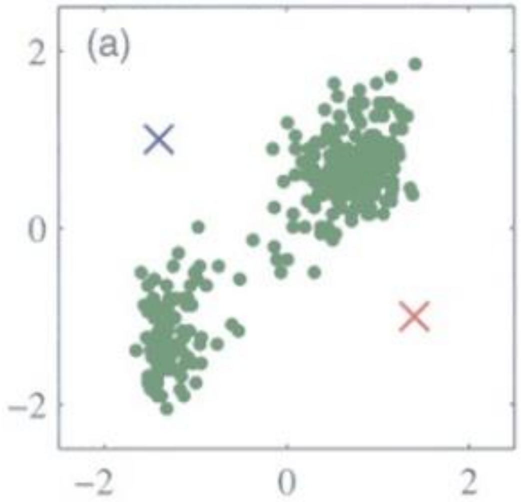
Clustering Algorithm: K-mean clustering

❑ **Initial group center (centroid)**

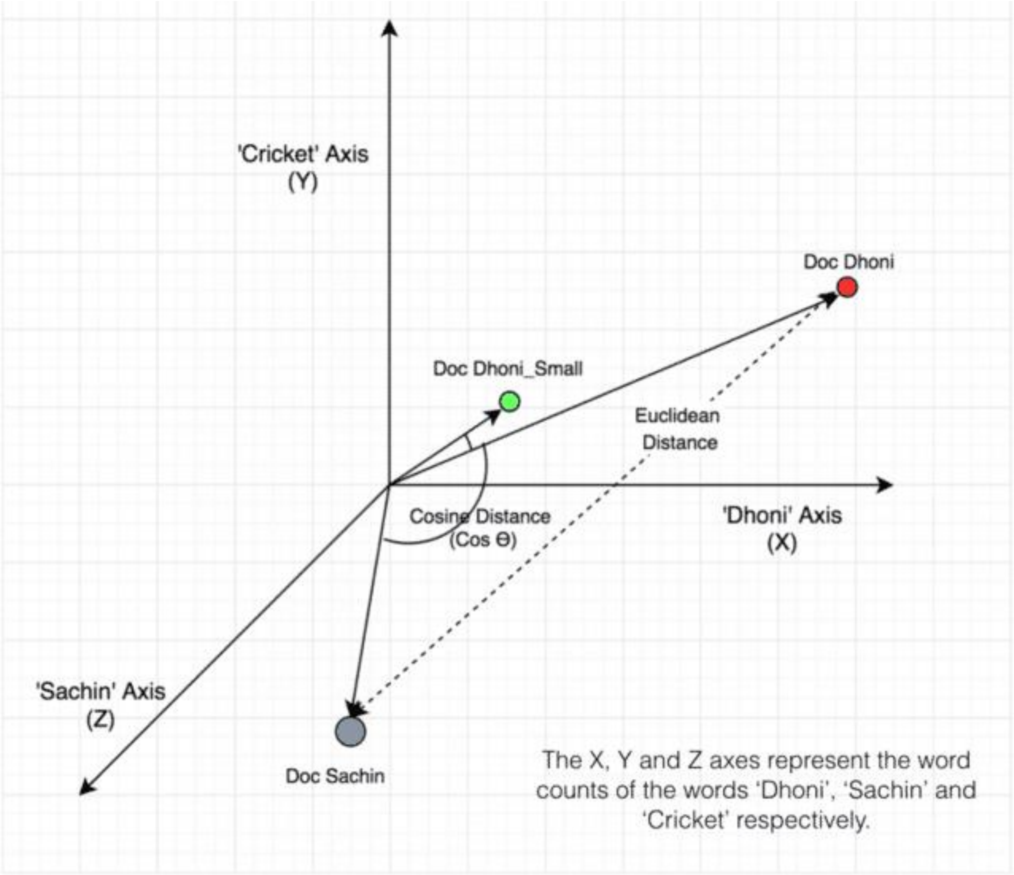
- **Random pickup from samples**



- **Random number**



Clustering Algorithm:
K-mean clustering



❑ Distance Measure

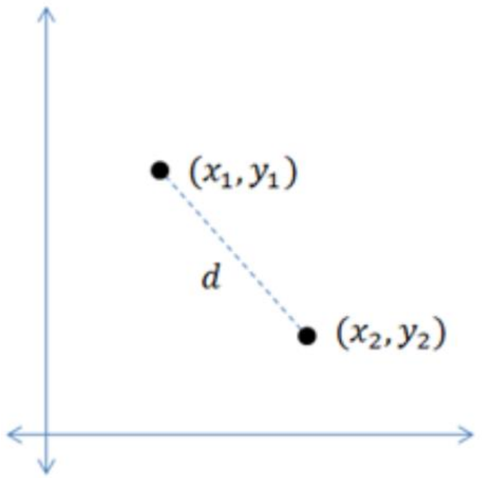
- **Euclidean Distance (Magnitude distance)**
  - **L2 Norm**

$$\sqrt{(x1-y1)^2 + (x2-y2)^2 + \ldots + (xN-yN)^2}$$

- **Cosine similarity (Direction Distance)**

$$Cos\theta = \frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|\ \|\vec{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2}\sqrt{\sum_1^n b_i^2}}$$

Clustering Algorithm:
K-mean clustering



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

❑ Distance Measure

- **Euclidean Distance (Magnitude distance)**
  - **L2 Norm**

$$dist_{L2}(a, b) = \|a - b\|$$

$$\sqrt{(x1 - y1)^2 \ + \ (x2 - y2)^2 \ + \ \ldots \ + \ (xN - yN)^2}$$

Clustering Algorithm:

K-mean clustering

When should we stop clustering process?

