NATIONAL ENGINEERING CENTER
University of the Philippines
Diliman, Quezon City

5.0 Unsupervised Learning Methodologies

Eugene Rex L. Jalao, Ph.D.

Associate Professor

Department Industrial Engineering and Operations Research

University of the Philippines Diliman

@thephdataminer

Module 3 of the Business Intelligence and Analytics Track of UP NEC and the UP Center of Business Intelligence

Module 3 Outline

1. Introduction to Data Mining
2. Data Preprocessing
   – Case Study on Big Data Preprocessing using R
3. Classification Methodologies
   – Case Study on Classification using R
4. Regression Methodologies
   – Case Study: Regression Analysis using R
5. **Unsupervised Learning**
   – **Case Study: Social Media Sentiment Analysis using R**

Outline for This Session

- Market Basket Analysis
- Sequential Pattern Mining
- Clustering
  - K-Means Clustering
  - Hierarchical Clustering
- Text Mining
- Social Media Sentiment Analysis
- Case Study

Unsupervised Learning

- Finding hidden patterns within data
- No Response/Class variable
- No guarantee that there are meaningful patterns
- No easy way to measure errors
- Most research on new algorithms

Association Rule Mining

- Given a set of transactions, find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction

Market-Basket transactions

TID	*Items*
1	**Bread, Milk**
2	**Bread, Diaper, Beer, Eggs**
3	**Milk, Diaper, Beer, Coke**
4	**Bread, Milk, Diaper, Beer**
5	**Bread, Milk, Diaper, Coke**

Example of Association Rules

{Diaper} $\rightarrow$ {Beer},
{Milk, Bread} $\rightarrow$ {Eggs,Coke},
{Beer, Bread} $\rightarrow$ {Milk},

Required Dataset Structure (Basket Format)

Items

Transactions

TID	*Items*
1	**Bread, Milk**
2	**Bread, Diaper, Beer, Eggs**
3	**Milk, Diaper, Beer, Coke**
4	**Bread, Milk, Diaper, Beer**
5	**Bread, Milk, Diaper, Coke**

Definition: Frequent Itemset

- Itemset
  - A collection of one or more items
    - Example: {Milk, Bread, Diaper}
  - k-itemset
    - An itemset that contains k items
- Support count (σ)
  - Frequency of occurrence of an itemset
  - E.g. σ({Milk, Bread, Diaper}) = 2
- Support
  - Fraction of transactions that contain an itemset
  - E.g. s({Milk, Bread, Diaper}) = 2/5
- Frequent Itemset
  - An itemset whose support is greater than or equal to a *minsup* threshold

TID	*Items*
1	**Bread, Milk**
2	**Bread, Diaper, Beer, Eggs**
3	**Milk, Diaper, Beer, Coke**
4	**Bread, Milk, Diaper, Beer**
5	**Bread, Milk, Diaper, Coke**

Definition: Association Rule

- Association Rule
  - An implication expression of the form $X \to Y$, where X and Y are itemsets
  - Example:
    $\{\text{Milk}, \text{Diaper}\} \to \{\text{Beer}\}$
- Rule Evaluation Metrics
  - Confidence (c)
    - Measures how often items in Y appear in transactions that contain X

TID	*Items*
1	**Bread, Milk**
2	**Bread, Diaper, Beer, Eggs**
3	**Milk, Diaper, Beer, Coke**
4	**Bread, Milk, Diaper, Beer**
5	**Bread, Milk, Diaper, Coke**

Example:

$$\{\text{Milk}, \text{Diaper}\} \Rightarrow \text{Beer}$$

$$s = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{|\text{T}|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{\sigma(\text{Milk}, \text{Diaper})} = \frac{2}{3} = 0.67$$

Association Rule Mining Task

- Given a set of transactions T, the goal of association rule mining is to find all rules having
  - support ≥ *minsup* threshold
  - confidence ≥ *minconf* threshold
- How to set the appropriate minsup threshold?
  - If minsup is set too high, we could miss itemsets involving interesting rare items (e.g., expensive products)
  - If minsup is set too low, it is computationally expensive and the number of itemsets is very large
- Using a single minimum support threshold may not be effective

Solving Association Rule Mining Problems

- Brute-force approach:
  - List all possible association rules
  - Compute the support and confidence for each rule
  - Disregard rules that fail the *minsup* and *minconf* thresholds

  $\Rightarrow$ Computationally prohibitive!

Mining Association Rules

TID	*Items*
1	**Bread, Milk**
2	**Bread, Diaper, Beer, Eggs**
3	**Milk, Diaper, Beer, Coke**
4	**Bread, Milk, Diaper, Beer**
5	**Bread, Milk, Diaper, Coke**

Example of Rules:

{Milk,Diaper} $\rightarrow$ {Beer} (s=0.4, c=0.67)
{Milk,Beer} $\rightarrow$ {Diaper} (s=0.4, c=1.0)
{Diaper,Beer} $\rightarrow$ {Milk} (s=0.4, c=0.67)
{Beer} $\rightarrow$ {Milk,Diaper} (s=0.4, c=0.67)
{Diaper} $\rightarrow$ {Milk,Beer} (s=0.4, c=0.5)
{Milk} $\rightarrow$ {Diaper,Beer} (s=0.4, c=0.5)

Observations:

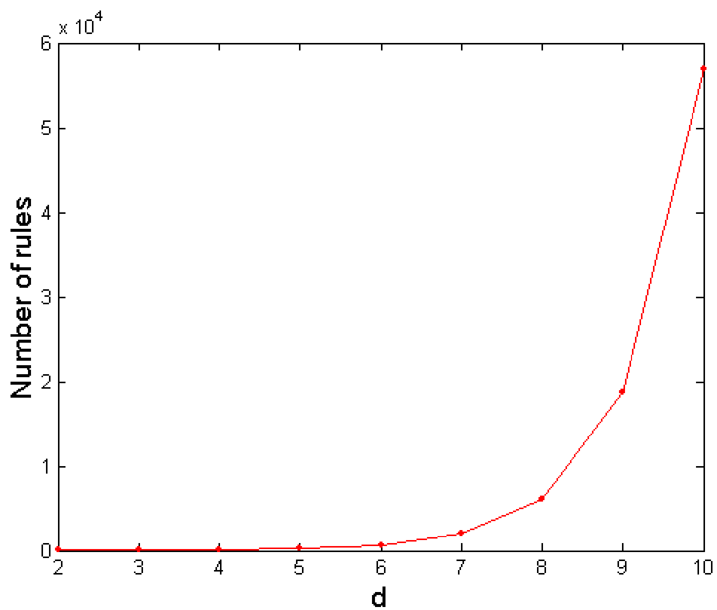
- All the above rules are binary partitions of the same itemset:
  {Milk, Diaper, Beer}
- Rules originating from the same itemset have identical support but can have different confidence
- Thus, we may decouple the support and confidence requirements

Mining Association Rules

- Two-step approach:
  - Frequent Itemset Generation
    - Generate all itemsets whose support $\geq$ minsup
  - Rule Generation
    - Generate high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset
- Frequent itemset generation is still computationally expensive

Computational Complexity

- Given d unique items:
  - Total number of itemsets = 2^d
  - Total number of possible association rules:



$$R = \sum_{k=1}^{d-1}\left[\binom{d}{k} \times \sum_{j=1}^{d-k}\binom{d-k}{j}\right]$$

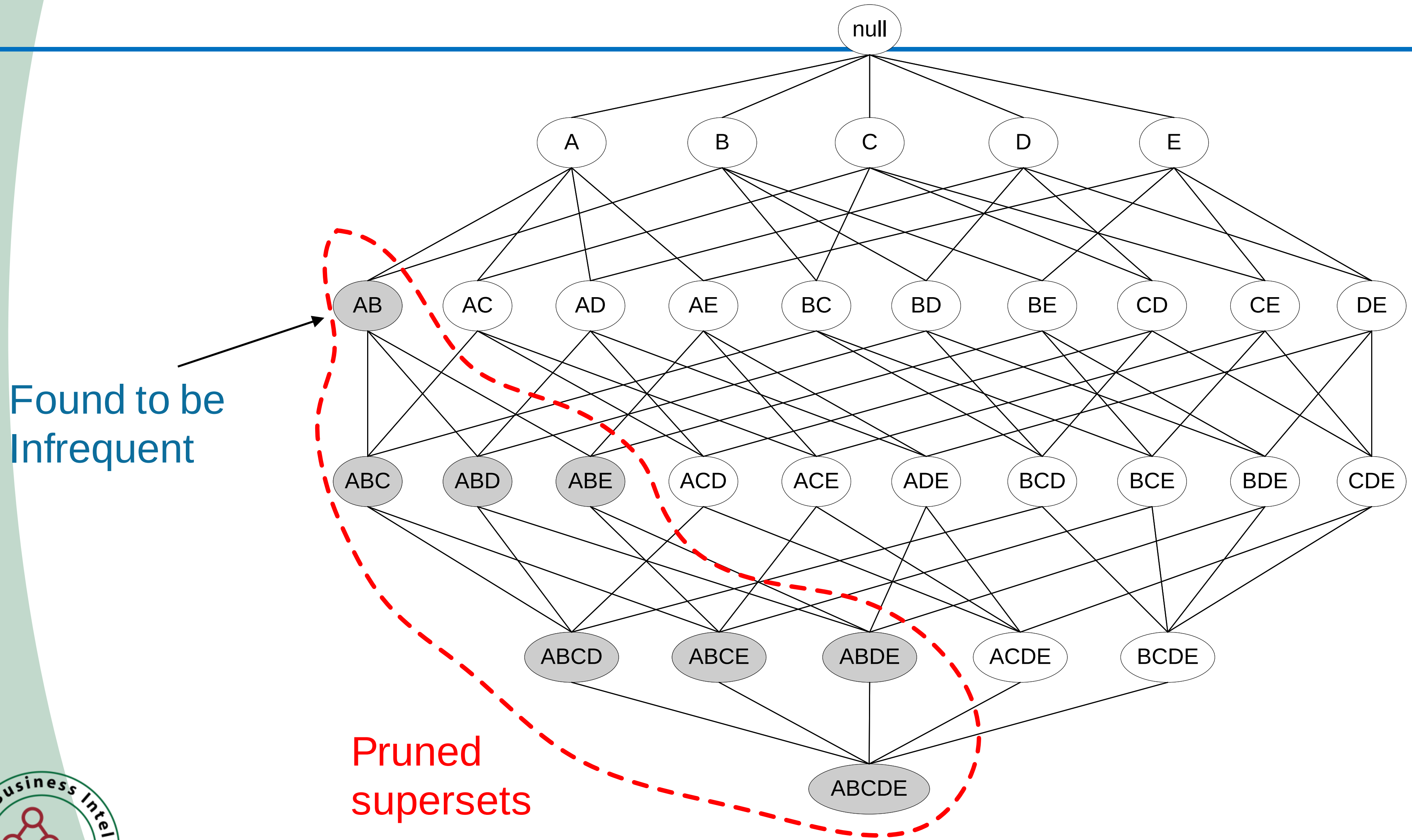
$$= 3^d - 2^{d+1} + 1$$

If d=6, R = 602 rules

Apriori Principle

- Apriori principle:
  - If an itemset is frequent, then all of its subsets must also be frequent
  - Support of an itemset never exceeds the support of its subsets
  - This is known as the anti-monotone property of support

Illustrating Apriori Principle



Illustrating Apriori Principle

Item	Count
Bread	**4**
Coke	**2**
Milk	**4**
Beer	**3**
Diaper	**4**
Eggs	**1**

Items (1-itemsets)

Itemset	Count
{Bread,Milk}	**3**
{Bread,Beer}	**2**
{Bread,Diaper}	**3**
{Milk,Beer}	**2**
{Milk,Diaper}	**3**
{Beer,Diaper}	**3**

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

Minimum Support = 3

Triplets (3-itemsets)

Itemset	Count
{Bread,Milk,Diaper}	**3**

If every subset is considered,

$^6C_1 + {}^6C_2 + {}^6C_3 = 41$

With support-based pruning,

$6 + 6 + 1 = 13$

Rule Generation

- Given a frequent itemset L, find all non-empty subsets $f \subset L$ such that $f \rightarrow L - f$ satisfies the minimum confidence requirement
  - If {A,B,C,D} is a frequent itemset, candidate rules:
    - $ABC \rightarrow D$, $ABD \rightarrow C$, $ACD \rightarrow B$, $BCD \rightarrow A$,
      $A \rightarrow BCD$, $B \rightarrow ACD$, $C \rightarrow ABD$, $D \rightarrow ABC$
      $AB \rightarrow CD$, $AC \rightarrow BD$, $AD \rightarrow BC$, $BC \rightarrow AD$,
      $BD \rightarrow AC$, $CD \rightarrow AB$,
- If $|L| = k$, then there are $2^k - 2$ candidate association rules (ignoring $L \rightarrow \varnothing$ and $\varnothing \rightarrow L$)

Rule Generation

- How to efficiently generate rules from frequent itemsets?
  - In general, confidence does not have an anti-monotone property

    $c(ABC \rightarrow D)$ can be larger or smaller than $c(AB \rightarrow D)$
  - But confidence of rules generated from the same itemset has an anti-monotone property
  - e.g., L = {A,B,C,D}:

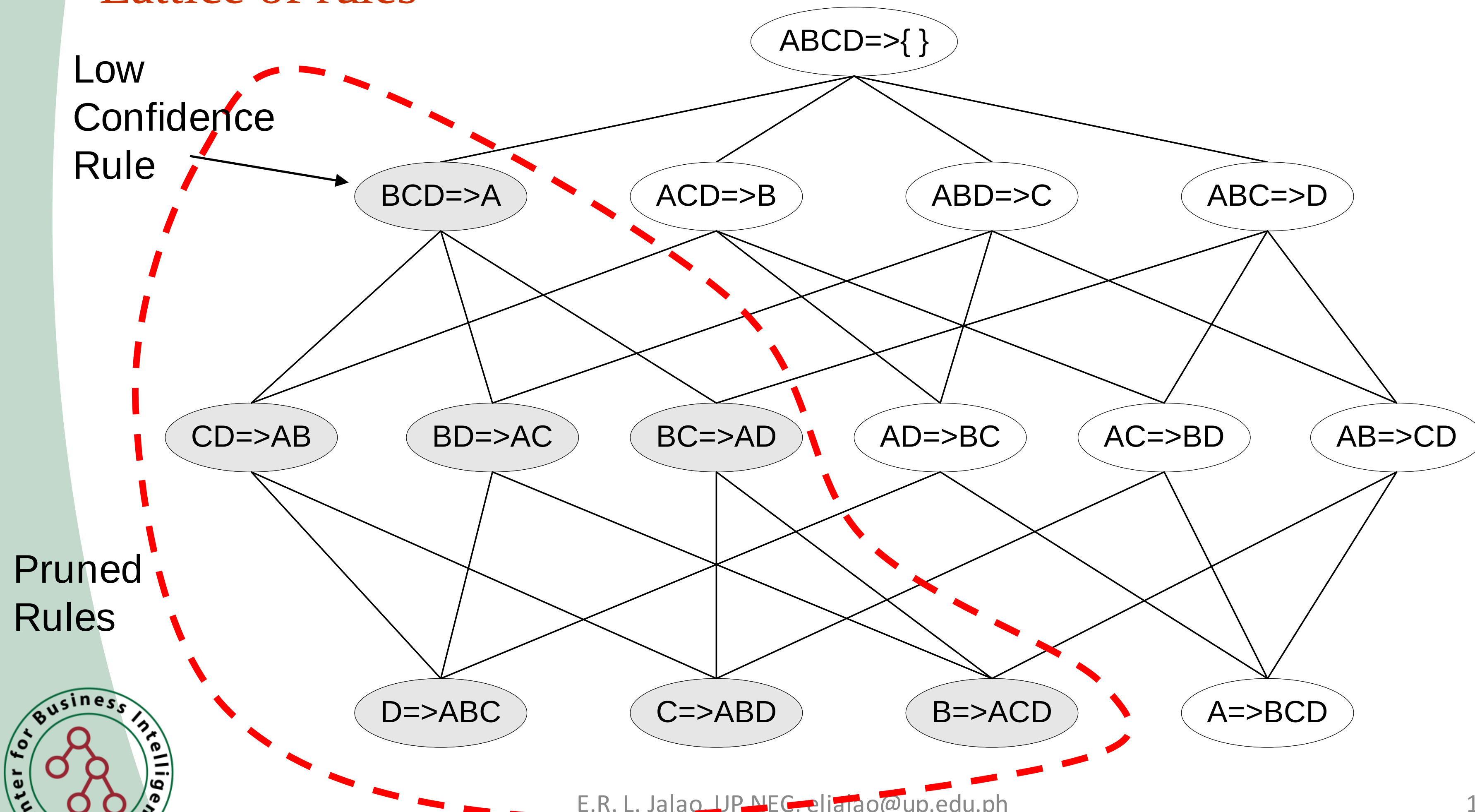
    $$c(ABC \rightarrow D) \geq c(AB \rightarrow CD) \geq c(A \rightarrow BCD)$$

