Chapter 2

Data

Data

- What are the various types of data?
- How is data represented?
- How do we improve data quality?
- How do we preprocess data for analysis?
- How do we measure similarity between data objects?

Typical Structured Data

With a well-defined set of attributes

- Collection of data objects (tuples/records) and their attributes
- An *attribute* is a property or characteristic of an *object*
  - Examples:
    - height/ weight of a person,
    - air pressure/ wind speed/ temperature/ lat-long at/of a location,
  - *Attribute* is also known as *variable*, *field*, *characteristic*, or *feature*
- A collection of attributes describe an object
  - *Object* is also known as *record*, *entity, tuple*.

Attributes

Taxpayer ID	Refund	Marital Status	Taxable Income	Tax fraud
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Schema
(the set of attributes)

Objects
(e.g., taxpayers)

Attribute Domains

- Each attribute has a domain, which is a set of possible values for that attribute (e.g., integer for “age”; string for “name”).
- Depending on the domain, attributes are of different types.

Types of Attributes

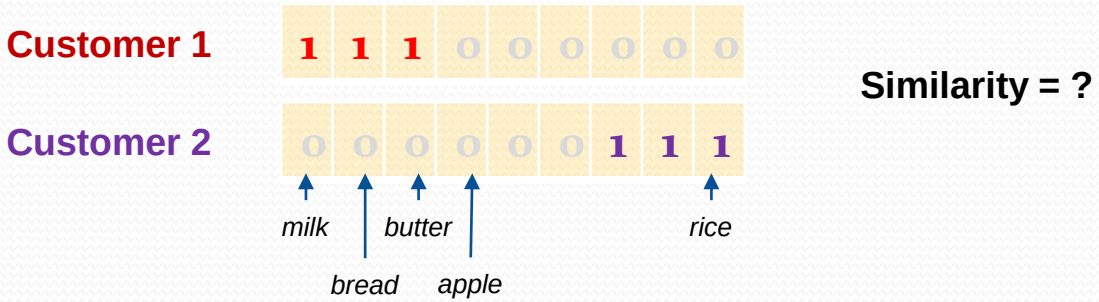
- *Binary*
  - Values are either *0* or *1*, used to denote the presence or absence of a feature
  - Examples: words in a document

Cat	Dog	Fish	Apple	…	Tiger
1	1	0	0	…	0

- *Nominal*
  - Values are (discrete) *names*
  - Examples: a person's name, ID numbers, eye color, zip codes
- *Ordinal*
  - Values are discrete names that are *ordered*
  - Examples: rankings (e.g., taste of potato chips on a scale from 1-10); grades {A, B, C, D, F}; height in {tall, medium, short}
- *Numerical*
  - Values are numbers representing quantities on which *arithmetic operators* (e.g., sum, average) can be applied.
  - Examples: height in meter, speed in km/h, sale amount in $$$.

Asymmetric attributes

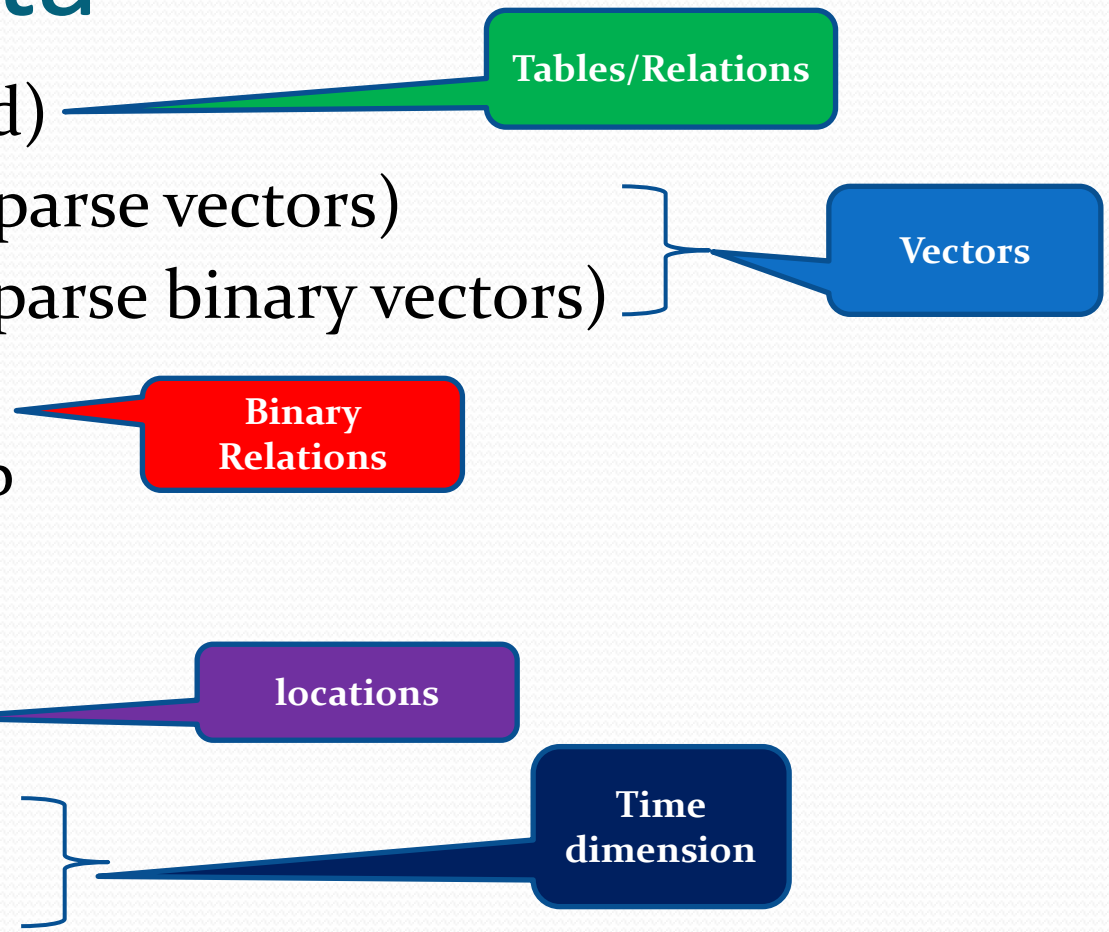
- Only presence (non-zero) values are important
- Example: assume that the set of items purchased in a supermarket transaction is modeled by a binary vector:
  - 1 for each item purchased
  - 0 otherwise
- When two transactions (sets of items purchased) are compared, only non-zero values are important.



Similarity = ?