Clustering Algorithm:
K-mean clustering

❑ **เงื่อนไขในการหยุดการจัดกลุ่ม**

- **Member**
  - ไม่มีการเปลี่ยนกลุ่มอีกแล้ว
  - อาจมีการเปลี่ยนกลุ่มอยู่บ้าง แต่
    - **Group center**
      - เปลี่ยนแปลงน้อยกว่าค่าที่ตั้งไว้ **(dcenter < Thc)**
- **Max iteration**

Clustering Algorithm: K-mean clustering

- เทคนิคการ update group center

Option#1: Update ตัวแทนกลุ่มทุกครั้งที่มีการเพิ่มสมาชิกเข้ากลุ่ม

G1={A}, G2={B} -> Initial

G1={A,C}, G2={B} -> คำนวนค่าตัวแทนกลุ่ม G1 ใหม่ และใช้ในการทดสอบการเป็นสมาชิกในรอบต่อไป

G1={A,C}, G2={B,D} -> คำนวนค่าตัวแทนกลุ่ม G2 ใหม่ และใช้ในการทดสอบการเป็นสมาชิกในรอบต่อไป

G1={A,C,E}, G2={B,D} -> คำนวนค่าตัวแทนกลุ่ม G1 ใหม่ และใช้ในการทดสอบการเป็นสมาชิกในรอบต่อไป

Option #2: Update ตัวแทนกลุ่มหลังการจับกลุ่มเสร็จสิ้นแล้ว

G1={A}, G2={B}

G1={A,C}, G2={B}

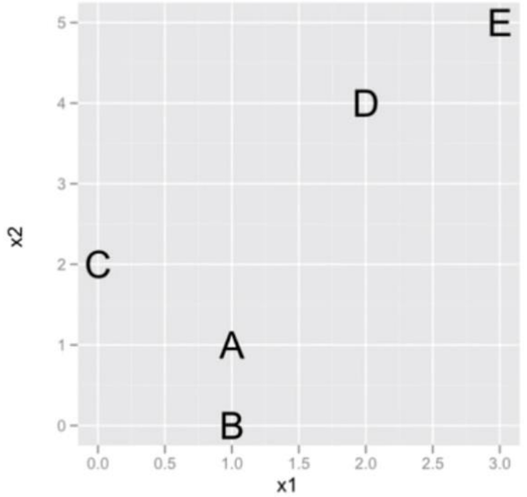
G1={A,C}, G2={B,D}

G1={A,C,E}, G2={B,D}

จับกลุ่มเสร็จแล้ว คำนวนค่าตัวแทนกลุ่ม G1 และ G2

Clustering Algorithm: K-mean clustering

i	X_1	X_2
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5



- เทคนิคการ update group center

Option#1: Update ตัวแทนกลุ่มทุกครั้งที่มีการเพิ่มสมาชิกเข้ากลุ่ม

Initial: G1= { }, G2 = { }

#1: G1 = { }, G2 = { }

#2: G1 = { }, G2 = { }

#3: G1 = { }, G2 = { }

#4: G1 = { }, G2 = { }

Option #2: Update ตัวแทนกลุ่มหลังการจับกลุ่มเสร็จสิ้นแล้ว

#1: G1={ }, G2={ }

#2: G1={ }, G2={ }

#3: G1={ }, G2={ }

#4: G1={ }, G2={ }

จับกลุ่มเสร็จแล้ว คำนวนค่าตัวแทนกลุ่ม G1 และ G2

ML: Recommendation System

Recommendation System:

- **Why recommendation system is needed?**
  - **Recommending a few items to a customer based on their needs and interests**
    - **can create a positive impact on the user experience and**
    - **lead to frequent visits.**

- **What would be best information used for recommendation?**
  - **Basic rules: not appropriate**
    - **Age / Gender / Education**
  - **Behavior (Livestyle): Mostly important and needed to be analyzed**

Recommendation System:

- **Recommendation can be based on**
  - **1) Most demanded / Best selling in store**
  - **2) Highest profit for business**
  - **3) Most popular in Group with Similar favourite**
  - **4) Previous history (past behavior)**
    - **Building smart and intelligent recommendation engines**
      - **by studying the past behavior of their users.**

- **1) & 2) are easily generated**
- **But 3) & 4) are needed to be analyzed with meaningful data collection**

Recommendation System:

- **Data Collection**
  - **Explicit:**
    - **user provided information (intentionally)**
      - **movie rating (Netflix) / hotel reviews (Agoda)**
      - **restaurant reviews (wongnai)**
  - **Implicit:**
    - **gather from available data streams (unintentionally)**
      - **previous orders and favorite items**
      - **search, clicks, like, or share history**

Recommendation System:
Association Analysis

- **Data Storage**
  - **The type of data plays an important role in**
    - **deciding the type of storage that has to be used.**
  - **This type of storage could include**
    - **a standard SQL database**
    - **a NoSQL database or**
    - **some kind of object storage**

Recommendation System:
SQL vs NoSQL

- **SQL vs NoSQL**
  - **SQL**
    - **Advantages:**
      - **support for ACID / reduces anomalies and protects the integrity of your database by suggesting precisely how transactions interact with the database.**
      - **a lot of tools come with better support, product suites and add-ons to manage these databases**
    - **Disadvantages: scalability with growing database**
  - **NoSQL**
    - **Ex: MongoDB, CouchDB, Cassandra, and Hbase**
    - **Advantages:**
      - **no limits on the types of data**
      - **designed to be scaled across multiple data centers**
      - **quickly create a database**
      - **ensure data doesn't become the bottleneck when all of the other components of your server-side application are designed to be seamless and fast.**
    - **Disadvantages:**
      - **lack of reporting tools for performance testing and analysis.**
      - **not yet 100% compatible with the SQL used in relational databases**
      - **lack of standardization / can cause a problem during migration**

https://medium.com/swlh/should-you-use-nosql-or-sql-db-or-both-349cb26c9add

Recommendation System:
Data filtering

- **Content based filtering**
  - **similar to the ones that a user has liked in the past.**
- **Collaborative filtering**
  - **User-User collaborative filtering**
  - **Item-Item collaborative filtering**
- **Hybrid filtering**
- **Association Analysis**

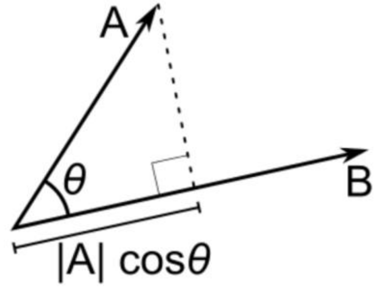
https://medium.com/swlh/should-you-use-nosql-or-sql-db-or-both-349cb26c9add