    Confidence is anti-monotone w.r.t. number of items on the RHS of the rule

    $$\frac{\#ABCD}{\#ABC} \geq \frac{\#ABCD}{\#AB} \geq \frac{\#ABCD}{\#A}$$

Rule Generation for Apriori Algorithm

Lattice of rules



Lift Ratio

- High Confidence Rules can sometimes be misleading because the confidence measure ignores the support
- Lift is a value that gives information about the increase in probability of the consequent given the antecedent part.
- Greater lift values indicate stronger associations.

$$Lift = \frac{c(A \rightarrow B)}{s(B)}$$

Lift Ratio Example

TID	*Items*
1	**Bread, Milk**
2	**Bread, Diaper, Beer, Eggs**
3	**Milk, Diaper, Beer, Coke**
4	**Bread, Milk, Diaper, Beer**
5	**Bread, Milk, Diaper, Coke**

Example:

$$\{\text{Milk}, \text{Diaper}\} \Rightarrow \text{Beer}$$

$$c = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{\sigma(\text{Milk}, \text{Diaper})} = \frac{2}{3}$$

$$s = \frac{\sigma(\text{Beer})}{|\text{T}|} = \frac{3}{5}$$

$$Lift(\{Milk, Diaper\} \rightarrow \{Beer\}) = \frac{c(\{Milk, Diaper\} \rightarrow \{Beer\})}{s(\{Beer\})}$$

$$= \frac{2/3}{3/5} = \frac{10}{9}$$

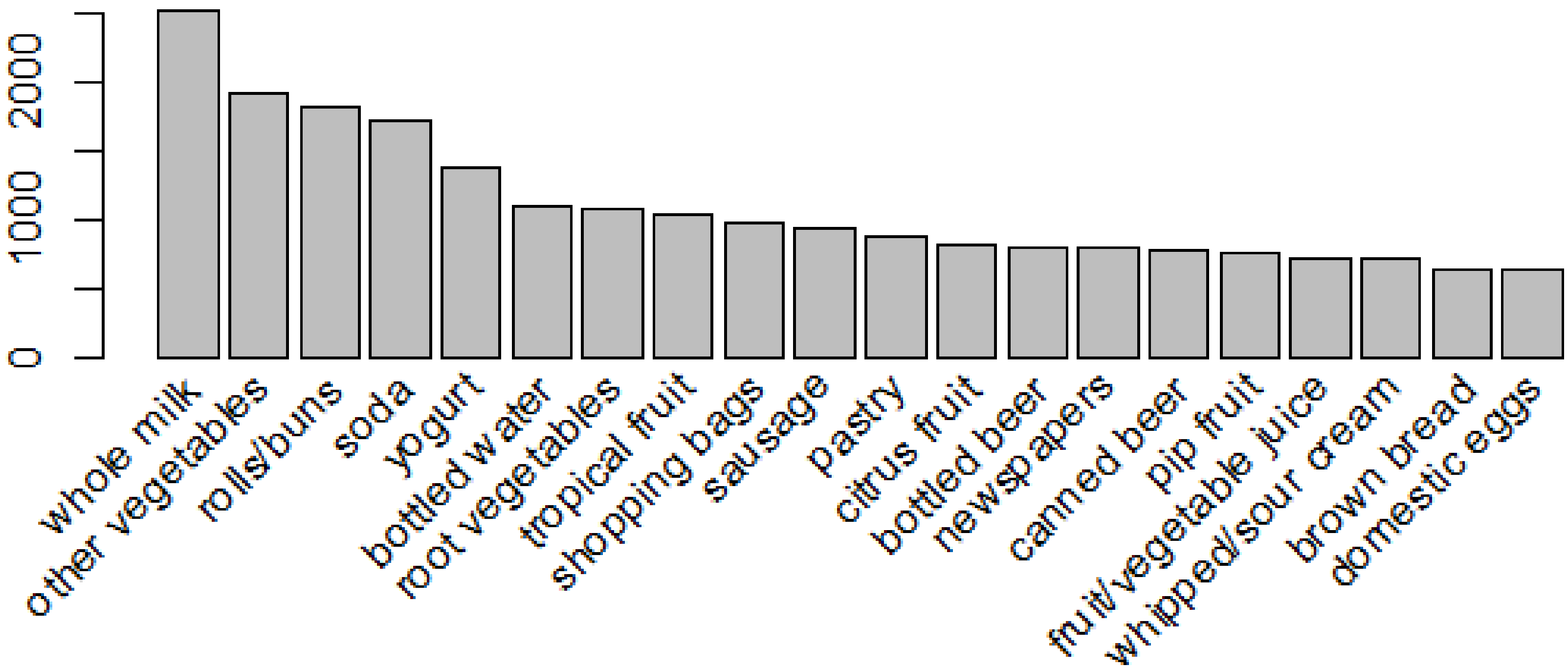
Business Scenario: Cross Selling

- Given a list of transactions from the supermarket's POS system, management would like to know which items can be bundled together as a promo
- Total Transactions: 9,835
- Total Unique Items: >161

Example: R Scripts

```
› library(arules)
› library(arulesViz)
› par(mar=c(2,2,2,2))
› groceries=read.transactions("groceries.csv",format="basket",sep=",")
› itemFrequencyPlot(groceries,topN=20,type="absolute")
```

Item Frequency Plot



Example: R Scripts

```
› rules = apriori(groceries, parameter =
  list(supp = 0.001, conf = 0.8))
› options(digits=2)
› inspect(rules[1:20])
› rules=sort(rules, by="confidence",
  decreasing=TRUE)
› inspect(rules[1:20])
```

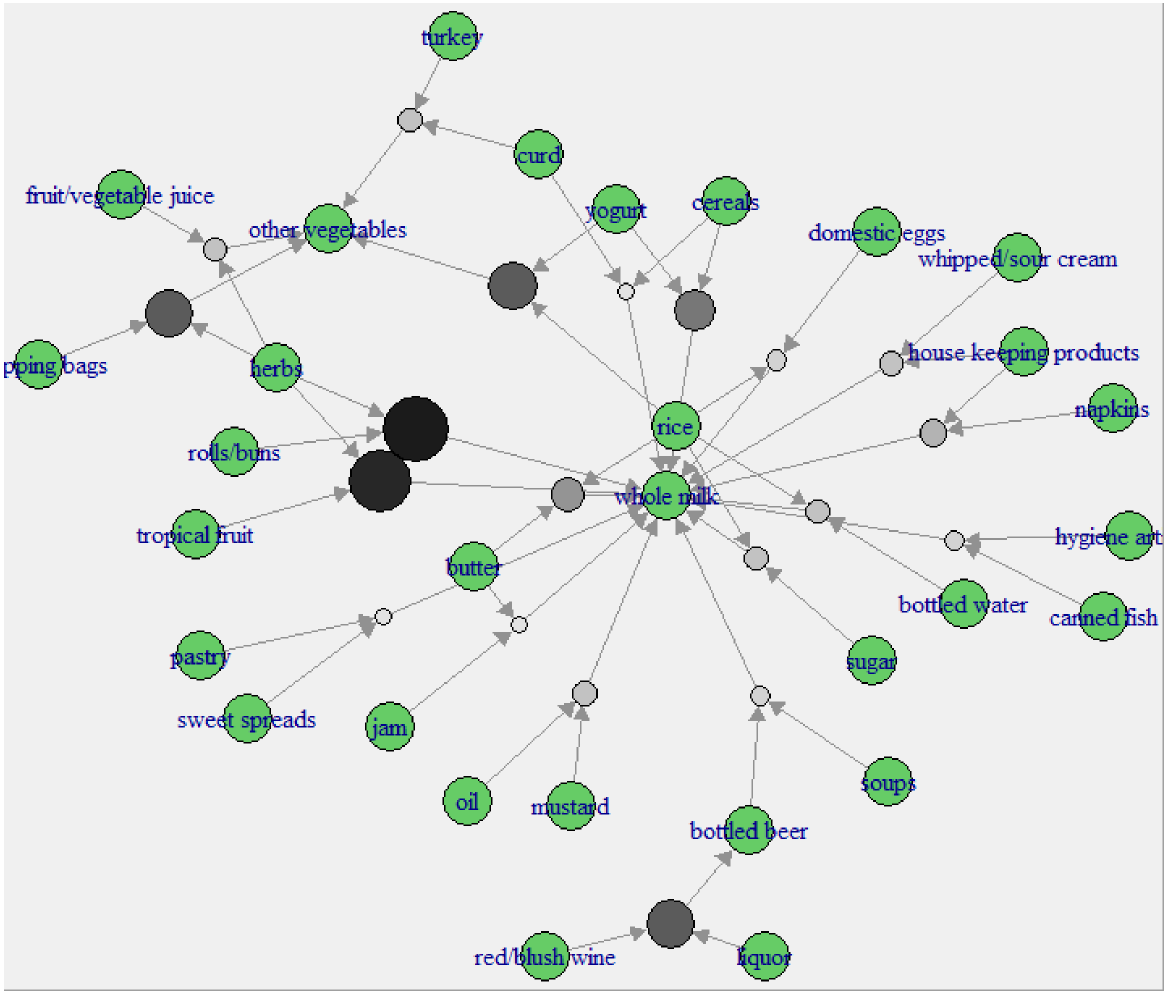
Rules

```
    lhs                        rhs                 support confidence lift
1   {rice,
     sugar}                 => {whole milk}         0.0012          1  3.9
2   {canned fish,
     hygiene articles}      => {whole milk}         0.0011          1  3.9
3   {butter,
     rice,
     root vegetables}       => {whole milk}         0.0010          1  3.9
4   {flour,
     root vegetables,
     whipped/sour cream}    => {whole milk}         0.0017          1  3.9
5   {butter,
     domestic eggs,
     soft cheese}           => {whole milk}         0.0010          1  3.9
```

Example: R Scripts

```
› plot(rules[1:20],method="graph",interac
  tive=TRUE,shading=T)
```

Association Graph



Outline for This Session

- Market Basket Analysis
- **Sequential Pattern Mining**
- Clustering
  - K-Means Clustering
  - Hierarchical Clustering
- Text Mining
- Social Media Sentiment Analysis
- Case Study

Formal Definition of a Sequence

- A sequence is an ordered list of elements (transactions)

$$s = < e_1 \; e_2 \; e_3 \; \ldots >$$

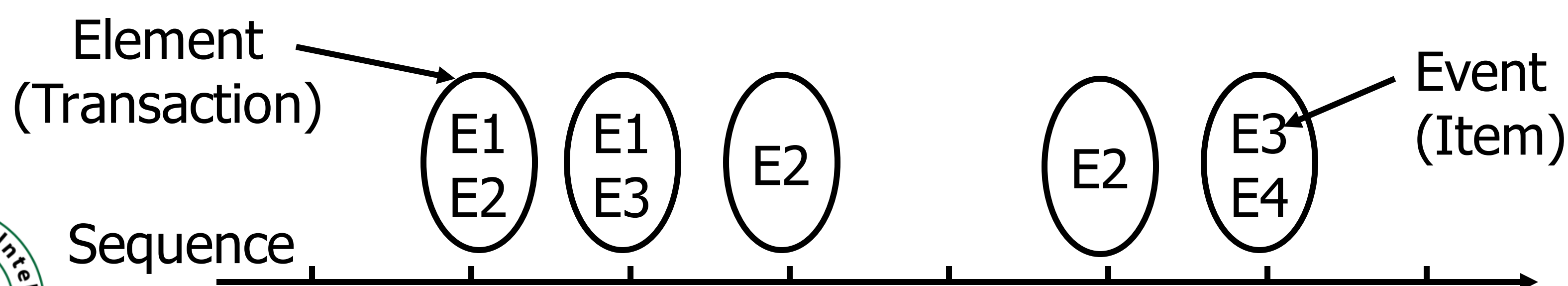
- Each element contains a collection of events (items)

$$e_i = \{i_1, i_2, \ldots, i_k\}$$

  - Each element is attributed to a specific time or location
- Length of a sequence, $|s|$, is given by the number of elements of the sequence
- A k-sequence is a sequence that contains k events (items)

