Data Science
Lecture 2:
Know your data & Frequent Pattern Mining

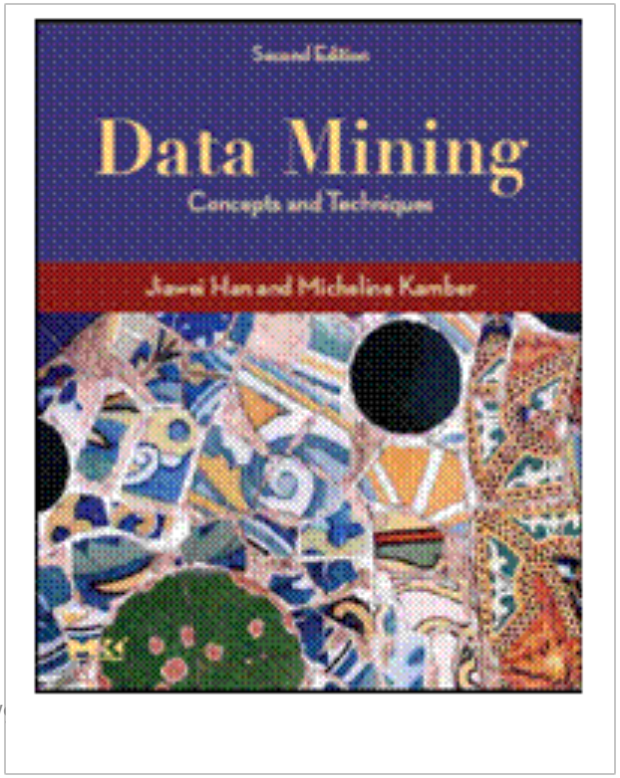
Speaker: Hong-Han Shuai (帥宏翰)
Department of Electrical and Computer engineering
National Chiao Tung University

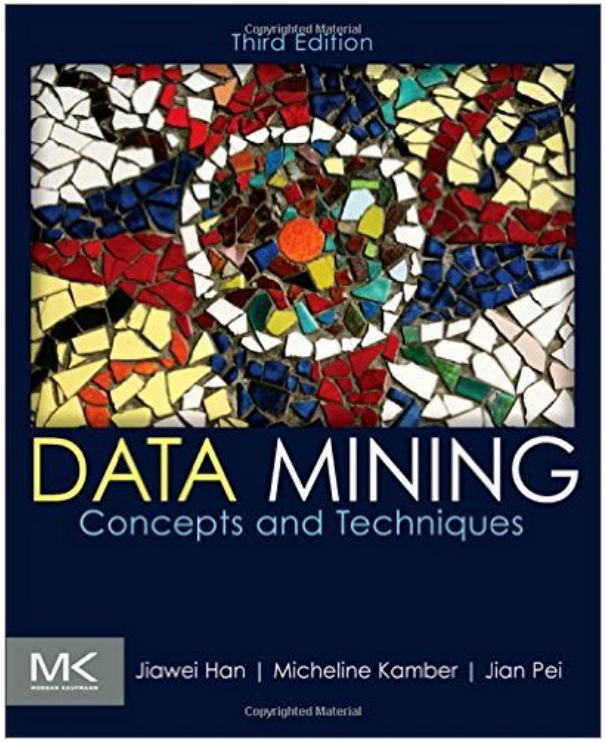
Open Data Project

- Data.gov: http://data.gov
- US Census Bureau: http://www.census.gov/data.html
- European Union Open Data Portal: http://open-data.europa.eu/en/data/
- Healthdata.gov: https://www.healthdata.gov/
- Google Finance: https://www.google.com/finance

Referred textbook

- Data Mining: Concepts and Techniques, Second (or Third) Edition
- Chapter 2: Getting to know your data





Exploring Data

- Data Objects and Attribute Types
- Basic Statistical Descriptions of Data
- Data Visualization
- Measuring Data Similarity and Dissimilarity
- Summary

Types of Data Sets

- Record
  - Relational records
  - Data matrix, e.g., numerical matrix, crosstabs
  - Document data: text documents: term-frequency vector
  - Transaction data
- Graph and network
  - World Wide Web
  - Social or information networks
  - Molecular Structures
- Ordered
  - Video data: sequence of images
  - Temporal data: time-series
  - Sequential Data: transaction sequences
  - Genetic sequence data
- Spatial, image and multimedia:
  - Spatial data: maps
  - Image data:
  - Video data:

	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

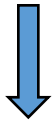
		OrgName1	OrgName2	OrgName3
ItemName1	DayValue	10	30	20
	WeekValue	20	10	10
ItemName2	DayValue	10	20	30
	WeekValue	30	10	90
ItemName3	DayValue	40	30	50
	WeekValue	50	90	30

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

Data Objects

- Data sets are made up of data objects.
- A **data object** represents an entity.
- Examples:
  - sales database: customers, store items, sales
  - medical database: patients, treatments
  - university database: students, professors, courses
- Also called *samples, examples, instances, data points, objects, tuples*.
- Data objects are described by **attributes**.
- Database rows -> data objects; columns ->attributes.

Attribute (feature)



Data row

Name	Thread pitch (mm)	Minor diameter tolerance	Nominal diameter (mm)	Head shape	Price for 50 screws	Available at factory outlet?	Number in stock	Flat or Phillips head?
M4	0.7	4g	4	Pan	$10.08	Yes	276	Flat
M5	0.8	4g	5	Round	$13.89	Yes	183	Both
M6	1	5g	6	Button	$10.42	Yes	1043	Flat
M8	1.25	5g	8	Pan	$11.98	No	298	Phillips
M10	1.5	6g	10	Round	$16.74	Yes	488	Phillips
M12	1.75	7g	12	Pan	$18.26	No	998	Flat
M14	2	7g	14	Round	$21.19	No	235	Phillips
M16	2	8g	16	Button	$23.57	Yes	292	Both
M18	2.1	8g	18	Button	$25.87	No	664	Both
M20	2.4	8g	20	Pan	$29.09	Yes	486	Both
M24	2.55	9g	24	Round	$33.01	Yes	982	Phillips
M28	2.7	10g	28	Button	$35.66	No	1067	Phillips
M36	3.2	12g	36	Pan	$41.32	No	434	Both
M50	4.5	15g	50	Pan	$44.72	No	740	Flat

Attributes

- **Attribute (**or **dimensions, features, variables**): a data field, representing a characteristic or feature of a data object.
  - *E.g., customer _ID, name, address*
- Types:
  - Nominal (類別，彼此間無順序, categorical)
  - Binary
  - Ordinal
  - Numeric: quantitative
    - Interval-scaled (no true zero point)
    - Ratio-scaled (Inherent zero point)

Attribute Types

- **Nominal:** categories, states, or "names of things"
  - *Hair_color* = {*auburn, black, blond, brown, grey, red, white*}
  - marital status, occupation, ID numbers, zip codes
- **Binary**
  - Nominal attribute with only 2 states (0 and 1)
  - Symmetric binary: both outcomes equally important
    - e.g., gender
  - Asymmetric binary: outcomes not equally important
    - e.g., medical test (positive vs. negative)
    - Convention: assign 1 to most important outcome (e.g., HIV positive)
- **Ordinal**
  - Values have a meaningful order (ranking) but magnitude between successive values is not known.
  - *Size* = {*small, medium, large*}, grades, army rankings

Numeric Attribute Types

- Quantity (integer or real-valued)
- **Interval**
  - Measured on a scale of **equal-sized units**
  - Values have order
    - E.g., *temperature in C˚or F˚, calendar dates*
  - No true zero-point
- **Ratio**
  - Inherent **zero-point**
  - We can speak of values as being an order of magnitude larger than the unit of measurement (10 K˚ is twice as high as 5 K˚).
    - e.g., *temperature in Kelvin, length, counts, monetary quantities*

Discrete vs. Continuous Attributes

- **Discrete Attribute**
  - Has only a finite or countably infinite set of values
    - E.g., zip codes, profession, or the set of words in a collection of documents
  - Sometimes, represented as integer variables
  - Note: Binary attributes are a special case of discrete attributes
- **Continuous Attribute**
  - Has real numbers as attribute values
    - E.g., temperature, height, or weight
  - Practically, real values can only be measured and represented using a finite number of digits
  - Continuous attributes are typically represented as floating-point variables

Exploring Data

- Data Objects and Attribute Types
- Basic Statistical Descriptions of Data
- Data Visualization
- Measuring Data Similarity and Dissimilarity
- Summary

Basic Statistical Descriptions of Data

- Motivation
  - To better understand the data: central tendency, variation and spread
- Data dispersion characteristics
  - median, max, min, quantiles, outliers, variance, etc.
- Numerical dimensions correspond to sorted intervals
  - Data dispersion: analyzed with multiple granularities of precision
  - Boxplot or quantile analysis on sorted intervals
- Dispersion analysis on computed measures
  - Folding measures into numerical dimensions
  - Boxplot or quantile analysis on the transformed cube

Measuring the Central Tendency

- Mean (algebraic measure) (sample vs. population):

  Note: *n* is sample size and *N* is population size.

  $$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \mu = \frac{\sum x}{N}$$

  - Weighted arithmetic mean:

  $$\bar{x} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$$

  - Trimmed mean: chopping extreme values

- Median:
  - Middle value if odd number of values, or average of the middle two values otherwise

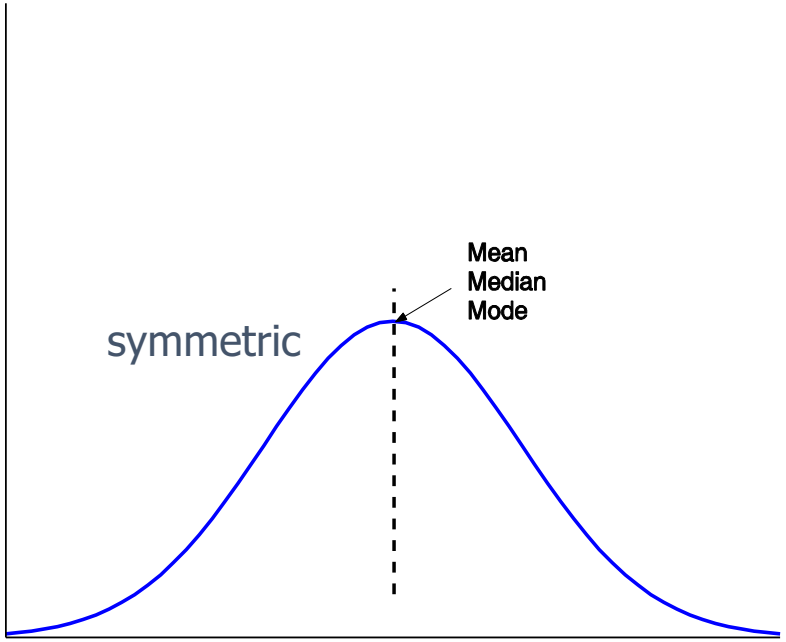
- Mode:
  - Value that occurs most frequently in the data
  - Unimodal, bimodal, trimodal (1, 2, and 3 modes in the dataset, respectively)
  - Empirical formula: $mean - mode = 3 \times (mean - median)$

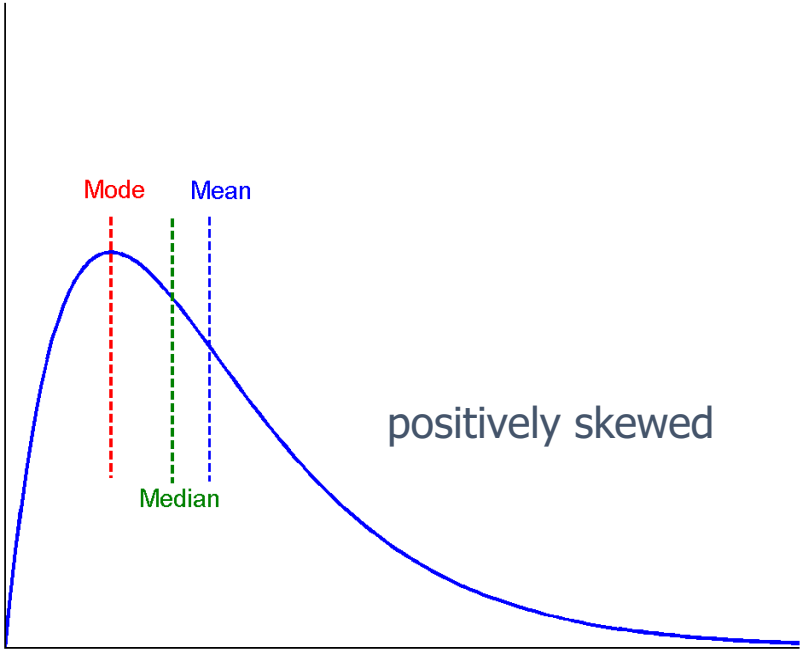
Questions

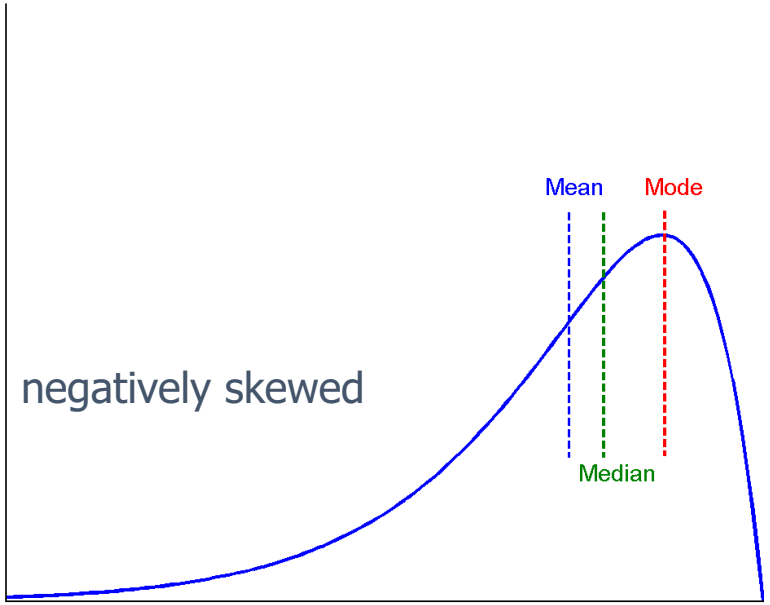
- Nominal — ?
- Ordinal — ?
- Interval-Ratio — ?

Symmetric vs. Skewed Data

- Median, mean and mode of symmetric, positively and negatively skewed data



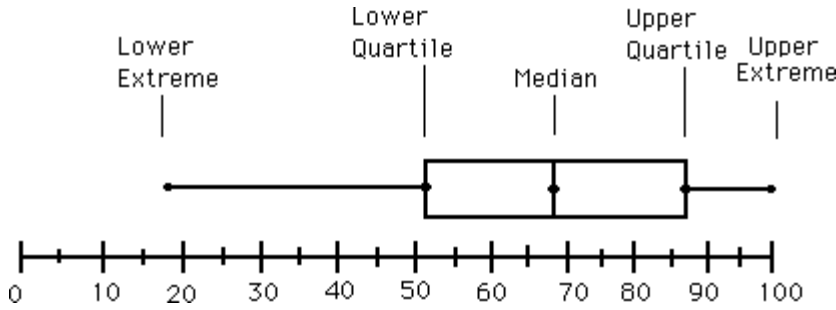




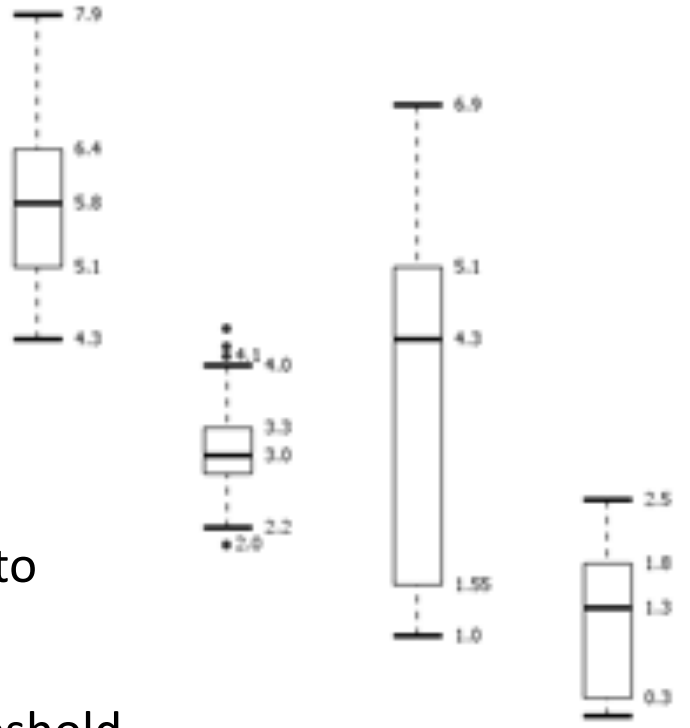
Measuring the Dispersion of Data

- Quartiles, outliers and boxplots
  - **Quartiles**: Q_1 (25^{th} percentile), Q_3 (75^{th} percentile)
  - **Inter-quartile range**: $IQR = Q_3 - Q_1$
  - **Five number summary**: min, Q_1, median, Q_3, max
  - **Boxplot**: ends of the box are the quartiles; median is marked; add whiskers, and plot outliers individually
  - **Outlier**: usually, a value higher/lower than 1.5 x IQR

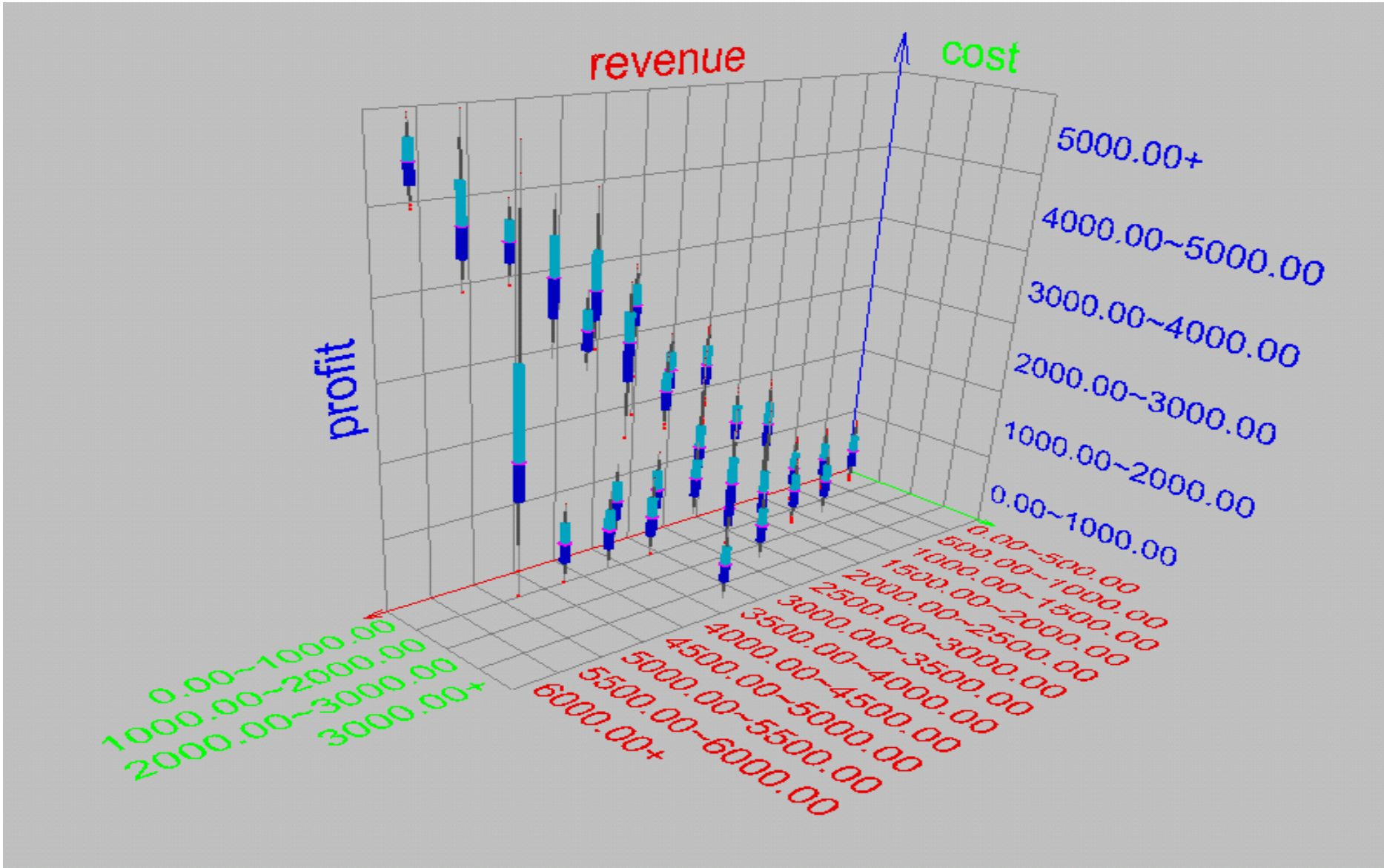
Boxplot Analysis



- **Five-number summary** of a distribution
  - Minimum, Q1, Median, Q3, Maximum
- **Boxplot**
  - Data is represented with a box
  - The ends of the box are at the first and third quartiles, i.e., the height of the box is IQR
  - The median is marked by a line within the box
  - Whiskers: two lines outside the box extended to Minimum and Maximum
  - Outliers: points beyond a specified outlier threshold (usually 1.5xIQR), plotted individually



Visualization of Data Dispersion: 3-D Boxplots



Variance and Standard Deviation

- Measure data dispersion, i.e., indicate how spread out a data distribution is
- Low standard deviation -> data observations very close to mean
- High standard deviation -> data are spread out over a large range of values

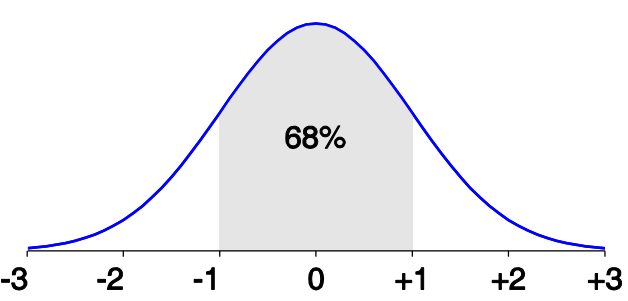
- Variance and standard deviation (*sample: s, population: σ)*
  - **Variance**: (algebraic, scalable computation)

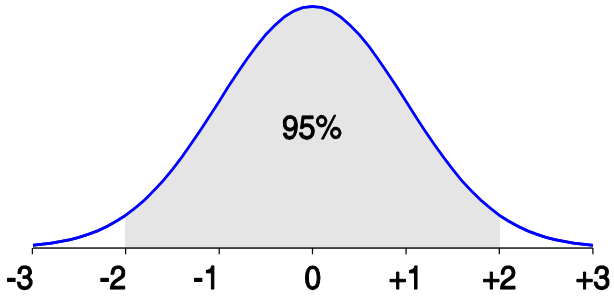
$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \frac{1}{n-1}[\sum_{i=1}^{n}x_i^2 - \frac{1}{n}(\sum_{i=1}^{n}x_i)^2] \qquad \sigma^2 = \frac{1}{N}\sum_{i=1}^{n}(x_i - \mu)^2 = \frac{1}{N}\sum_{i=1}^{n}x_i^2 - \mu^2$$

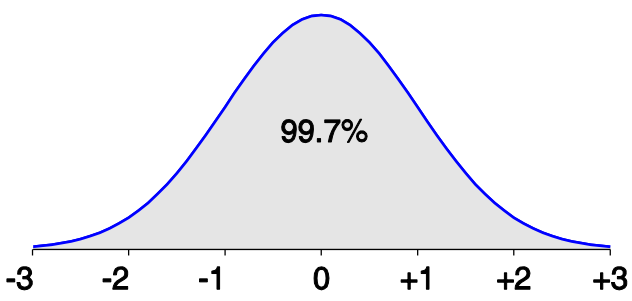
  - **Standard deviation** *s (or σ)* is the square root of variance s^2 *(or* σ^2*)*