Recommendation System:
Content based filtering

- **similar to the ones that a user has liked in the past.**
  - **Ex. if a person has liked the movie "Inception",**
    - **then recommend movies that fall under the same genre.**
- **how does the algorithm understand which genre to pick and recommend movies from?**
  - **generate profile vector contains the past behavior of the user**
    - **Ex. movies liked/disliked and the ratings given by users**
  - **generate Item vector contains the details of each movie, like genre, cast, director**
  - **compare similarity between profile vectors or item vectors**
    - **Cosine similarity / Euclidian Distance**

https://medium.com/swlh/should-you-use-nosql-or-sql-db-or-both-349cb26c9add

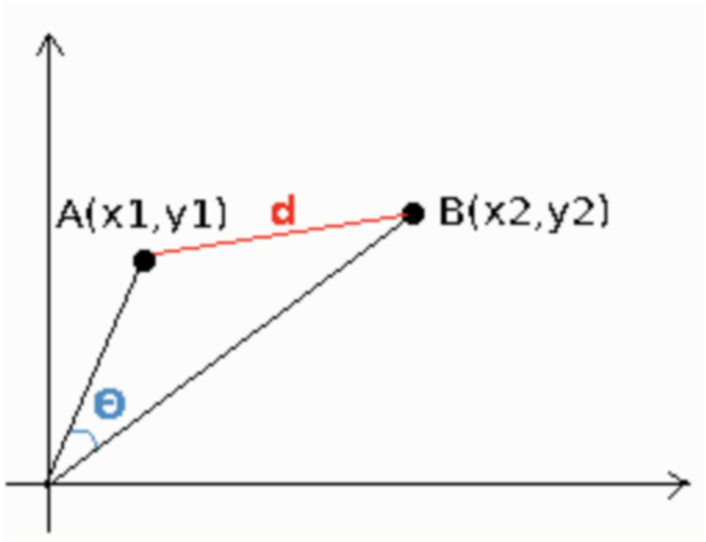
Recommendation System:
Content based filtering



- **Cosine similarity**
  - **calculates the cosine of th**
    - **measurement of orientation and not magnitude**
    - **ignore magnitude difference**

$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

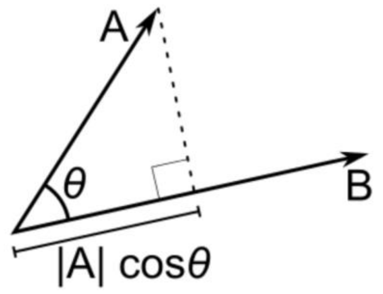
- **Euclidian Distance**
  - **Calculate line distance between two points in Euclidian space**
    - **Measure of magnitude difference**



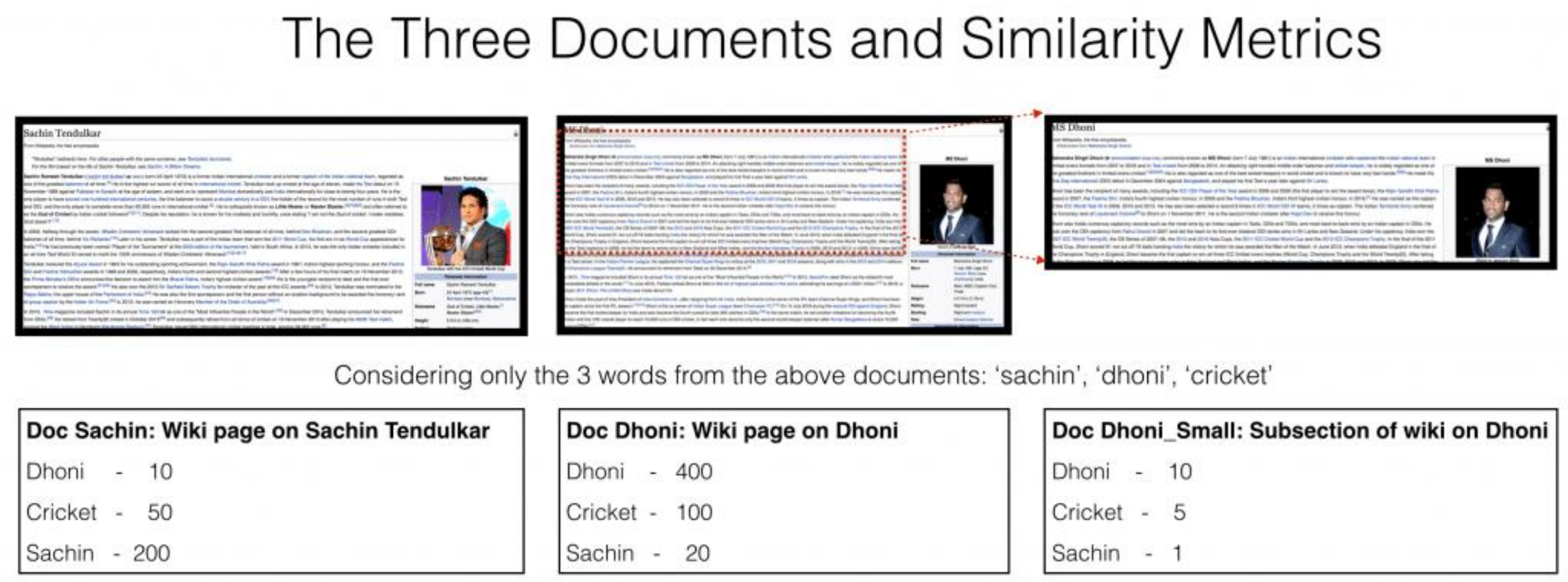
$$\text{Euclidean Distance} = \sqrt{(x_1 - y_1)^2 + \ldots + (x_N - y_N)^2}$$

https://medium.com/@mark.rethana/building-a-song-recommendation-system-using-cosine-similarity-and-euclidian-distance-748fdfc832fd