Types of data

- Record (structured)
- Document data (sparse vectors)
- Set-valued data (sparse binary vectors)
- Graph
  - World Wide Web
  - Social networks
- Spatial-temporal
  - Spatial data
  - Time series data
  - Sequence data



Document data

- Bag-of-Words (BoW) Model
  - Each document is considered a collection (a set) of terms.
    - "*A cat sits on a mat*"
    - → {"*a*":2; "*cat*":1; "*sits*":1; "*on*":1; "*mat*":1}
    - → {"*cat*":1; "*sit*":1; "*mat*":1} (with stop-word removal and stemming/lemmatization)
  - Each *document* is modeled as a *vector*; Each *component* of the vector corresponds to a *term* in a given dictionary.
  - The term-frequency model (TF): the value of each component is the number of times the corresponding term occurs in the document.

Document data

A term vector

	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

Document Data

- Term vectors could be used to measure the similarity of documents.
- Problem with the TF-model: commonly occurring words (e.g., stop words) dominate the vector.
- TF-IDF model:

Term frequency of *t*

Inverse document frequency of *t*

$$TF(t) \times IDF(t),$$

where,

$$IDF(t) = \log \frac{|D|}{|D_t|}$$

Number of documents

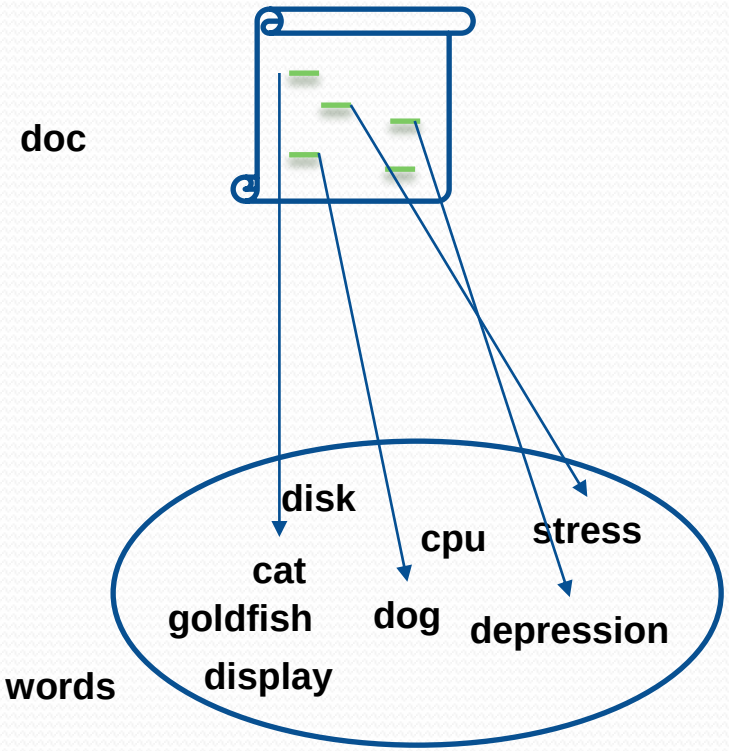
Number of documents containing *t*

Document Data

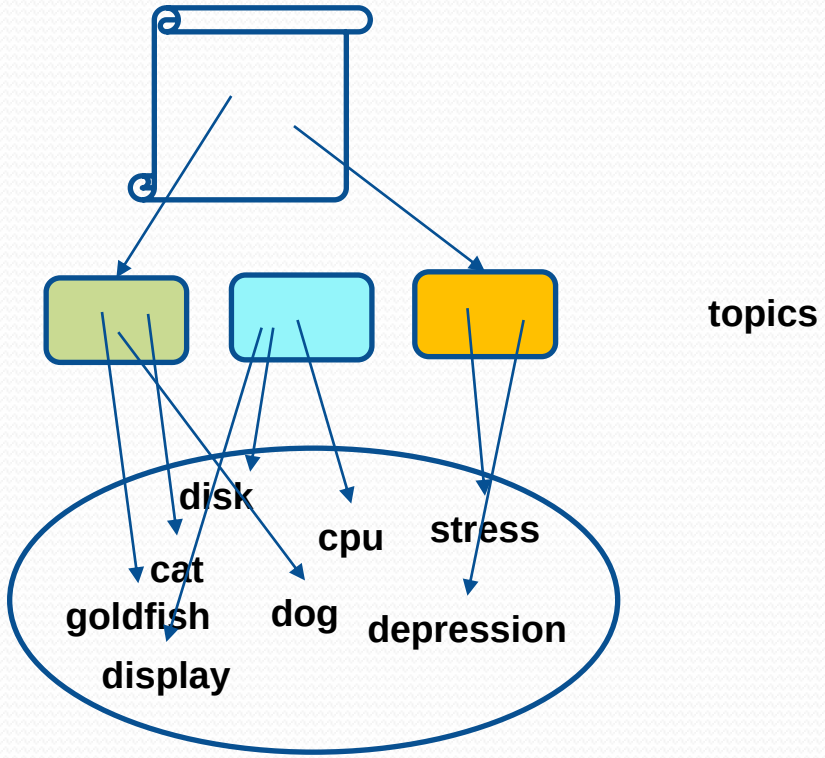
- Topic modelling
  - Analyze a document corpus to identify clusters of words.
  - Each word cluster expresses a latent (hidden) topic
  - It gives the probability that a specific topic is covered in a document.
  - A document can cover multiple topics.

Document Data

- BoW model

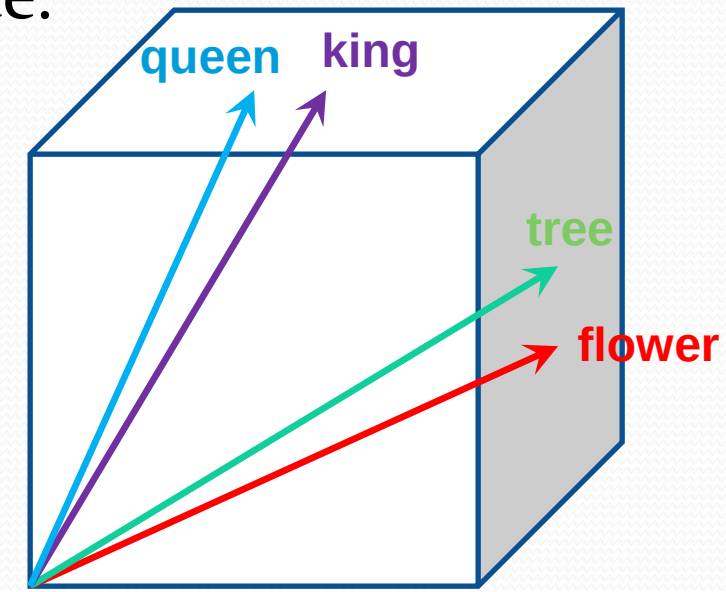


- Topic model



Document Data

- Word Embedding
- Learn a vector representation of each word such that words with related/similar meanings are closer in the vector space.
- Based on the "*distributional hypothesis*"
  (words with similar contexts are semantically related)
- Word2vec, GloVe, BERT