Properties of Normal Distribution Curve

- The normal (distribution) curve
  - From μ–σ to μ+σ: contains about 68% of the measurements (μ: mean, σ: standard deviation)
  - From μ–2σ to μ+2σ: contains about 95% of it
  - From μ–3σ to μ+3σ: contains about 99.7% of it





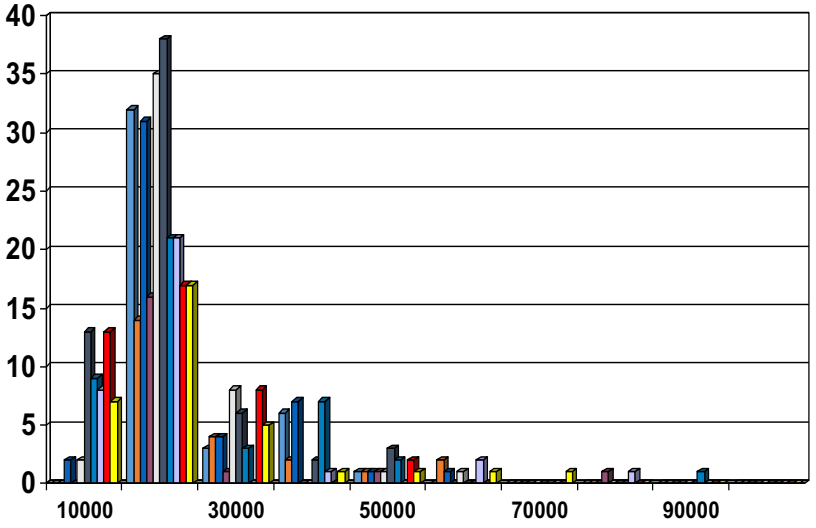


Graphic Displays of Basic Statistical Descriptions

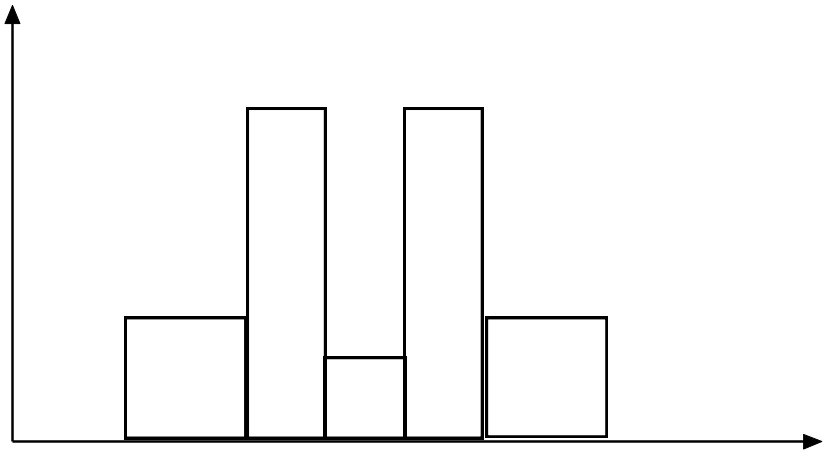
- **Boxplot**: graphic display of five-number summary
- **Histogram**: x-axis are values, y-axis repres. frequencies
- **Quantile plot**: each value x_i is paired with f_i indicating that approximately 100 f_i % of data are $\leq x_i$
- **Quantile-quantile (q-q) plot**: graphs the quantiles of one univariant distribution against the corresponding quantiles of another
- **Scatter plot**: each pair of values is a pair of coordinates and plotted as points in the plane

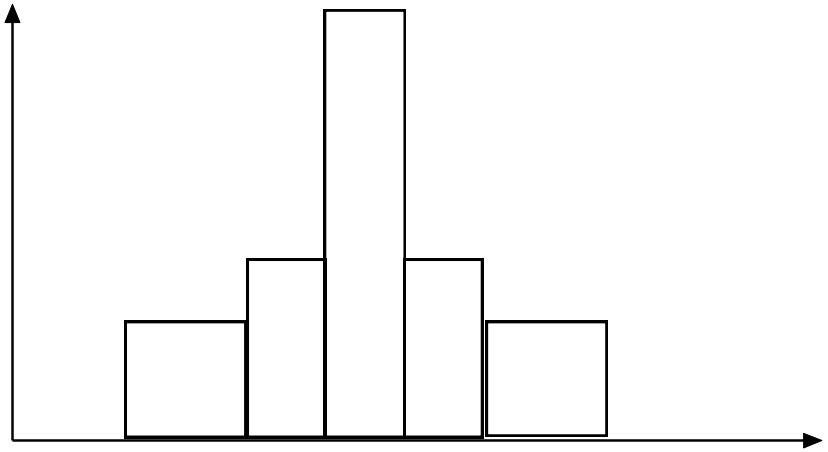
Histogram Analysis

- Histogram: Graph display of tabulated frequencies, shown as bars
- It shows what proportion of cases fall into each of several categories
- Differs from a bar chart in that it is the *area* of the bar that denotes the value, not the height as in bar charts, a crucial distinction when the categories are not of uniform width
- The categories are usually specified as non-overlapping intervals of some variable. The categories (bars) must be adjacent



Histograms Often Tell More than Boxplots

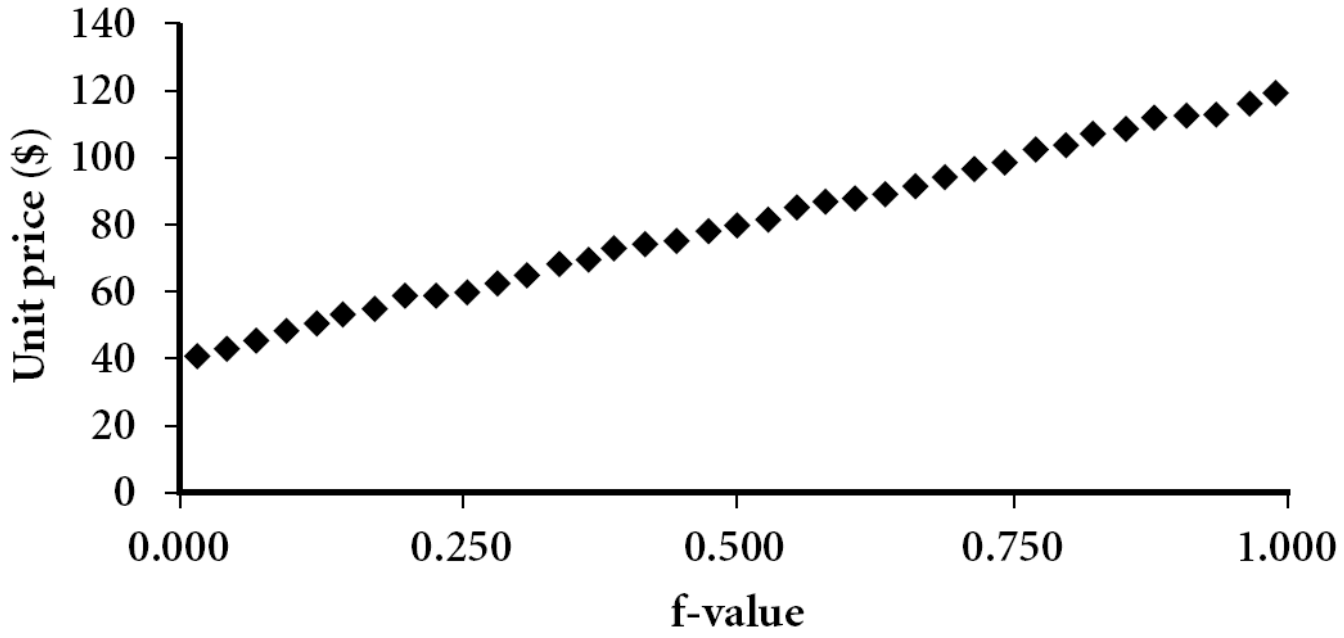




- The two histograms shown in the left may have the same boxplot representation
  - The same values for: min, Q1, median, Q3, max
- But they have rather different data distributions

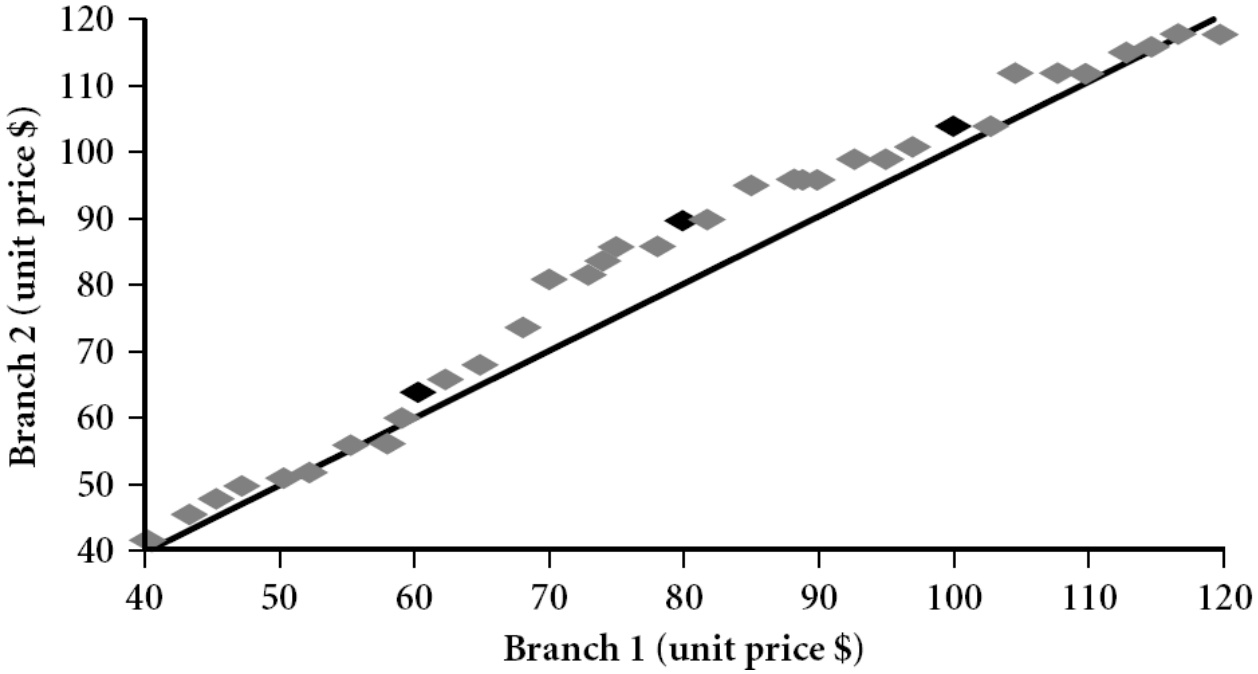
Quantile Plot

- Displays all of the data (allowing the user to assess both the overall behavior and unusual occurrences)
- Plots **quantile** information
  - For a data x_i data sorted in increasing order, f_i indicates that approximately 100 f_i% of the data are below or equal to the value x_i



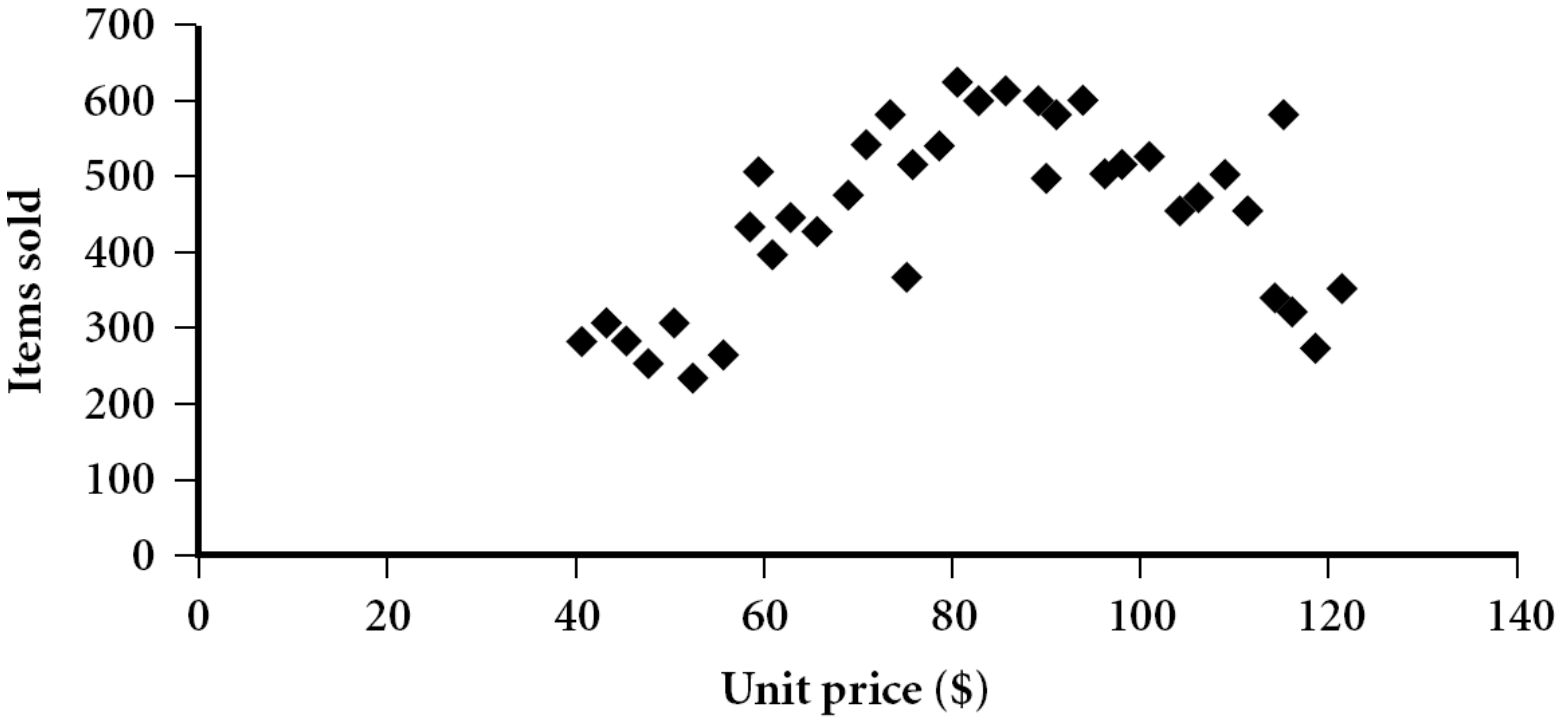
Quantile-Quantile (Q-Q) Plot

- Graphs the quantiles of one univariate distribution against the corresponding quantiles of another
- View: Is there is a shift in going from one distribution to another?
- Example shows unit price of items sold at Branch 1 vs. Branch 2 for each quantile. Unit prices of items sold at Branch 1 tend to be lower than those at Branch 2.

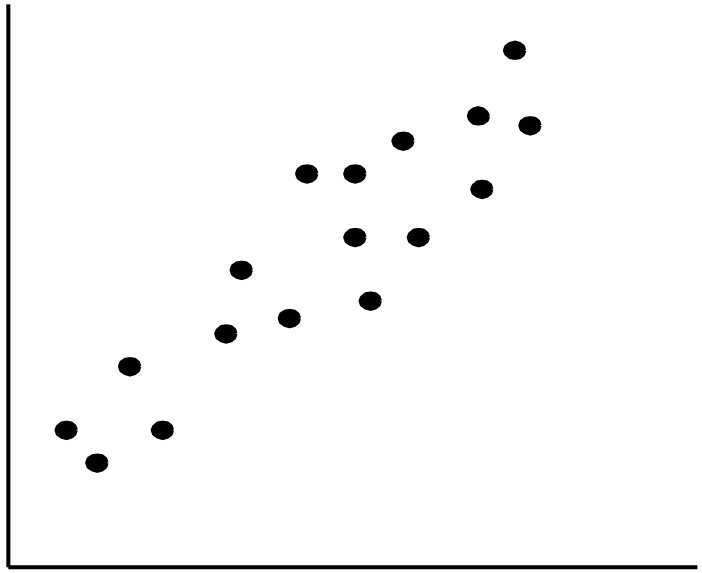


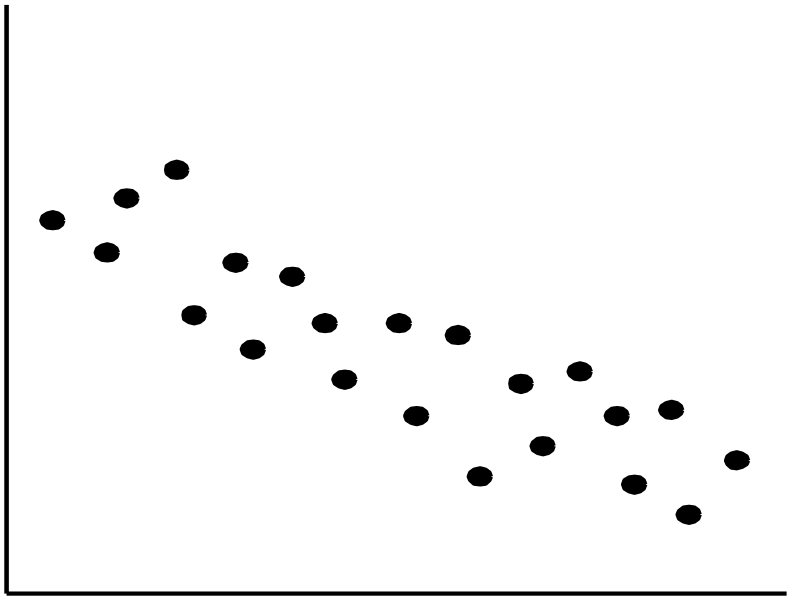
Scatter plot

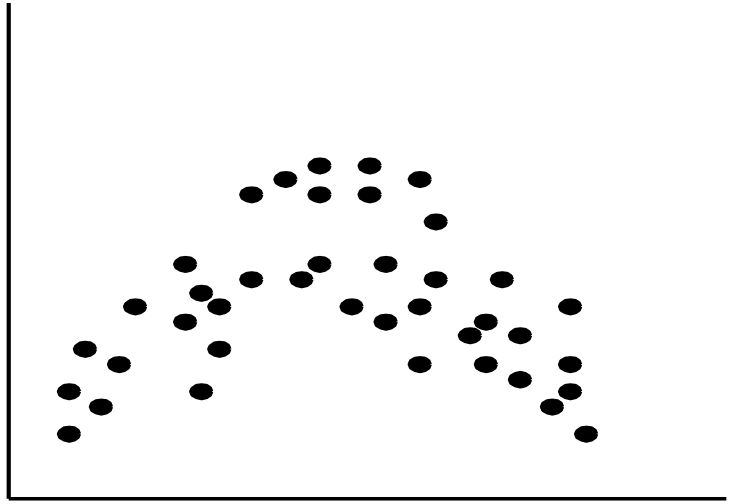
- Provides a first look at bivariate data to see clusters of points, outliers, etc
- Each pair of values is treated as a pair of coordinates and plotted as points in the plane



Positively and Negatively Correlated Data

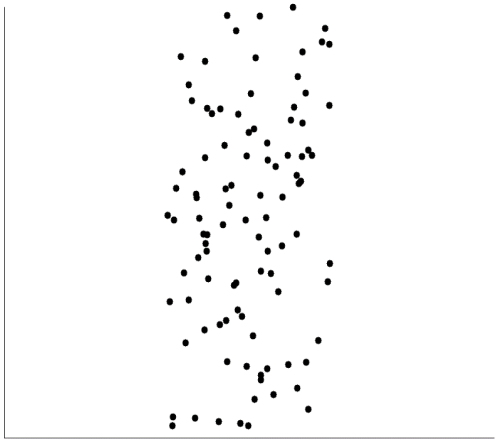


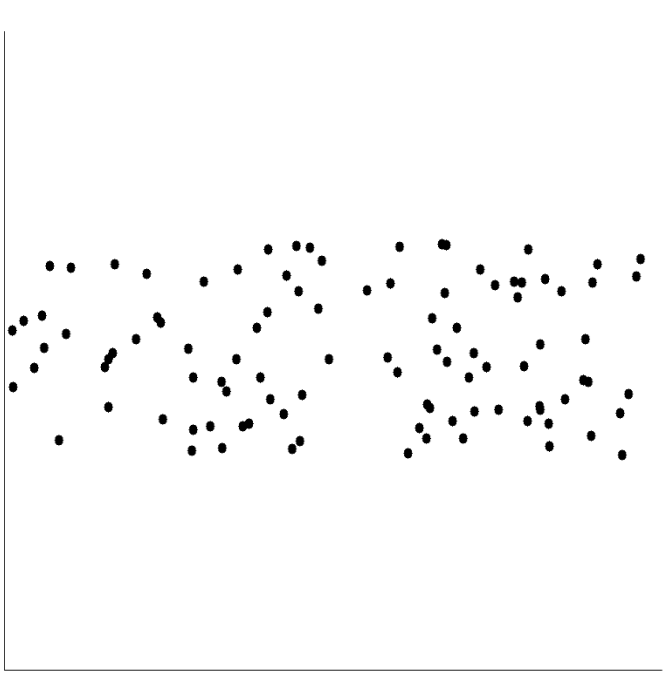


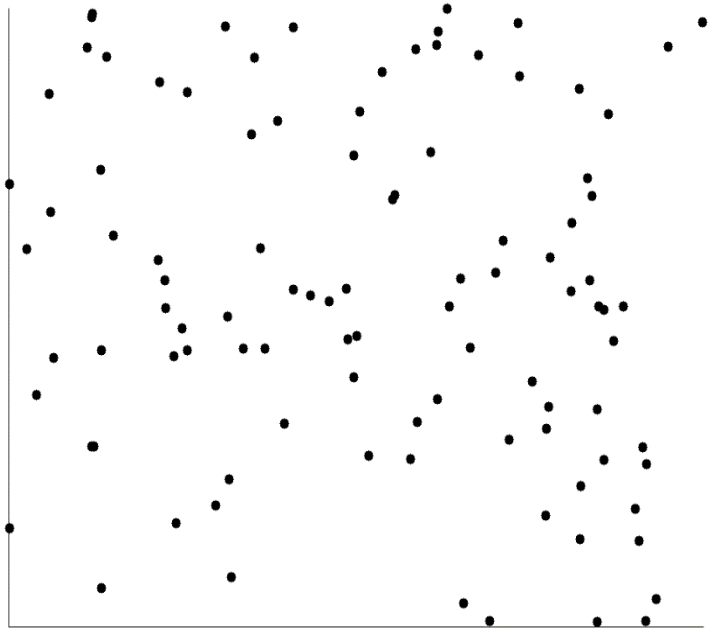


- The left half fragment is positively correlated
- The right half is negative correlated