Recommendation System:
Content based filtering

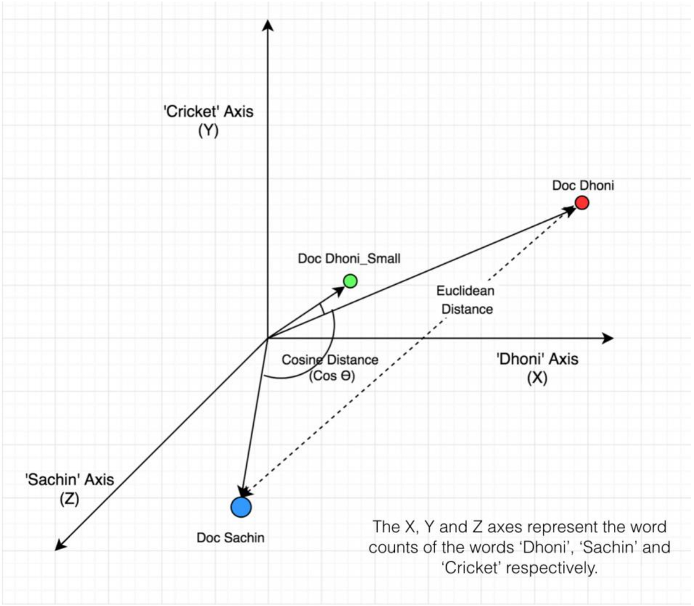


- **Cosine similarity**



https://www.machinelearningplus.com/nlp/cosine-similarity/

Recommendation System:
Content based filtering



- **Cosine similarity**

$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

Document - Term Matrix (Word Counts)

Word Counts	"Dhoni"	"Cricket"	"Sachin"
Doc Sachin	10	50	200
Doc Dhoni	400	100	20
Doc Dhoni_Small	10	5	1

Word counts	Cosine	Euclidian
Doc Sachin & Doc Dhoni		
Doc Dhoni & Doc Dhoni_Small		
Doc Sachin & Doc Dhoni_Small		

https://www.machinelearningplus.com/nlp/cosine-similarity/

Recommendation System:
Collaborative based filtering (User-User)

- **User-User collaborative filtering**
  - **finds the similarity score between users**
    - **Pearson's Correlation**

User/Movie	x1	x2	x3	x4	x5	Mean User Rating
A	4	1	–	4	–	3
B	–	4	–	2	3	3
C	–	1	–	4	4	3

$$sim(u,v) = \frac{\sum(r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum(r_{ui} - \bar{r}_u)^2}\sqrt{\sum(r_{vi} - \bar{r}_v)^2}}$$

  - **base**
    - **picks out the most similar users and**
    - **recommends products which these similar users have liked or bought previously.**

Recommendation System:
Collaborative based filtering (User-User)

- **Pearson's Correlation**

$$sim(u,v) = \frac{\sum(r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum(r_{ui} - \bar{r}_u)^2}\sqrt{\sum(r_{vi} - \bar{r}_v)^2}}$$

User/Movie	x1	x2	x3	x4	x5	Mean User Rating
A	4	1	–	4	–	3
B	–	4	–	2	3	3
C	–	1	–	4	4	3

Correlation	$\sum(r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)$	$\sqrt{\sum(r_{ui} - \bar{r}_u)^2}\sqrt{\sum(r_{vi} - \bar{r}_v)^2}$	**sim (u,v)**
AB			
BC			
AC			

Recommendation System:
Collaborative based filtering (User-User)

- **User-User collaborative filtering**
  - **useful when the number of users is less.**
  - **not effective with a large number of users**
    - **take a lot of time to compute the similarity between all user pairs.**

  - **One way of reducing similarity complexity**
    - **Select a threshold similarity and choose all the users above that value**
      - **Binary similarity measurement**
    - **Randomly select the users**
    - **Arrange the neighbors in descending order of their similarity value and choose top-N users**
    - **Use clustering for choosing neighbors**

Recommendation System:
Collaborative based filtering (Item-Item)

- **Item-Item collaborative filtering**
  - **finds the similarity score between items**
    - **Pearson's Correlation**

$$sim(u,v) = \frac{\sum(r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum(r_{ui} - \bar{r}_u)^2}\sqrt{\sum(r_{vi} - \bar{r}_v)^2}}$$

  - **base**
    - **picks out the most similar items with the item use**
    - **recommend similar items which are liked by the u**

User/Movie	x1	x2	x3	x4	x5
A	4	1	2	4	4
B	2	4	4	2	1
C	–	1	–	3	4
Mean Item Rating	3	2	3	3	3

User/Movie	A	B	C	Mean Item Rating
x1	4	2	–	3
x2	1	4	1	2
x3	2	4	–	3
x4	4	2	3	3
x5	4	1	4	3

Recommendation System:
Collaborative based filtering (Item-Item)

- **Pearson's Correlation**

$$sim(u,v) = \frac{\sum(r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum(r_{ui} - \bar{r}_u)^2}\sqrt{\sum(r_{vi} - \bar{r}_v)^2}}$$

User/Movie	A	B	C	Mean Item Rating
x1	4	2	–	3
x2	1	4	1	2
x3	2	4	–	3
x4	4	2	3	3
x5	4	1	4	3

	$\sum(r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)$	$\sqrt{\sum(r_{ui} - \bar{r}_u)^2}\sqrt{\sum(r_{vi} - \bar{r}_v)^2}$		**sim (u,v)**
X(1,2)				
X(1,3)				
X(1,4)				
X(1,5)				

Recommendation System:
Association Analysis

- **Uncover associations between items**
  - **Looking for combinations of items that occur together frequently in transactions**
    - **Find relationships in large data sets.**
    - **Find association across different product segments**
- **using this asset to gain competitive advantage.**
  - **gain competitive customer insights by**
    - **knowing how your products are associated in different segments;**
    - **differentiate discount offers based on market basket analysis.**

Recommendation System:
Association Analysis

- **Transaction Examples**

Transaction #	Shirts	Trousers	Ties
001	1	1	1
002	0	1	0
003	1	0	1
004	1	0	1
005	1	1	0

TID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

TID	Bread	Milk	Diapers	Beer	Eggs	Cola
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