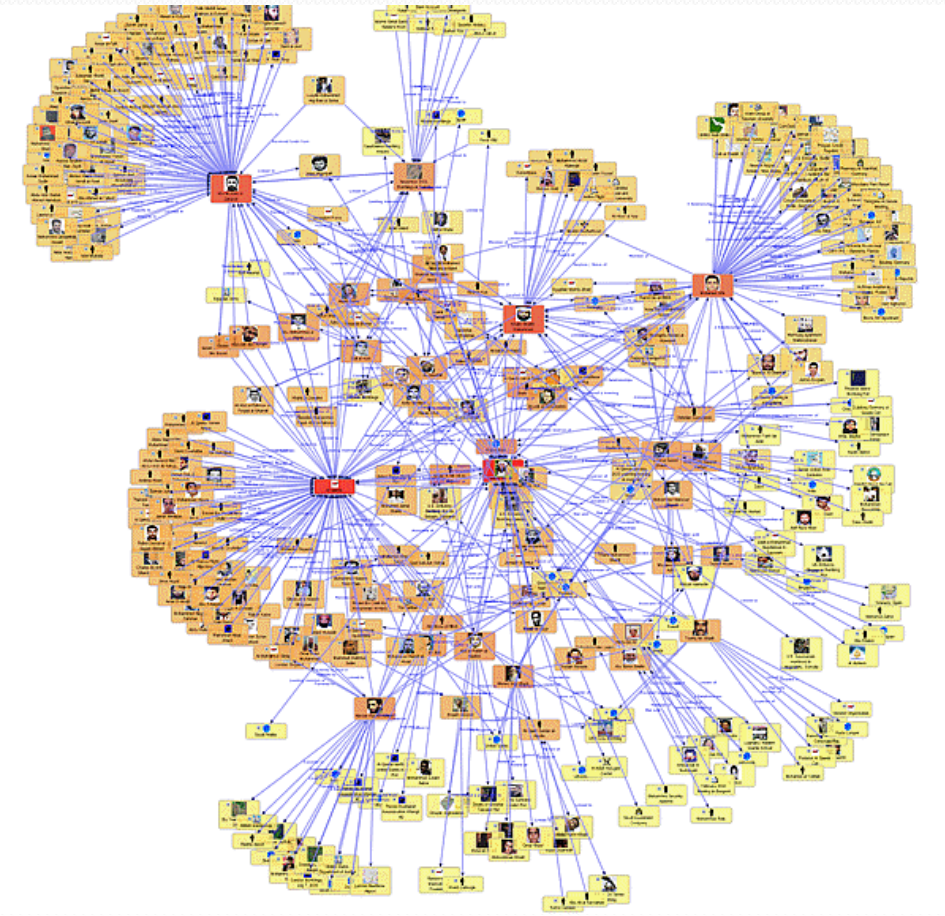
Set-valued Data (Transactional Data)

- Each record (or a "transaction") is a set of items.
- Example: market-basket data. The set of products purchased by a customer during one shopping trip constitute a transaction, while the individual products that were purchased are the items.
- Often, data is represented as "*inverted lists*", in the form *item* → {*a list of transaction ids*}

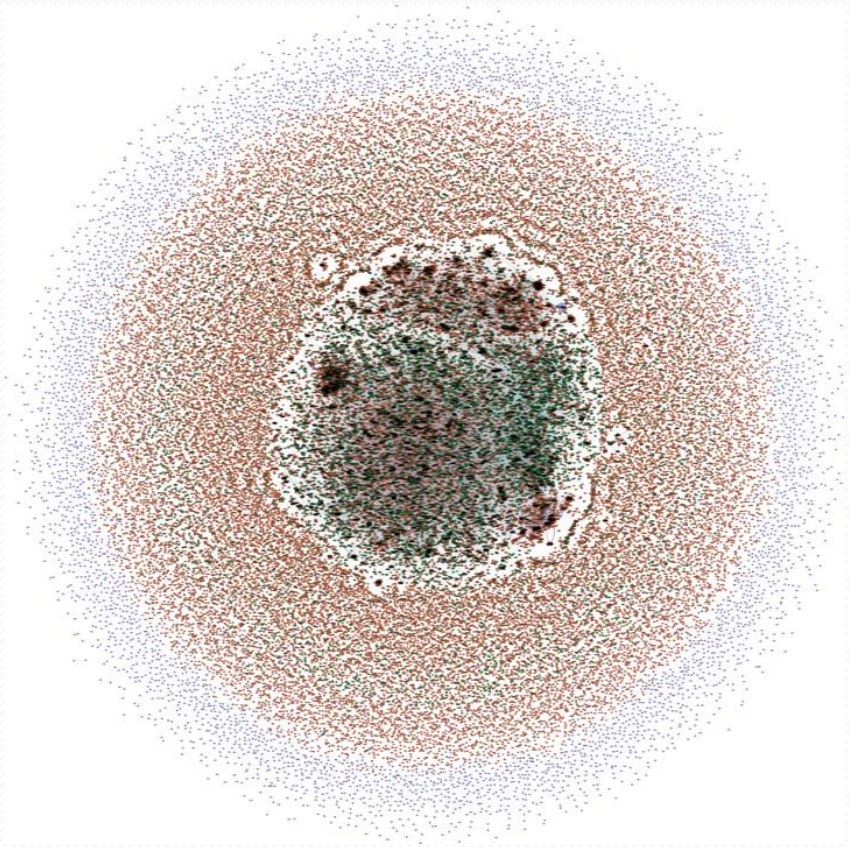
TID	*Items*
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Graph Data