Uncorrelated Data







Exploring Data

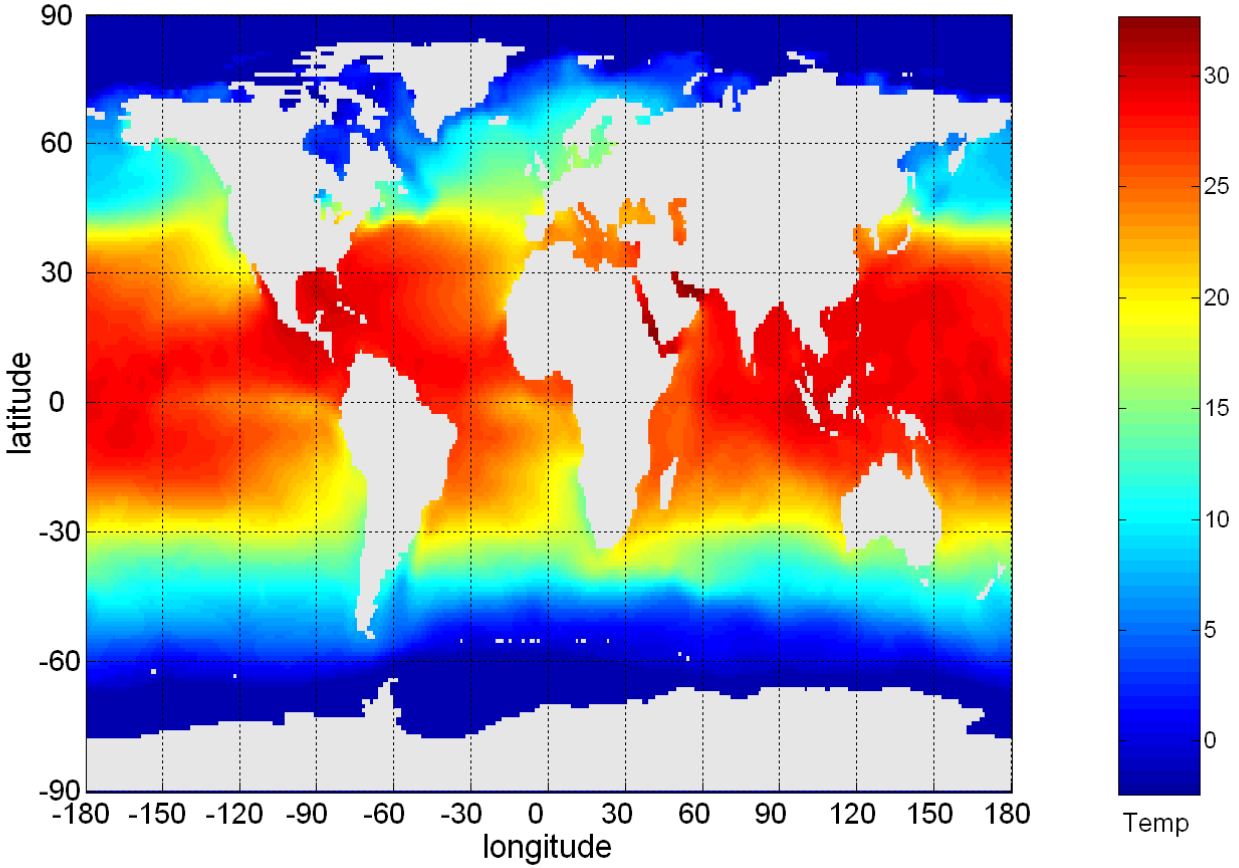
- Data Objects and Attribute Types
- Basic Statistical Descriptions of Data
- Data Visualization
- Measuring Data Similarity and Dissimilarity
- Summary

Data Visualization

- Why data visualization?
  - Gain insight into an information space by mapping data onto graphical primitives
  - Provide qualitative overview of large data sets
  - Search for patterns, trends, structure, irregularities, relationships among data
  - Help find interesting regions and suitable parameters for further quantitative analysis
  - Provide a visual proof of computer representations derived

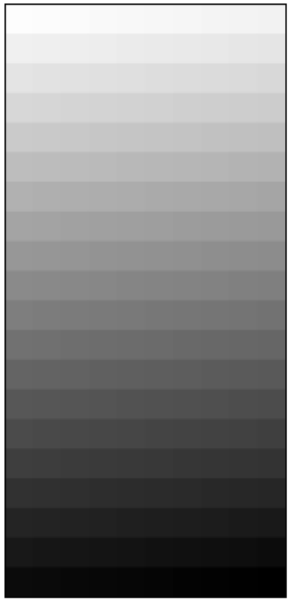
Example: Sea Surface Temperature

- The following shows the Sea Surface Temperature (SST) for July 1982
  - Tens of thousands of data points are summarized in a single figure

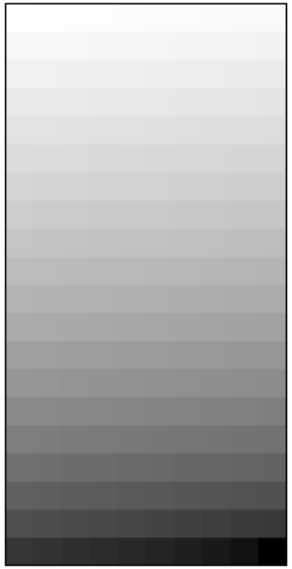


Pixel-Oriented Visualization Techniques

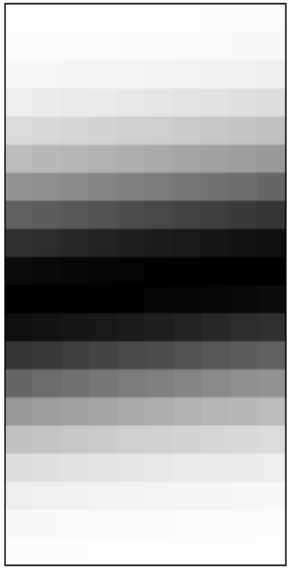
- For a data set of m dimensions, create m windows on the screen, one for each dimension
- The m dimension values of a record are mapped to m pixels at the corresponding positions in the windows
- The colors of the pixels reflect the corresponding values



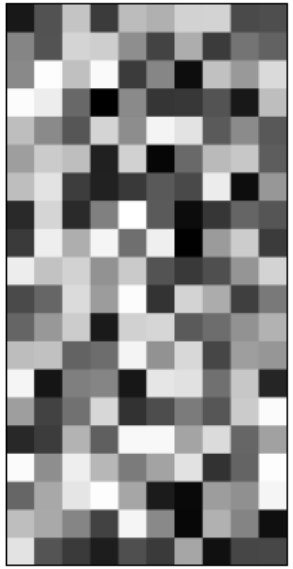
(a) Income



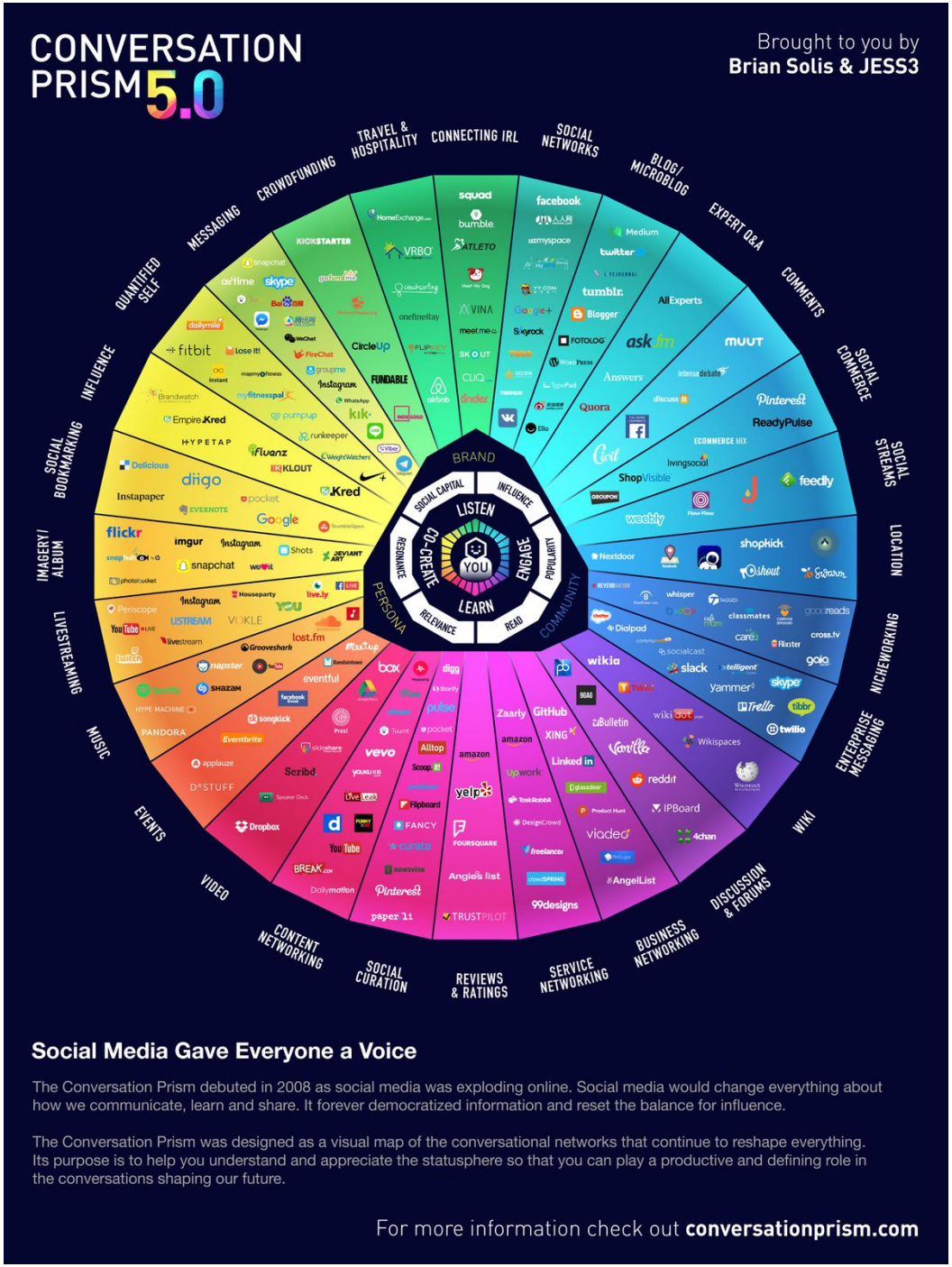
(b) Credit Limit

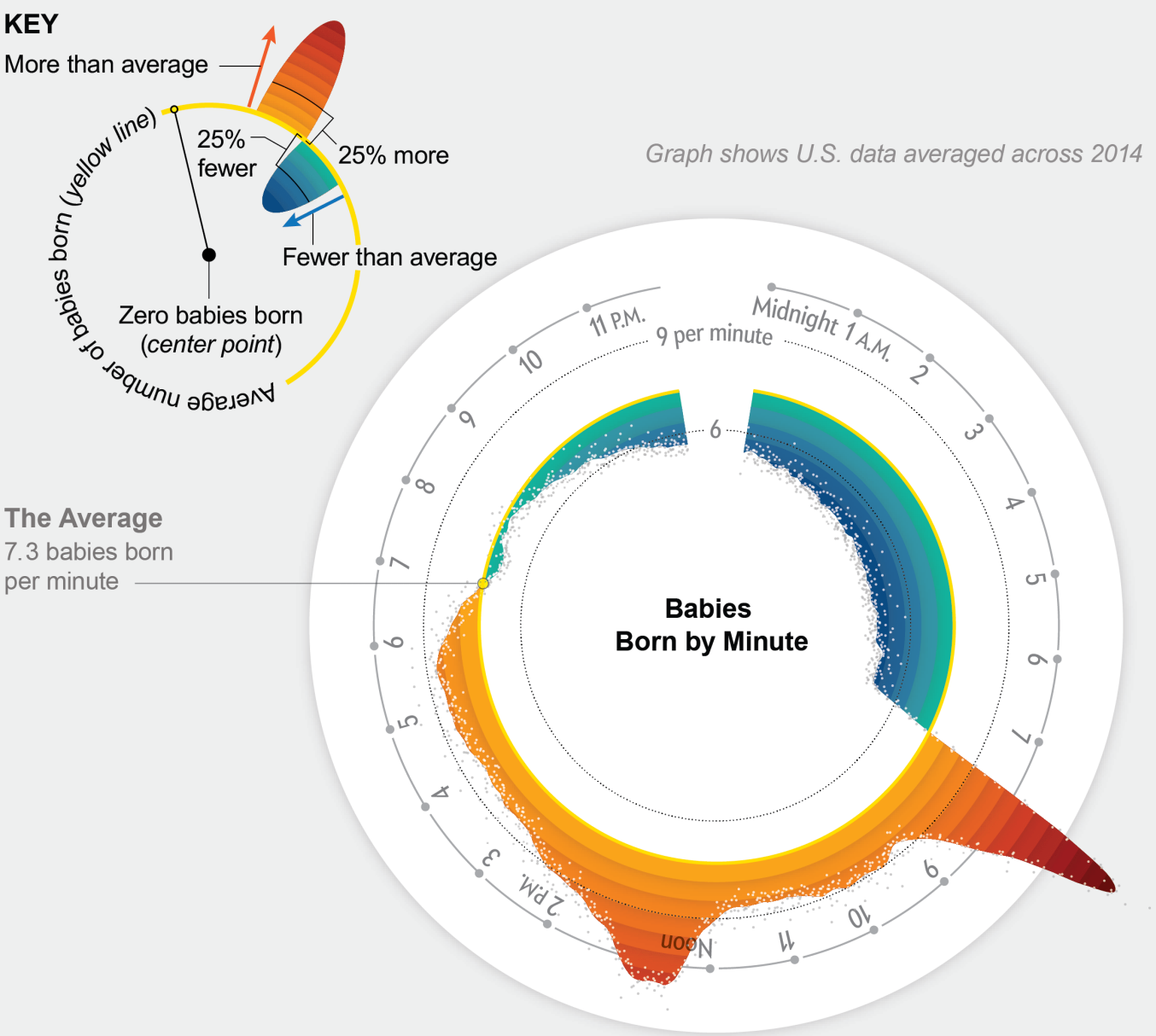


(c) transaction volume



(d) age

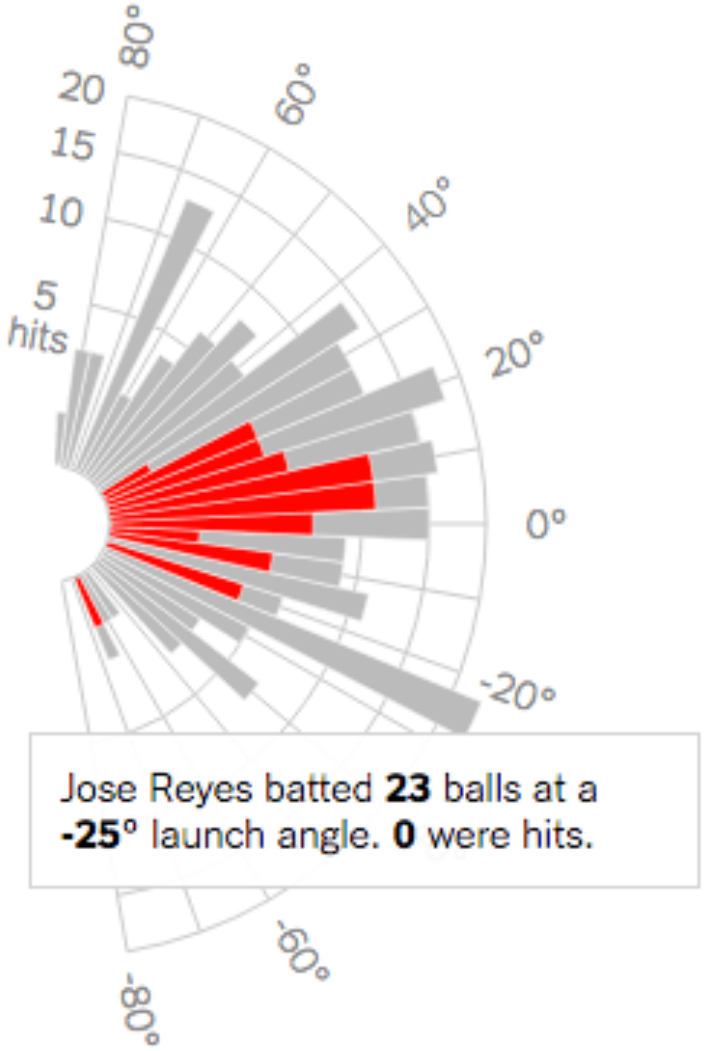

CONVERSATION PRISM 5.0
Brought to you by Brian Solis & JESS3
TRAVEL & HOSPITALITY
CONNECTING IRL
SOCIAL NETWORKS
BLOG/ MICROBLOG
EXPERT Q&A
COMMENTS
SOCIAL COMMERCE
SOCIAL STREAMS
LOCATION
NICHENETWORKING
ENTERPRISE MESSAGING
WIKI
DISCUSSION & FORUMS
BUSINESS NETWORKING
SERVICE NETWORKING
REVIEWS & RATINGS
SOCIAL CURATION
CONTENT NETWORKING
VIDEO
EVENTS
MUSIC
LIVESTREAMING
IMAGERY / ALBUM
SOCIAL BOOKMARKING
INFLUENCE
QUANTIFIED SELF
MESSAGING
CROWDFUNDING
BRAND
SOCIAL CAPITAL
INFLUENCE
LISTEN
CO-CREATE
ENGAGE
YOU
LEARN
RESONANCE
POPULARITY
RELEVANCE
READ
PERSONA
COMMUNITY
Social Media Gave Everyone a Voice
The Conversation Prism debuted in 2008 as social media was exploding online. Social media would change everything about how we communicate, learn and share. It forever democratized information and reset the balance for influence.
The Conversation Prism was designed as a visual map of the conversational networks that continue to reshape everything. Its purpose is to help you understand and appreciate the statusphere so that you can play a productive and defining role in the conversations shaping our future.
For more information check out conversationprism.com


KEY
More than average
25% fewer
25% more
Average number of babies born (yellow line)
Fewer than average
Zero babies born
(center point)
Graph shows U.S. data averaged across 2014
The Average
7.3 babies born
per minute
9 per minute
Midnight
1 A.M.
2
3
4
5
6
7
8
9
10
11
11 P.M.
10
9
8
7
6
5
4
3
2 P.M.
Noon
11
10
9
Babies
Born by Minute

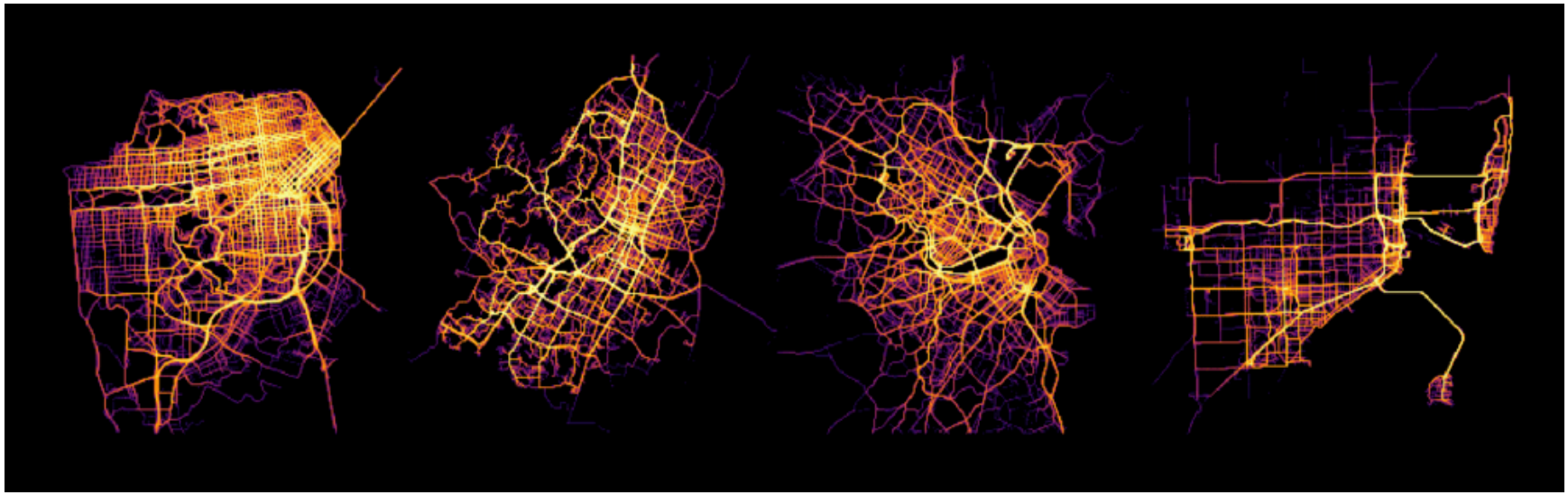
Jose Reyes Mets

Balls in play
Hits

Frank Franklin II/Associated Press



Space, Time and Groceries



Exploring Data

- Data Objects and Attribute Types
- Basic Statistical Descriptions of Data
- Data Visualization
- Measuring Data Similarity and Dissimilarity
- Summary

Similarity and Dissimilarity

- **Similarity**
  - Numerical measure of how alike two data objects are
  - Value is higher when objects are more alike
  - Often falls in the range [0,1]
- **Dissimilarity** (e.g., distance)
  - Numerical measure of how different two data objects are
  - Lower when objects are more alike
  - Minimum dissimilarity is often 0
  - Upper limit varies
- **Proximity** refers to a similarity or dissimilarity

Data Matrix and Dissimilarity Matrix

- Data matrix
  - n data points with p dimensions
  - Two modes

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

- Dissimilarity matrix
  - n data points, but registers only the distance
  - A triangular matrix
  - Single mode

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

Proximity Measure for Nominal Attributes

- Can take 2 or more states, e.g., red, yellow, blue, green (generalization of a binary attribute)
- Method 1: Simple matching
  - *m*: # of matches, *p*: total # of variables

$$d(i,j) = \frac{p - m}{p}$$

- Method 2: Use a large number of binary attributes
  - creating a new binary attribute for each of the *M* nominal states

Proximity Measure for Binary Attributes

- A contingency table for binary data

Object *i* \ Object *j*	1	0	sum
1	q	r	$q+r$
0	s	t	$s+t$
sum	$q+s$	$r+t$	p

- Distance measure for symmetric binary variables:

$$d(i,j) = \frac{r+s}{q+r+s+t}$$

- Distance measure for asymmetric binary variables:

$$d(i,j) = \frac{r+s}{q+r+s}$$

- Jaccard coefficient (*similarity* measure for *asymmetric* binary variables):

$$sim_{Jaccard}(i,j) = \frac{q}{q+r+s}$$

- Note: Jaccard coefficient is the same as "coherence":

$$coherence(i,j) = \frac{sup(i,j)}{sup(i) + sup(j) - sup(i,j)} = \frac{q}{(q+r)+(q+s)-q}$$

Dissimilarity between Binary Variables

- Example

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

- Gender is a symmetric attribute
  (assume similarity is computed based only on asymmetric attributes)
- The remaining attributes are asymmetric binary
- Let the values Y and P be 1, and the value N 0

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.75$$

Distance on Numeric Data: Minkowski Distance

- *Minkowski distance*: A popular distance measure

$$d(i, j) = \sqrt[h]{|x_{i1} - x_{j1}|^h + |x_{i2} - x_{j2}|^h + \cdots + |x_{ip} - x_{jp}|^h}$$

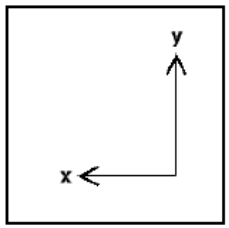
where $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ are two p-dimensional data objects, and h is the order (the distance so defined is also called L-h norm)

- Properties
  - d(i, j) > 0 if i ≠ j, and d(i, i) = 0 (Positive definiteness)
  - d(i, j) = d(j, i) (Symmetry)
  - d(i, j) ≤ d(i, k) + d(k, j) (Triangle Inequality)
- A distance that satisfies these properties is a metric

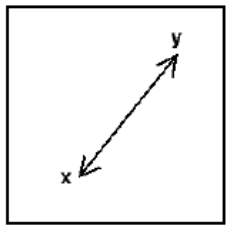
Special Cases of Minkowski Distance

- *h* = 1: Manhattan (city block, L_1 norm) distance
  - E.g., the Hamming distance: the number of bits that are different between two binary vectors

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \ldots + |x_{ip} - x_{jp}|$$