Recommendation System:
Association Analysis

- **How can we measure how items are related?**
  - **Support**
    - **measure how strong the rule is**
      - **Looking rules with**
        - **strong/ high support**
      - **Rule with low support**
        - **might occur simply by chance**
    - **measure how often**
      - **both X,Y occur together from total N transactions**
    - **support = 60% is a fairly high value**
      - **For real world problems with several product groups,**
        - **support of 1% or at times even lower depending upon the nature of your problem is also useful.**

$$Rule: X \Rightarrow Y$$

$$Support = \frac{frq(X,Y)}{N}$$

$$Confidence = \frac{frq(X,Y)}{frq(X)}$$

$$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$$

Recommendation System:
Association Analysis

- **How can we measure how items are related?**
  - **Confident**
    - **measure reliability of rules**
    - **For a given rule X→ Y ,**
      - **the higher the confidence,**
      - **the more likely it is for Y to be present**
        - **In transactions that contain X.**
    - **Confidence also provides an estimate of the conditional probability of Y given X.**
    - **Suggest a strong co-occurrence relationship between items**
      - **How often X,Y occur together compare to how often X occurs**

$$Rule:\ X \Rightarrow Y$$

$$Support = \frac{frq(X,Y)}{N}$$

$$Confidence = \frac{frq(X,Y)}{frq(X)}$$

$$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$$

Recommendation System:
Association Analysis

- **How can we measure how items are related?**
  - **Lift**
    - **how likely item Y is purchased**
      - **when item X is purchased,**
      - **while controlling for how popular item Y is.**
    - **Lift value greater than 1 means**
      - **item Y is likely to be bought if item X is bought,**
    - **Lift value less than 1 means**
      - **item Y is unlikely to be bought if item X is bought.**

$$Rule:\ X \Rightarrow Y$$

$$Support = \frac{frq(X,Y)}{N}$$

$$Confidence = \frac{frq(X,Y)}{frq(X)}$$

$$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$$

Recommendation System:
Association Analysis

- **How can we measure how items are related?**

$$Rule:\ X \Rightarrow Y$$

$$Support = \frac{frq(X,Y)}{N}$$

$$Confidence = \frac{frq(X,Y)}{frq(X)}$$

$$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$$

	A	B	C	D	E
T1					
T2					
T3					
T4					
T5					

Recommendation System:

Association Analysis

$$Rule:\ X \Rightarrow Y$$

$$Support = \frac{frq(X,Y)}{N}$$

$$Confidence = \frac{frq(X,Y)}{frq(X)}$$

$$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$$

- **How can we measure how items are related?**

	A	B	C	D	E
T1	1	1	1	0	0
T2	1	0	1	1	0
T3	0	1	1	1	0
T4	1	0	0	1	1
T5	0	1	1	0	0
Frq()					

	Support	Confident	Lift
{A} -> {C}			
{C} -> {A}			
{A,C} -> {D}			
{A,B,C} -> {D}			

Recommendation System:
Association Analysis

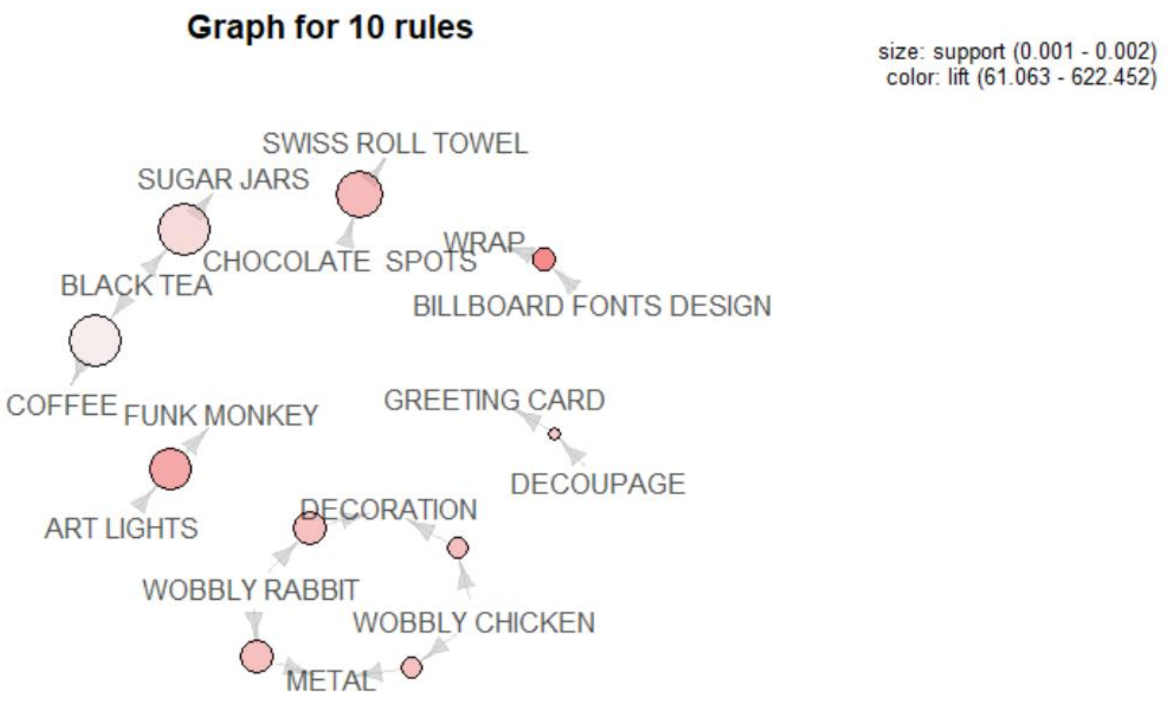
$$Rule: X \Rightarrow Y$$

$$Support = \frac{frq(X,Y)}{N}$$

$$Confidence = \frac{frq(X,Y)}{frq(X)}$$

$$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$$

- **How can we visualize association rules?**
  - **in term of a graph**
    - **Size of node:**
      - **Strength of rule**
    - **Edge:**
      - **Relations between rules**