Examples of Sequence Data

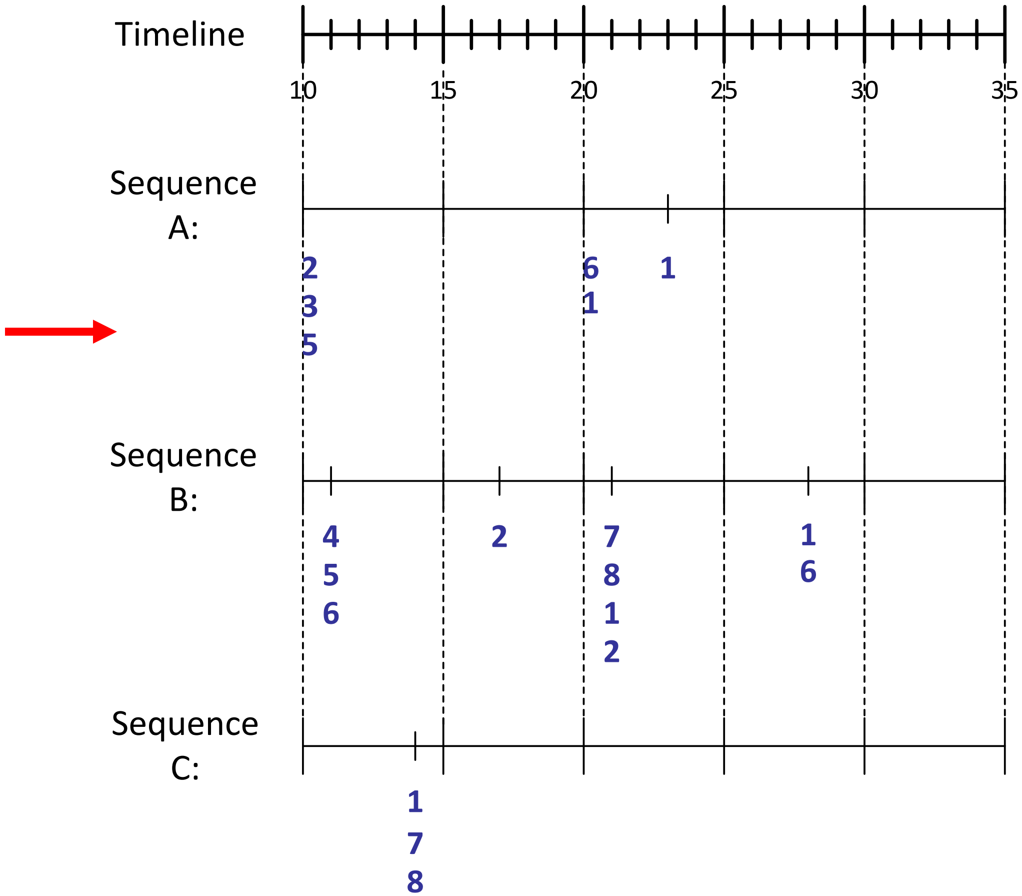
Sequence Database	Sequence	Element (Transaction)	Event (Item)
Customer	Purchase history of a given customer	A set of items bought by a customer at time t	Books, diary products, CDs, etc
Web Data	Browsing activity of a particular Web visitor	A collection of files viewed by a Web visitor after a single mouse click	Home page, index page, contact info, etc
Event data	History of events generated by a given sensor	Events triggered by a sensor at time t	Types of alarms generated by sensors
Genome sequences	DNA sequence of a particular species	An element of the DNA sequence	Bases A,T,G,C



Sequence Dataset Structure

Sequence Database:

Sequence	Timestamp	Events
1	10	2,3,5
1	20	6,1
1	23	1
2	11	4,5,6
2	17	2
2	21	7,8,1,2
2	28	1,6
3	14	1,8,7



Formal Definition of a Subsequence

- A sequence $< a_1 \, a_2 \, \ldots \, a_n >$ is contained in another sequence $< b_1 \, b_2 \, \ldots \, b_m >$ $(m \geq n)$ is called a subsequence if $a_i \subseteq b_i$

Data sequence	Subsequence	Contain?
< {2,4} {3,5,6} {8} >	< {2} {3,5} >	Yes
< {1,2} {3,4} >	< {1} {2} >	No
< {2,4} {2,4} {2,5} >	< {2} {4} >	Yes

- The support of a subsequence w is defined as the fraction of data sequences that contain w
- A sequential pattern is a frequent subsequence (i.e., a subsequence whose $support \geq minsup$)

Sequential Pattern Mining: Example

Sequence	Timestamp	Events
1	1	1,2,4
1	2	2,3
1	3	5
2	1	1,2
2	2	2,3,4
3	1	1,2
3	2	2,3,4
3	3	2,4,5
4	1	2
4	2	3,4
4	3	4,5
5	1	1,3
5	2	2,4,5

$$Minsup = 50\%$$

Examples of Frequent Subsequences:

< {1,2} >	s=60%
< {2,3} >	s=60%
< {2,4}>	s=80%
< {3} {5}>	s=80%
< {1} {2} >	s=80%
< {2} {2} >	s=60%
< {1} {2,3} >	s=60%
< {2} {2,3} >	s=60%
< {1,2} {2,3} >	s=60%

Sequential Pattern Mining: Definition

- Given:
  - a database of sequences
  - a user-specified minimum support threshold, *minsup*
- Task:
  - Find all subsequences with $support \geq minsup$

The SPADE Algorithm

- SPADE (Sequential PAttern Discovery using Equivalent Class) developed by Zaki 2001
- A vertical format sequential pattern mining method
- A sequence database is mapped to a large set of
  - Item: <SID, EID>
- Sequential pattern mining is performed by
  - growing the subsequences (patterns) one item at a time by Apriori candidate generation

Business Scenario

- Given the calls of subscribers to a hotline, a leading telecommunications company would like to profile its customers in terms of the sequence of their call transactions.
- This is done to reduce the number of calls to the hotline and divert other transactions to other self service channels.

Example

```
› library(arulesSequences)
› SequenceData =
  read.csv("hotline.csv",stringsAsFactors=FALSE)
› SequenceData$count = rep(1,nrow(SequenceData))
› SequenceData = subset(SequenceData,
  select=c("SUBID","TRANSID","count","TRANS"))
› write.table(SequenceData, "playdata.txt", sep="\t",
  col.names = F, row.names = F)
› Transactions = read_baskets("playdata.txt", info =
  c("sequenceID","eventID","SIZE"), sep="\t")
› SequenceRules = cspade(Transactions, parameter =
  list(support = 0.01), control = list(verbose = TRUE))
› summary(SequenceRules)
› as(SequenceRules, "data.frame")
```

Some Sequence Rules

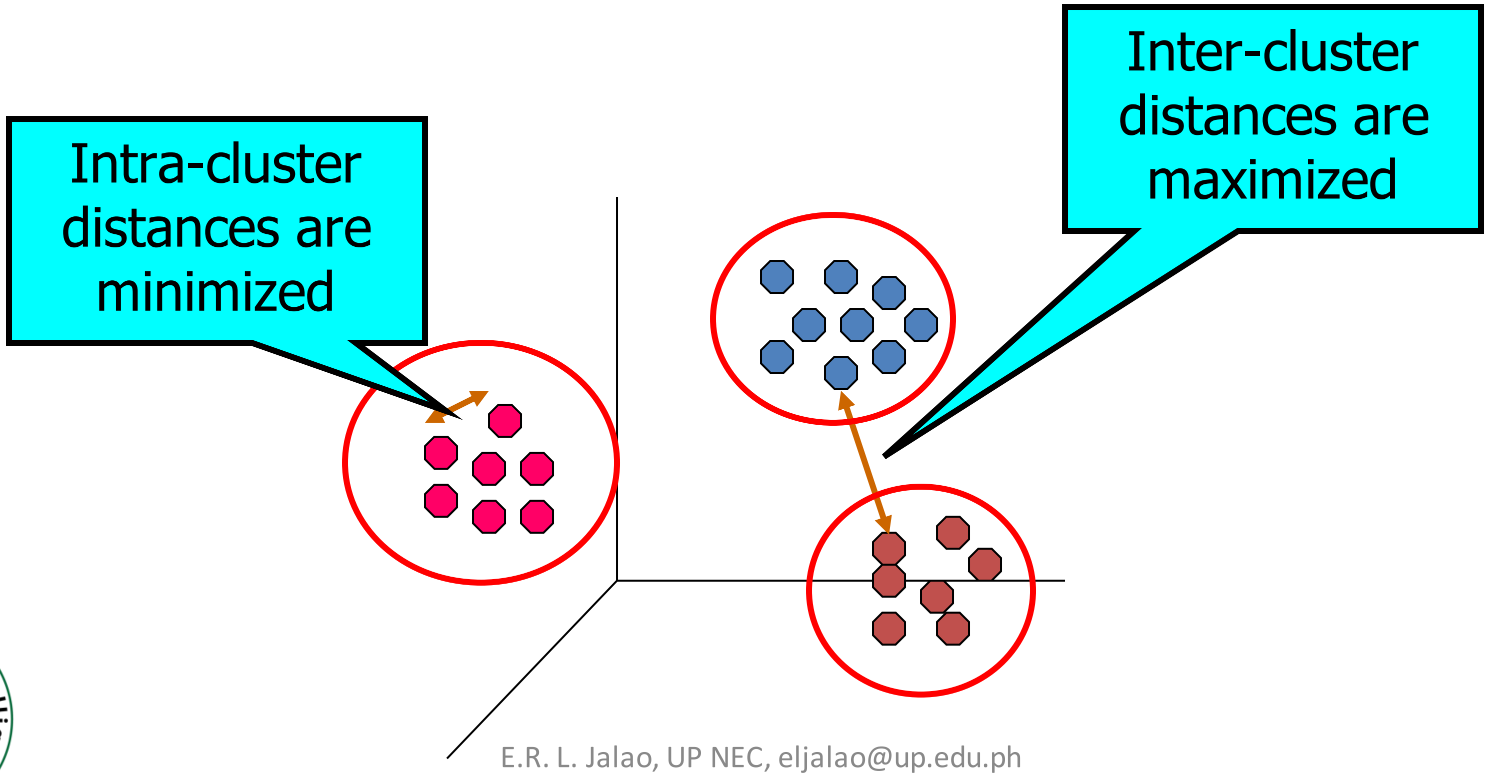
```
21 <{"DEVICE CONFIGURATION"},{"SUCCESSFUL NOT INTERESTED"}> 0.02270376
22     <{"DEVICE CONFIGURATION"},{"SUCCESSFUL INTERESTED"}> 0.03499870
23      <{"MECHANICS PROCEDURE"},{"SUCCESSFUL INTERESTED"}> 0.01403326
24                         <{"SHORT CALL"},{"SHORT CALL"}> 0.01384324
25        <{"MECHANICS PROCEDURE"},{"MECHANICS PROCEDURE"}> 0.01272424
26      <{"DEVICE CONFIGURATION"},{"DEVICE CONFIGURATION"}> 0.02080357
27     <{"SUCCESSFUL INTERESTED"},{"DEVICE CONFIGURATION"}> 0.01027511
28          <{"UNCOMPLETED CALL"},{"DEVICE CONFIGURATION"}> 0.01059180
29                <{"ACCOUNT DETAILS"},{"BILLING INQUIRY"}> 0.01052846
30                <{"BILLING INQUIRY"},{"BILLING INQUIRY"}> 0.02450542
31          <{"AFTERSALES REQUEST"},{"AFTERSALES REQUEST"}> 0.01010620
32             <{"BILLING INQUIRY"},{"AFTERSALES REQUEST"}> 0.01342098
33                <{"BILLING INQUIRY"},{"ACCOUNT DETAILS"}> 0.01133780
```

Outline for This Session

- Market Basket Analysis
- Sequential Pattern Mining
- **Clustering**
  - K-Means Clustering
  - Hierarchical Clustering
- Text Mining
- Social Media Sentiment Analysis
- Case Study

What is Cluster Analysis?

- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups



Examples of Clustering Applications

- **Marketing:** Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- **Land use:** Identification of areas of similar land use in an earth observation database
- **Insurance:** Identifying groups of motor insurance policy holders with a high average claim cost
- **City-planning:** Identifying groups of houses according to their house type, value, and geographical location
- **Earth-quake studies:** Observed earth quake epicenters should be clustered along continent faults

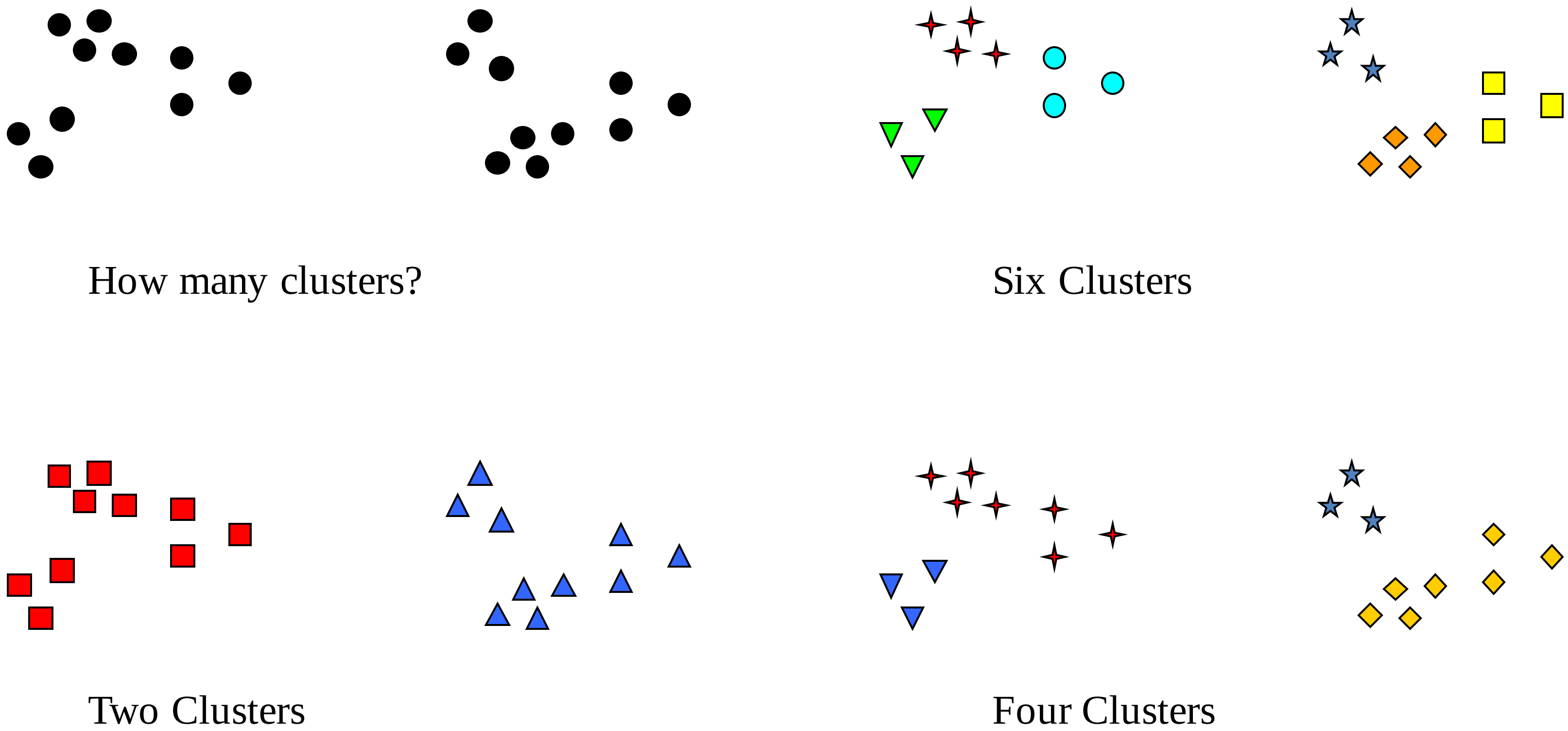
Examples of Clustering Applications

- Understanding
  - Group related documents for browsing, group genes and proteins that have similar functionality, or group stocks with similar price fluctuations
- Summarization
  - Reduce the size of large data sets
    - Reduce 1 Million Rows to 10,000 rows.

What is not Cluster Analysis?