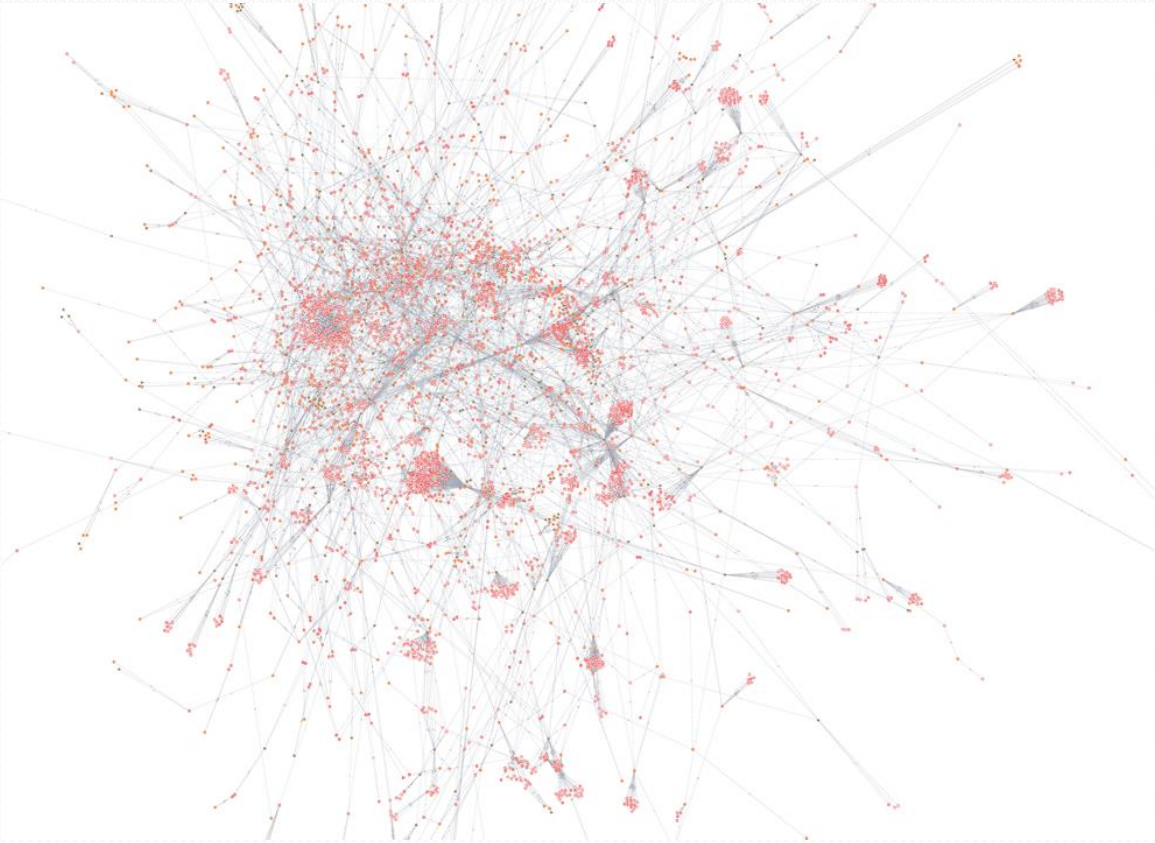
- Examples:
  - Web
  - Social networks
  - Citation networks
  - Knowledge graphs
  - User-item interactions
- How to identify “important” nodes?
- How to measure the “similarity” of nodes?



A Legal Citation Network

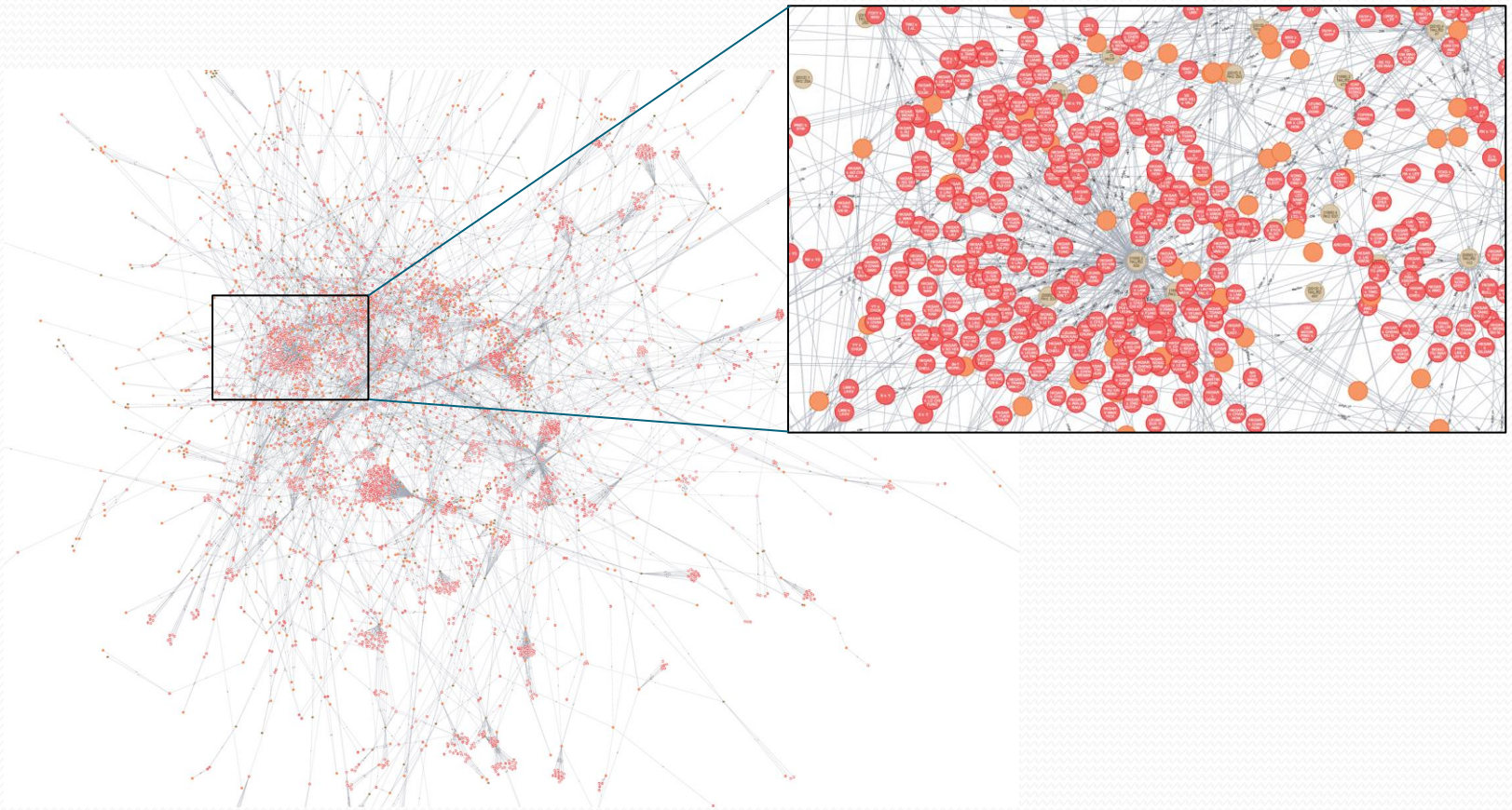


83k Judgments



5k Judgments

Example of a highly-cited judgment



Ordered Data

- Sequences of events
  - E.g., a sequence of purchase transactions
    (A B), (C), (D E F)
  - Log analysis

- Spatiotemporal data
  - E.g., trajectory data (a sequence of timestamped locations)

Other unstructured data

- Example: images, sound files, video, documents
  - Extracting features from them (e.g., object recognition, voice recognition, annotations, etc.)
  - Representing them as record (structured) data based on the features they contain
  - Other descriptive data, such as tags, text descriptions, etc.