Recommendation System:
Association Analysis

$$Rule: X \Rightarrow Y$$

$$Support = \frac{frq(X,Y)}{N}$$

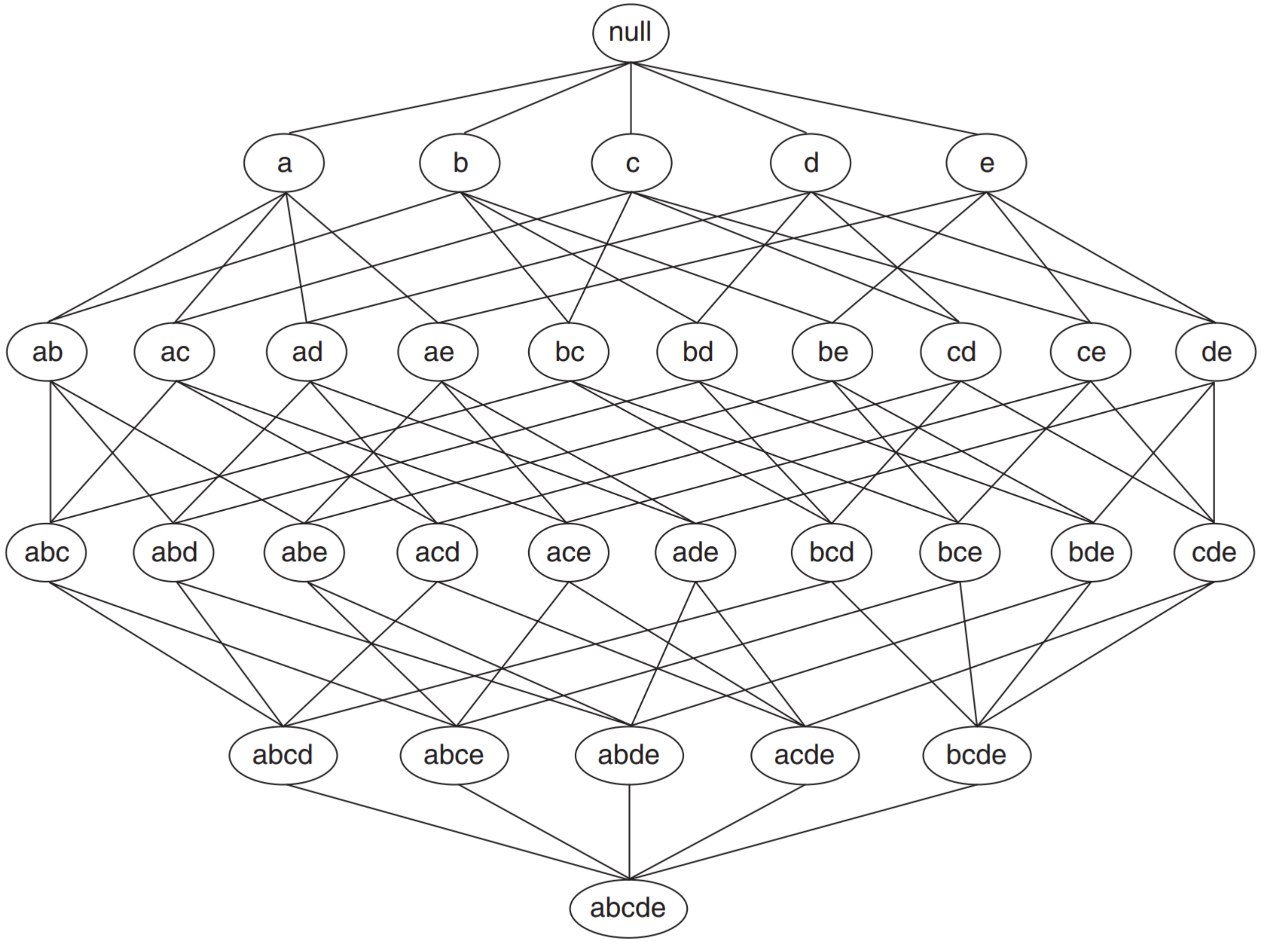
$$Confidence = \frac{frq(X,Y)}{frq(X)}$$

$$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$$

- **How can we solve realword problem?**
  - **Generate item lattice (Tree structure of Item combination rules)**
  - **Select only desired combination**
  - **Calculate strength of association rules**
    - **support / confident / lift**

Recommendation System:
Association Analysis

- **How can we solve realword problem?**
  - **Generate item lattice (Tree structure of Item combination rules)**
    - **Brute-force combination**
    - **Exponential growth**
      - **With large number of items**



Recommendation System:

Association Analysis

- **Pruning:**
  - **Select only desired combination**
  - **Perform Tree pruning**
    - **Apriori support pruning**

Candidate 1-Itemsets

Item	Count
Beer	3
Bread	4
Cola	2
Diapers	4
Milk	4
Eggs	1

6×6 = 36 combination

Minimum support count = 3

Candidate 2-Itemsets

Itemset	Count
{Beer, Bread}	2
{Beer, Diapers}	3
{Beer, Milk}	2
{Bread, Diapers}	3
{Bread, Milk}	3
{Diapers, Milk}	3