- Supervised classification
  - Have class label information
- Simple segmentation
  - Dividing students into different registration groups alphabetically, by last name
- Results of a query
  - Groupings are a result of an external specification
- Graph partitioning
  - Some mutual relevance and synergy, but areas are not identical

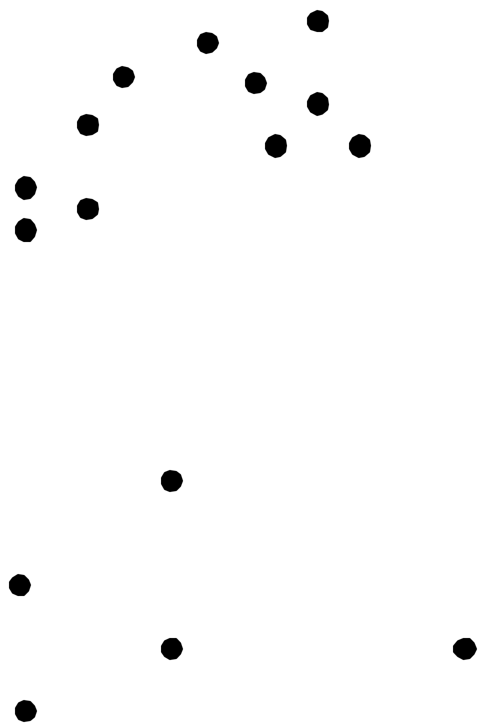
Notion of a Cluster can be Ambiguous



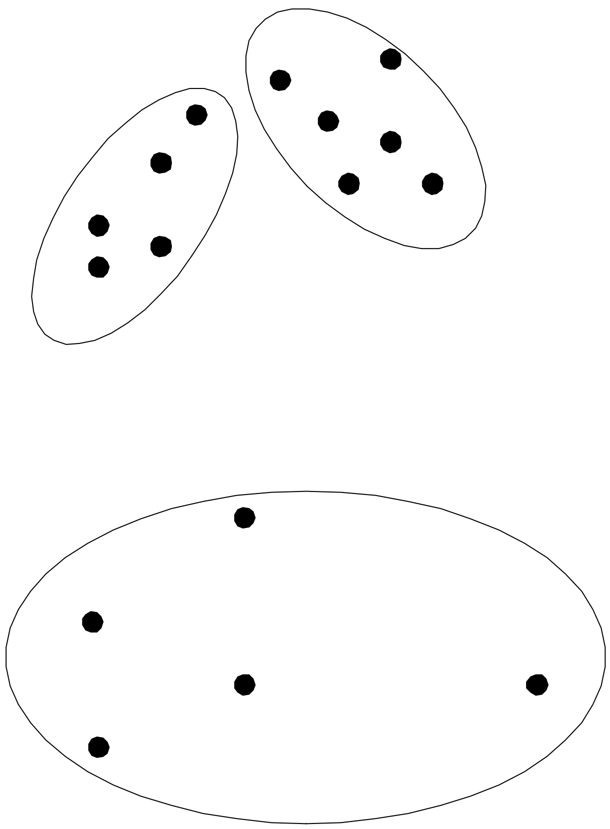
Types of Clustering Methodologies

- Partitional Clustering
  - A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset
- Hierarchical clustering
  - A set of nested clusters organized as a hierarchical tree

Partitional Clustering

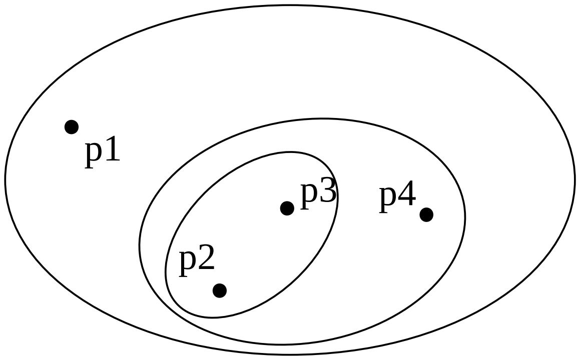


Original Points

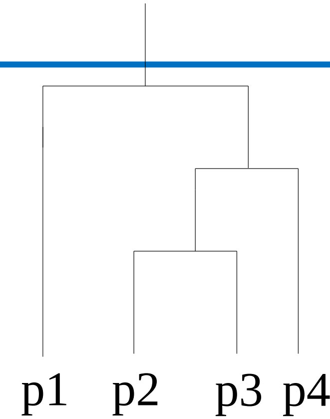


A Partitional Clustering

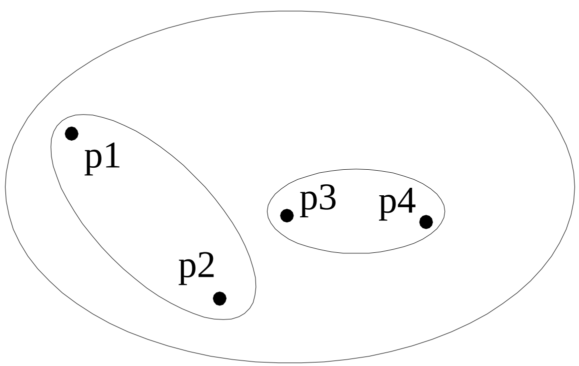
Hierarchical Clustering



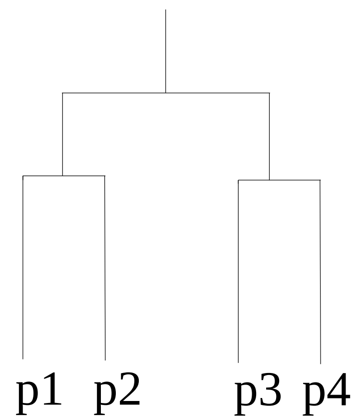
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

Outline for This Session

- Market Basket Analysis
- Sequential Pattern Mining
- Clustering
  - **K-Means Clustering**
  - Hierarchical Clustering
- Text Mining
- Social Media Sentiment Analysis
- Case Study

K-means Clustering

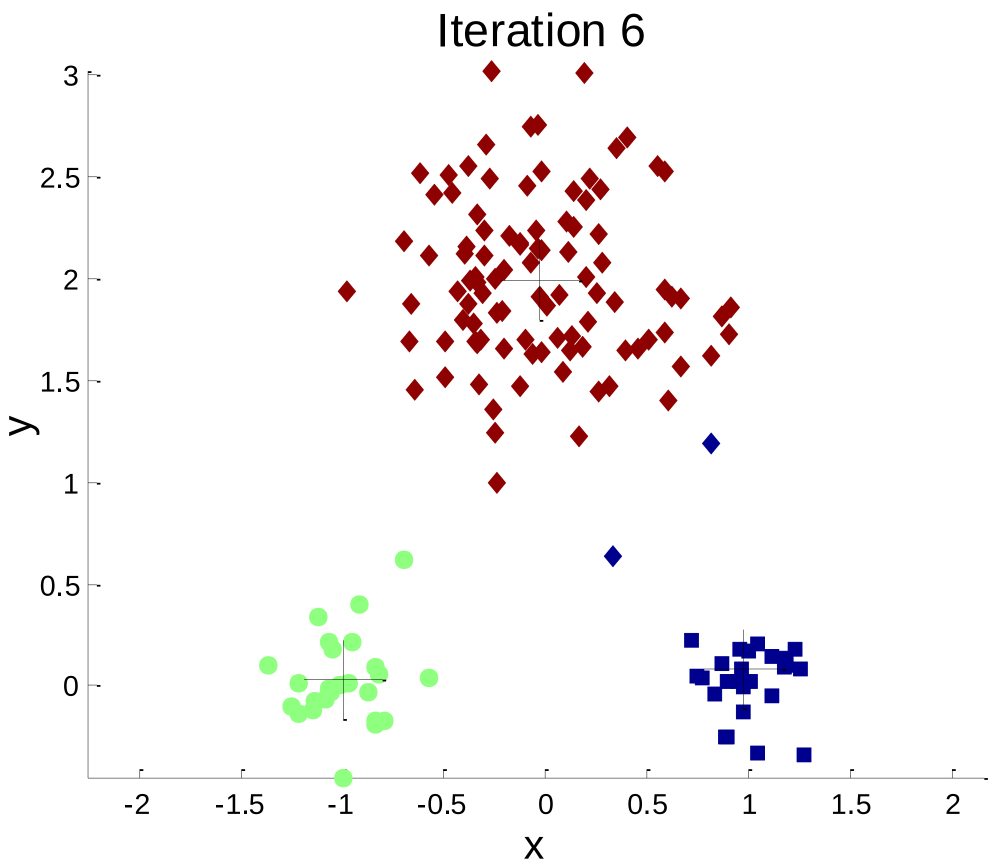
- Partitional clustering approach
  - Each cluster is associated with a centroid (center point)
  - Each point is assigned to the cluster with the closest centroid
  - Number of clusters, K, must be specified
  - The basic algorithm is very simple

```
Select K points as the initial centroids.
repeat
    Form K clusters by assigning all points to the closest centroid.
    Recompute the centroid of each cluster.
until The centroids don't change
```

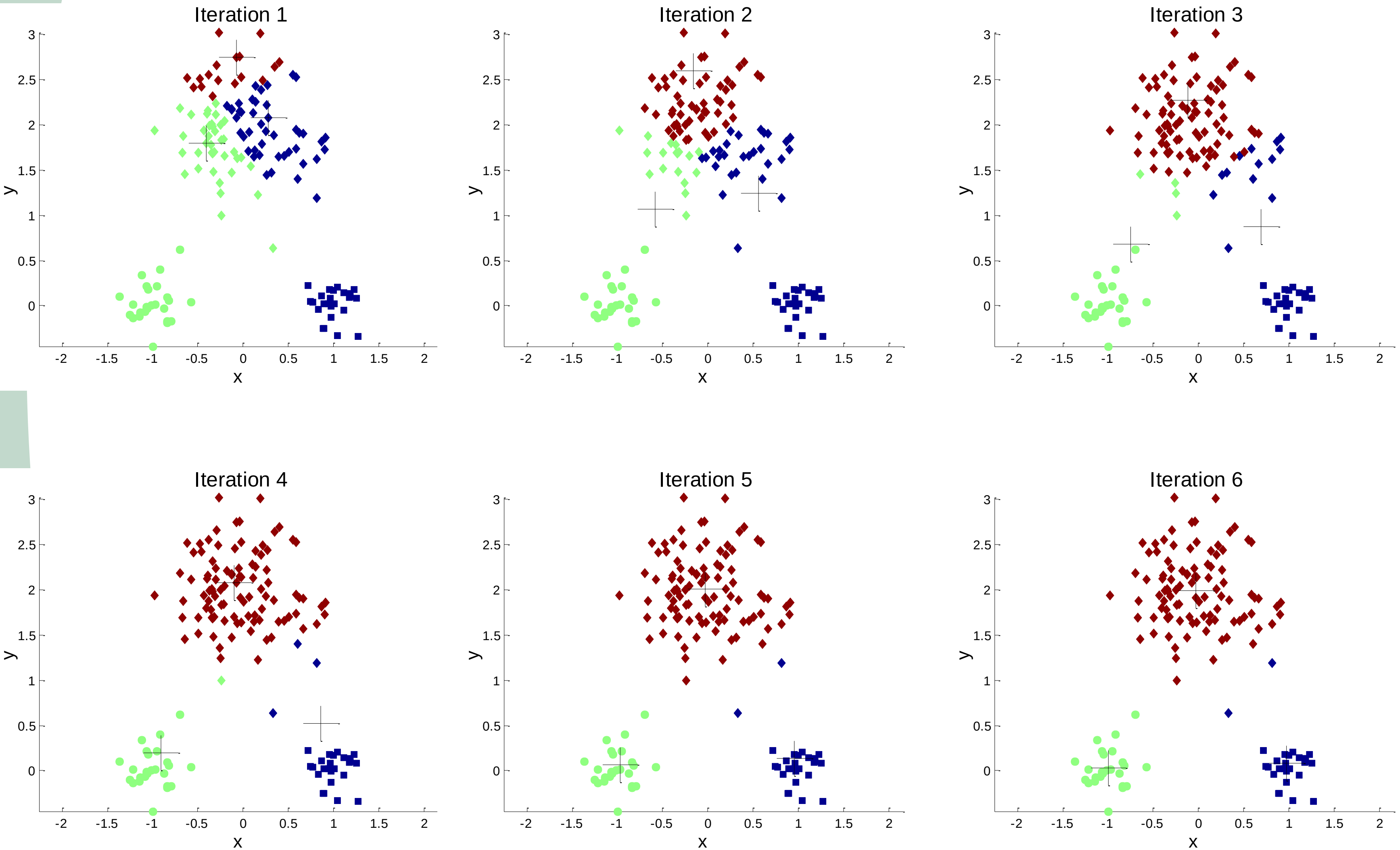
K-means Clustering – Details

- Initial centroids are often chosen randomly.
  - Clusters produced vary from one run to another.
- The centroid is the mean of the points in the cluster.
- Closeness is measured by Euclidean distance, cosine similarity, correlation, etc.
- K-means will converge for common similarity measures mentioned above.
- Convergence happens in the first few iterations.
  - Often the stopping condition is changed to 'Until relatively few points change clusters'

Overview of K-Means



Overview of K-Means



Pre-processing and Post-processing

- Pre-processing
  - Normalize the data
  - Eliminate outliers
- Post-processing
  - Eliminate small clusters that may represent outliers
  - Split 'loose' clusters, i.e., clusters
  - Merge clusters that are 'close'

Business Scenario

- Given 33 cars and their respective profiles in terms of: mpg, cylinders, displacement, horsepower, weight, number of gears, carburators, etc.
- Which cars are similar in terms of these factors?
- Which cars can be grouped together?

Example: Cars DataSet

```
› cars=read.csv("cars.csv")
› rownames(cars) = cars[,1]
› cars = cars[,c(2:12)]
› fit = kmeans(cars, 5)
› aggregate(cars,by=list(fit$cluster),FUN
  =mean)
› carswithclusters = data.frame(cars,
  fit$cluster)
› carswithclusters
```