Manhattan



Euclidean

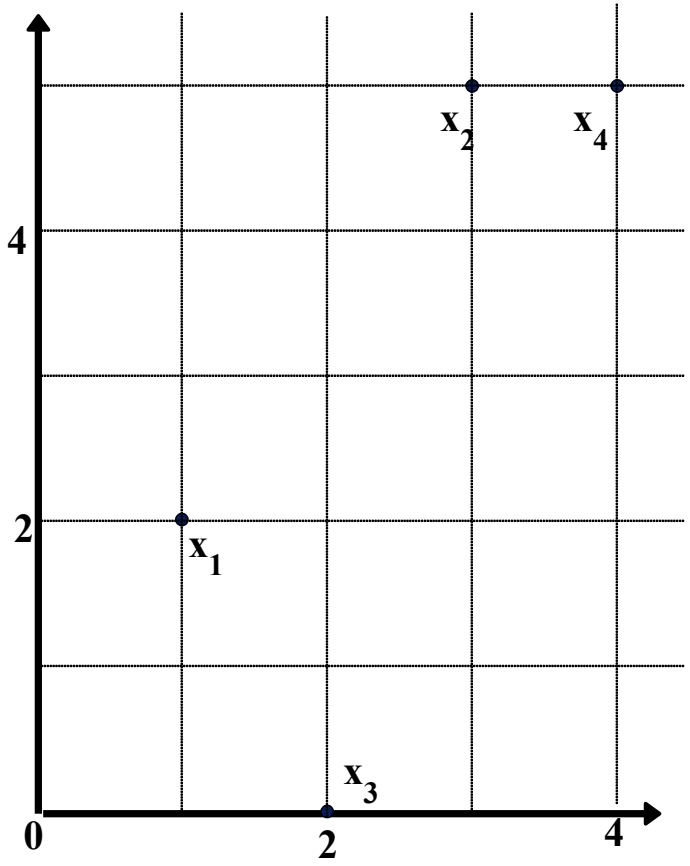
- *h* = 2: (L_2 norm) Euclidean distance

$$d(i,j) = \sqrt{(|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \ldots + |x_{ip} - x_{jp}|^2)}$$

- $h \to \infty$. "supremum" (L_{max} norm, L_∞ norm) distance.
  - This is the maximum difference between any component (attribute) of the vectors

$$d(i,j) = \lim_{h \to \infty} \left(\sum_{f=1}^{p} |x_{if} - x_{jf}|^h \right)^{\frac{1}{h}} = \max_{f}^{p} |x_{if} - x_{jf}|$$

Example:
Data Matrix and Dissimilarity Matrix



Data Matrix

point	attribute1	attribute2
x1	1	2
x2	3	5
x3	2	0
x4	4	5

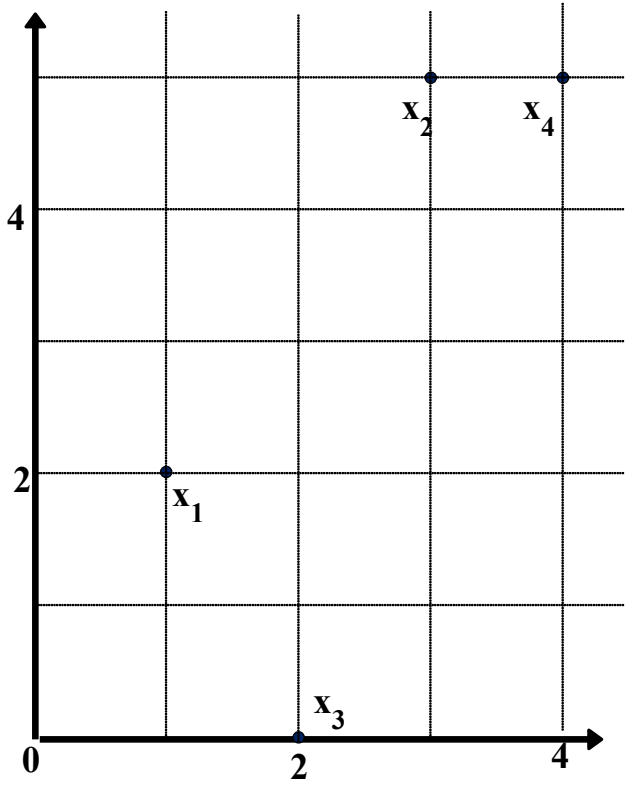
Dissimilarity Matrix

(with Euclidean Distance)

	x1	*x2*	*x3*	*x4*
x1	0			
x2	3.61	0		
x3	5.1	5.1	0	
x4	4.24	1	5.39	0

Example: Minkowski Distance

point	attribute 1	attribute 2
x1	1	2
x2	3	5
x3	2	0
x4	4	5



Dissimilarity Matrices

Manhattan (L_1)

L	x1	x2	x3	x4
x1	0			
x2	5	0		
x3	3	6	0	
x4	6	1	7	0

Euclidean (L_2)

L2	x1	x2	x3	x4
x1	0			
x2	3.61	0		
x3	2.24	5.1	0	
x4	4.24	1	5.39	0

Supremum

L_∞	x1	x2	x3	x4
x1	0			
x2	3	0		
x3	2	5	0	
x4	3	1	5	0

Cosine Similarity

- A **document** can be represented by thousands of attributes, each recording the *frequency* of a particular word (such as keywords) or phrase in the document.

Document	*team*	*coach*	*hockey*	*baseball*	*soccer*	*penalty*	*score*	*win*	*loss*	*season*
Document1	5	0	3	0	2	0	0	2	0	0
Document2	3	0	2	0	1	1	0	1	0	1
Document3	0	7	0	2	1	0	0	3	0	0
Document4	0	1	0	0	1	2	2	0	3	0

- Other vector objects: gene features in micro-arrays, …
- Applications: information retrieval, biologic taxonomy, gene feature mapping, ...
- Cosine measure: If d_1 and d_2 are two vectors (e.g., term-frequency vectors), then

$$\cos(d_1, d_2) = (d_1 \bullet d_2) / ||d_1|| \, ||d_2|| ,$$

where $\bullet$ indicates vector dot product, $||d||$: the length of vector d

Example: Cosine Similarity

- $\cos(d_1, d_2) = (d_1 \bullet d_2) / ||d_1|| \, ||d_2||$,
  where $\bullet$ indicates vector dot product, $||d|$: the length of vector d

- Ex: Find the **similarity** between documents 1 and 2.

d_1 = (5, 0, 3, 0, 2, 0, 0, 2, 0, 0)
d_2 = (3, 0, 2, 0, 1, 1, 0, 1, 0, 1)

$d_1 \bullet d_2$ = 5*3+0*0+3*2+0*0+2*1+0*1+0*1+2*1+0*0+0*1 = 25
$||d_1|| = (5*5+0*0+3*3+0*0+2*2+0*0+0*0+2*2+0*0+0*0)^{0.5} = (42)^{0.5} = 6.481$
$||d_2|| = (3*3+0*0+2*2+0*0+1*1+1*1+0*0+1*1+0*0+1*1)^{0.5} = (17)^{0.5} = 4.12$
$\cos(d_1, d_2) = 0.94$

Exploring Data

- Data Objects and Attribute Types
- Basic Statistical Descriptions of Data
- Data Visualization
- Measuring Data Similarity and Dissimilarity
- Summary

Summary

- Data attribute types: nominal, binary, ordinal, interval-scaled, ratio-scaled
- Many types of data sets, e.g., numerical, text, graph, Web, image.
- Gain insight into the data by:
  - Basic statistical data description: central tendency, dispersion, graphical displays
  - Data visualization: map data onto graphical primitives
  - Measure data similarity
- Above steps are the beginning of data preprocessing.
- Many methods have been developed but still an active area of research.

Data Mining: an Overview

What is Data Mining?

- **Knowledge discovery in databases**
  - Extraction of interesting (non-trivial, implicit, previously unknown and potentially useful) patterns or knowledge from huge amount of data.
- Alternative names:
  - Knowledge discovery (mining) in databases (KDD), knowledge extraction, data/pattern analysis, data archeology, data dredging, information harvesting, business intelligence, etc.

Why Data Mining?

- The Explosive Growth of Data: from terabytes to petabytes (10^{15} B= 1 million GB)
  - Data collection and data availability
    - Automated data collection tools, database systems, Web, computerized society
  - Major sources of abundant data
    - Business: Web, e-commerce, transactions, stocks, …
    - Science: Remote sensing, bioinformatics, scientific simulation, …
    - **Society and everyone**: news, digital cameras, YouTube, Facebook
- We are drowning in data, but starving for knowledge!
- "Necessity is the mother of invention"—Data mining—Automated analysis of massive data sets

Why Not Traditional Data Analysis?

- Tremendous amount of data
  - Algorithms must be highly scalable to handle such as terabytes of data
- High-dimensionality of data
  - Many features are extracted to describe the data
- High complexity of data
- New and sophisticated applications

Evolution of Database Technology

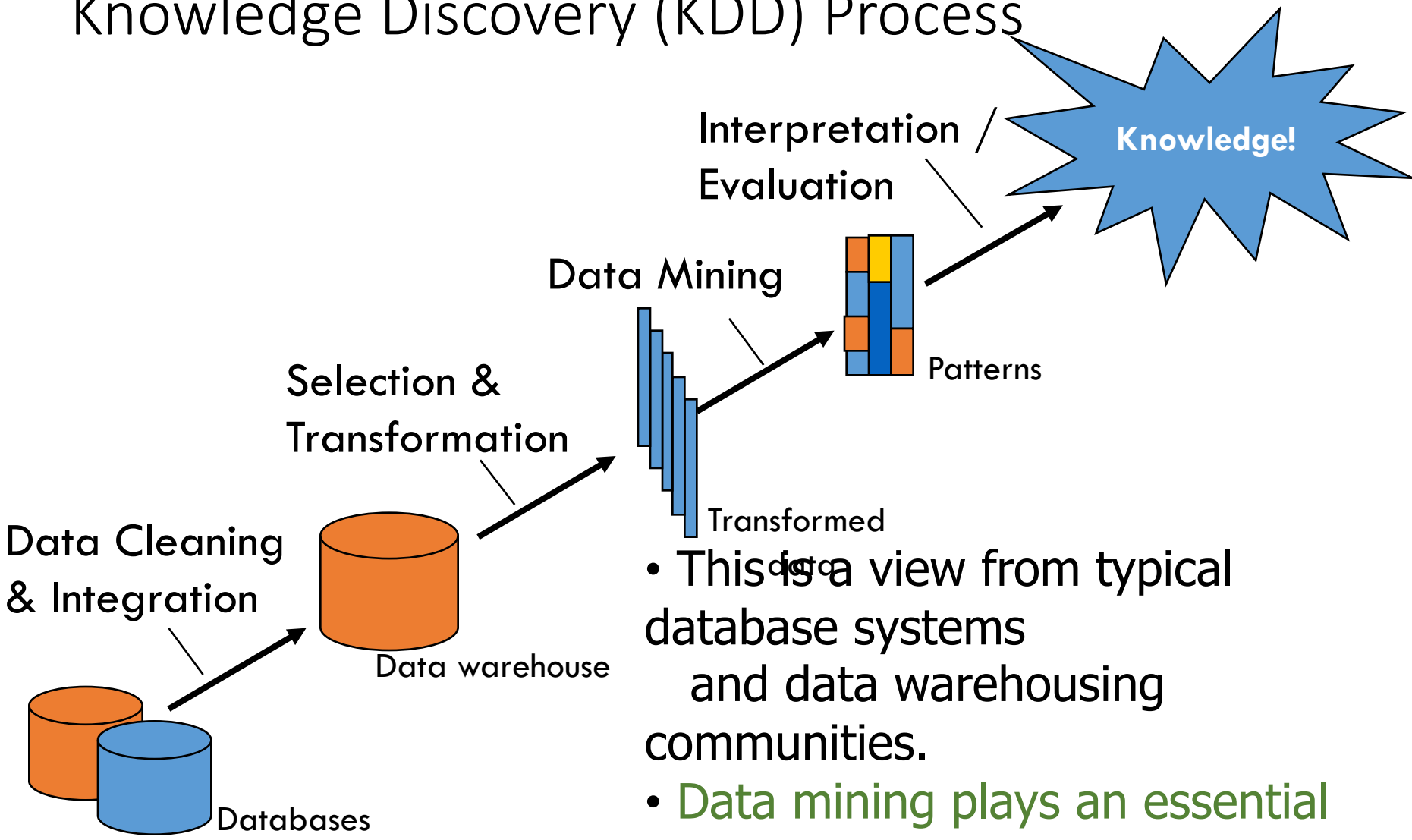
- 1960s:
  - Data collection, database creation, IMS and network DBMS
- 1970s:
  - Relational data model, relational DBMS implementation
- 1980s:
  - RDBMS, advanced data models (extended-relational, OO, deductive, etc.)
  - Application-oriented DBMS (spatial, scientific, engineering, etc.)

- 1990s:
  - Data mining, data warehousing, multimedia databases, and Web databases
- 2000s
  - Stream data management and mining
  - Data mining and its applications
  - Web technology (XML, data integration) and global information systems
- 2010s
  - Cloud computing
  - Big data/Data science

Data Mining: On What Kinds of Data?

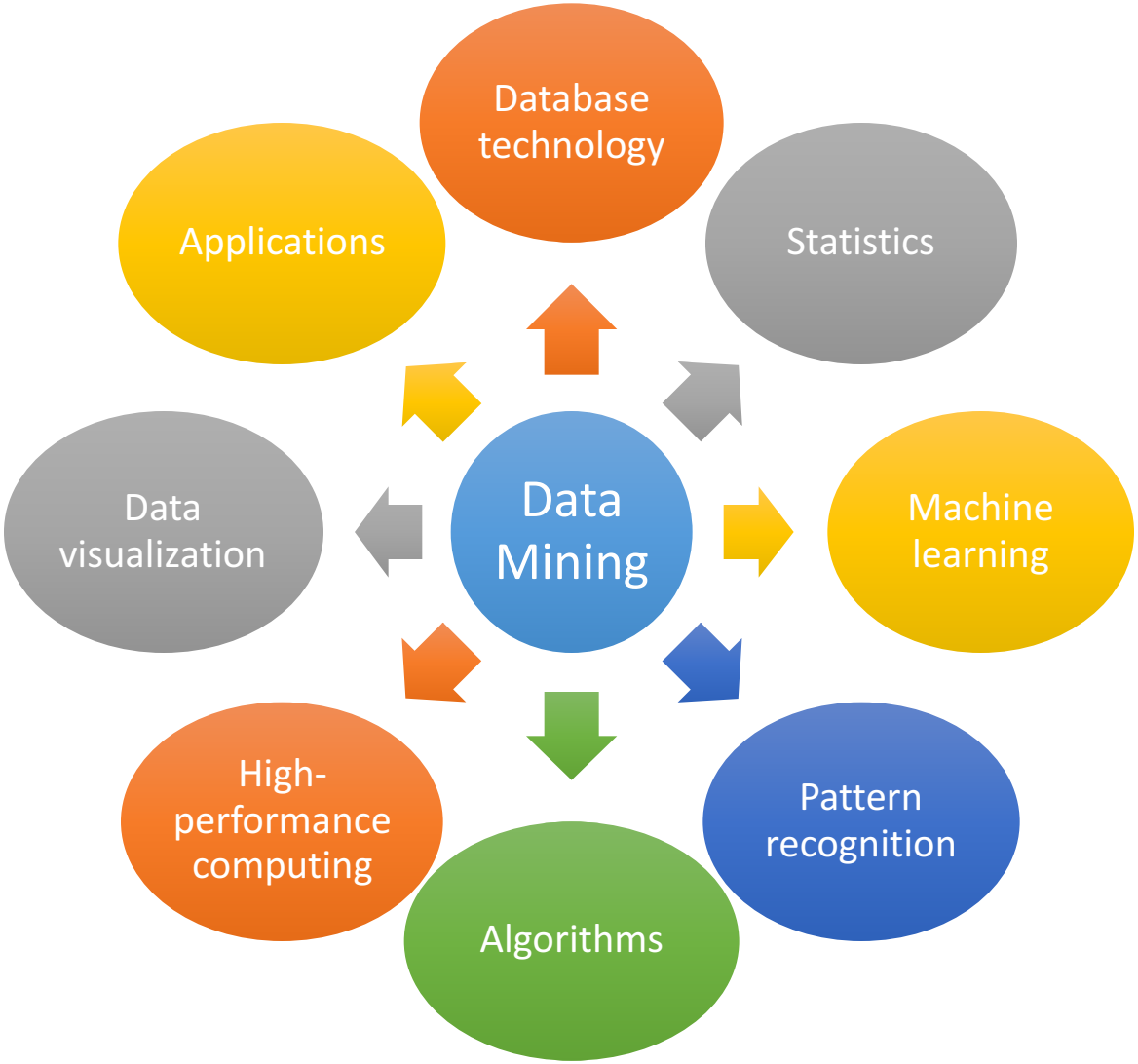
- Database-oriented data sets and applications
  - Relational database, data warehouse, transactional database
- Advanced data sets and advanced applications
  - Data streams and sensor data
  - Time-series data, temporal data, sequence data (incl. bio-sequences)
  - Structure data, graphs, social networks and multi-linked data
  - Object-relational databases
  - Heterogeneous databases and legacy databases
  - Spatial data and spatiotemporal data
  - Multimedia database
  - Text databases
  - The World-Wide Web

Knowledge Discovery (KDD) Process

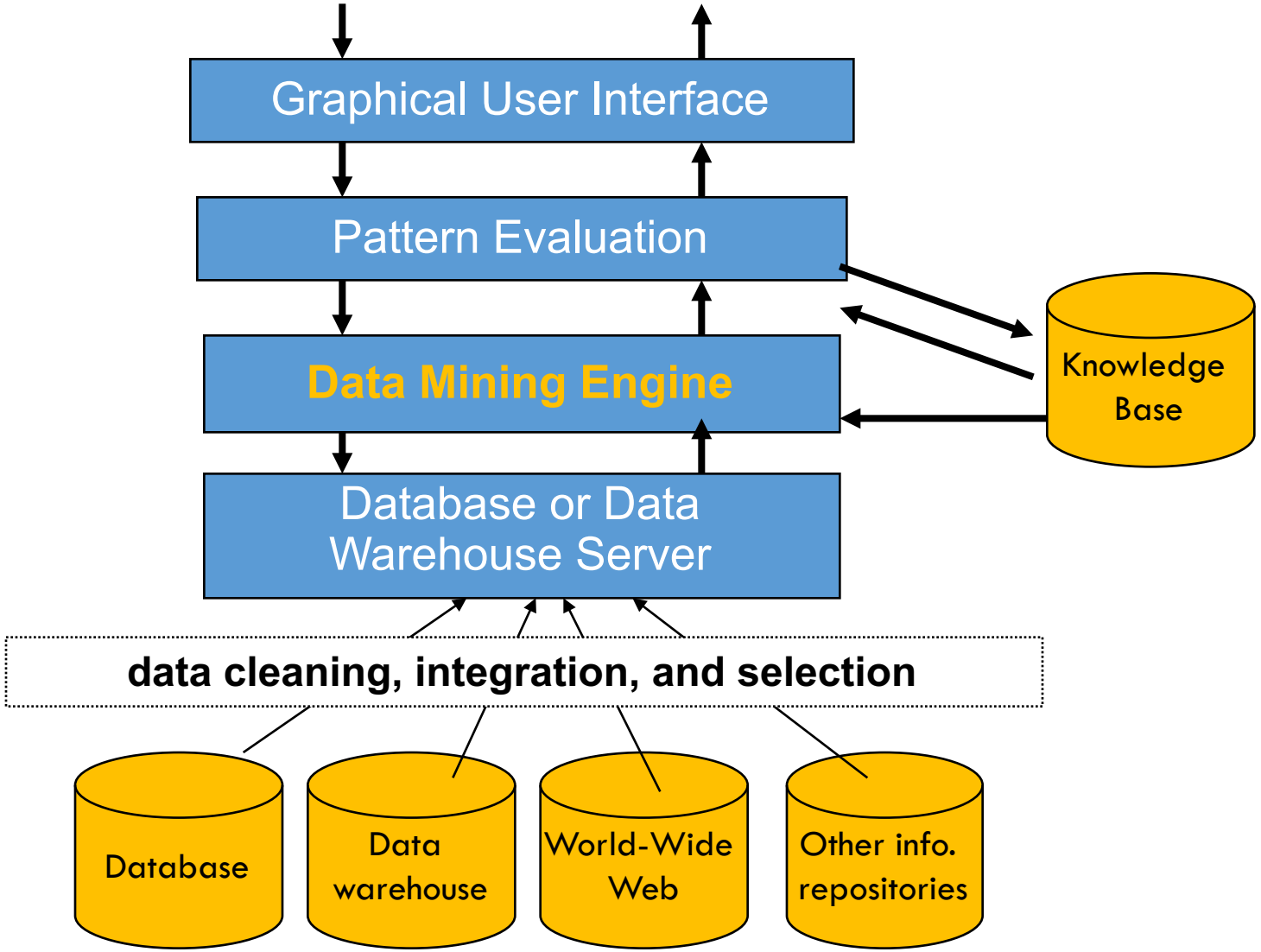


- This is a view from typical database systems and data warehousing communities.
- Data mining plays an essential role in the knowledge discovery process.

Data Mining: Confluence of Multiple Disciplines



Typical Data Mining System



Multi-Dimensional View of Data Mining

- **Data to be mined**
  - Relational, data warehouse, transactional, stream, object-oriented/relational, active, spatial, time-series, text, multi-media, heterogeneous, legacy, WWW
- **Knowledge to be mined**
  - Characterization, discrimination, association, classification, clustering, trend/deviation, outlier analysis, etc.
  - Multiple/integrated functions and mining at multiple levels
- **Techniques utilized**
  - Database-oriented, machine learning, statistics, visualization, etc.
- **Applications adapted**
  - Retail, telecommunication, banking, fraud analysis, bio-data mining, stock market analysis, text mining, Web mining, etc.

Mining Capabilities (1/4)

- Multi-dimensional concept description: Characterization and discrimination
  - Generalize, summarize, and contrast data characteristics
- Frequent patterns (or frequent itemsets), association
  - Diaper → Beer [0.5%, 75%] (support, confidence)

Mining Capabilities (2/4)

- Classification and prediction
  - Construct models (functions) that describe and distinguish classes or concepts for future prediction
    - E.g., classify countries based on (climate), or classify cars based on (gas mileage)
  - Predict some unknown or missing numerical values

Mining Capabilities (3/4)

- Clustering
  - Class label is unknown: Group data to form new categories (i.e., clusters), e.g., cluster houses to find distribution patterns
  - Maximizing intra-class similarity & minimizing interclass similarity
- Outlier analysis
  - Outlier: Data object that does not comply with the general behavior of the data
  - Noise or exception? Useful in fraud detection, rare events analysis

Mining Capabilities (4/4)

- Time and ordering, trend and evolution analysis
  - Trend and deviation: e.g., regression analysis
  - Sequential pattern mining: e.g., digital camera → large SD memory
  - Periodicity analysis
  - Motifs and biological sequence analysis
    - Approximate and consecutive motifs
  - Similarity-based analysis

More Advanced Mining Techniques

- Data stream mining
  - Mining data that is ordered, time-varying, potentially infinite.
- Information network analysis
  - Social networks: actors (objects, nodes) and relationships (edges)
    - e.g., author networks, terrorist networks
  - Multiple heterogeneous networks
    - A person could be multiple information networks: friends, family, classmates, …
  - Links carry a lot of semantic information: Link mining

More Advanced Mining Techniques (cont.)

- Graph mining
  - Finding frequent subgraphs (e.g., chemical compounds), trees (XML), substructures (web fragments)
- Web mining
  - Web is a big information network: from PageRank to Google
  - Analysis of Web information networks
    - Web community discovery, opinion mining, usage mining, …

Challenges for Data Mining

- Handling of different types of data (**heterogeneous data**)
- **Efficiency** and **scalability** of mining algorithms
- **Usefulness** and **certainty** of mining results
- **Expression** of various kinds of mining results
- Interactive mining at multiple abstraction levels
- Mining information from different sources of data
- Protection of **privacy** and data **security**

Brief Summary

- **Data mining**: Discovering interesting patterns and knowledge from massive amount of data
- A natural evolution of database technology, in great demand, with wide applications
- A KDD process includes data cleaning, data integration, data selection, transformation, data mining, pattern evaluation, and knowledge presentation
- Mining can be performed in a variety of data
- Data mining functionalities: characterization, discrimination, association, classification, clustering, outlier and trend analysis, etc.