6 combination

มีครึ่ง

Itemsets removed because of low support

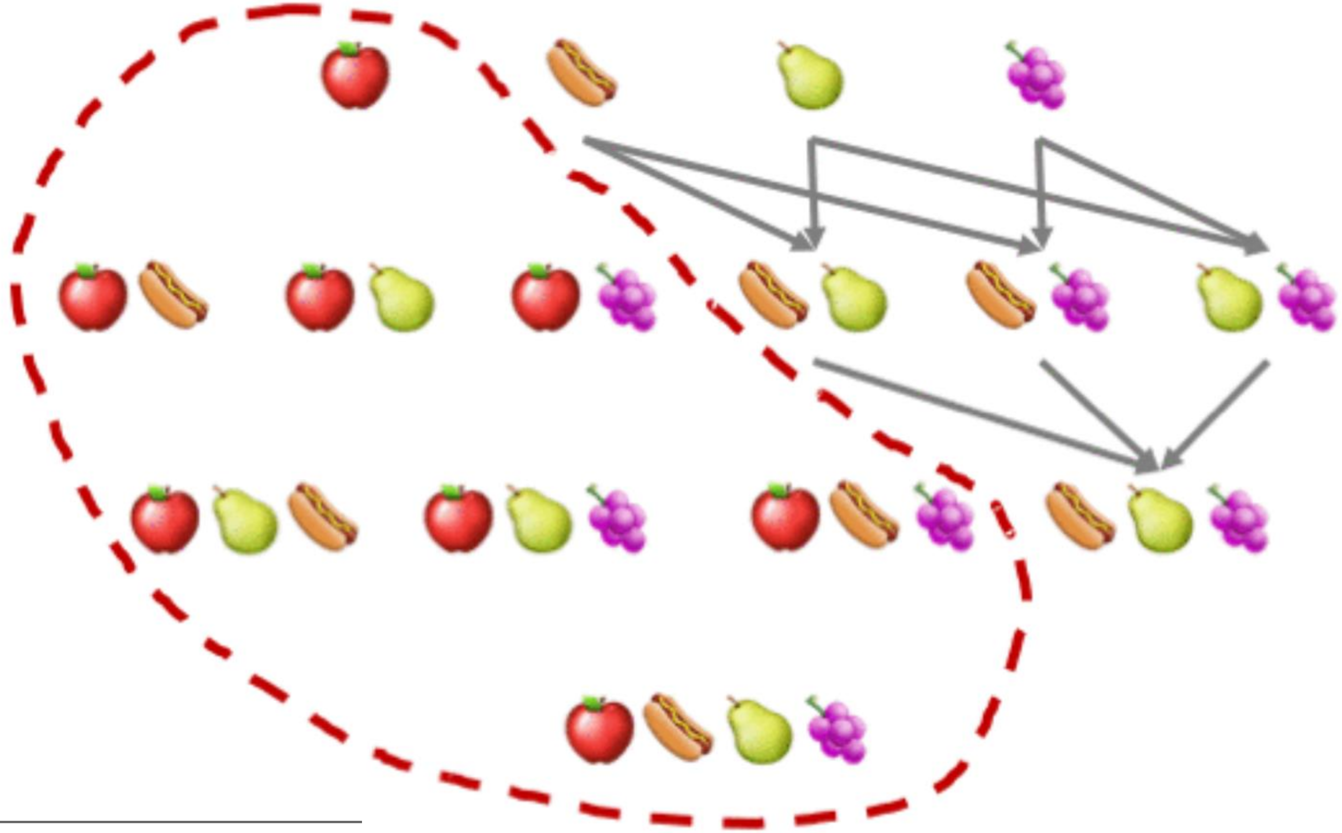
4 × 4 = 16

Candidate 3-Itemsets

Itemset	Count
{Bread, Diapers, Milk}	2

Recommendation System:
Association Analysis

- **Pruning:**
  - **Select only desired combination**
  - **Perform Tree pruning**
    - **Apriori support pruning**
      - **eliminate the itemsets with low support**



If an itemset is frequent, then all of its subsets must also be frequent.

Recommendation System:
Association Analysis

- **Apriori support pruning**
  - **Advantage:**
    - **Easy to compute**
  - **Disadvantage:**
    - **very slow and inefficient,**
    - **especially when memory capacity is limited and the number of transactions is large.**
- **Frequent Pattern tree pruning (FP-Growth Pruning)**
  - **Advantage:**
    - faster
      - **improves upon the Apriori algorithm quite significantly**
      - **the FPgrowth algorithm only needs two passes on a dataset.**
  - **Disadvantage:**
    - **expensive to build, since if the dataset is big it may not fit in memory.**

Recommendation System:
Association Analysis

- **FP-Growth Pruning**

TID	Items
1	E, A, D, B
2	D, A, C, E, B
3	C, A, B. E
4	B, A, D
5	D
6	D,B
7	A,D,E
8	B,C

Item	Frequency	
A	5	3
B	6	1
C	3	5
D	6	2
E	4	4

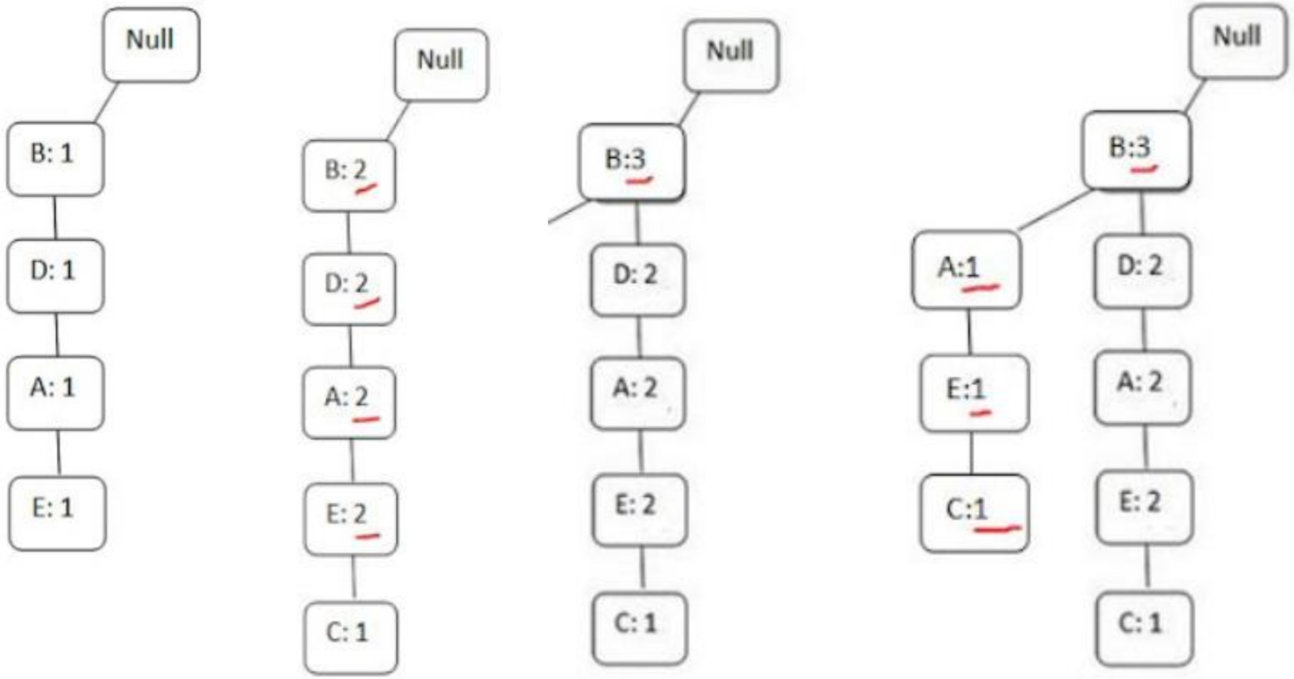
TID	Items	Ordered Items
1	E, A, D, B	B,D,A,E
2	D, A, C, E, B	B,D,A,E,C
3	C, A, B. E	B,A,E,C
4	B, A, D	B,D,A
5	D	D
6	D,B	B,D
7	A,D,E	D,A,E
8	B,C	B,C

http://hareenlaks.blogspot.com/2011/06/fp-tree-example-how-to-identify.html

Recommendation System:
Association Analysis

- **FP-Growth Pruning**

TID	Items	Ordered Items
1	E, A, D, B	B,D,A,E
2	D, A, C, E, B	B,D,A,E,C
3	C, A, B. E	B,A,E,C
4	B, A, D	B,D,A
5	D	D
6	D,B	B,D
7	A,D,E	D,A,E
8	B,C	B,C