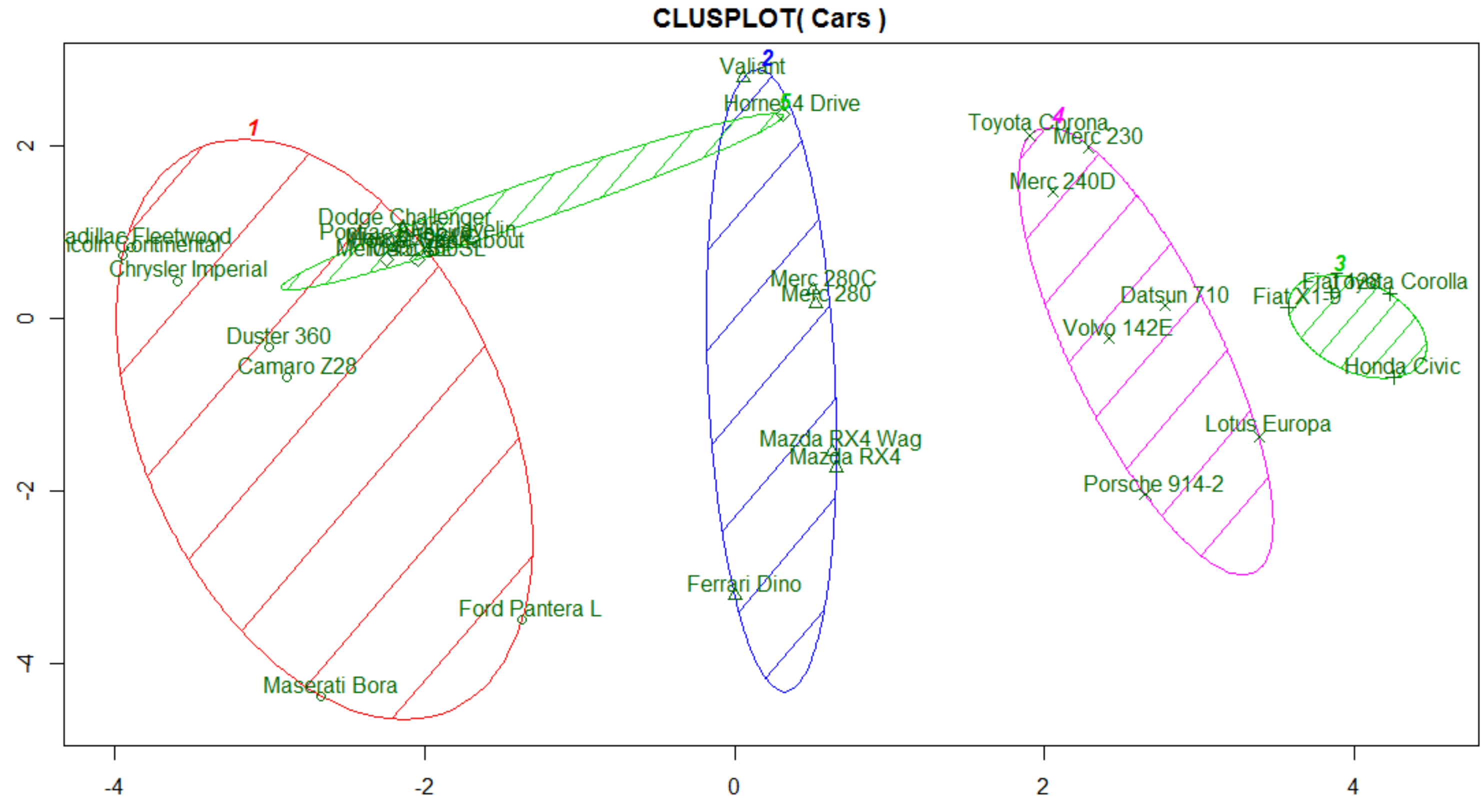
Cluster Means and Assignments

	Group.1	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
1	1	15	8.0	388	232	3.3	4.2	16	0.00	0.22	3.4	4.0
2	2	19	6.0	171	124	3.7	3.1	18	0.50	0.50	4.0	3.8
3	3	31	4.0	76	62	4.3	1.9	19	1.00	1.00	4.0	1.2
4	4	24	4.0	122	94	3.9	2.5	19	0.86	0.57	4.1	1.7
5	5	17	7.7	285	158	3.0	3.6	18	0.17	0.00	3.0	2.3

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb	fit.cluster
Mazda RX4	21	6	160	110	3.9	2.6	16	0	1	4	4	2
Mazda RX4 Wag	21	6	160	110	3.9	2.9	17	0	1	4	4	2
Datsun 710	23	4	108	93	3.8	2.3	19	1	1	4	1	4
Hornet 4 Drive	21	6	258	110	3.1	3.2	19	1	0	3	1	5
Hornet Sportabout	19	8	360	175	3.1	3.4	17	0	0	3	2	1
Valiant	18	6	225	105	2.8	3.5	20	1	0	3	1	2

Visualization

```
› library(cluster)
› clusplot(cars, fit$cluster, color=TRUE,
  shade=TRUE, labels=2, lines=0)
```

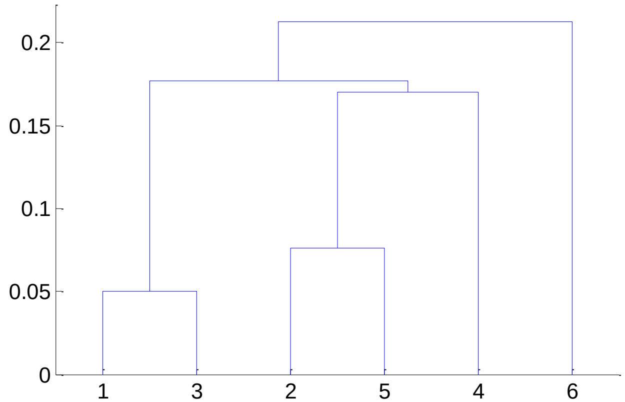


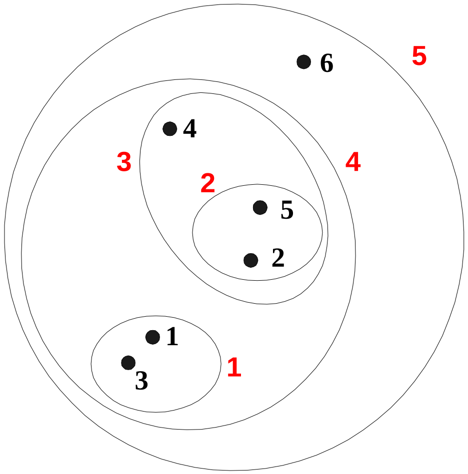
Outline for This Session

- Market Basket Analysis
- Sequential Pattern Mining
- Clustering
  - K-Means Clustering
  - **Hierarchical Clustering**
- Text Mining
- Social Media Sentiment Analysis
- Case Study

Hierarchical Clustering

- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequences of merges or splits





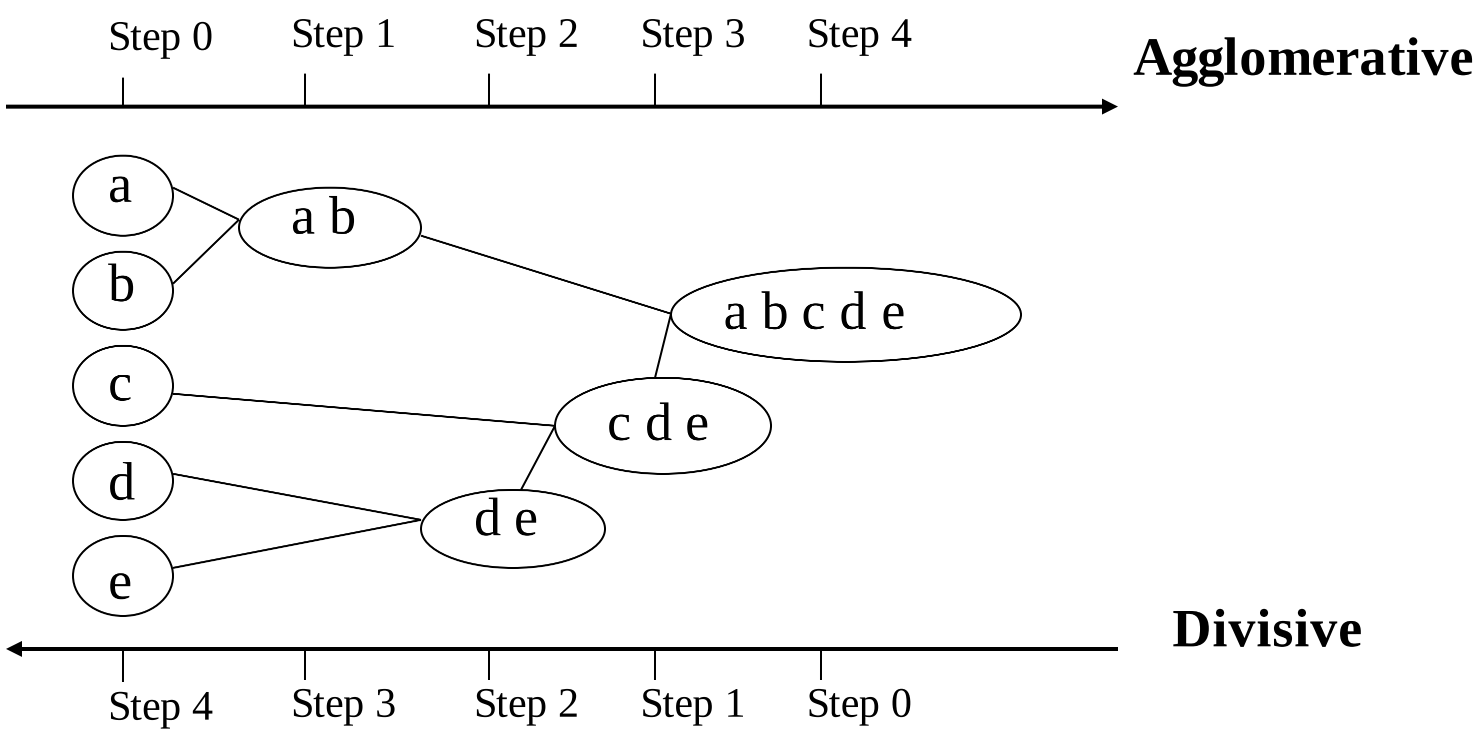
Strengths of Hierarchical Clustering

- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by 'cutting' the dendogram at the proper level

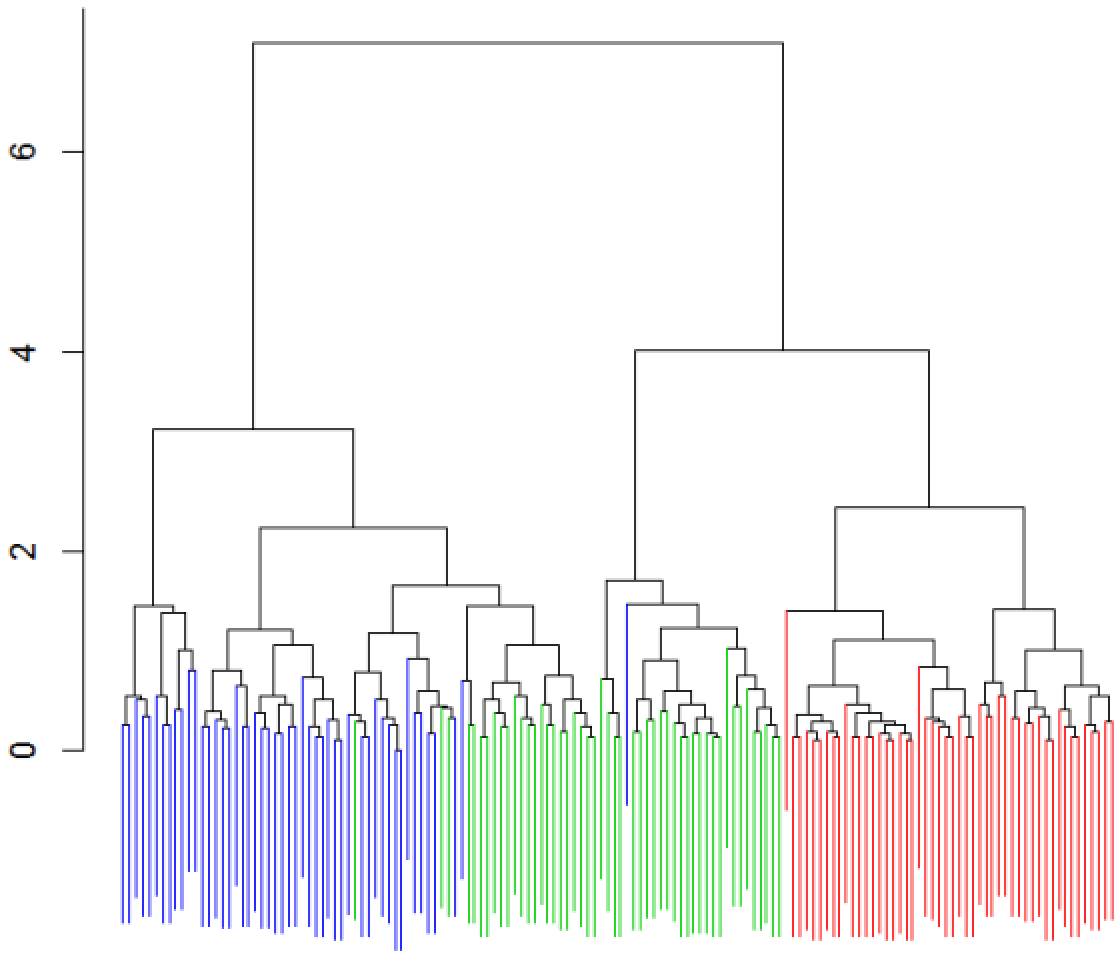
- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)

Hierarchical Clustering

- Illustrative Example
  - Agglomerative and divisive clustering on the data set {a, b, c, d ,e }

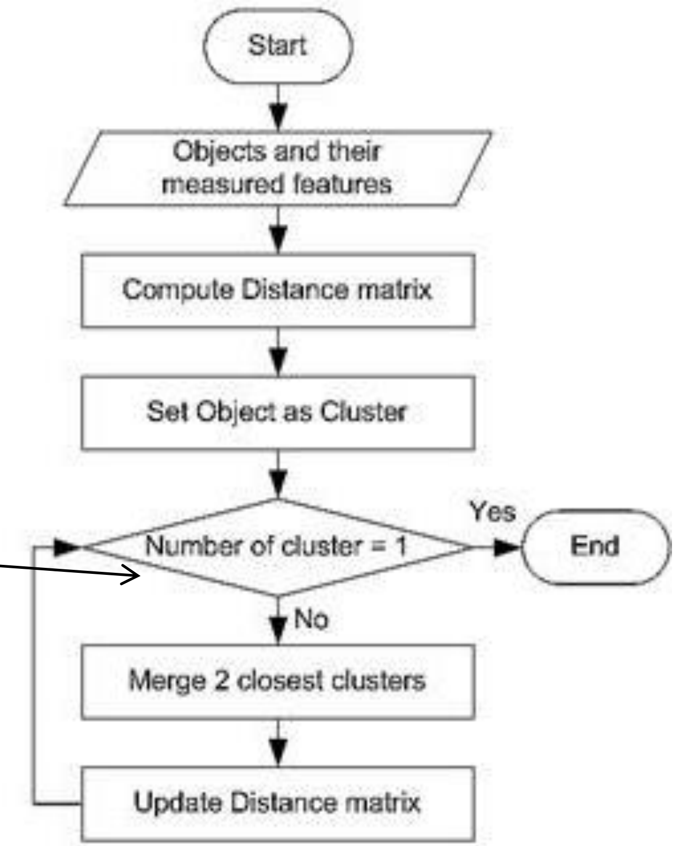


Example



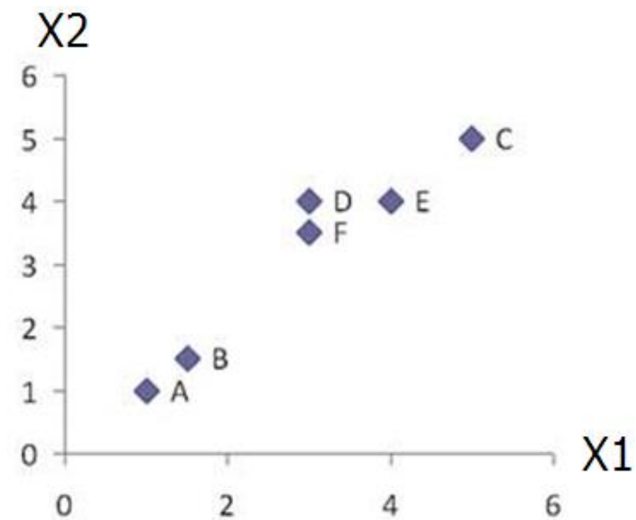
Agglomerative Algorithm

- The Agglomerative algorithm is carried out in three steps:
  - Convert all object features into a **distance matrix**
  - Set each object **as a cluster** (thus if we have *N* objects, we will have *N* clusters at the beginning)
  - Repeat until number of cluster is one (or known # of clusters)
    - Merge two closest clusters
    - Update "distance matrix"