References

- W. Cleveland, Visualizing Data, Hobart Press, 1993
- T. Dasu and T. Johnson. Exploratory Data Mining and Data Cleaning. John Wiley, 2003
- U. Fayyad, G. Grinstein, and A. Wierse. Information Visualization in Data Mining and Knowledge Discovery, Morgan Kaufmann, 2001
- L. Kaufman and P. J. Rousseeuw. Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons, 1990.
- H. V. Jagadish, et al., Special Issue on Data Reduction Techniques. Bulletin of the Tech. Committee on Data Eng., 20(4), Dec. 1997
- D. A. Keim. Information visualization and visual data mining, IEEE trans. on Visualization and Computer Graphics, 8(1), 2002
- D. Pyle. Data Preparation for Data Mining. Morgan Kaufmann, 1999
- S. Santini and R. Jain," Similarity measures", IEEE Trans. on Pattern Analysis and Machine Intelligence, 21(9), 1999
- E. R. Tufte. The Visual Display of Quantitative Information, 2nd ed., Graphics Press, 2001
- C. Yu , et al., Visual data mining of multimedia data for social and behavioral studies, Information Visualization, 8(1), 2009

Frequent Pattern Mining

Outline

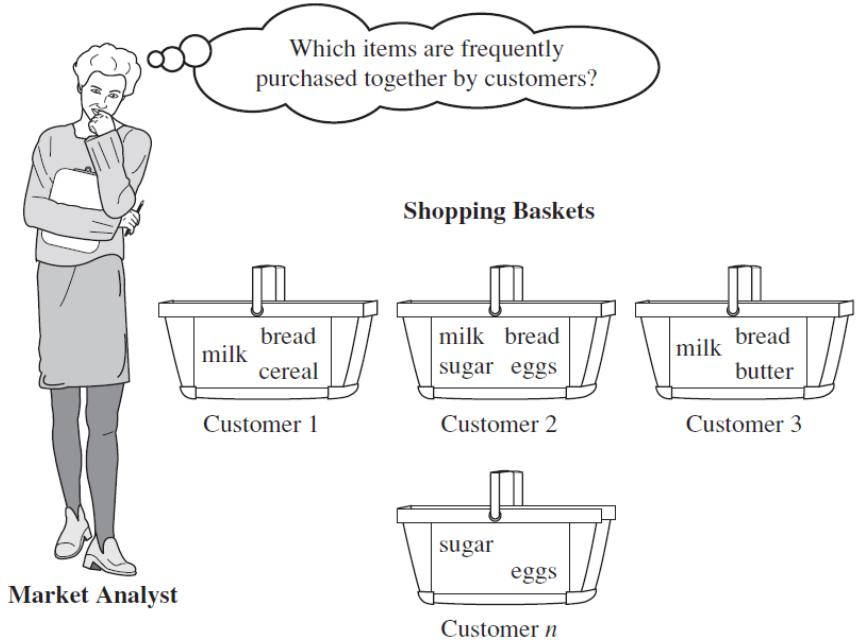
- **Basic Concepts**
- Frequent Itemset Mining Methods
- Which Patterns Are Interesting?—Pattern Evaluation Methods
- Summary

What Is Frequent Pattern Analysis?

- Frequent pattern: a pattern (a set of items, subsequences, substructures, etc.) that occurs frequently in a data set
- First proposed by Agrawal, Imielinski, and Swami [AIS93] in the context of frequent itemsets and association rule mining
- Motivation: Finding inherent regularities in data
  - What products were often purchased together?— Beer and diapers?!
  - What are the subsequent purchases after buying a PC?
  - What kinds of DNA are sensitive to this new drug?
  - Can we automatically classify web documents?

What Is Frequent Pattern Analysis?

- Applications
  - Basket data analysis
  - cross-marketing
  - catalog design
  - sale campaign analysis
  - Web log (click stream) analysis
  - DNA sequence analysis



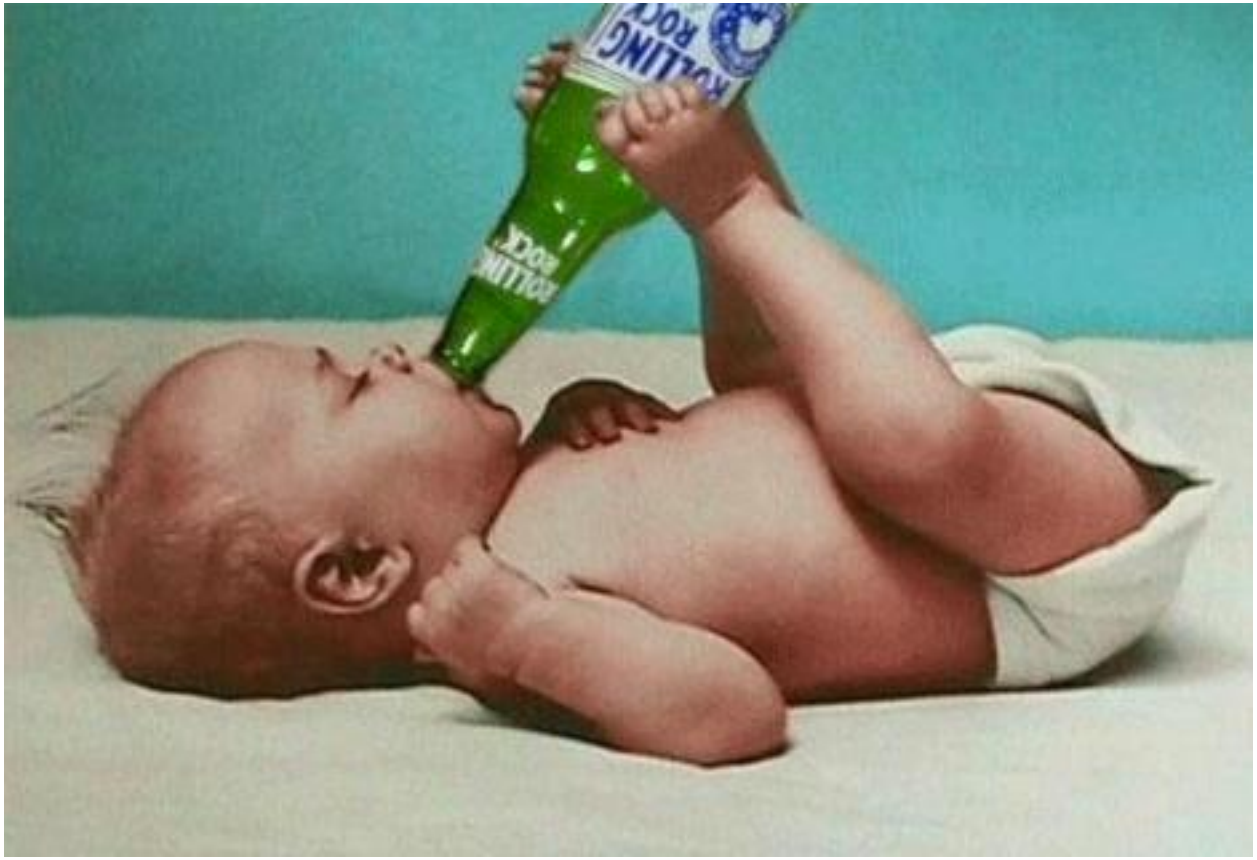
Why Is Freq. Pattern Mining Important?

- Freq. pattern: An intrinsic and important property of datasets

- Foundation for many essential data mining tasks
  - Association, correlation, and causality analysis
  - Sequential, structural (e.g., sub-graph) patterns
  - Pattern analysis in spatiotemporal, multimedia, time-series, and stream data
  - Classification: discriminative, frequent pattern analysis
  - Cluster analysis: frequent pattern-based clustering


BIG PAK
HUGGIES
30527 00
NEW!
¡NUEVO!
NOUVEAU!
ComfortFit
4
60
Labatt
IMPORTED
BLUELIGHT
CANADIAN PILSENER

Of course!

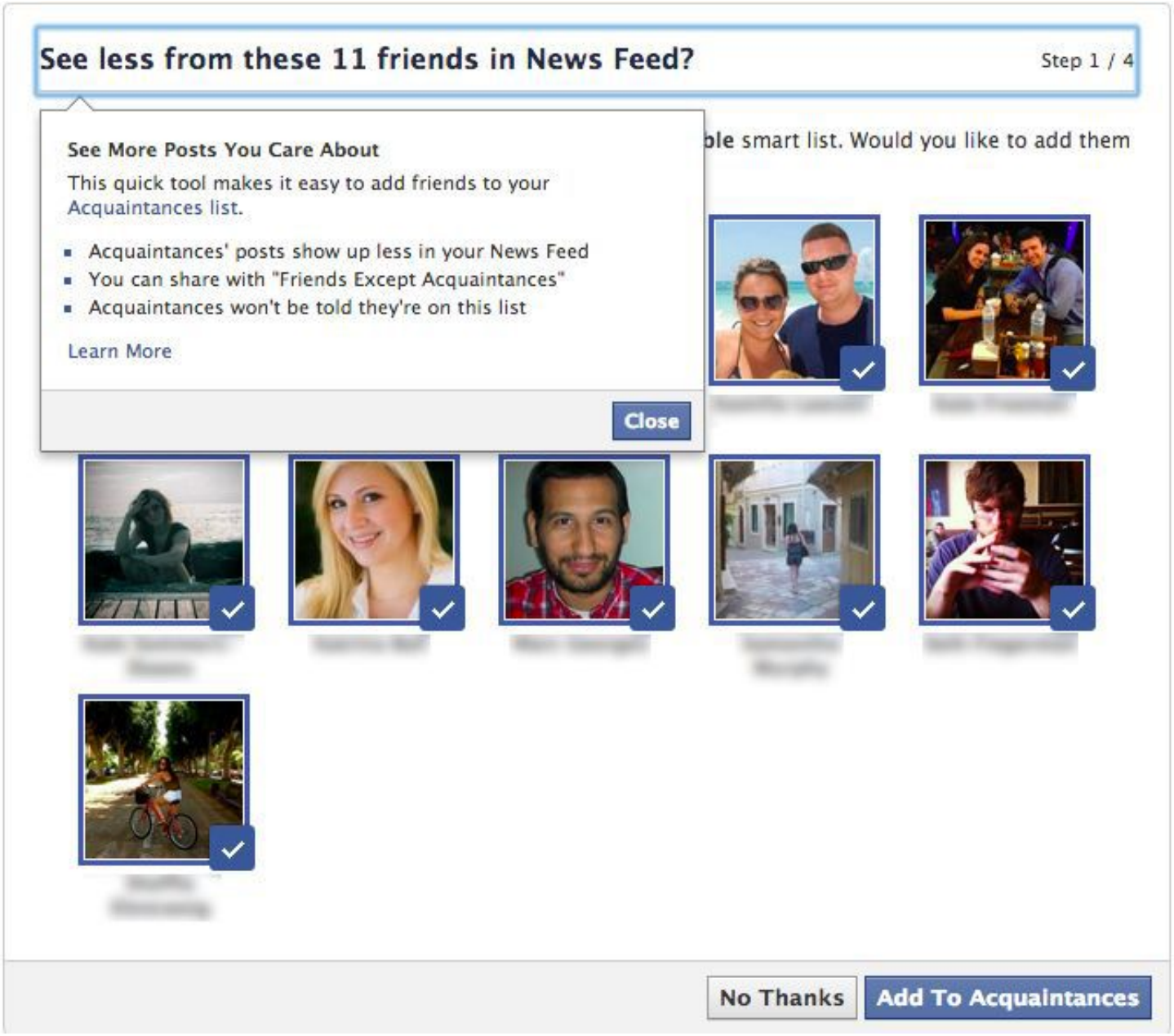


Here's the story!

- **Walmart: Friday, Diapers, Beer**

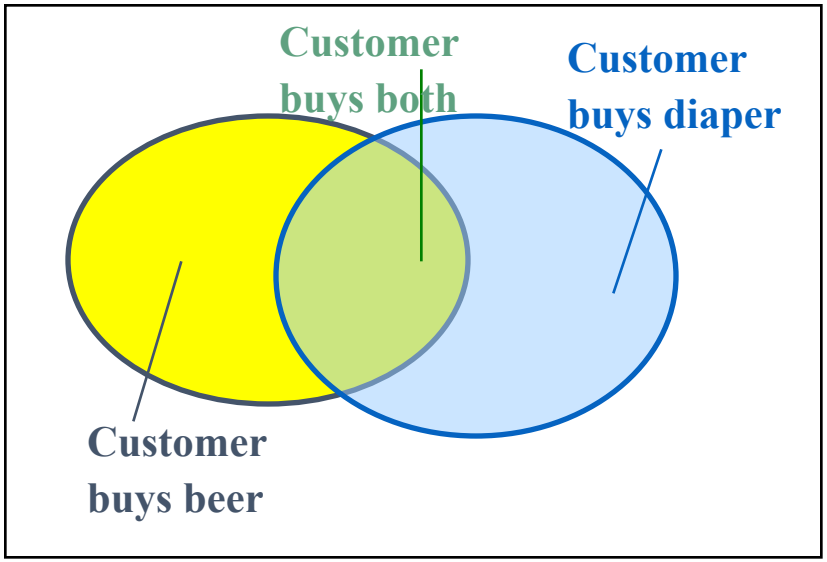
- Candidate 1: Thomas Blischok: Osco Drug, 1992
- Candidate 2: John Earle : Osco Drug, 1988
- Candidate 3: Tom Fawcett: Kdnuggets, 2000
- 化妝品與賀卡
- Candidate 4: ARonny Kohavi: Kdnuggets, 2000

- *Never let truth get in the way of a good story.*
  - *Mark Twain*


See less from these 11 friends in News Feed?
Step 1 / 4
See More Posts You Care About
This quick tool makes it easy to add friends to your Acquaintances list.
Acquaintances' posts show up less in your News Feed
You can share with "Friends Except Acquaintances"
Acquaintances won't be told they're on this list
Learn More
Close
ble smart list. Would you like to add them
No Thanks
Add To Acquaintances

Basic Concepts: Frequent Patterns

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



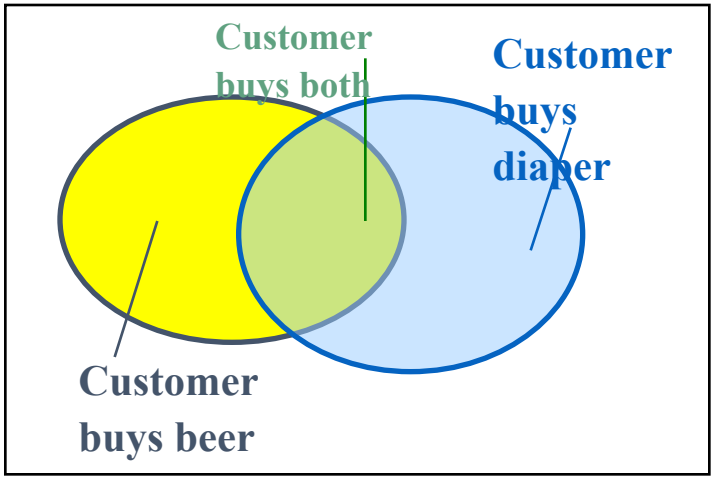
- itemset: A set of one or more items
- k-itemset $X = \{x_1, \ldots, x_k\}$
- *(absolute) support*, or, *support count* of X: Frequency or occurrence of an itemset X
- *(relative) support*, *s*, is the fraction of transactions that contains X (i.e., the probability that a transaction contains X)
- An itemset X is *frequent* if X's support is no less than a *minsup* threshold

Basic Concepts: Frequent Patterns

No error, is definition

This means "X and Y" appear at the same time

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



- support, *s*, probability that a transaction contains $X \cup Y$
- confidence, *c*, conditional probability that a transaction having X also contains *Y*

- support(Beer ∪ Diaper)=3/5=60%
- confidence(Beer=>Diaper)=3/3=100%

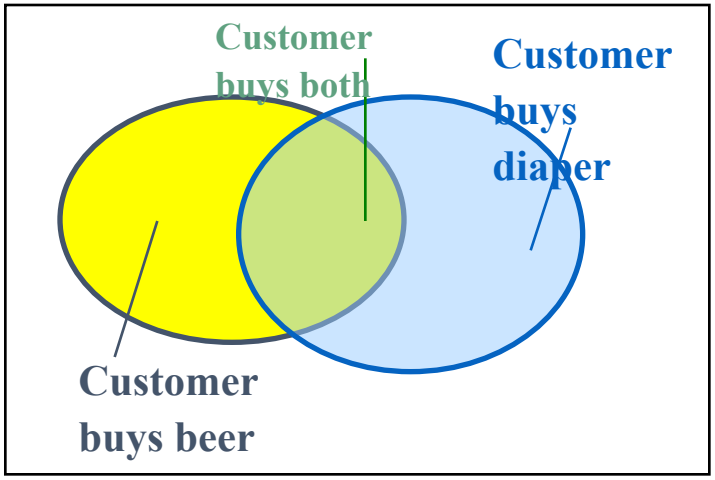
$$support(A \Rightarrow B) = P(A \cup B)$$

$$confidence(A \Rightarrow B) = P(B|A).$$

$$confidence(A \Rightarrow B) = P(B|A) = \frac{support(A \cup B)}{support(A)} = \frac{support\_count(A \cup B)}{support\_count(A)}$$

Basic Concepts: Association Rules

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



- Find all the rules $X \rightarrow Y$ with minimum support and confidence

Let minsup = 50%, minconf = 50%

Frequent Pattern:

Beer:3, Nuts:3, Diaper:4, Eggs:3, {Beer, Diaper}:3, ~~{Nuts, Diaper}:2~~

- Association rules: (many more!)
  - *Beer* → *Diaper* (60%, 100%)
  - *Diaper* → *Beer* (60%, 75%)

Closed Patterns and Max-Patterns

- A long pattern contains a combinatorial number of sub-patterns, e.g., $\{a_1, \ldots, a_{100}\}$ contains $\binom{100}{1} + \binom{100}{2} + \ldots + \binom{1}{1}\binom{0}{0}\binom{0}{0} = 2^{100} - 1 = 1.27*10^{30}$ sub-patterns!
- Solution: *Mine closed patterns and max-patterns instead*
- An itemset X is closed if X is *frequent* and there exists *no super-pattern* Y Ↄ X, *with the same support* as X
- An itemset X is a max-pattern if X is frequent and there exists no frequent super-pattern Y Ↄ X
- Closed pattern is a lossless compression of freq. patterns
  - Reducing the # of patterns and rules

Closed Patterns and Max-Patterns

- Example: A transaction database with 2 transactions $\{<a_1, a_2, \ldots, a_{100}>; <a_1, a_2, \ldots, a_{50}>\}$
- Let ***min_sup = 1***
- Closed freq. itemsets: $C=\{\{a_1, a_2, \ldots, a_{100}\}:1, \{a_1, a_2, \ldots, a_{50}\}:2\}$
- Only 1 maximal freq. itemset: $M=\{\{a_1, a_2, \ldots, a_{100}\}:1\}$
- Closed freq. itemsets contains complete info
  - From C, we derive 1) $\{a_2, a_{45}\}:2$ (i.e., $\{a_2, a_{45}\} \in \{a_1, a_2, \ldots, a_{50}\}:2$), 2) $\{a_8, a_{55}\}:1$ (i.e., $\{a_8, a_{55}\} \in \{a_1, a_2, \ldots, a_{100}\}:1$)
- However, from maximal freq. itemset, we only know that $\{a_2, a_{45}\}$ and $\{a_8, a_{55}\}$ are frequent, but no information of their supports

Computational Complexity of Frequent Itemset Mining

- How many itemsets are potentially to be generated in the worst case?
  - The number of frequent itemsets to be generated is sensitive to the min_sup threshold
  - When min_sup is low, an exponential number of frequent itemsets
  - The worst case: M^N where M: # distinct items, and N: max length of transactions $\binom{M}{N} = M \times (M-1) \times \cdots \times (M-N+1)/N!$
- The worst case complexty vs. the expected probability
  - Eg. Suppose Walmart has 10^4 kinds of products
    - The chance to pick up one product 10^{-4}
    - The chance to pick up a particular set of 10 products: ~10^{-40}
    - What is the chance this particular set of 10 products to be frequent 10^3 times in 10^9 transactions?

Outline

- Basic Concepts
- **Frequent Itemset Mining Methods**
- Which Patterns Are Interesting?—Pattern Evaluation Methods
- Summary

Scalable Frequent Itemset Mining Methods

- **Apriori**: A Candidate Generation-and-Test Approach
- **FPGrowth**: A Frequent Pattern-Growth Approach
- **ECLAT**: Frequent Pattern Mining with Vertical Data Format

First Consider a Brute-Force Approach

- Incrementally construct 1-itemset, 2-itemset, …, k-itemset, (k+1)-itemset

- What are the main problems?
  - 1) Unnecessary itemset examination, i.e., constructing too many non-frequent itemsets
    E.g., 1-itemset: I1, …, I5,
    2-itemset: {I1, I2}, {I1, I3}, …
  - 2) Generating duplicate itemsets:
    {I1, I2}, {I1, I3}, {I2, I3} => cross product may result in multiple {I1, I2, I3}

TID	*List of item_IDs*
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

The Downward Closure Property and Scalable Mining Methods

- The downward closure property of frequent patterns
  - Any subset of a frequent itemset must be frequent
  - If **{beer, diaper, nuts}** is frequent, so is **{beer, diaper}**
  - i.e., every transaction having {beer, diaper, nuts} also contains {beer, diaper}

- Scalable mining methods: Three major approaches
  - Apriori (Agrawal & Srikant@VLDB'94)
  - Freq. pattern growth (FPgrowth—Han, Pei & Yin @SIGMOD'00)
  - Vertical data format approach (Charm—Zaki & Hsiao @SDM'02)

Apriori: A Candidate Generation & Test Approach

- Apriori pruning principle: If there is any itemset which is infrequent, its superset should not be generated/tested!
- Method:
  - Initially, scan DB once to get frequent 1-itemset
  - Generate length (k+1) candidate itemsets from length k frequent itemsets
    - Joining 2 k-itemsets: 2 k-itemsets I1, I2 could be joined into a (k+1)-itemset if I1 and I2 have the same first (k-1) items
    - E.g., {A, B, C, D, E} join {A, B, C, D, F} = {A, B, C, D, E, F}
      {A, B, C, D, E} cannot join {A, B, C, E, F}
      => Avoid generating duplicate itemsets
  - Test the candidates against DB
  - Terminate when no frequent or candidate set can be generated

The Apriori Algorithm—An Example

min. support =2 tx's (50%)