Data Quality

- Data should be of good quality for correct data analysis
- Examples of data quality problems:
  - noise and outliers
  - missing values
  - duplicate and inconsistent data

Unusual data
(could be incorrect)

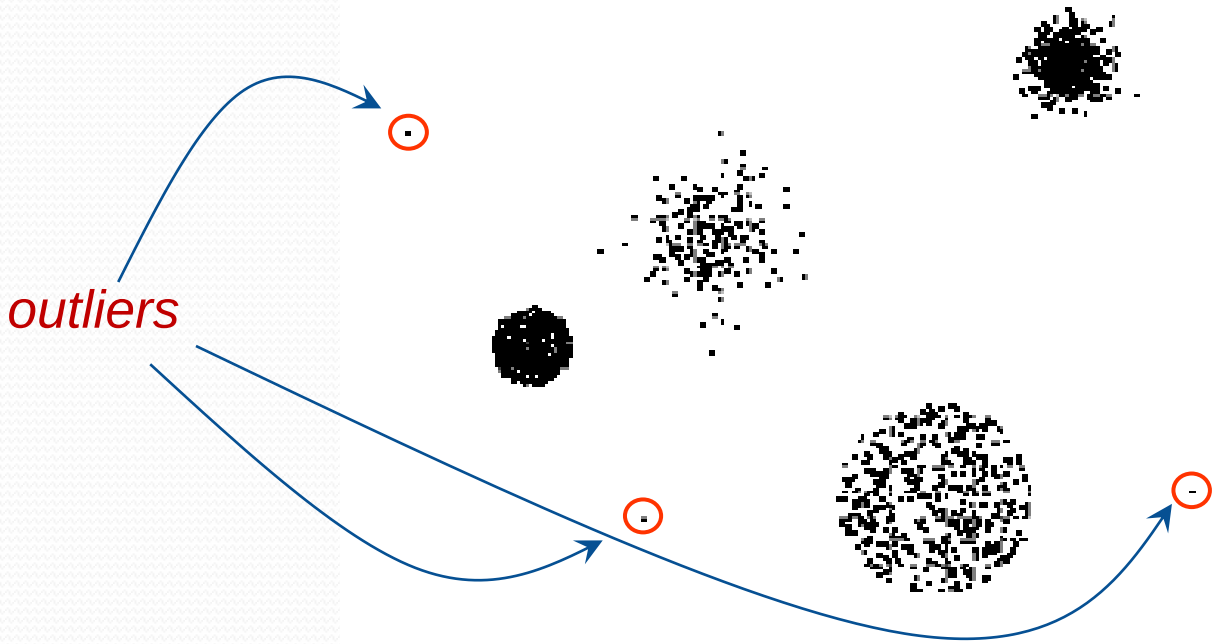
Incorrect data
(could look normal)

Noise and Outliers

- Noise in data could occur due to error in data collection
  - e.g., a student of age 123 years old (but is actually 23)
  - their presence would affect the accuracy of the results
  - should be detected and removed
- Outliers are data that deviate so much from the norm that they may represent some very rare or exceptional cases (or just noise).
- Outliers are not necessarily noise
  - e.g., no one can ever guarantee that no one lives for more than 123 years
  - outliers could be useful or not

Outliers

- Outliers are data objects with characteristics that are considerably different than most of the other data objects in the data set



Outliers

- Example: Credit-card fraud detection relies a lot on outliers (exceptional cases, abnormal spending patterns) detection to discover possible cases of fraud.
- Example: An insurance company uses outlier detection to discover a fraudulent case in which a customer claims maternity compensation twice in half a year time.
- Outliers are not the norm. So, rules about them may or may not be useful.

Detecting Outliers

- One method to detect outliers is to perform cluster analysis.
  - By definition, outliers are data points that are not close to, or similar to, other data points in the data set.
  - A cluster is a group of data points that are similar to each other.
  - Points that do not fall into any cluster could be flagged as outliers.
- Distance-based methods:
  - Simple kNN method: For a data object (point) *p*, the distance to its *k*-th nearest neighbor is taken as *p*'s outlier score. The higher the score, the more likely that *p* is an outlier.
- Model-based methods. Assume data follows certain model. Then, find data points that deviate from the model prediction.

Handling Noise

- No good ways
- Perform outlier detection and then check manually whether the outliers are likely noise
- Data verification
  - E.g., cross validating HITs (*Human Intelligence Tasks*) for data obtained with crowdsourcing

Missing Values

- A common and difficult problem
  - a customer may refuse to supply certain info
  - a change of the database schema may also lead to missing values
- In general, no good solutions

Missing Values

- Possible actions:
  - Ignore records with missing values
    - simple; ok if only a small fraction of the data contains missing values
  - Find the missing data
    - may require expensive and time-consuming labor
  - Guess
    - e.g., fill in the missing values with averages
  - Use a special symbol, say, *null*, to denote a missing value.

Erroneous and Duplicate Data

- Data set may include erroneous values
  - E.g., negative age
  - Detect and treat them as missing values
  - Sometimes error-correction codes can be used to find correct value
- Data set may include data objects that are duplicates, or almost duplicates of one another
  - Major issue when merging data from heterogeneous sources
  - E.g., Same person with multiple email addresses

Data Preprocessing

- is needed to *integrate data* into a data warehouse
  - e.g., a car dealer's database may use a *string*, "red" say, to represent color, another dealer's database may use a *numeric* color code, say 4
  - data may be inconsistent, e.g., a customer may have two phone lines, one registered under the name "P. Chan", another under the name "Peter Chan".
  - naming inconsistency: one db uses "customer-id", another uses "cust-id" as attribute name
  - numerical attributes may be recorded using different unit measures (e.g., HKD, USD, ...)

Data Preprocessing

- is needed to *clean* the *data*
  - E.g., noise due to entry error
- is needed to *reduce* the size of the *data*
  - Raw data may have “too much” details and redundancy
- is needed to *transform the data* into a format that is more suitable for data mining
  - E.g., a document is converted to a TF-IDF vector.