Example

- Problem: clustering analysis with agglomerative algorithm



	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5

data matrix

$$d_{AB} = \left((1-1.5)^2 + (1-1.5)^2\right)^{\frac{1}{2}} = \sqrt{\tfrac{1}{2}} = 0.7071$$

$$d_{DF} = \left((3-3)^2 + (4-3.5)^2\right)^{\frac{1}{2}} = 0.5$$

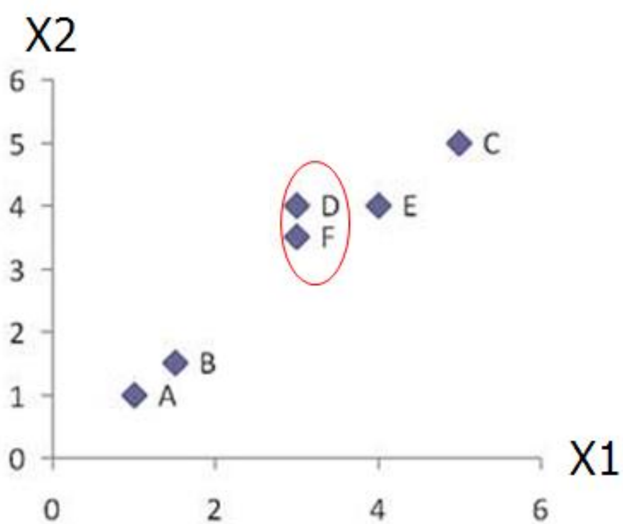
Euclidean distance

Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

distance matrix

Example

- Merge two closest clusters (iteration 1)

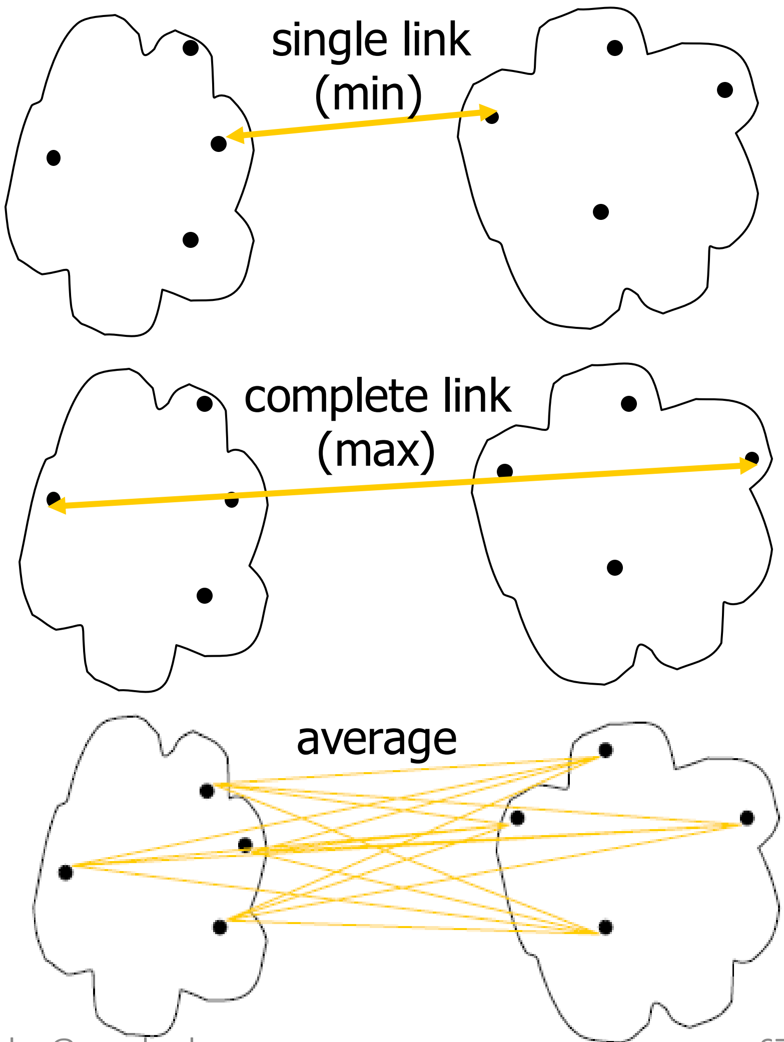


Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	?	4.24
B	0.71	0.00	4.95	?	3.54
C	5.66	4.95	0.00	?	1.41
D, F	?	?	?	0.00	?
E	4.24	3.54	1.41	?	0.00

Cluster Distance Measures

- **Single link:** smallest distance between an element in one cluster and an element in the other, i.e., $d(Ci, Cj) = \min\{d(xip, xjq)\}$
- **Complete link:** largest distance between an element in one cluster and an element in the other, i.e., $d(Ci, Cj) = \max\{d(xip, xjq)\}$
- **Average:** avg distance between elements in one cluster and elements in the other, i.e., $d(Ci, Cj) = avg\{d(xip, xjq)\}$



d(C, C)=0

Example

- Update distance matrix (iteration 1)
-

Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

$$d_{(D,F)\mapsto A} = \min\left(d_{DA}, d_{FA}\right) = \min\left(3.61, 3.20\right) = 3.20$$

$$d_{(D,F)\mapsto B} = \min\left(d_{DB}, d_{FB}\right) = \min\left(2.92, 2.50\right) = 2.50$$

$$d_{(D,F)\mapsto C} = \min\left(d_{DC}, d_{FC}\right) = \min\left(2.24, 2.50\right) = 2.24$$

$$d_{E\to(D,F)} = \min\left(d_{ED}, d_{EF}\right) = \min\left(1.00, 1.12\right) = 1.00$$

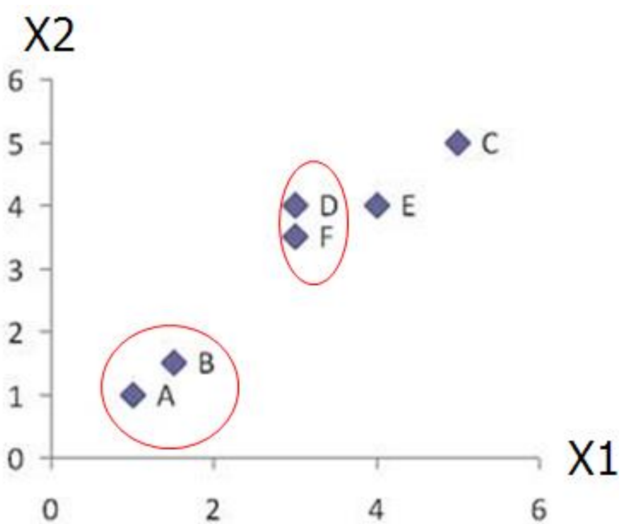
Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	?	4.24
B	0.71	0.00	4.95	?	3.54
C	5.66	4.95	0.00	?	1.41
D, F	?	?	?	0.00	?
E	4.24	3.54	1.41	?	0.00

Min Distance (Single Linkage)

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	1.00
E	4.24	3.54	1.41	**1.00**	0.00

Example

- Merge two closest clusters (iteration 2)
-



Min Distance (Single Linkage)

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	1.00
E	4.24	3.54	1.41	**1.00**	0.00

Dist	A,B	C	(D, F)	E
A,B	0	?	?	?
C	**?**	0	2.24	1.41
(D, F)	**?**	2.24	0	1.00
E	**?**	1.41	1.00	0

Example

- Update distance matrix (iteration 2)

Min Distance (Single Linkage)

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	1.00
E	4.24	3.54	1.41	**1.00**	0.00

$$d_{C\rightarrow(A,B)} = \min\left(d_{CA}, d_{CB}\right) = \min\left(5.66, 4.95\right) = 4.95$$

$$d_{(D,F)\rightarrow(A,B)} = \min\left(d_{DA}, d_{DB}, d_{FA}, d_{FB}\right) = \min\left(3.61, 2.92, 3.20, 2.50\right) = 2.50$$

$$d_{E\rightarrow(A,B)} = \min\left(d_{EA}, d_{EB}\right) = \min\left(4.24, 3.54\right) = 3.54$$

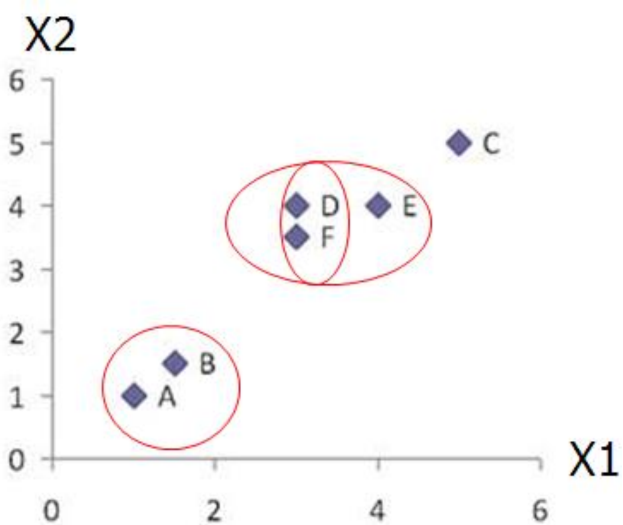
Dist	A,B	C	(D, F)	E
A,B	0	?	?	?
C	**?**	0	2.24	1.41
(D, F)	**?**	2.24	0	1.00
E	**?**	1.41	1.00	0

Min Distance (Single Linkage)

Dist	A,B	C	(D, F)	E
A,B	0	4.95	2.50	3.54
C	**4.95**	0	2.24	1.41
(D, F)	**2.50**	2.24	0	1.00
E	**3.54**	1.41	1.00	0

Example

- Merge two closest clusters/update distance matrix (iteration 3)



Min Distance (Single Linkage)

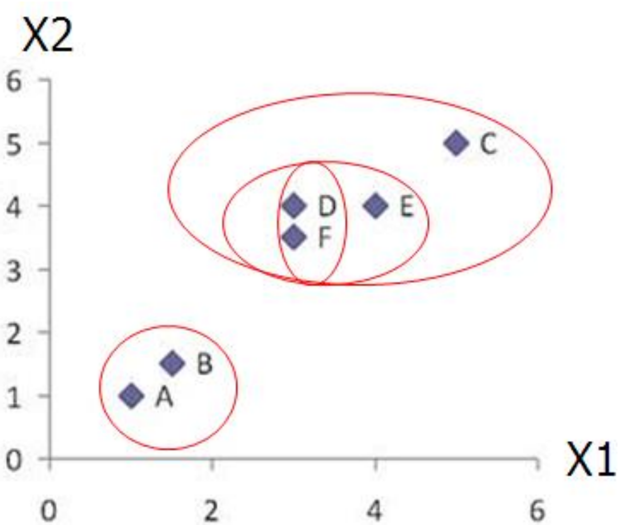
Dist	A,B	C	(D, F)	E
A,B	0	4.95	2.50	3.54
C	**4.95**	0	2.24	1.41
(D, F)	**2.50**	2.24	0	1.00
E	**3.54**	1.41	1.00	0

Min Distance (Single Linkage)

Dist	(A,B)	C	(D, F), E
(A,B)	0.00	4.95	2.50
C	4.95	0.00	1.41
(D, F), E	**2.50**	**1.41**	0.00

Example

- Merge two closest clusters/update distance matrix (iteration 4)
-



Min Distance (Single Linkage)

Dist	(A,B)	C	(D, F), E
(A,B)	0.00	4.95	2.50
C	4.95	0.00	1.41
(D, F), E	**2.50**	**1.41**	0.00

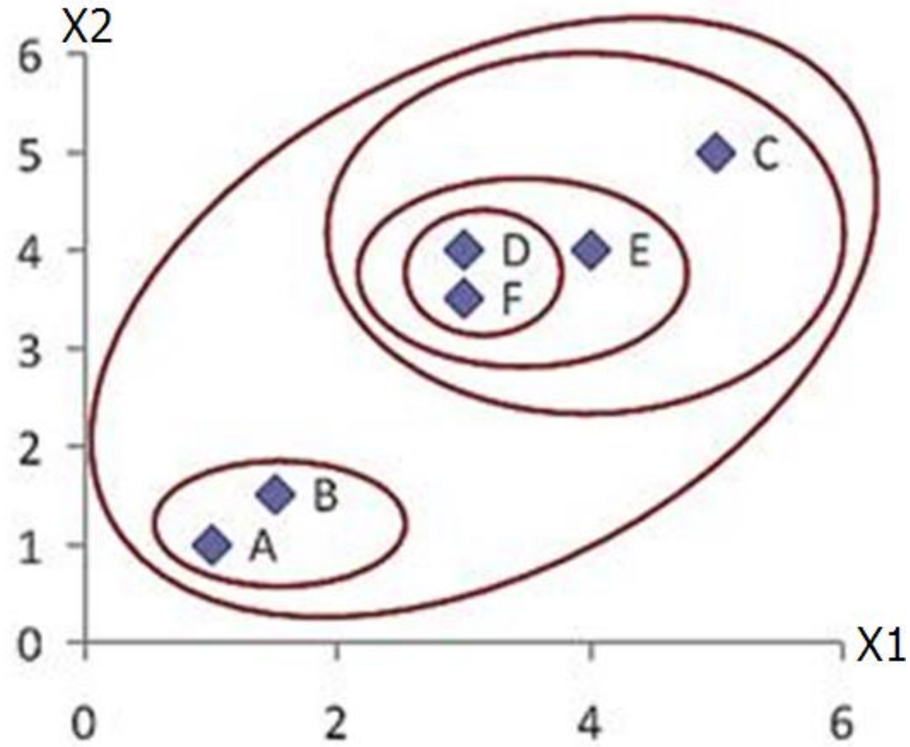
Min Distance (Single Linkage)

Dist	(A,B)	((D, F), E),C
(A,B)	0.00	2.50
((D, F), E),C	**2.50**	0.00

Example

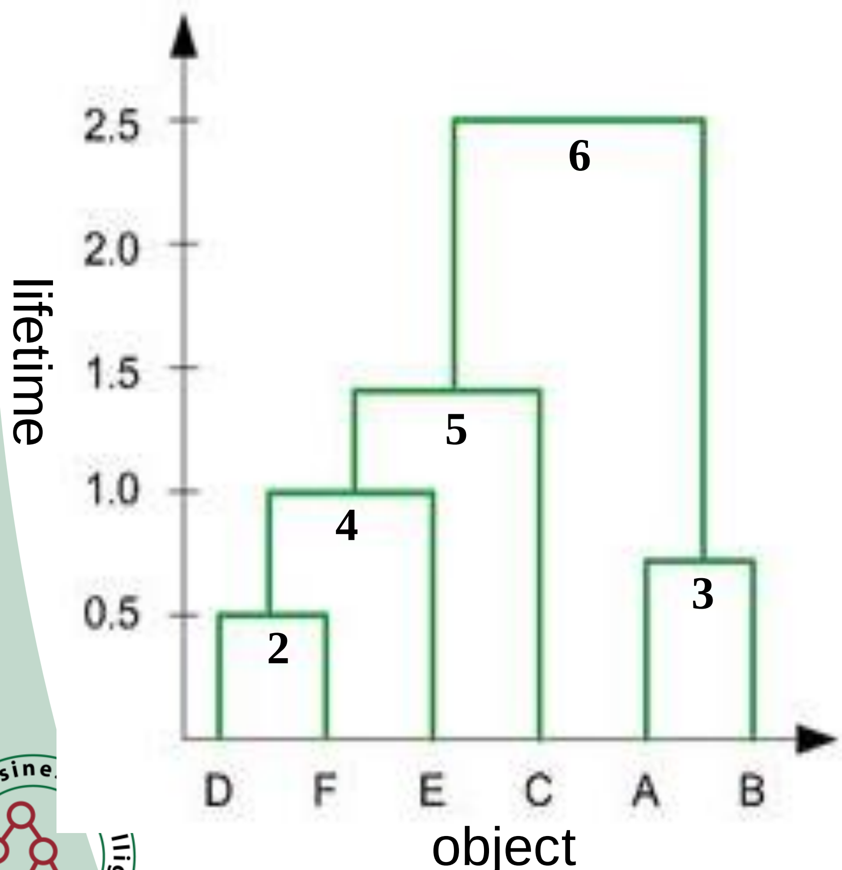
- Final result (meeting termination condition)