Database TDB

Tid	Items
100	A, C, D
200	B, C, E
300	A, B, C, E
400	B, E

1st scan

C_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

L_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

C_2

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

2nd scan

C_2

Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

L_2

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

C_3

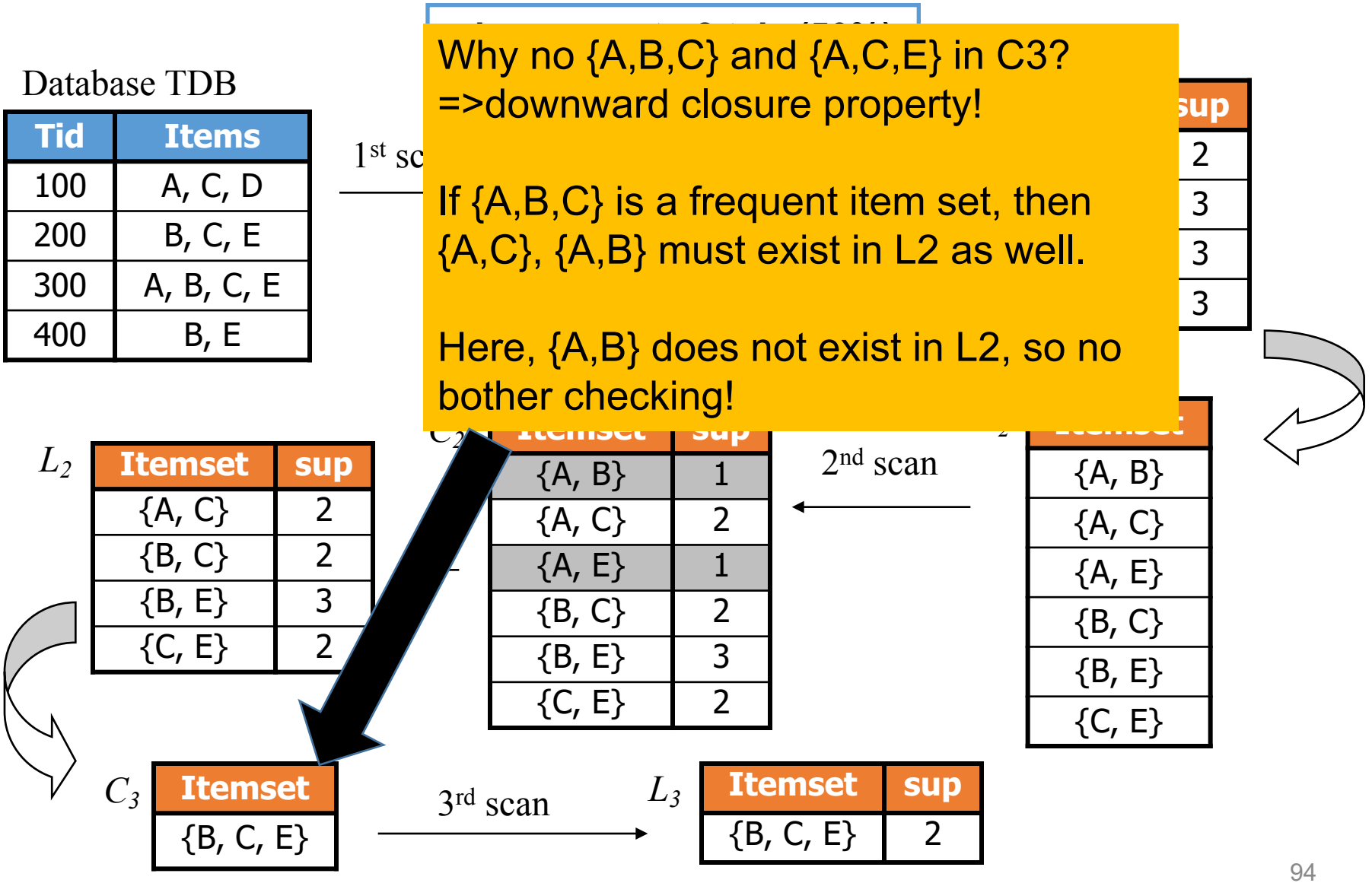
Itemset
{B, C, E}

3rd scan

L_3

Itemset	sup
{B, C, E}	2

The Apriori Algorithm—An Example



Database TDB

Tid	Items
100	A, C, D
200	B, C, E
300	A, B, C, E
400	B, E

From Large Itemsets to Rules

- Recall that confidence is defined as:

$$confidence(A \Rightarrow B) = P(B|A) = \frac{support\_count(A \cup B)}{support\_count(A)}$$

L_3

Itemset	sup
{B, C, E}	2

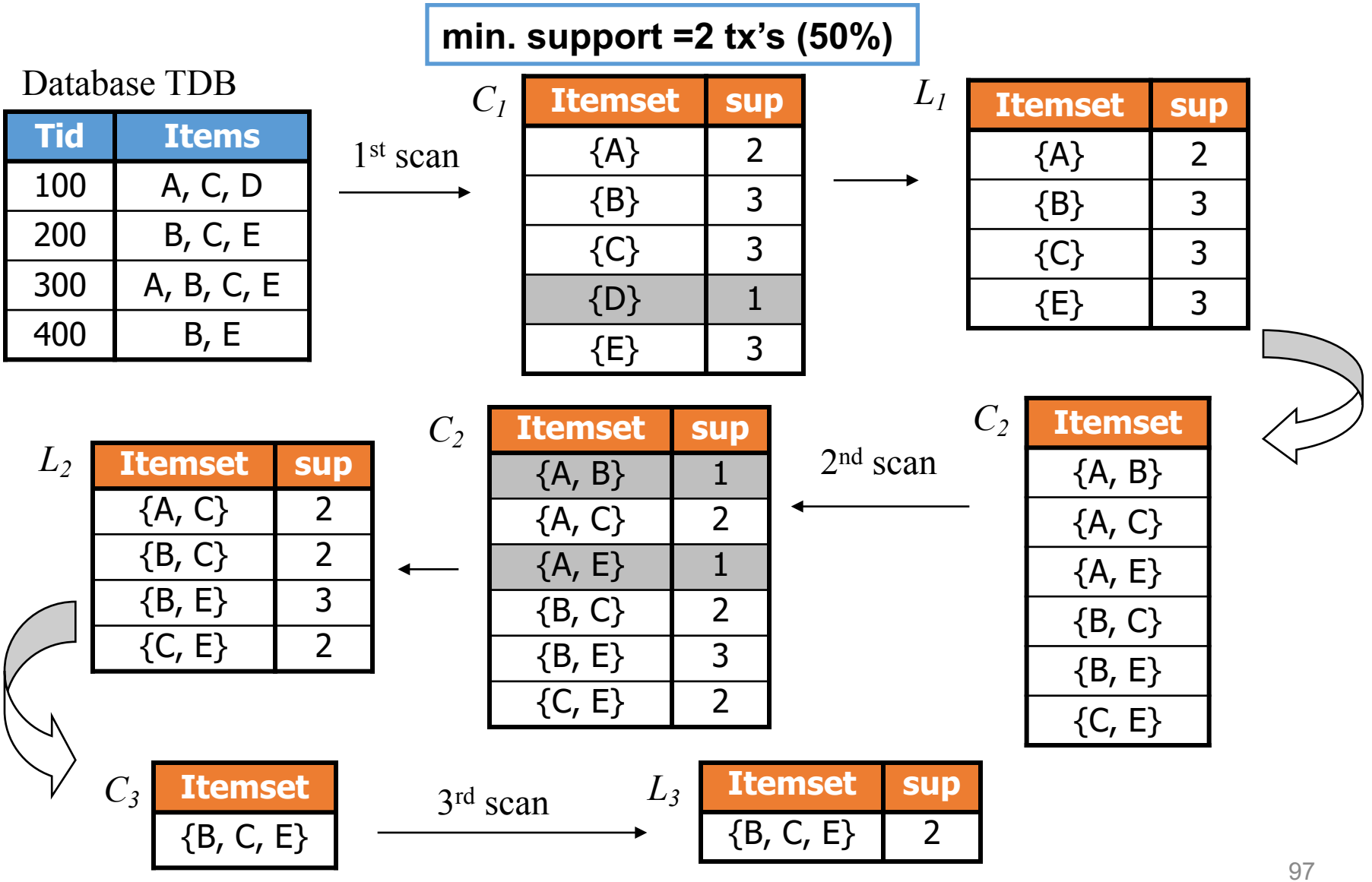
- For each large itemset *l*
  - For each subset *s* of *l*, if (*sup(l) / sup(s)* ≥ *min_conf*)
    - output the rule ***s=> (l-s)***
      - conf. = sup(l)/sup(s)
      - support = sup(l)

- E.g., *l*={B, C, E} with support =2 is a frequent 3-item set, assume min_conf=80%
- Let *s*={C,E}, *sup(l) / sup(s)* =2/2=100%>80%
- Therefore, {C,E}=>{B} is an association rule with support = 50%, confidence = 100%

Redundant Rules

- For the same support and confidence, if we have a rule {a,d} →{c,e,f,g}, do we have:
  - {a,d} →{c,e,f} ? — Yes!
  - {a} →{c,e,f,g} ? — ~~Yes!~~ **NO!**
    - **Consider the example in previous page**
    - *l*={B, C, E}, *s*={C,E}, then {C,E}->{B} is an association rule
    - **However,** *l*={B, C, E}, *s*={C}, i.e., {C}->{B} is not
  - {a,d,c} →{e,f,g} ? — Yes!
  - {a} →{c,d,e,f,g} ? — No!

The Apriori Algorithm—An Example



The Apriori Algorithm (Pseudo-Code)

C_k: Candidate itemset of size k
L_k : frequent itemset of size k

L_1 = {frequent items};
for (k = 1; L_k !=∅; k++) **do begin**
  C_{k+1} = candidates generated from L_k;
  for each transaction t in database do
    increment the count of all candidates in C_{k+1} that are contained in t
  L_{k+1} = candidates in C_{k+1} with min_support
  end
return $\cup_k L_k$;

Implementation of Apriori

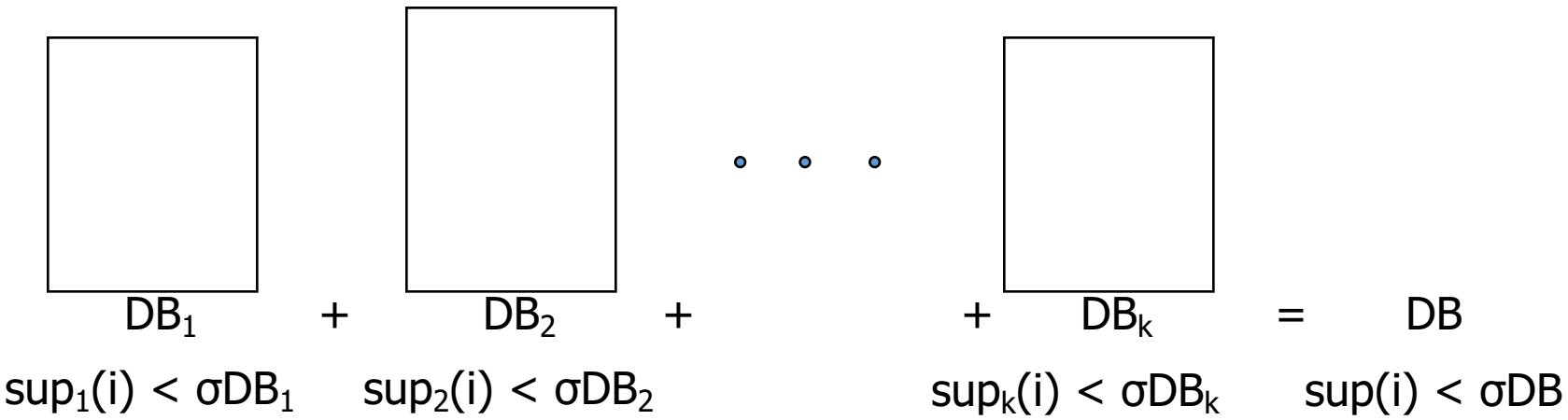
- How to generate candidates?
  - Step 1: self-joining L_k
  - Step 2: pruning
- Example of Candidate-generation
  - L_3={*abc, abd, acd, ace, bcd*}
  - Self-joining: $L_3 * L_3$
    - *abcd* from *abc* and *abd*
    - *acde* from *acd* and *ace*
  - Pruning:
    - *acde* is removed because *ade* is not in L_3
  - C_4 = {*abcd*}

Further Improvement of the Apriori Method

- Major computational challenges
  - Multiple scans of transaction database
  - Huge number of candidates (especially C2)
  - Tedious workload of support counting for candidates
- Improving Apriori: general ideas
  - Reduce passes of transaction database scans
  - Shrink number of candidates
  - Facilitate support counting of candidates

Partition: Scan Database Only Twice

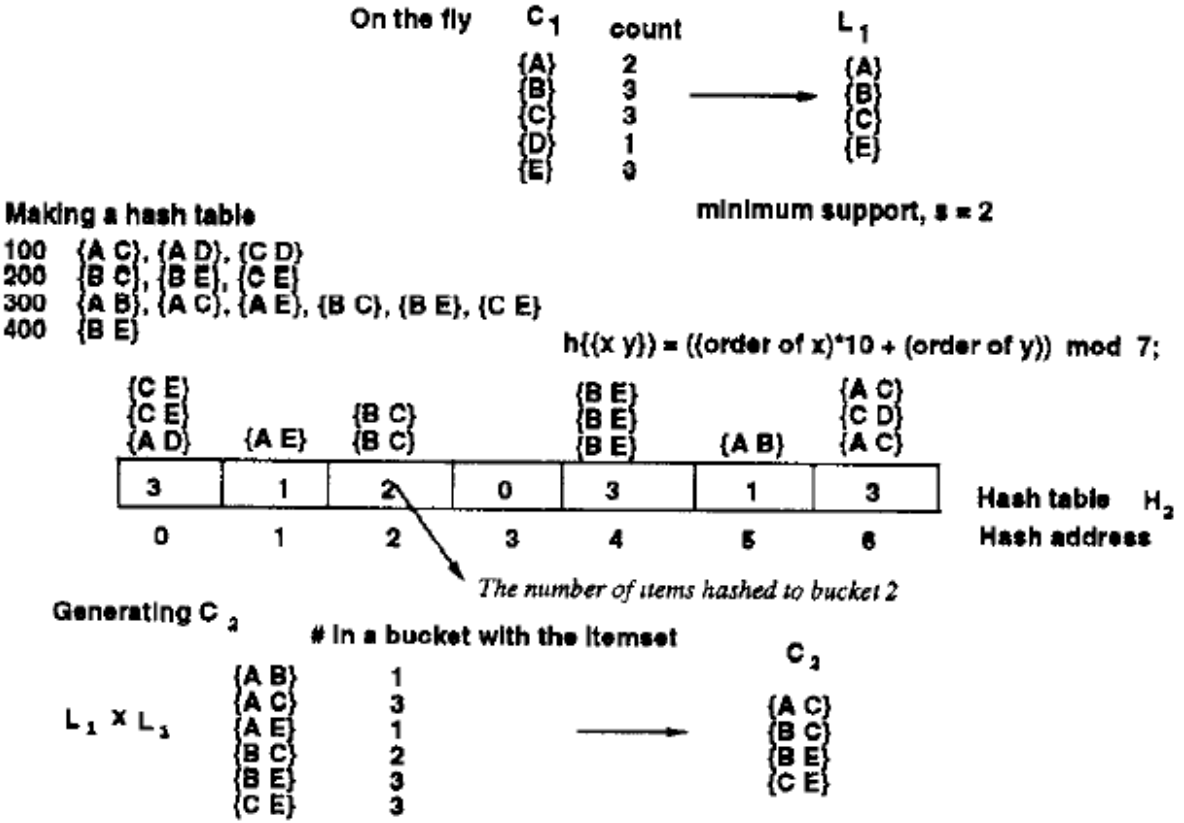
- Partition original DB into ***k*** small DB
- Any itemset that is potentially frequent in DB must be frequent in at least one of the partitions of DB
  - Scan 1: partition database and find local frequent patterns
  - Scan 2: consolidate global frequent patterns
- If the minimum support ratio is ***min_sup***, the ***min_sup*** for a partition DB_i, i.e., min_sup_i, is set to $\#tx(DB_i)$



Example Improvement 2- DHP

- DHP (direct hashing with pruning): Apriori + hashing
  - Use hash-based method to **reduce the size of C_2**.
  - Allow effective **reduction on tx database size** (tx number and each tx size.)

Tid	Items
100	A, C, D
200	B, C, E
300	A, B, C, E
400	B, E



J. Park, M.-S. Chen, and P. Yu.
An effective hash-based algorithm for mining association rules. In *SIGMOD'95*.

Scalable Frequent Itemset Mining Methods

- **Apriori**: A Candidate Generation-and-Test Approach
- **FPGrowth**: A Frequent Pattern-Growth Approach
- **ECLAT**: Frequent Pattern Mining with Vertical Data Format

Pattern-Growth Approach: Mining Frequent Patterns Without Candidate Generation

- Bottlenecks of the Apriori approach
  - Breadth-first (i.e., level-wise) search
  - Candidate generation and test
    - Often generates a huge number of candidates
- The FPGrowth Approach
  - Depth-first search
  - Avoid explicit candidate generation
- Major philosophy: Grow long patterns from short ones using local frequent items only
  - abc is a frequent pattern
  - Get all transactions having abc, i.e., project DB on abc: DB|abc
  - d is a local frequent item in DB|abc → abcd is a frequent pattern

Construct FP-tree from a Transaction Database

TID	Items bought	(ordered) frequent items
100	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, l, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o, w}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

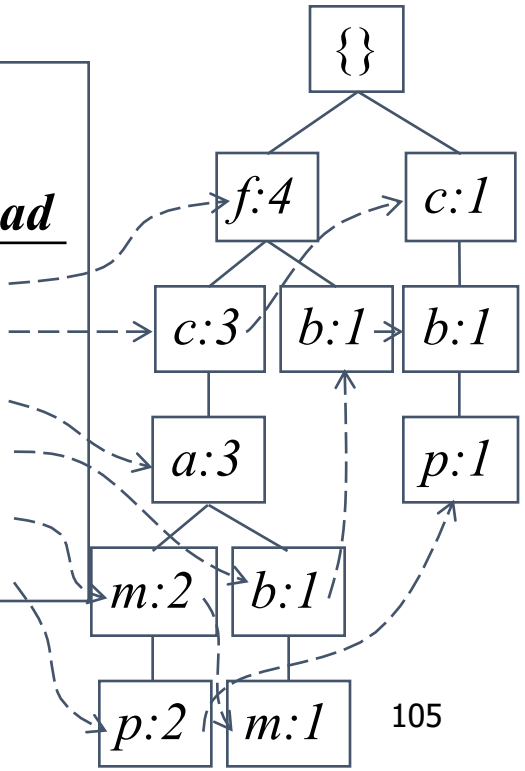
min_support = 3

F-list = f-c-a-b-m-p

1. Scan DB once, find frequent 1-itemset (single item pattern)
2. Sort frequent items in frequency descending order, F-list
3. Scan DB again, items in each transaction are processed in F-list order, construct FP-tree (see the next slide)

Header Table

Item	frequency	head
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	



Constructing FP-tree

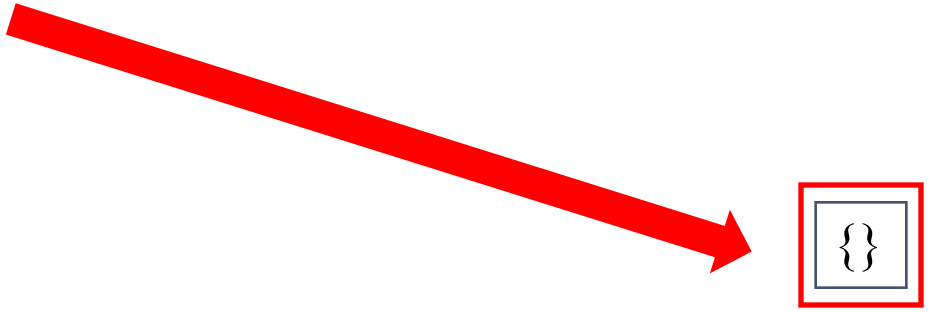
TID	Items bought	(ordered) frequent items
100	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, l, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o, w}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

min_support = 3

F-list = f-c-a-b-m-p

Constructing FP-tree

1. Create the root of the tree



Constructing FP-tree

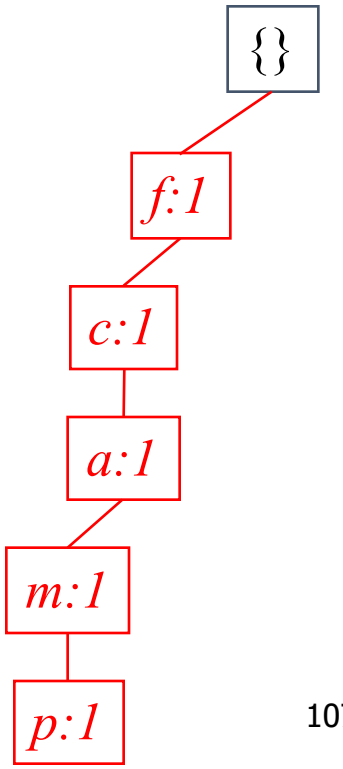
TID	Items bought	(ordered) frequent items
100	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, l, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o, w}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

min_support = 3

F-list = f-c-a-b-m-p

Constructing FP-tree

1. Create the root of the tree
2. Scan and process each tx in L order – constructing each branch of the transaction.
   - TID100 {f,c,a,m,p}



Constructing FP-tree

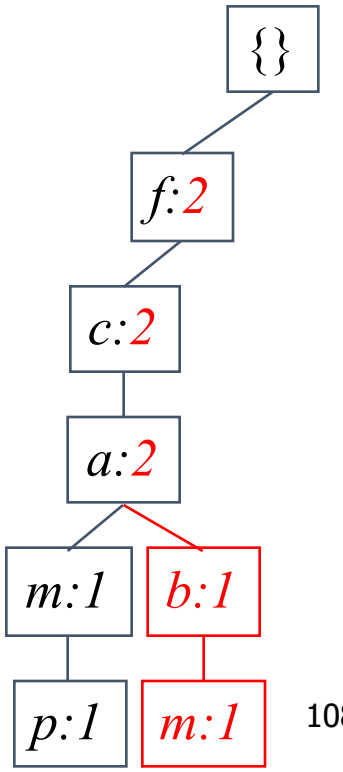
TID	Items bought	(ordered) frequent items
100	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, l, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o, w}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

min_support = 3

F-list = f-c-a-b-m-p

Constructing FP-tree

1. Create the root of the tree
2. Scan and process each tx in L order – constructing each branch of the transaction.
   - TID100 {f,c,a,m,p}
   - TID200 {f,c,a,b,m}



Constructing FP-tree

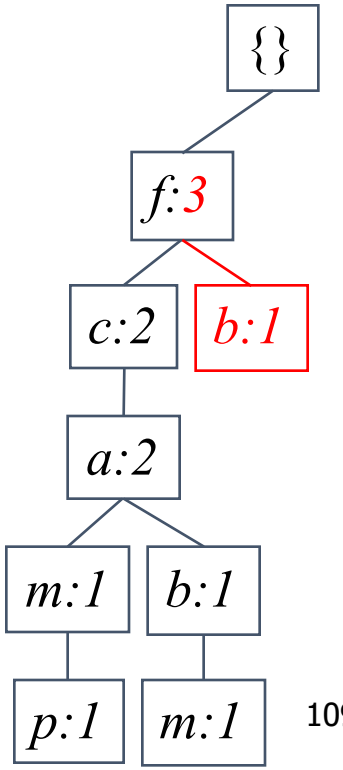
TID	Items bought	(ordered) frequent items
100	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, l, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o, w}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

min_support = 3

F-list = f-c-a-b-m-p

Constructing FP-tree

1. Create the root of the tree
2. Scan and process each tx in L order – constructing each branch of the transaction.
   - TID100 {f,c,a,m,p}
   - TID200 {f,c,a,b,m}
   - TID300 {f,b}



Constructing FP-tree

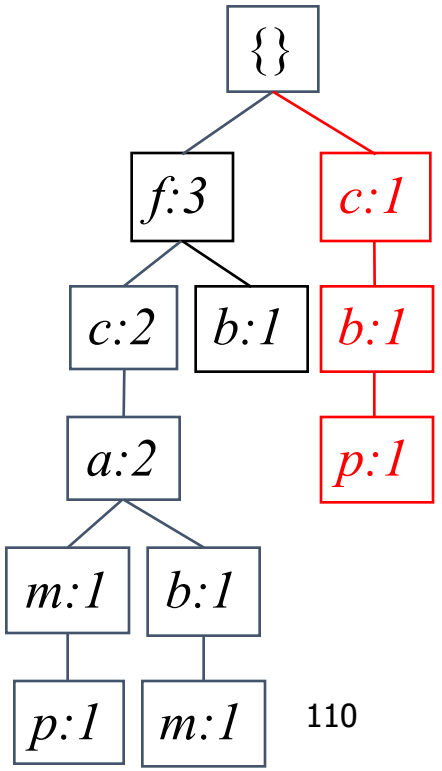
TID	Items bought	(ordered) frequent items
100	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, l, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o, w}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

min_support = 3

F-list = f-c-a-b-m-p

Constructing FP-tree

1. Create the root of the tree
2. Scan and process each tx in L order – constructing each branch of the transaction.
   - TID100 {f,c,a,m,p}
   - TID200 {f,c,a,b,m}
   - TID300 {f,b}
   - TID400 {c,b,p}