Data Cleaning

- to fill in missing data
- to resolve inconsistency
- to remove noisy data

Data Reduction and Transformation

- Data reduction refers to wisely reducing the size of the data set so that data mining algorithms could be executed more efficiently.
- One should be very careful so that the reduction would not affect the analytical results much.
- Typical approaches:
  - Use fewer records (sampling)
  - Use fewer attributes (feature selection)

Data Reduction and Transformation

- Aggregation
- Generalization
- Sampling
- Dimensionality Reduction
- Feature creation
- Feature selection
- Discretization and Binarization
- Normalization

Aggregation

- Reduce the resolution (or granularity) of data by deriving summary statistics of raw data.
- A busy stock market may process tens or hundreds of trade transactions per second.
- To analyze the data for patterns, it may not be necessary to consider all the details of each transaction.
- Perhaps grouping all transactions that occur in 10-second intervals (and compute the turnover sum) would provide enough resolution (on the time domain) for pattern analysis.

Generalization

- Reduce the cardinality of an attribute's domain.
- Example:
  - A customer database may contain information about the ages of the customers.
  - We want to know how customers' ages affect their buying patterns.
  - We could map the values of the attribute "age" into another attribute "age-group" which takes on one of the following values: "kids, teens, thirties, middle-ages, seniors".
  - The attribute "age-group" is a higher-level concept (or a generalization) of the attribute "age".

Generalization

- Generalization may make rules (1) *more easily found*, (2) *more concise* and (3) *more interpretable*.
- E.g.,
  - "People of age 11 like to play video game"
    "People of age 12 like to play video game"
    ....
  - "Teens like to play video game"
  - Which rule do you like better?

Sampling

- Sampling is the main technique employed for data selection.
  - It is often used for both the preliminary investigation of the data and the final data analysis.
- Statisticians sample because obtaining the entire set of data of interest is too expensive or time consuming.
- Sampling is used in data mining because processing the entire set of data of interest is too expensive or time consuming.

Sampling

- A sample is representative if it has approximately the same property (of interest) as the original set of data.
- With representative sample, using a sample will work almost as well as using the entire data set.

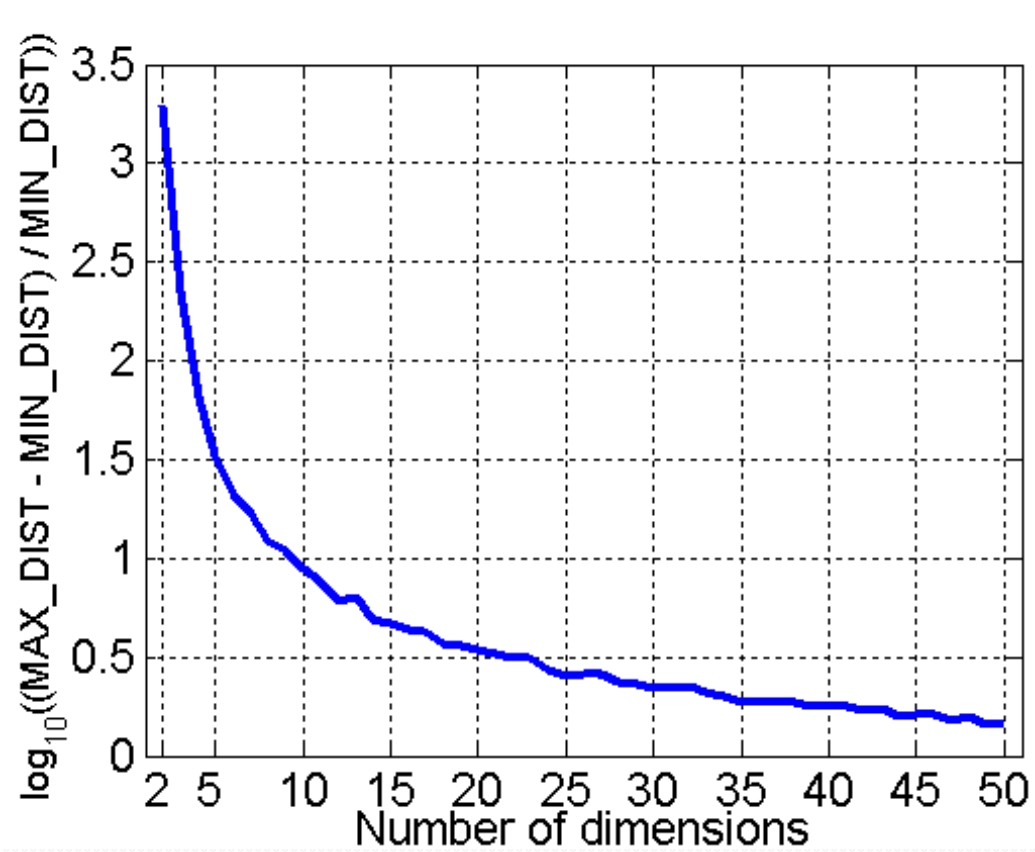
Types of Sampling

- Simple Random Sampling
  - Equal probability of selecting any particular object
- Sampling without replacement
  - As each object is selected, it is removed from the population
- Sampling with replacement
  - Objects are not removed from the population as they are selected for the sample.
    - In sampling with replacement, the same object can be picked up more than once
    - Simpler mathematical analysis
- Stratified sampling
  - Split the data into groups (called strata); then draw random samples from each group

Dimensionality Reduction

- Typical data is high-dimensional (i.e., many attributes)
- Remove attributes without much affecting the mining results.
- e.g., if customer account numbers are randomly assigned, they may not be associated with any property of the customers. So, account numbers could be excluded from the analysis.
- e.g., if two attributes A and B are highly correlated, including only one of them may not affect the results of data mining much (e.g., height and foot-size; number of words and number of characters in a document)

Curse of Dimensionality



- **Randomly generate 500 points**
- **Compute difference between max and min distance between any pair of points**

- When dimensionality increases, data becomes *increasingly sparse* in the space that it occupies
- Definitions of *density* and *distance* between points, which is critical for clustering and outlier detection, become less meaningful
- Sometimes, it is observed that including too many features degrade model performance.
- Data Mining algorithms will become very slow if there are too many features in the data.

Dimensionality Reduction

- Purpose:
  - Avoid curse of dimensionality
  - Reduce amount of time and memory required by data mining algorithms
  - Less complex rules $\Rightarrow$ more interpretable
- Techniques
  - Feature creation
  - Correlation analysis
  - Principal components analysis (PCA)
  - Feature selection

Feature creation

- new attributes are derived from existing attributes
- e.g., a database table may record a person's *height* and *weight*. To understand the association between "overweight" and certain illness, we need a measure of "overweight". *BMI*, which is equal to $weight/height^2$, could be a good indicator. We can derive this new attribute before a data mining algorithm is applied.