	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5



Example

- Dendrogram tree representation



1. In the beginning we have 6 clusters: A, B, C, D, E and F
2. We merge clusters D and F into cluster (D, F) at distance 0.50
3. We merge cluster A and cluster B into (A, B) at distance 0.71
4. We merge clusters E and (D, F) into ((D, F), E) at distance 1.00
5. We merge clusters ((D, F), E) and C into (((D, F), E), C) at distance 1.41
6. We merge clusters (((D, F), E), C) and (A, B) into ((((D, F), E), C), (A, B)) at distance 2.50
7. The last cluster contain all the objects, thus conclude the computation

Cluster Similarity: Ward's Method

- Similarity of two clusters is based on the increase in squared error when two clusters are merged
  - Similar to group average if distance between points is distance squared
- Less susceptible to noise and outliers
- Biased towards globular clusters
- Hierarchical analogue of K-means
  - Can be used to initialize K-means

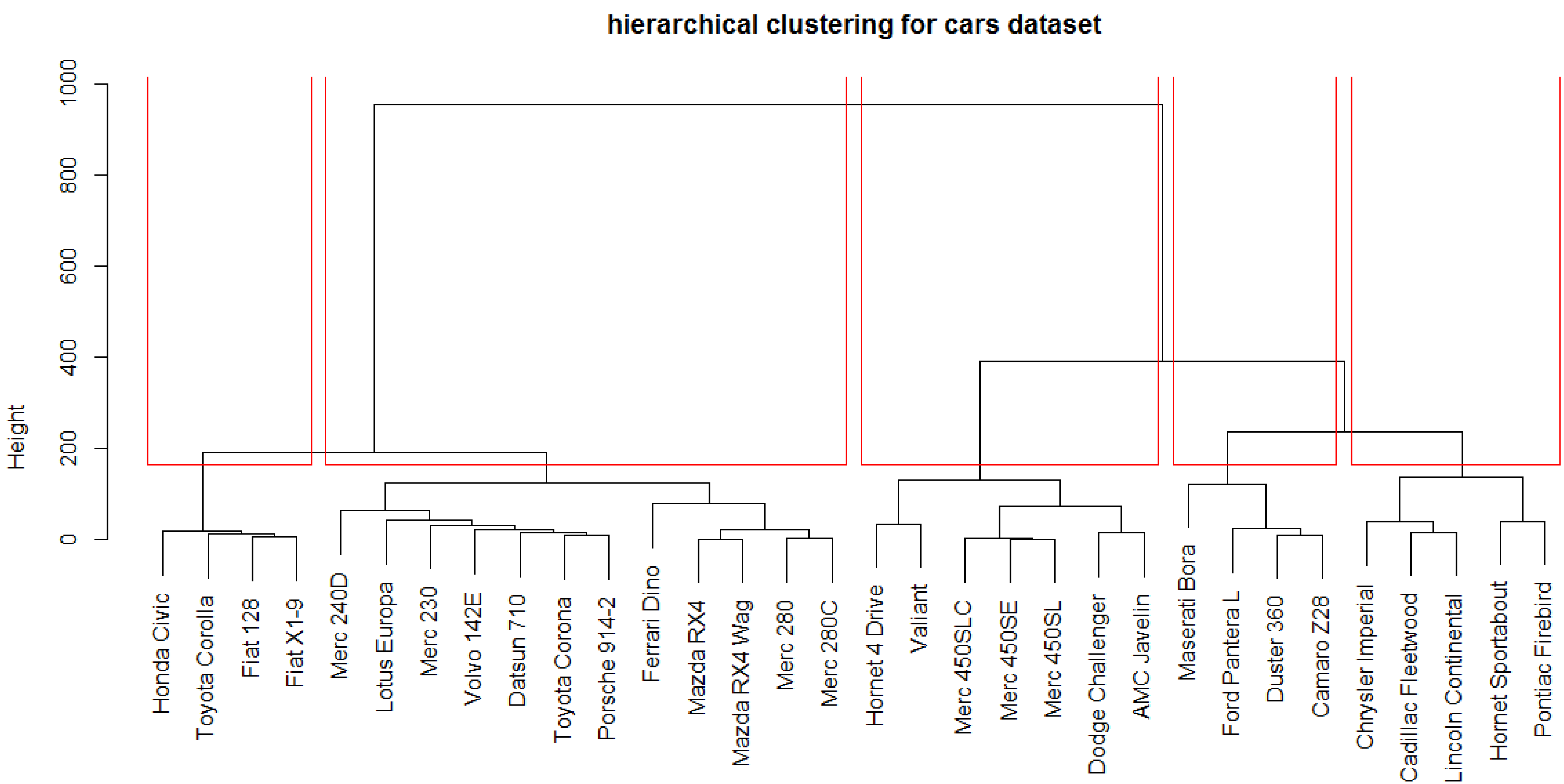
Hierarchical Clustering: Problems and Limitations

- Once a decision is made to combine two clusters, it cannot be undone
- No objective function is directly minimized
- Different schemes have problems with one or more of the following:
  - Sensitivity to noise and outliers
  - Difficulty handling different sized clusters and convex shapes
  - Breaking large clusters

R Code

- `cars=read.csv("cars.csv")`
- `rownames(cars) = cars[,1]`
- `cars = cars[,c(2:12)]`
- `d = dist(cars, method = "euclidean")`
- `fit = hclust(d, method="ward")`
- `plot(fit, main="hierarchical clustering for cars dataset")`
- `groups = cutree(fit, k=5)`
- `rect.hclust(fit, k=5, border="red")`

Example of Clustering



Outline for This Session

- Market Basket Analysis
- Sequential Pattern Mining
- Clustering
  - K-Means Clustering
  - Hierarchical Clustering
- **Text Mining**
- Social Media Sentiment Analysis
- Case Study

Text Mining Concepts

- 85-90 percent of all corporate data is in some kind of unstructured form (e.g., text)
- Unstructured corporate data is doubling in size every 18 months
- Tapping into these information sources is not an option, but a need to stay competitive
- Answer: text mining
  - A semi-automated process of extracting knowledge from unstructured data sources
  - a.k.a. text data mining or knowledge discovery in textual databases

Text Mining Concepts

- Benefits of text mining are obvious especially in text-rich data environments
  - e.g., law (court orders), academic research (research articles), finance (quarterly reports), medicine (discharge summaries), biology (molecular interactions), technology (patent files), marketing (customer comments), etc.
- Electronic communization records (e.g., Email)
  - Spam filtering
  - Email prioritization and categorization
  - Automatic response generation

Text Mining Terminology

- Unstructured or semistructured data
  - Data does not have a predetermined format and stored in documents
- Corpus (Corpha)
  - Large collection of structured texts for knowledge discovery
- Stemming
  - The process of reducing inflected words to their stem. Stemmer, stemming, stemmed are all based on the root stem.

Text Mining Terminology

- Stop Words
  - Words that are filtered out prior to or after processing of natural language data (a, am, the, of…)
- Term
  - A single word or phrase extracted from the corpus
- Tokenizing
  - A token is a categorized block of text in a sentence. The assignment of meanings to blocks of text is called tokenizing
- Term-by-document matrix
  - Occurrence matrix

Bag-of-Tokens Approaches

Documents

Four score and seven years ago our fathers brought forth on this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation, or …

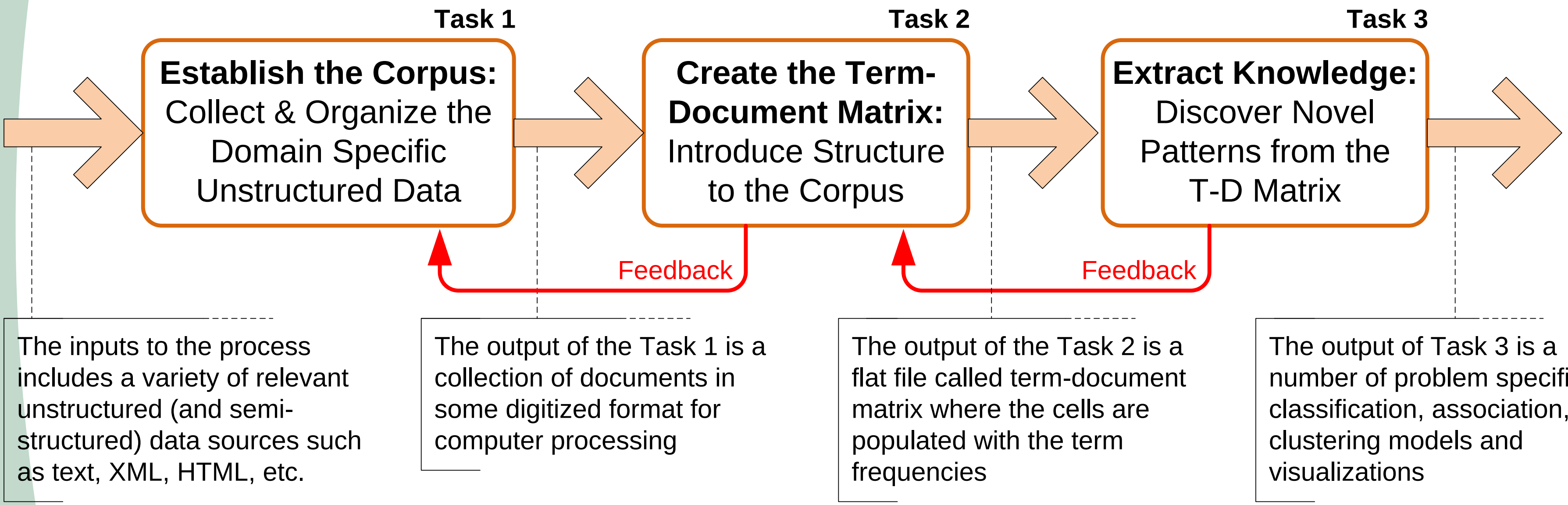
Feature Extraction

Token Sets

nation – 5
civil - 1
war – 2
men – 2
died – 4
people – 5
Liberty – 1
God – 1
…

Loses all order-specific information!
Severely limits context!

Text Mining Process



The three-step text mining process

Text Mining Process

- Step 1: Establish the corpus
  - Collect all relevant unstructured data
    - (e.g., textual documents, XML files, emails, Web pages, short notes, voice recordings…)
  - Digitize, standardize the collection
    - (e.g., all in ASCII text files)
  - Place the collection in a common place
    - (e.g., in a flat file, or in a directory as separate files)

Text Mining Process

- Step 2: Create the Term–by–Document Matrix

Documents \ Terms	investment risk	project management	software engineering	development	SAP	...
Document 1	1			1		
Document 2		1				
Document 3			3		1	
Document 4		1				
Document 5			2	1		
Document 6	1			1		
...						

Text Mining Process

- Step 2: Create the Term–by–Document Matrix (TDM), cont.
  - Should all terms be included?
    - Stop words, include words
    - Synonyms, homonyms
    - Stemming
  - What is the best representation of the indices (values in cells)?
    - Row counts; binary frequencies; log frequencies;
    - Inverse document frequency

Text Mining Process

- Step 3: Extract patterns/knowledge
  - Classification (text categorization)
  - Clustering (natural groupings of text)
    - Improve search recall
    - Improve search precision
    - Scatter/gather
    - Query-specific clustering
  - Association
  - Trend Analysis (…)

Business Scenario

- Identify the most common words in a sample of 1000 reviews of popular fee apps from the iTunes Store

Example: Creating a Word Cloud

```
› library(wordcloud)
› library(tm)
› reviews <- read.csv("reviews.csv",
  stringsAsFactors=FALSE)
› review_source <- VectorSource(reviews$text)
› corpus <- Corpus(review_source)
› summary(corpus)
› corpus <- tm_map(corpus,
  content_transformer(tolower))
› corpus <- tm_map(corpus, removePunctuation)
› corpus <- tm_map(corpus, stripWhitespace)
› corpus <- tm_map(corpus, removeWords,
  stopwords("english"))
```

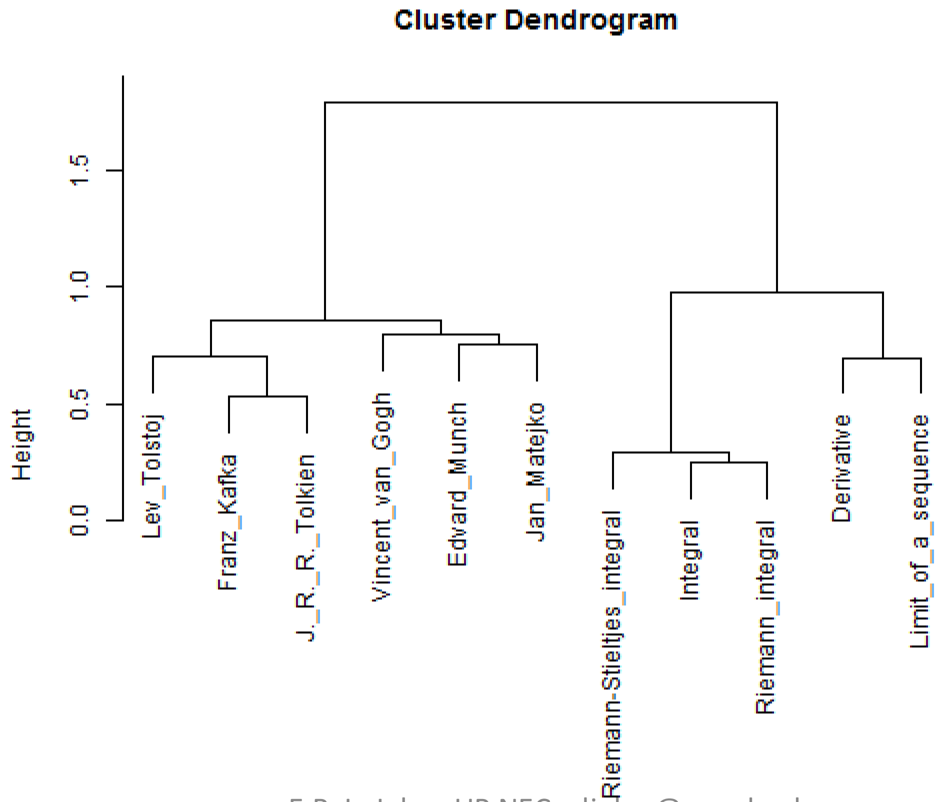
Example: Creating a Word Cloud

```
› corpus <- tm_map(corpus, removeWords, c("game"))
› dtm <- DocumentTermMatrix(corpus)
› dtm2 <- as.matrix(dtm)
› frequency <- colSums(dtm2)
› frequency <- sort(frequency, decreasing=TRUE)
› head(frequency,14)
› words <- names(frequency)
› wordcloud(words[1:100],
  frequency[1:100],colors=brewer.pal(8, "Dark2"))
```

Sample Result

Other Examples of Text Mining

- With **Hierarchical Clustering** (http://www.rexamine.com/2014/06/text-mining-in-r-automatic-categorization-of-wikipedia-articles/)



Outline for This Session

- Market Basket Analysis
- Sequential Pattern Mining
- Clustering
  - K-Means Clustering
  - Hierarchical Clustering
- Text Mining
- **Social Media Sentiment Analysis**
- Case Study

Introduction

- Two main types of textual information.
  - Facts and Opinions
    - Note: factual statements can imply opinions too.
- Most current text information processing methods (e.g., web search, text mining) work with factual information.
- Sentiment analysis or opinion mining
  - computational study of opinions, sentiments and emotions expressed in text.
- Why now?
  - Mainly because of the Web; huge volumes of opinionated text.