Constructing FP-tree

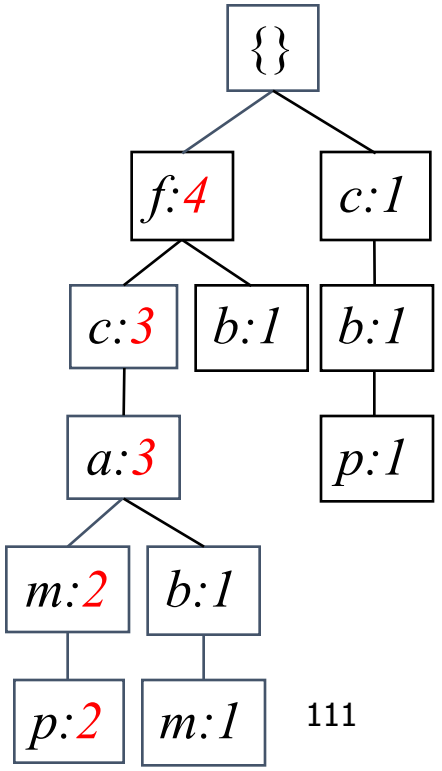
TID	Items bought	(ordered) frequent items
100	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, l, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o, w}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

min_support = 3

F-list = f-c-a-b-m-p

Constructing FP-tree

1. Create the root of the tree
2. Scan and process each tx in L order – constructing each branch of the transaction.
   - TID100 {f,c,a,m,p}
   - TID200 {f,c,a,b,m}
   - TID300 {f,b}
   - TID400 {c,b,p}
   - TID500 {f,c,a,m,p}



Constructing FP-tree

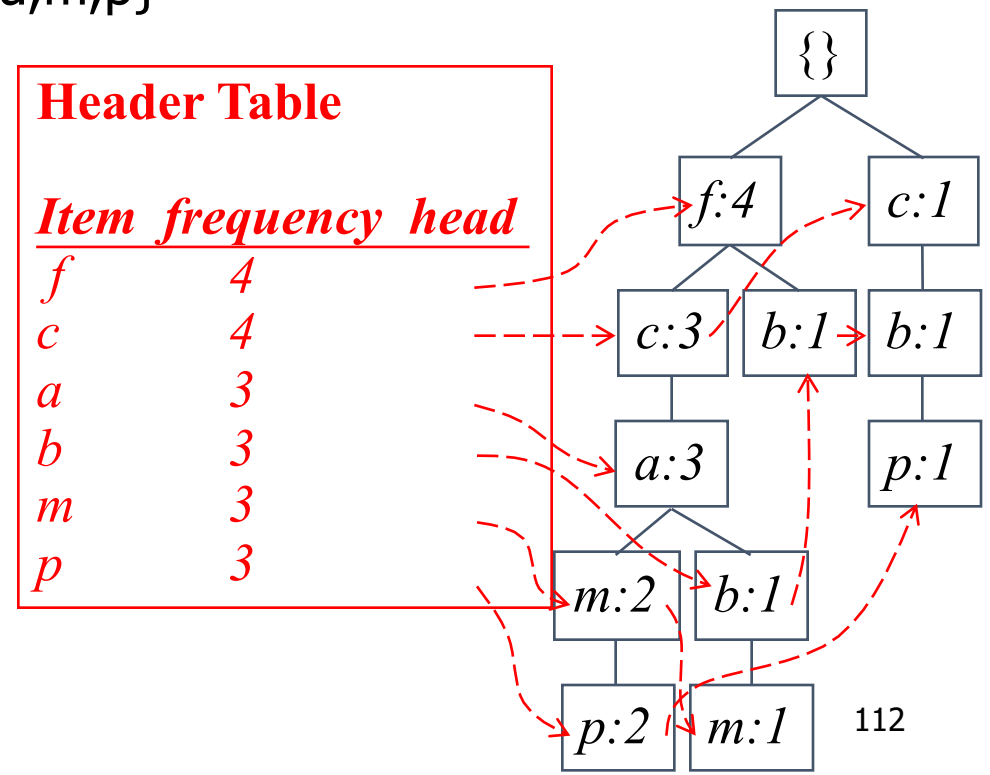
TID	Items bought	(ordered) frequent items
100	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, l, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o, w}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

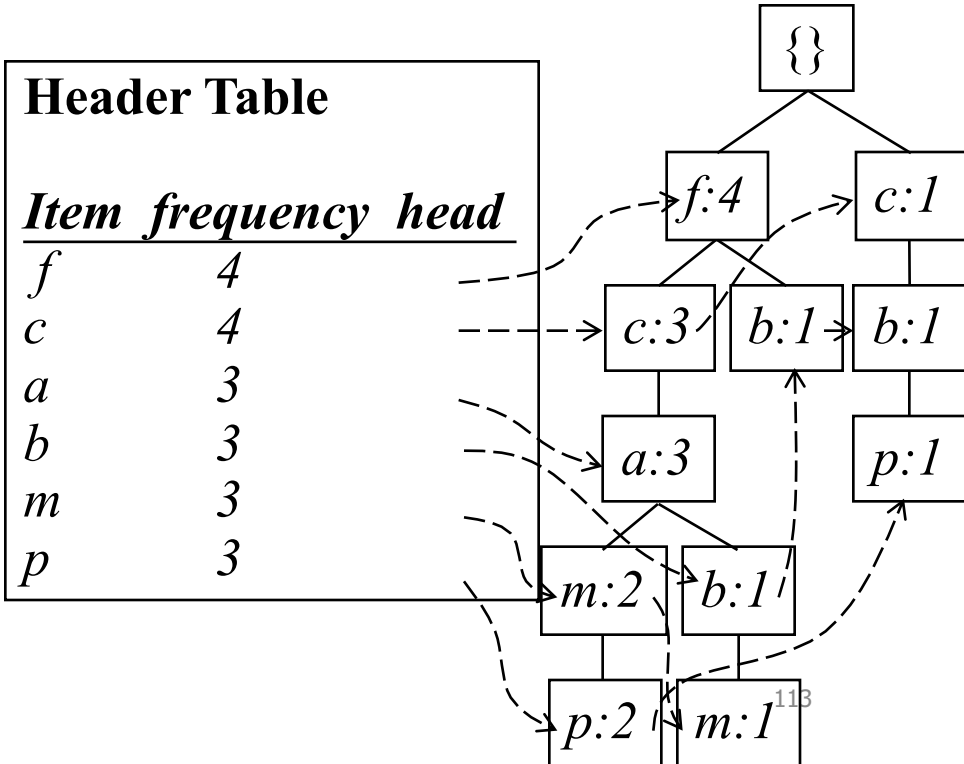
min_support = 3

F-list = f-c-a-b-m-p

Constructing FP-tree

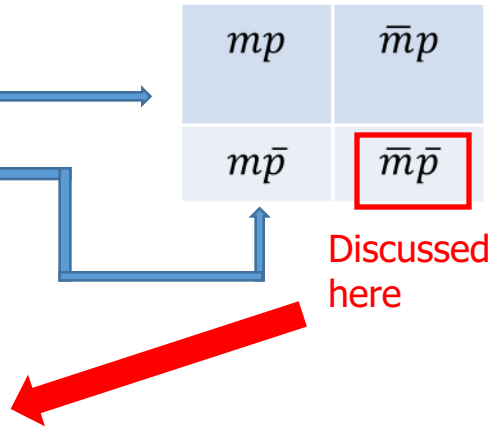
1. Create the root of the tree
2. Scan and process each tx in L order – constructing each branch of the transaction. E.g., TID100 {f,c,a,m,p}
3. Add Header Table and node-links to facilitate traversal




Header Table
Item frequency head
f 4
c 4
a 3
b 3
m 3
p 3
{}
f:4
c:1
c:3
b:1
b:1
a:3
p:1
m:2
b:1
p:2
m:1

Partition Patterns and Databases

- Frequent patterns can be partitioned into subsets according to f-list
  (recall that f-list is sorted with frequency)
  - F-list = f-c-a-b-m-p
  - Patterns containing p
  - Patterns having m but no p
  - …
  - Patterns having c but no a nor b, m, p
  - Pattern f
- Completeness and non-redundency

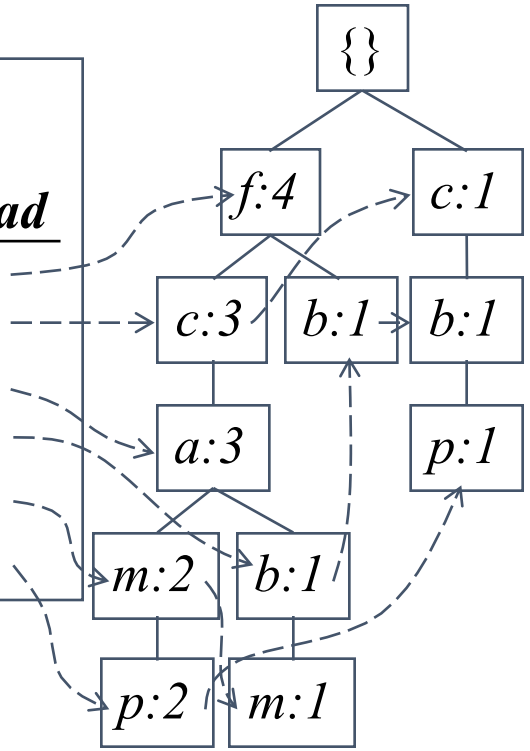


Find Patterns Having P From P-conditional Database

- Starting at the frequent item header table in the FP-tree
- Start with the last item
- Traverse the FP-tree by following the link of each frequent item *p*
- Accumulate all of *transformed prefix paths* of item *p* to form *p*'s conditional pattern base

Header Table

Item	frequency	head
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	



***Conditional* pattern bases**

item	*cond. pattern base*
c	*f:3*
a	*fc:3*
b	*fca:1, f:1, c:1*
m	*fca:2, fcab:1*
p	*fcam:2, cb:1*

***Conditional* pattern bases**

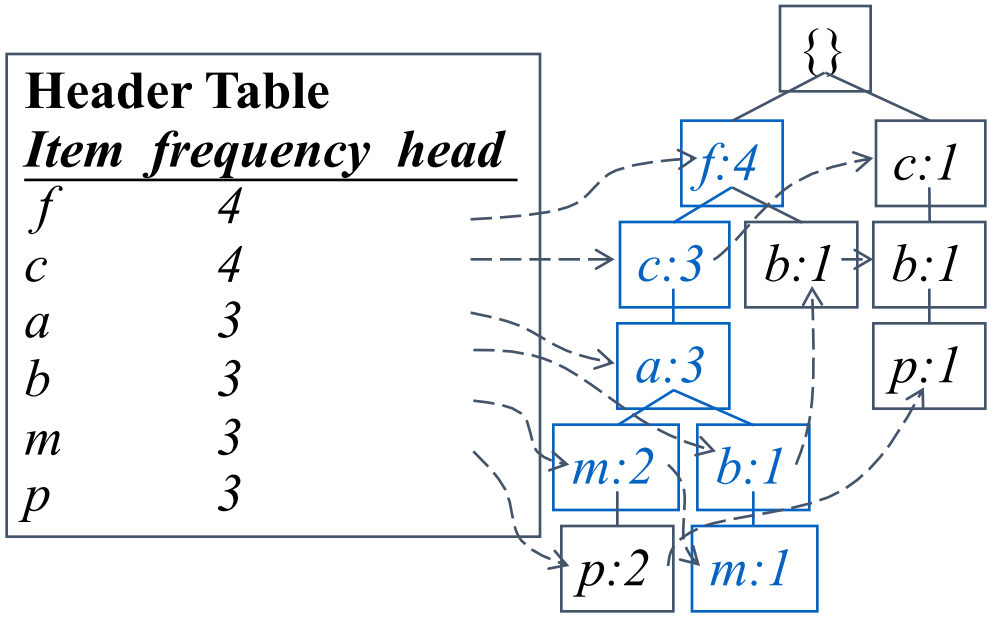
item	*cond. pattern base*
c	*f:3*
a	*fc:3*
b	*fca:1, f:1, c:1*
m	*fca:2, fcab:1*
p	*fcam:2, cb:1*

From Conditional Pattern-bases to Conditional FP-trees

- For each pattern-base
  - Accumulate the count for each item in the base
  - Construct the FP-tree for the frequent items of the pattern base
  - Example as follows (min_sup=2)

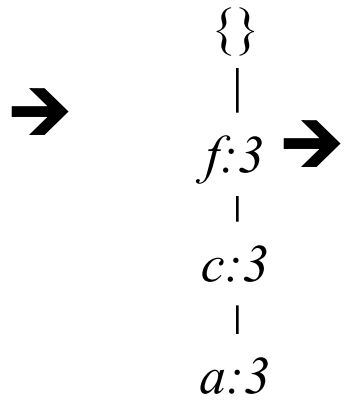
***Conditional* pattern bases**

item	*cond. pattern base*
c	*f:3*
a	*fc:3*
b	*fca:1, f:1, c:1*
m	*fca:2, fcab:1*
p	*fcam:2, cb:1*



***m-conditional* pattern base:**

fca:2, fcab:1



***m-conditional* FP-tree**

All frequent patterns relate to *m*

***m*,**

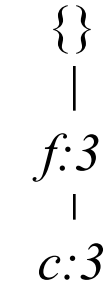
fm, cm, am,

fcm, fam, cam,

fcam

Recursion: Mining Each Conditional FP-tree

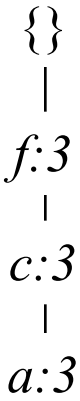
Cond. pattern base of "am": (fc:3)



***am-conditional* FP-tree**

Cond. pattern base of "cm": (f:3)

***cm-conditional* FP-tree**



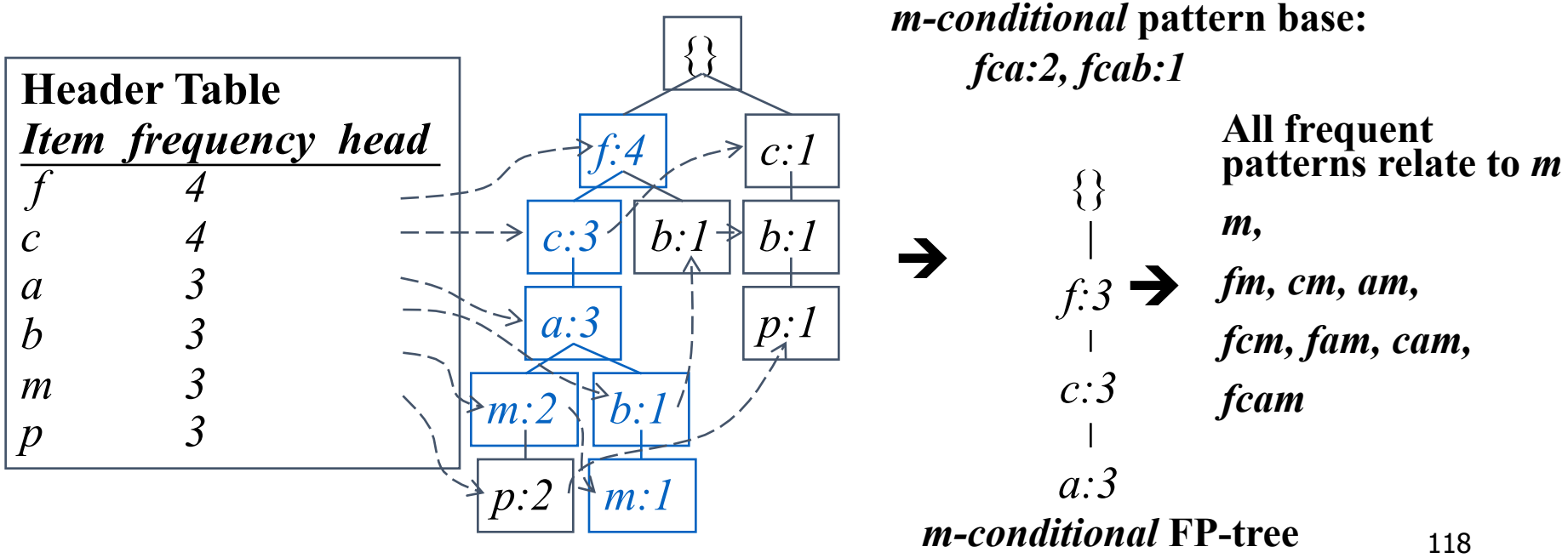
***m-conditional* FP-tree**

Cond. pattern base of "cam": (f:3)

***cam-conditional* FP-tree**

From Conditional Pattern-bases to Conditional FP-trees

- Suppose ***min_sup*** = 2 (count)
- m-conditional pattern base: fca:2, fcab:1
- Conditional FP-Tree for m: fca:3
- Frequent patterns generated: *m, fm, cm , am, fcm, fam, cam, fcam*



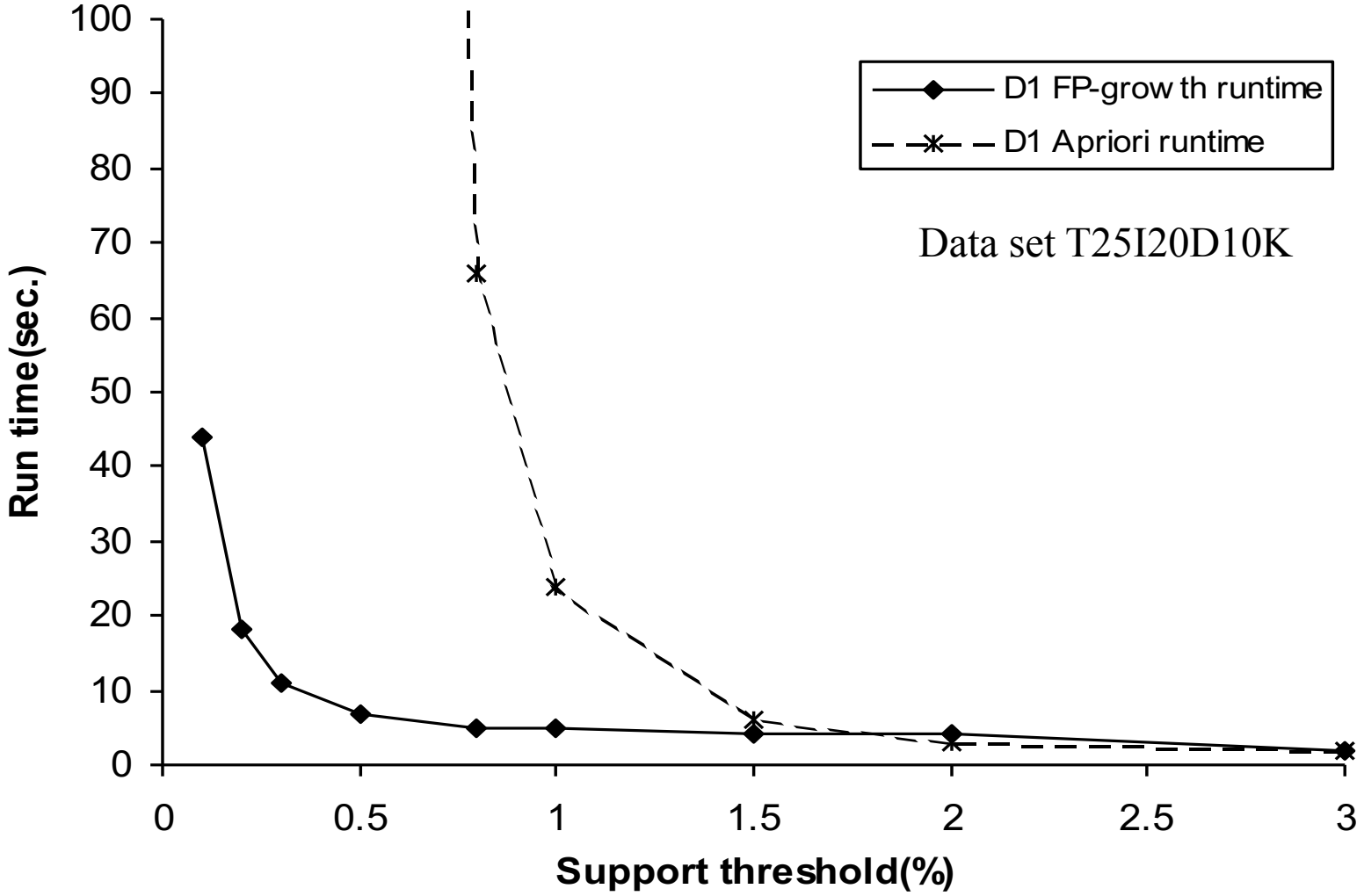
Benefits of the FP-tree Structure

- Completeness
  - Preserve complete information for frequent pattern mining
  - Never break a long pattern of any transaction
- Compactness
  - Reduce irrelevant info—infrequent items are gone
  - Items in frequency descending order: the more frequently occurring, the more likely to be shared
  - Never be larger than the original database (not including node-links and the *count* field)

The Frequent Pattern Growth Mining Method

- Idea: Frequent pattern growth
  - Recursively grow frequent patterns by pattern and database partition
- Method
  - For each frequent item, construct its conditional pattern-base, and then its conditional FP-tree
  - Repeat the process on each newly created conditional FP-tree
  - Until the resulting FP-tree is empty, or it contains only one path—single path will generate all the combinations of its sub-paths, each of which is a frequent pattern

Performance of FPGrowth in Large Datasets



FP-Growth vs. Apriori

Advantages of the Pattern Growth Approach

- Divide-and-conquer:
  - Decompose both the mining task and DB according to the frequent patterns obtained so far
  - Lead to focused search of smaller databases
- Other factors
  - No candidate generation, no candidate test
  - Compressed database: FP-tree structure
  - No repeated scan of entire database
  - Basic ops: counting local freq items and building sub FP-tree, no pattern search and matching
- A good open-source implementation and refinement of FPGrowth
  - FPGrowth+ (Grahne and J. Zhu, FIMI'03)

ECLAT: Frequent Pattern Mining with Vertical Data Format

ECLAT: Mining by Exploring Vertical Data Format

- Vertical format: $t(AB) = \{T_{11}, T_{25}, \ldots\}$
  - tid-list: list of trans.-ids containing an itemset
- Deriving frequent patterns based on vertical intersections
  - t(X) = t(Y): X and Y always happen together
  - $t(X) \subset t(Y)$: transaction having X always has Y

TID	*List of item_IDs*
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

itemset	*TID_set*
I1	{T100, T400, T500, T700, T800, T900}
I2	{T100, T200, T300, T400, T600, T800, T900}
I3	{T300, T500, T600, T700, T800, T900}
I4	{T200, T400}
I5	{T100, T800}

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Original database (horizontal)

Assume ***min_sup = 2***

itemset	TID_set
I1	{T100, T400, T500, T700, T800, T900}
I2	{T100, T200, T300, T400, T600, T800, T900}
I3	{T300, T500, T600, T700, T800, T900}
I4	{T200, T400}
I5	{T100, T800}

Vertical data format

Intersect each frequent 1-itemsets, leading to 8 non-empty 2-itemsets

itemset	TID_set
{I1, I2}	{T100, T400, T800, T900}
{I1, I3}	{T500, T700, T800, T900}
~~{I1, I4}~~	~~{T400}~~
{I1, I5}	{T100, T800}
{I2, I3}	{T300, T600, T800, T900}
{I2, I4}	{T200, T400}
{I2, I5}	{T100, T800}
~~{I3, I5}~~	~~{T800}~~

Join and intersect 2-item sets, similar to Apriori

itemset	TID_set
{I1, I2, I3}	{T800, T900}
{I1, I2, I5}	{T100, T800}

Strong Points of Mining with Vertical Data Format

- Apriori property can be leveraged
- No need to scan the DB to find support of (k+1)-itemsets
- diffset can be used to reduce the storage overhead of vertical data format
- Using diffset to accelerate mining
  - Only keep track of differences of tids
  - $t(X) = \{T_1, T_2, T_3\}$, $t(XY) = \{T_1, T_3\}$
  - Diffset (XY, X) = $\{T_2\}$

Outline

- **Basic Concepts**
- Frequent Itemset Mining Methods
- **Which Patterns Are Interesting?—Pattern Evaluation Methods**
- Summary