Correlation

- Correlation measures the linear relationship between attributes
- Pearson's correlation is a normalized definition of covariance
- Usually visualized using a correlation matrix

$$corr(x, y) = \frac{coVariance(x, y)}{s_x s_y}$$

$$coVariance(x, y) = \frac{1}{n-1} \sum_{k=1}^{n} (x_k - \bar{x})(y_k - \bar{y})$$

$$s_x = \sqrt{\frac{1}{n-1} \sum_{k=1}^{n} (x_k - \bar{x})^2}$$

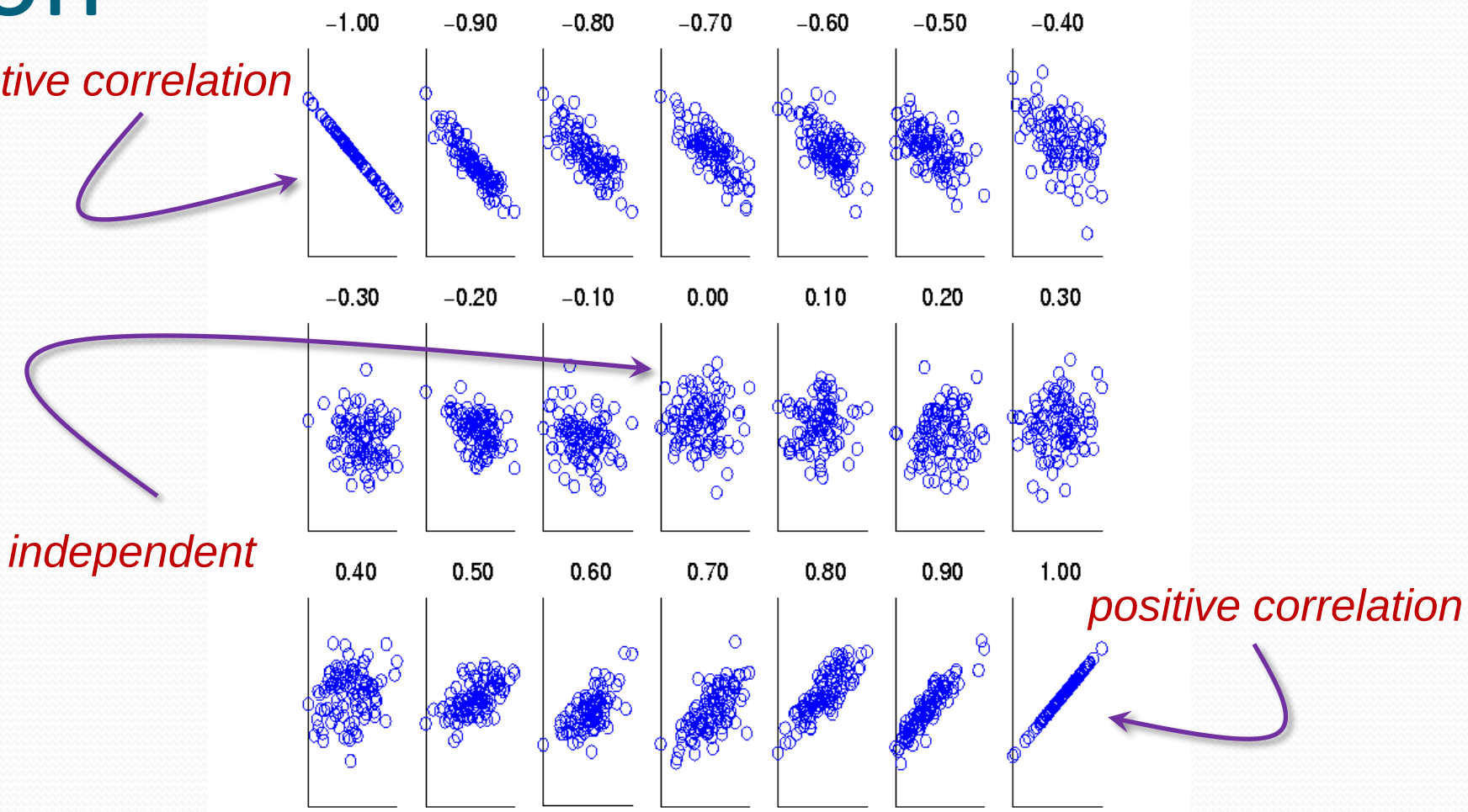
means of x and y

n records in total

x, y values of the k-th record

s.d. of attribute x

Correlation



Principal Components Analysis (PCA)

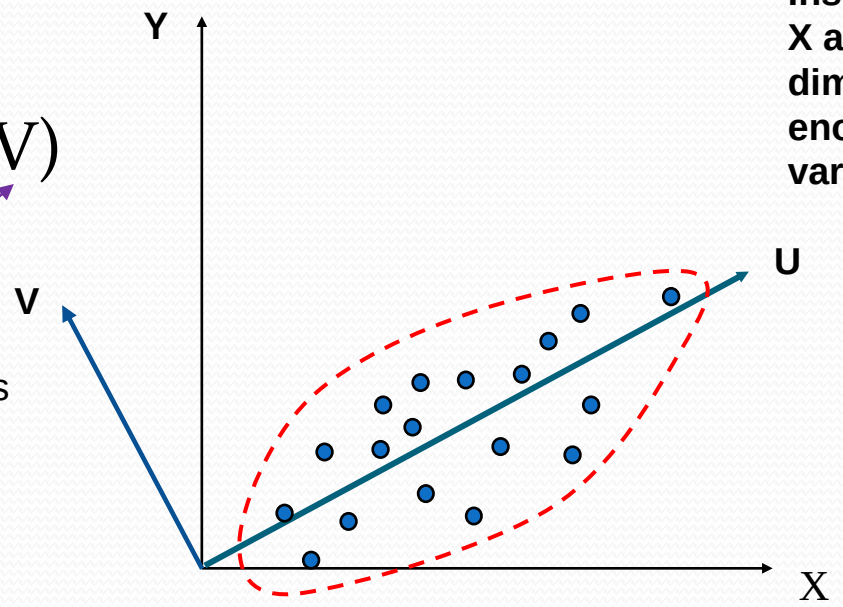
- Input: An *n*-dimensional dataset of objects
- Goal: Find a new set of dimensions that better captures the variability of data
  - First dimension captures as much variability as possible
  - Second dimension should be orthogonal to first and capture as much as possible from the remaining variability
  - …
- Data are "rotated" to a new set of axes, based on variability
- Total variability is preserved but new attributes are uncorrelated

Principal Components Analysis

- Total variance is preserved.
- Var(X) + Var(Y) = Var(U) + Var(V)

original dimensions

transformed dimensions



Instead of using attributes X and Y, using (the derived) dimension U is perhaps good enough to capture the variability in the data.

Why PCA?