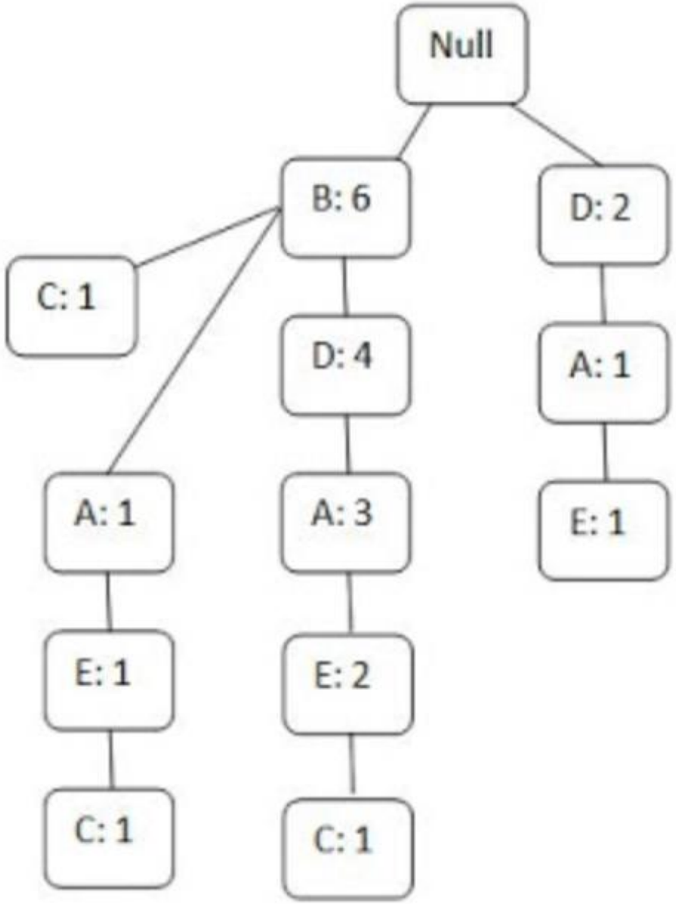


http://hareenlaks.blogspot.com/2011/06/fp-tree-example-how-to-identify.html

Recommendation System:
Association Analysis

- **FP-Growth Pruning**

TID	Items	Ordered Items
1	E, A, D, B	B,D,A,E
2	D, A, C, E, B	B,D,A,E,C
3	C, A, B. E	B,A,E,C
4	B, A, D	B,D,A
5	D	D
6	D,B	B,D
7	A,D,E	D,A,E
8	B,C	B,C



http://hareenlaks.blogspot.com/2011/06/fp-tree-example-how-to-identify.html

Recommendation System:
Association Analysis

TID	Items	Ordered Items
1	E, A, D, B	B,D,A,E
2	D, A, C, E, B	B,D,A,E,C
3	C, A, B. E	B,A,E,C
4	B, A, D	B,D,A
5	D	D
6	D,B	B,D
7	A,D,E	D,A,E
8	B,C	B,C

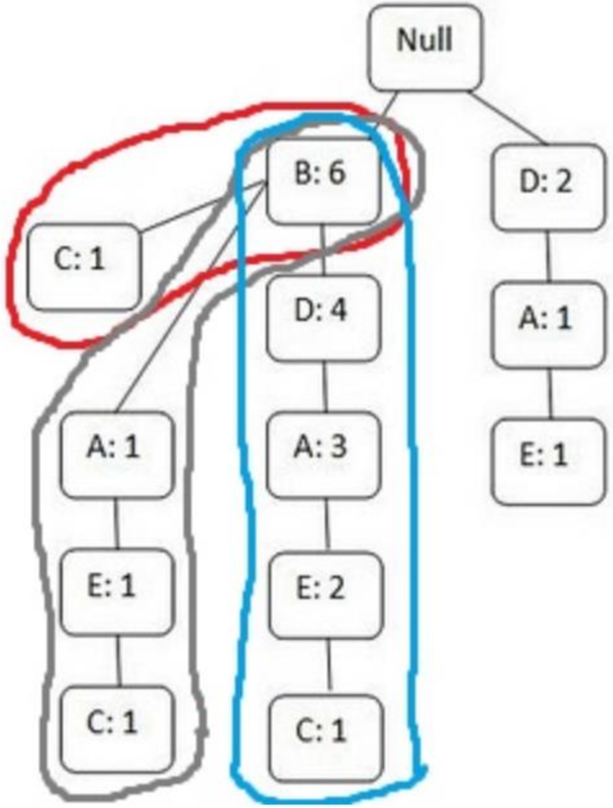
- **FP-Growth Pruning**

Conditional Pattern Base of C

BDAE: 1 - The branch which surrounded in Blue.

B: 1 - The branch which surrounded in Red.

BAE: 1 - The branch which surrounded in Brown.



B:4,D:4,A:3,E:2
Delete D, A, E
Frequent branch: C:3 / BC: 3

https://hareenlaks.blogspot.com/2011/10/how-to-identify-frequent-patterns-from.html

conclusion

- **Content-based recommendation**
  - **Recommend items that similar to what a user has liked in the past**
- **Collaborative-based recommendation**
  - **User-User collaborative filtering**
    - **Recommend items according to similar User**
  - **Item-Item collaborative filtering**
    - **Recommend items according to similar items**
- **Association Analysis**
  - **Check item association**
    - **Recommend item with high support / confidence / lift**
      - **For a probotion set**