Strong rules are not necessarily interesting

- Game: transactions containing computer games
  Video: transactions containing videos
- Of 10,000 transactions:
  6,000 include games, 7,500 include videos, 4,000 include both game and video
- A **strong** association rule is thus derived (min_sup=30%, min_conf=60%):

$$buys(X, \text{“}computer\ games\text{”}) \Rightarrow buys(X, \text{“}videos\text{”})$$

$$[support = 40\%, confidence = 66\%].$$

- This rule is **MISLEADING**, because probability of videos is 75% > 66% (buying game and video together)

- In fact, games and videos are negatively associated
  Buying one actually decreases the likelihood of buying the other

Interestingness Measure: Correlations (Lift)

- *games* $\Rightarrow$ ***not*** *video* [20%, 33.3%] is more accurate, although with lower support and confidence
- Measure of dependent/correlated events: lift

  If the occurrence of A is independent of B => $P(A \cup B) = P(A)P(B)$

$$\text{lift}(A, B) = \frac{P(A \cup B)}{P(A)P(B)}$$

$\text{lift}(A, B) < 1$ => A, B are negative correlated
$\text{lift}(A, B) > 1$ => A, B are positively correlated
$\text{lift}(A, B) = 1$ => A, B are independent

$$lift(G, V) = \frac{4000/10000}{6000/10000 * 7500/10000} = 0.89$$

$$lift(G, \neg V) = \frac{2000/10000}{6000/10000 * 2500/10000} = 1.33$$

	Game	Not game	Sum (row)
Video	4000	3500	7500
Not video	2000	500	2500
Sum(col.)	6000	4000	10000

Interesting Measure: Chi-square (χ^2)

	game	$\overline{game}$	Σ_{row}
video	4000 (4500)	3500 (3000)	7500
$\overline{video}$	2000 (1500)	500 (1000)	2500
Σ_{col}	6000	4000	10,000

$$\chi^2 = \Sigma \frac{(observed - expected)^2}{expected} = \frac{(4000-4500)^2}{4500} + \frac{(3500-3000)^2}{3000}$$
$$+ \frac{(2000-1500)^2}{1500} + \frac{(500-1000)^2}{1000} = 555.6.$$

χ^2>1, and the observed value of *(game, video)=4000*, which is less than the expected value *4,500*

$\Rightarrow$ Buying game and buying video are negatively correlated

Consistent with the conclusion derived from the analysis of the lift measure

Pattern Evaluation Measures

- **all_confidence:** $all\_conf(A,B) = \frac{sup(A \cup B)}{max\{sup(A), sup(B)\}} = min\{P(A|B), P(B|A)\}$
  - Minimum confidence of "A=>B" and "B=>A"
- **max_confidence:** $max\_conf(A, B) = max\{P(A \mid B), P(B \mid A)\}$
  - Maximum confidence of "A=>B" and "B=>A"
- **Kulczynski:** $Kulc(A, B) = \frac{1}{2}(P(A|B) + P(B|A))$
  - Average confidence of "A=>B" and "B=>A"
- **Cosine:** $cosine(A, B) = \frac{P(A \cup B)}{\sqrt{P(A) \times P(B)}} = \frac{sup(A \cup B)}{\sqrt{sup(A) \times sup(B)}} = \sqrt{P(A|B) \times P(B|A)}.$

Comparison of Interestingness Measures

m and **c**:

- positively correlated in D_1, D_2,
  i.e., $mc(10{,}000) > \bar{m}c(1{,}000) = m\bar{c}(1{,}000)$
- negatively correlated in D_3,
- neutral in D_4

	milk	$\overline{milk}$	Σ_{row}
coffee	mc	$\bar{m}c$	c
$\overline{coffee}$	$m\bar{c}$	$\bar{m}\bar{c}$	$\bar{c}$
Σ_{col}	m	$\bar{m}$	Σ

Data Set	*mc*	$\bar{m}c$	$m\bar{c}$	$\bar{m}\bar{c}$	χ^2	*lift*	*all_conf.*	*max_conf.*	*Kulc.*	*cosine*
D_1	10,000	1000	1000	100,000	90557	9.26	0.91	0.91	0.91	0.91
D_2	10,000	1000	1000	100	0	1	0.91	0.91	0.91	0.91
D_3	100	1000	1000	100,000	670	8.44	0.09	0.09	0.09	0.09
D_4	1000	1000	1000	100,000	24740	25.75	0.5	0.5	0.5	0.5
D_5	1000	100	10,000	100,000	8173	9.18	0.09	0.91	0.5	0.29
D_6	1000	10	100,000	100,000	965	1.97	0.01	0.99	0.5	0.10

Comparison of Interestingness Measures

m and **c**:

- positively correlated in D_1, D_2,
  i.e., $mc(10{,}000) > \bar{m}c(1{,}000) = m\bar{c}(1{,}000)$
- negatively correlated in D_3,
- neutral in D_4

All the four new measures show m and c are strongly positively associated

Data Set	mc	$\bar{m}c$	$m\bar{c}$	$\bar{m}\bar{c}$	χ^2	lift	all_conf.	max_conf.	Kulc.	cosine
D_1	10,000	1000	1000	100,000	90557	9.26	0.91	0.91	0.91	0.91
D_2	10,000	1000	1000	100	0	1	0.91	0.91	0.91	0.91
D_3	100	1000	1000	100,000	670	8.44	0.09	0.09	0.09	0.09
D_4	1000	1000	1000	100,000	24740	25.75	0.5	0.5	0.5	0.5
D_5	1000	100	10,000	100,000	8173	9.18	0.09	0.91	0.5	0.29
D_6	1000	10	100,000	100,000	965	1.97	0.01	0.99	0.5	0.10

Comparison of Interestingness Measures

m and **c**:

- positively correlated in D_1, D_2,
- negatively correlated in D_3,
- neutral in D_4

χ^2 and lift generate dramatically different measures
Due to their sensitivity to $\overline{mc}$

In real-world scenarios, $\overline{mc}$ is usually huge and unstable

Data Set	mc	$\overline{m}c$	$m\overline{c}$	$\overline{mc}$	χ^2	lift	all_conf.	max_conf.	Kulc.	cosine
D_1	10,000	1000	1000	100,000	90557	9.26	0.91	0.91	0.91	0.91
D_2	10,000	1000	1000	100	0	1	0.91	0.91	0.91	0.91
D_3	100	1000	1000	100,000	670	8.44	0.09	0.09	0.09	0.09
D_4	1000	1000	1000	100,000	24740	25.75	0.5	0.5	0.5	0.5
D_5	1000	100	10,000	100,000	8173	9.18	0.09	0.91	0.5	0.29
D_6	1000	10	100,000	100,000	965	1.97	0.01	0.99	0.5	0.10

Comparison of Interestingness Measures

m and **c**:

- positively correlated in D_1, D_2,
- negatively correlated in D_3,
- neutral in D_4

All the four new measures show m and c are strongly nagtively associated

Data Set	mc	$\overline{m}c$	$m\overline{c}$	$\overline{mc}$	χ^2	lift	all_conf.	ma onf.	Kulc.	cosine
D_1	10,000	1000	1000	100,000	90557	9.26	0.91	0.9	0.91	0.91
D_2	10,000	1000	1000	100	0	1	0.91	0.91	0.91	0.91
D_3	100	1000	1000	100,000	670	8.44	0.09	0.09	0.09	0.09
D_4	1000	1000	1000	100,000	24740	25.75	0.5	0.5	0.5	0.5
D_5	1000	100	10,000	100,000	8173	9.18	0.09	0.91	0.5	0.29
D_6	1000	10	100,000	100,000	965	1.97	0.01	0.99	0.5	0.10

Comparison of Interestingness Measures

m and **c**:

- positively correlated in D_1, D_2,
- negatively correlated in D_3,
- neutral in D_4

χ^2 and lift: values are between D_1 and D_2

Data Set	mc	$\overline{m}c$	$m\overline{c}$	$\overline{m}\overline{c}$	χ^2	lift	all_conf.	max_conf.	Kulc.	cosine
D_1	10,000	1000	1000	100,000	90557	.26	0.91	0.91	0.91	0.91
D_2	10,000	1000	1000	100	0	1	0.91	0.91	0.91	0.91
D_3	100	1000	1000	100,000	670	8.44	0.09	0.09	0.09	0.09
D_4	1000	1000	1000	100,000	24740	25.75	0.5	0.5	0.5	0.5
D_5	1000	100	10,000	100,000	8173	9.18	0.09	0.91	0.5	0.29
D_6	1000	10	100,000	100,000	965	1.97	0.01	0.99	0.5	0.10

Comparison of Interestingness Measures

m and **c**:

- positively correlated in D_1, D_2,
- negatively correlated in D_3,
- neutral in D_4

χ^2 and lift: show that D4 is positive associated between m and c

Data Set	mc	$\overline{m}c$	$m\overline{c}$	$\overline{m}\overline{c}$	χ^2	lift	all_conf.	max_conf.	Kulc.	cosine
D_1	10,000	1000	1000	100,000	90557	.26	0.91	0.91	0.91	0.91
D_2	10,000	1000	1000	100	0	1	0.91	0.91	0.91	0.91
D_3	100	1000	1000	100,000	670	8.44	0.09	0.09	0.09	0.09
D_4	1000	1000	1000	100,000	24740	25.75	0.5	0.5	0.5	0.5
D_5	1000	100	10,000	100,000	8173	9.18	0.09	0.91	0.5	0.29
D_6	1000	10	100,000	100,000	965	1.97	0.01	0.99	0.5	0.10

Comparison of Interestingness Measures

m and **c**:

- positively correlated in D_1, D_2,
- negatively correlated in D_3,
- neutral in D_4

It is neutral as indicated by the four measures.
A customer buys coffee (or milk), the probability of buying milk (of coffee) is exactly 50%

Data Set	mc	$\overline{m}c$	$m\overline{c}$	$\overline{m}\overline{c}$	χ^2	lift	all_conf.	max_conf.	Kulc.	cosine
D_1	10,000	1000	1000	100,000	90557	9.26	0.91	0.91	0.91	0.91
D_2	10,000	1000	1000	100	0	1	0.91	0.91	0.91	0.91
D_3	100	1000	1000	100,000	670	8.44	0.09	0.09	0.09	0.09
D_4	1000	1000	1000	100,000	24740	25.75	0.5	0.5	0.5	0.5
D_5	1000	100	10,000	100,000	8173	9.18	0.09	0.91	0.5	0.29
D_6	1000	10	100,000	100,000	965	1.97	0.01	0.99	0.5	0.10

Why are lift and χ^2 so poor?

- Null-transactions
  - Transaction that does not contain any of the itemsets being examined
  - E.g., $\overline{mc}$ is the number of null-transactions
- lift and χ^2 are strongly influenced by $\overline{mc}$

- The other four measures are good indicators
  - Their definitions remove the influence of $\overline{mc}$

Which Null-Invariant Measure Is Better?

- IR (Imbalance Ratio): measure the imbalance of two itemsets A and B in rule implications

$$IR(A,B) = \frac{|sup(A) - sup(B)|}{sup(A) + sup(B) - sup(A \cup B)}$$

- **c** occurs strongly suggests **m** occurs also
- **m** occurs strongly suggests **c** unlikely occur

Diverse outcomes!!

Data Set	mc	$\overline{m}c$	$m\overline{c}$	$\overline{mc}$	χ^2	lift	all_conf.	m_conf.	Kulc.	cosine
D_1	10,000	1000	1000	100,000	90557	9.26	0.91	0.	0.91	0.91
D_2	10,000	1000	1000	100	0	1	0.91	0.	0.91	0.91
D_3	100	1000	1000	100,000	670	8.44	0.09	0	0.09	0.09
D_4	1000	1000	1000	100,000	24740	25.75	0.5	0.5	0.5	0.5
D_5	1000	100	10,000	100,000	8173	9.18	0.09	0.91	0.5	0.29
D_6	1000	10	100,000	100,000	965	1.97	0.01	0.99	0.5	0.10

Which Null-Invariant Measure Is Better?

- IR (Imbalance Ratio): measure the imbalance of two itemsets A and B in rule implications

$$IR(A,B) = \frac{|sup(A) - sup(B)|}{sup(A) + sup(B) - sup(A \cup B)}$$

Kulczynski and Imbalance Ratio (IR) together present a clear picture for all the three datasets D_4 through D_6

D_4 is balanced & neutral (IR(m,c)=0 => perfect balanced)

D_5 is imbalanced & neutral (IR(m,c)=0.89 => imbalanced)

D_6 is very imbalanced & neutral (IR(m,c)=0.99 => very skewed)

Data Set	mc	$\overline{m}c$	$m\overline{c}$	$\overline{m}\overline{c}$	χ^2	*lift*	*all_conf.*	*max_conf.*	*Kulc.*	*cosine*
D_1	10,000	1000	1000	100,000	90557	9.26	0.91	0.91	0.91	0.91
D_2	10,000	1000	1000	100	0	1	0.91	0.91	0.91	0.91
D_3	100	1000	1000	100,000	670	8.44	0.09	0.09	0.09	0.09
D_4	1000	1000	1000	100,000	24740	25.75	0.5	0.5	0.5	0.5
D_5	1000	100	10,000	100,000	8173	9.18	0.09	0.91	0.5	0.29
D_6	1000	10	100,000	100,000	965	1.97	0.01	0.99	0.5	0.10

Outline

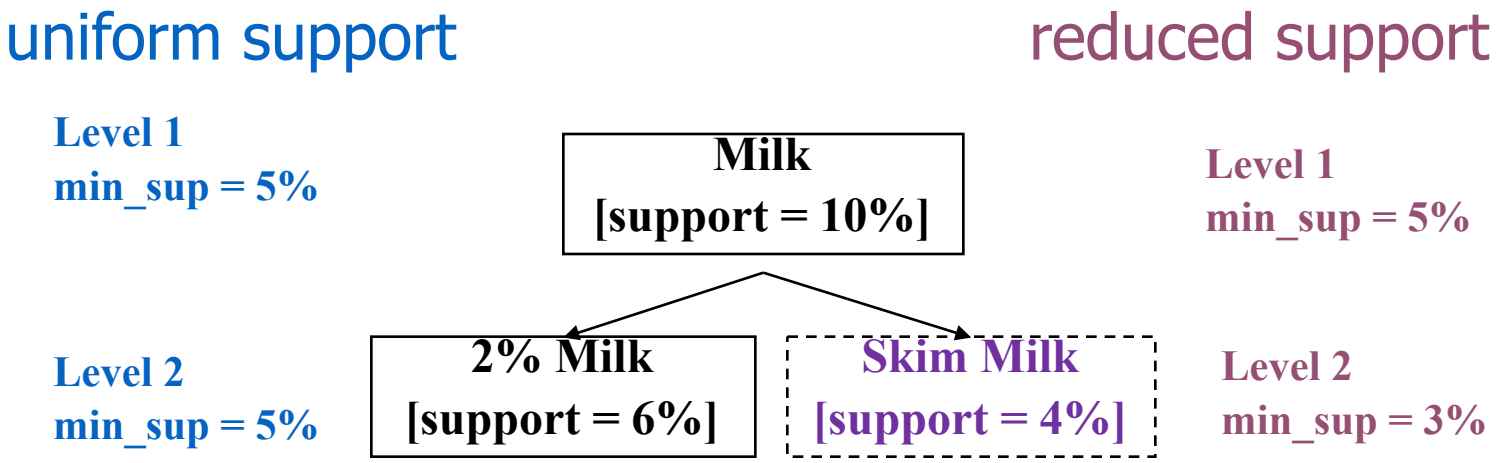
- Basic Concepts
- Frequent Itemset Mining Methods
- Which Patterns Are Interesting?—Pattern Evaluation Methods
- **Summary**

Summary

- Basic concepts: association rules, support-confident framework, closed and max-patterns
- Scalable frequent pattern mining methods
  - Apriori (Candidate generation & test)
  - Projection-based (FPgrowth, CLOSET+, ...)
  - Vertical format approach (ECLAT, CHARM, ...)
- Which patterns are interesting?
  - Pattern evaluation methods

Mining Multiple-Level Association Rules

- Items often form hierarchies
- Flexible support settings
  - Items at the lower level are expected to have lower support
- Exploration of *shared* multi-level mining (Agrawal & Srikant@VLB'95, Han & Fu@VLDB'95)



Multi-level Association: Flexible Support and Redundancy filtering

- Flexible min-support thresholds: Some items are more valuable but less frequent
  - Use non-uniform, group-based min-support
  - E.g., {diamond, watch, camera}: 0.05%; {bread, milk}: 5%; …
- Redundancy Filtering: Some rules may be redundant due to “ancestor” relationships between items
  - milk $\Rightarrow$ wheat bread [support = 8%, confidence = 70%]
  - 2% milk $\Rightarrow$ wheat bread [support = 2%, confidence = 72%]

  The first rule is an ancestor of the second rule
- A rule is *redundant* if its support is close to the “expected” value, based on the rule’s ancestor

Mining Multi-Dimensional Association

- Single-dimensional rules:

  $buys(X, \text{“milk”}) \Rightarrow buys(X, \text{“bread”})$

- Multi-dimensional rules: ≥ 2 dimensions or predicates
  - Inter-dimension assoc. rules (*no repeated predicates*)

    $age(X,\text{”19-25”}) \wedge occupation(X,\text{“student”}) \Rightarrow buys(X, \text{“coke”})$
  - hybrid-dimension assoc. rules (*repeated predicates*)

    $age(X,\text{”19-25”}) \wedge buys(X, \text{“popcorn”}) \Rightarrow buys(X, \text{“coke”})$

- Categorical Attributes: finite number of possible values, no ordering among values—data cube approach
- Quantitative Attributes: Numeric, implicit ordering among values—discretization, clustering, and gradient approaches

Mining Quantitative Associations

Techniques can be categorized by how numerical attributes, such as age or salary are treated

1. Static discretization based on predefined concept hierarchies (data cube methods)
2. Dynamic discretization based on data distribution (quantitative rules, e.g., Agrawal & Srikant@SIGMOD96)
3. Clustering: Distance-based association (e.g., Yang & Miller@SIGMOD97)
   - One dimensional clustering then association
4. Deviation: (such as Aumann and Lindell@KDD99)

   Sex = female => Wage: mean=$7/hr (overall mean = $9)

Negative and Rare Patterns

- Rare patterns: Very low support but interesting
  - E.g., buying Rolex watches
  - Mining: Setting individual-based or special group-based support threshold for valuable items
- Negative patterns
  - Since it is unlikely that one buys Ford Expedition (an SUV car) and Toyota Prius (a hybrid car) together, Ford Expedition and Toyota Prius are likely negatively correlated patterns
- Negatively correlated patterns that are infrequent tend to be more interesting than those that are frequent

Defining Negative Correlated Patterns (I)

- Definition 1 (support-based)
  - If itemsets X and Y are both frequent but rarely occur together, i.e.,

    $sup(X \cup Y) < sup(X) * sup(Y)$
  - Then X and Y are negatively correlated
- Problem: A store sold two kinds of packages A and B (100 for each), only one transaction containing both A and B.
  - When there are in total 200 transactions, we have

    $s(A \cup B) = 0.005$, $s(A) * s(B) = 0.25$, $s(A \cup B) < s(A) * s(B)$
  - When there are 10^5 transactions, we have

    $s(A \cup B) = 1/10^5$, $s(A) * s(B) = 1/10^3 * 1/10^3$, $s(A \cup B) > s(A) * s(B)$
  - Where is the problem? —Null transactions, i.e., the support-based definition is not null-invariant!

Defining Negative Correlated Patterns (II)

- Definition 2 (negative itemset-based)
  - X is a *negative itemset* if (1) X = $\bar{A}$ U B, where B is a set of positive items, and $\bar{A}$ is a set of negative items, $|\bar{A}| \geq 1$, and (2) $s(X) \geq \mu$
  - Itemsets X is negatively correlated, if

$$s(X) < \prod_{i=1}^{k} s(x_i), where\ x_i \in X,\ and\ s(x_i)\ is\ the\ support\ of\ x_i$$

- This definition suffers a similar null-invariant problem
- Definition 3 (Kulzynski measure-based) If itemsets X and Y are frequent, but $(P(X|Y) + P(Y|X))/2 < \epsilon$, where ϵ is a negative pattern threshold, then X and Y are negatively correlated.
- Ex. For the same package problem, when no matter there are 200 or 10^5 transactions, if $\epsilon = 0.01$, we have

$$(P(A|B) + P(B|A))/2 = (0.01 + 0.01)/2 < \epsilon$$

Ref: Basic Concepts of Frequent Pattern Mining

- (Association Rules) R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. SIGMOD'93
- (Max-pattern) R. J. Bayardo. Efficiently mining long patterns from databases. SIGMOD'98
- (Closed-pattern) N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. ICDT'99
- (Sequential pattern) R. Agrawal and R. Srikant. Mining sequential patterns. ICDE'95

Ref: Apriori and Its Improvements

- R. Agrawal and R. Srikant. Fast algorithms for mining association rules. VLDB'94
- H. Mannila, H. Toivonen, and A. I. Verkamo. Efficient algorithms for discovering association rules. KDD'94
- A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association rules in large databases. VLDB'95
- J. S. Park, M. S. Chen, and P. S. Yu. An effective hash-based algorithm for mining association rules. SIGMOD'95
- H. Toivonen. Sampling large databases for association rules. VLDB'96
- S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket analysis. SIGMOD'97
- S. Sarawagi, S. Thomas, and R. Agrawal. Integrating association rule mining with relational database systems: Alternatives and implications. SIGMOD'98

Ref: Depth-First, Projection-Based FP Mining

- R. Agarwal, C. Aggarwal, and V. V. V. Prasad. A tree projection algorithm for generation of frequent itemsets. J. Parallel and Distributed Computing, 2002.
- G. Grahne and J. Zhu, Efficiently Using Prefix-Trees in Mining Frequent Itemsets, Proc. FIMI'03
- B. Goethals and M. Zaki. An introduction to workshop on frequent itemset mining implementations. *Proc. ICDM'03 Int. Workshop on Frequent Itemset Mining Implementations (FIMI'03),* Melbourne, FL, Nov. 2003
- J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. SIGMOD' 00
- J. Liu, Y. Pan, K. Wang, and J. Han. Mining Frequent Item Sets by Opportunistic Projection. KDD'02
- J. Han, J. Wang, Y. Lu, and P. Tzvetkov. Mining Top-K Frequent Closed Patterns without Minimum Support. ICDM'02
- J. Wang, J. Han, and J. Pei. CLOSET+: Searching for the Best Strategies for Mining Frequent Closed Itemsets. KDD'03

Ref: Vertical Format and Row Enumeration Methods

- M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. Parallel algorithm for discovery of association rules. DAMI:97.
- M. J. Zaki and C. J. Hsiao. CHARM: An Efficient Algorithm for Closed Itemset Mining, SDM'02.
- C. Bucila, J. Gehrke, D. Kifer, and W. White. DualMiner: A Dual-Pruning Algorithm for Itemsets with Constraints. KDD’02.
- F. Pan, G. Cong, A. K. H. Tung, J. Yang, and M. Zaki , CARPENTER: Finding Closed Patterns in Long Biological Datasets. KDD'03.
- H. Liu, J. Han, D. Xin, and Z. Shao, Mining Interesting Patterns from Very High Dimensional Data: A Top-Down Row Enumeration Approach, SDM'06.

Ref: Mining Correlations and Interesting Rules

- S. Brin, R. Motwani, and C. Silverstein. Beyond market basket: Generalizing association rules to correlations. SIGMOD'97.
- M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. I. Verkamo. Finding interesting rules from large sets of discovered association rules. CIKM'94.
- R. J. Hilderman and H. J. Hamilton. *Knowledge Discovery and Measures of Interest*. Kluwer Academic, 2001.
- C. Silverstein, S. Brin, R. Motwani, and J. Ullman. Scalable techniques for mining causal structures. VLDB'98.
- P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the Right Interestingness Measure for Association Patterns. KDD'02.
- E. Omiecinski. Alternative Interest Measures for Mining Associations. TKDE'03.
- T. Wu, Y. Chen, and J. Han, "Re-Examination of Interestingness Measures in Pattern Mining: A Unified Framework", Data Mining and Knowledge Discovery, 21(3):371-397, 2010