Introduction – User-Generated media

- Importance of opinions:
  - Opinions are important because whenever we need to make a decision, we want to hear others' opinions.
  - In the past
    - Individuals: opinions from friends and family
    - businesses: surveys, focus groups, consultants …
- Word-of-mouth on the Web
  - User-generated media: One can express opinions on anything in reviews, forums, discussion groups, blogs ...
  - Opinions of global scale: No longer limited to:
    - Individuals: one's circle of friends
    - Businesses: Small scale surveys, tiny focus groups, etc.

A Fascinating Problem!

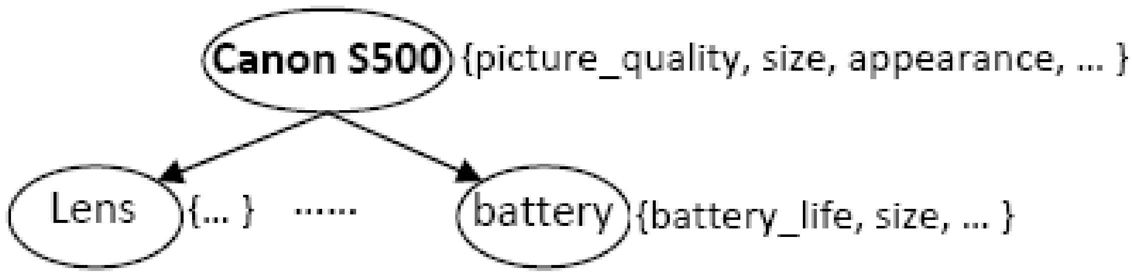
- Intellectually challenging & major applications.
  - A popular research topic in recent years in NLP and Web data mining.
  - 20-60 companies in USA alone
- It touches every aspect of NLP and yet is restricted and confined.
  - Little research in NLP/Linguistics in the past.
- Potentially a major technology from NLP.
  - But "not yet" and not easy!
  - Data sourcing and data integration are hard too!

An Example Review

- "I bought an iPhone a few days ago. It was such a nice phone. The touch screen was really cool. The voice quality was clear too. Although the battery life was not long, that is ok for me. However, my mother was mad with me as I did not tell her before I bought the phone. She also thought the phone was too expensive, and wanted me to return it to the shop. …"
- What do we see?
  - Opinions, targets of opinions, and opinion holders

Target Object (Liu, Web Data Mining book, 2006)

- Definition: An object o is a product, person, event, organization, or topic. o is represented as
  - a hierarchy of components, sub-components, and so on.
  - Each node represents a component and is associated with a set of attributes of the component.



- An opinion can be expressed on any node or attribute of the node.
- We use the term features to represent both components and attributes.

What is an Opinion? (Liu, a Ch. in NLP handbook)

- An opinion is a quintuple

$$(o_j, f_{jk}, s_{oijkl}, h_i, t_l)$$

- where
  - o_j is a target object.
  - f_{jk} is a feature of the object o_j.
  - s_{oijkl} is the sentiment value of the opinion of the opinion holder h_i on feature f_{jk} of object o_j at time t_l.
  - h_i is an opinion holder.
  - t_l is the time when the opinion is expressed.

Objective – structure the unstructured

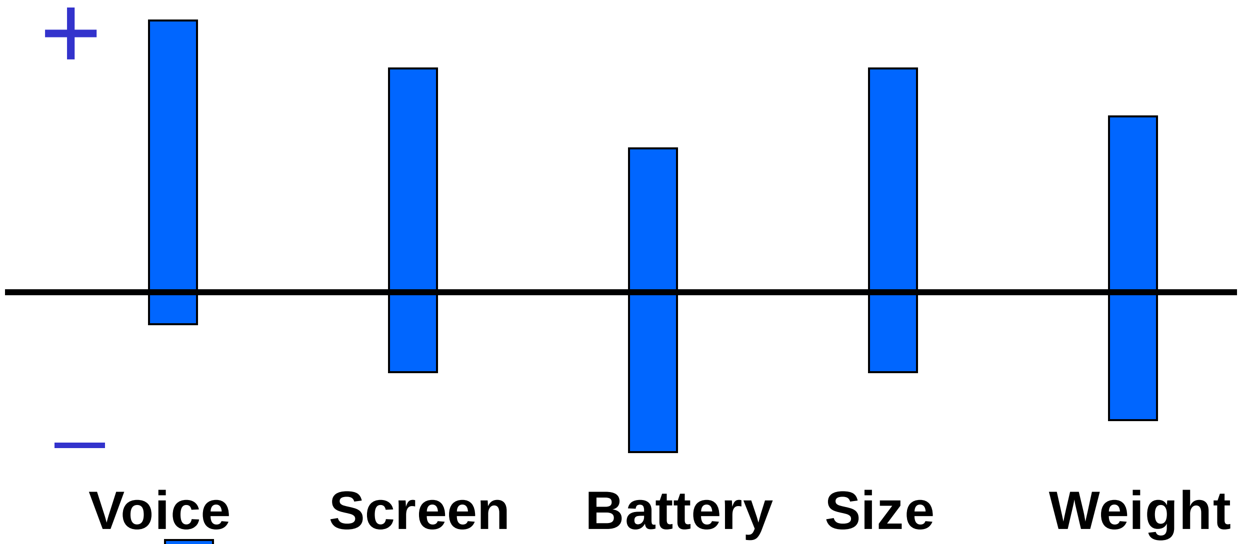
- Objective: Given an opinionated document
  - Discover all quintuples $(o_j, f_{jk}, s_{oijkl}, h_i, t_l)$
    - i.e., mine the five corresponding pieces of information in each quintuple, and
- With the quintuples,
  - Unstructured Text $\rightarrow$ Structured Data
    - Traditional data and visualization tools can be used to slice, dice and visualize the results in all kinds of ways
    - Enable qualitative and quantitative analysis.

Sentiment Classification: doc-level (Pang and Lee, et al 2002 and Turney 2002)

- Classify a document (e.g., a review) based on the overall sentiment expressed by opinion holder
  - Classes: Positive, or negative (and neutral)
  - In the model, $(o_j, f_{jk}, s_{oijkl}, h_i, t_l)$
  - It assumes
    - Each document focuses on a single object and contains opinions from a single opinion holder.

Visual Comparison (Liu et al. WWW-2005)

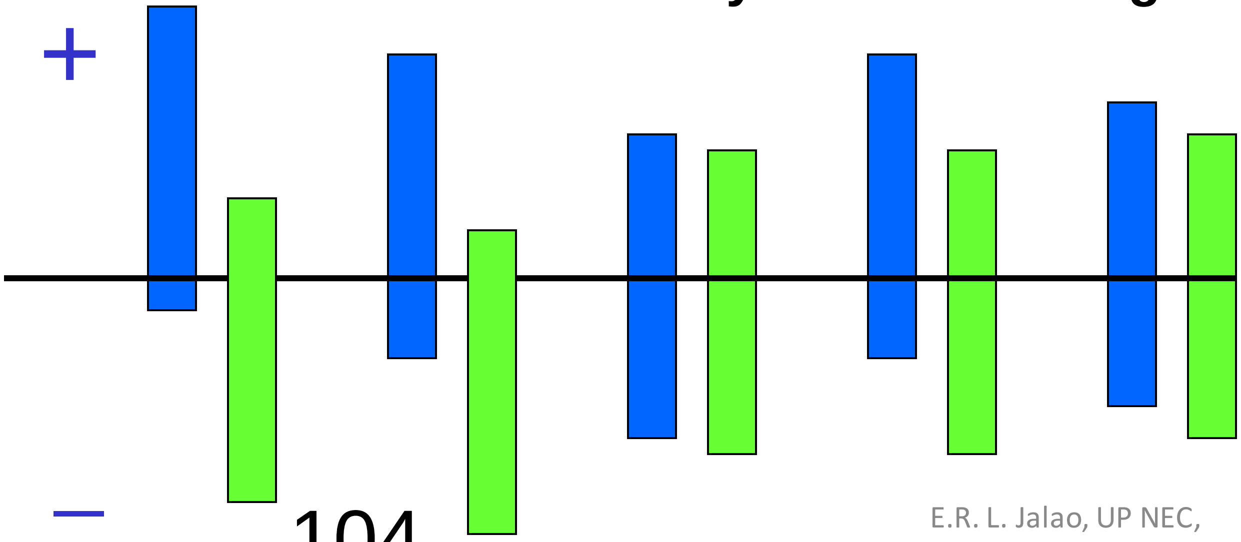
- Summary of reviews of Cell Phone 1



- Comparison of reviews of

Cell Phone 1

Cell Phone 2



E.R. L. Jalao, UP NEC,
eljalao@up.edu.ph

Outline for This Session

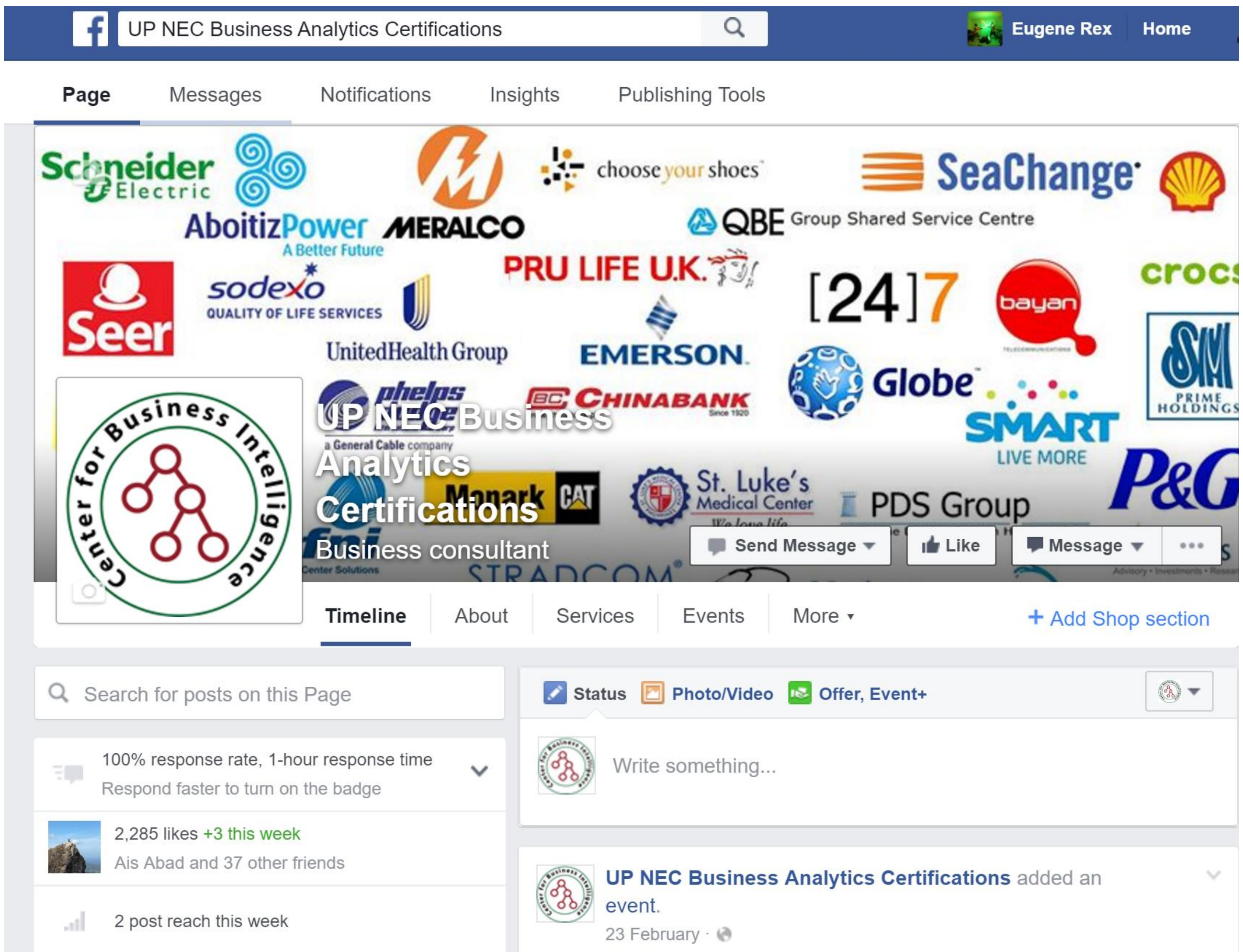
- Market Basket Analysis
- Sequential Pattern Mining
- Clustering
  - K-Means Clustering
  - Hierarchical Clustering
- Text Mining
- Social Media Sentiment Analysis
- **Case Study**

Certification Exam

- Coverage:
  - BI Analyst: All Three Modules
  - Data Mining Analyst: Module 1 and Module 3
  - Data Warehousing Analyst: Module 1 and Module 2
- Type
  - Multiple Choice, Concept Based Questions (From Notes)
  - 3 Hours for BI Analyst
  - 2 Hours for Data Mining and Data Warehousing Analyst

Facebook Page

- https://www.facebook.com/upnecanalytics/



R Users Group

http://www.meetup.com/R-Users-Group-Philippines

Case Study 4

- Web Mining of Twitter Data
- Text Mining of Tweets for Sensitivity Analysis
  - Compare iPhone 6 and Samsung Galaxy S6

Outline for This Session

- Market Basket Analysis
- Sequential Pattern Mining
- Clustering
  - K-Means Clustering
  - Hierarchical Clustering
- Text Mining
- Social Media Sentiment Analysis
- Case Study

References

- Turban et. al. 2011, Decision Support and Business Intelligence Systems (9th Ed., Prentice Hall)
- Tan et al. Intro to Data Mining Notes
- Notes and Datasets from Montgomery, Peck and Vining, Introduction to Linear Regression Analysis 4th Ed. Wiley
- Notes from G. Runger, ASU IEE 578
- Trevor Hastie, Rob Tibshirani, Friedman: Elements of Statistical Learning (2nd Ed.) 2009
- Ke Chen, http://studentnet.cs.manchester.ac.uk/ugt/COMP24111/materials/slides/Hierarchical.ppt
- Gabadinho, Alexis, et al. "Mining sequence data in R with the TraMineR package: A users guide for version 1.2." *Geneva: University of Geneva* (2009).
- Minqing Hu and Bing Liu, 2004, "Mining and Summarizing Customer Reviews." *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004), Aug 22-25, 2004, Seattle, Washington, USA*.
- https://deltadna.com/blog/text-mining-in-r-for-term-frequency/