- If most of the data variability can be captured by the first *k* dimensions (or "*components*"):
  - Use only first *k* components and discard the others
    - Original dimensions: $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, \ldots$
    - After transformation (in descending variability): $y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, \ldots$
  - The transformed data does not lose much information but have lower dimensionality
  - Search/indexing/analysis is easier for such data
- Rule of thumb: keep the minimum number of components that capture at least 85% of the total variability (as measured by sum of variances) in the data.

Keep these if they represent ≥ 85% of total variability

Feature Selection

- Remove redundant and/or irrelevant features
- Redundant features
  - duplicate much or all of the information contained in one or more other attributes
  - Example: purchase price of a product and the amount of sales tax paid
- Irrelevant features
  - contain no information that is useful for the data mining task at hand
  - Example: customer name is often irrelevant to the task of predicting customers' buying habit.

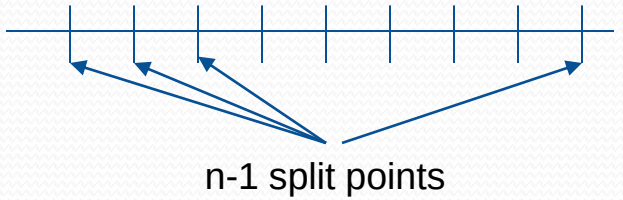
Feature Selection

- Techniques:
  - Brute-force approach:
    - Try all possible feature subsets as input to data mining method
  - Embedded approach:
    - Feature selection occurs naturally as part of the data mining algorithm
  - Filter approach:
    - Features are selected before data mining algorithm is run
  - Wrapper approach:
    - Use the output of a data mining algorithm as feedback to revise the set of attributes used. Example: ablation test

Discretization and Binarization

- Discretization: Convert a numerical attribute into an ordinal/nominal attribute
- Binarization: The resulting attribute is binary.

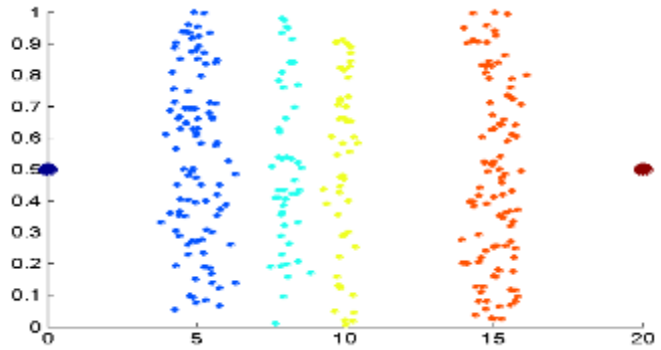
Discretization of Continuous Data



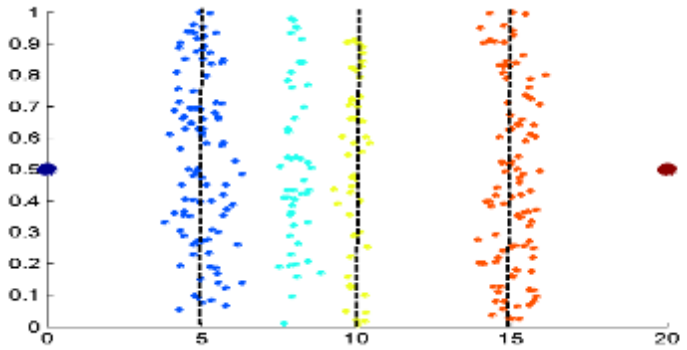
- Continuous data domain is split into ranges
- *n*-1 split points define *n* ranges
- Ranges are values of a discrete attribute
- E.g., height (in cm) discretized into ranges as follows:
  - 0-60, 60-80, 80-100, 100-120, 120-140, 140-160, 160-170, 170-180, 180-190, 190-

10 values

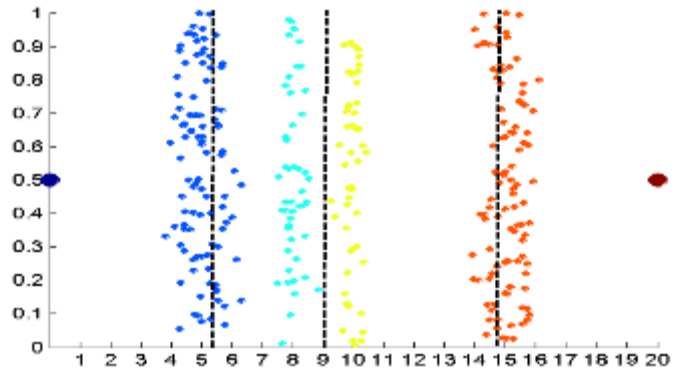
Discretization Examples



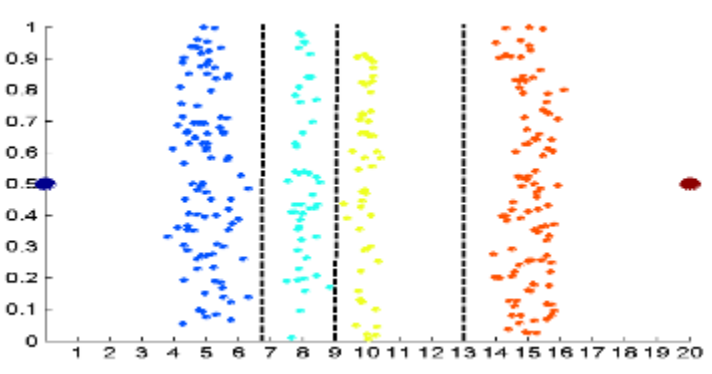
Data



Equal interval width



Equal frequency



K-means

Normalization

- the process of converting numerical numbers to a common range (e.g., [0,1])
- e.g., weight in kg; salary in HKD
  wanna measure how similar two persons are based on their weights and salaries?

Normalization

- min-max normalization

$$x' = (x - min)/(max - min)$$

- z-score normalization

$$x' = (x - \bar{X})/\sigma_X$$

Maximum and minimum values of attribute x found in data

Similarity and Dissimilarity

- *Similarity*
  - Numerical measure of how alike two data objects are.
  - Is higher when objects are more alike.
  - Often falls in the range [0,1]
- *Dissimilarity (distance measure)*
  - 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 how "close" two objects are. It can be expressed by either similarity or dissimilarity.

Similarity/Dissimilarity for Single Attributes

p and *q* are the attribute values for two data objects.

Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & \text{if } p = q \\ 1 & \text{if } p \neq q \end{cases}$	$s = \begin{cases} 1 & \text{if } p = q \\ 0 & \text{if } p \neq q \end{cases}$
Ordinal	$d = \frac{\lvert p-q \rvert}{n-1}$ (values mapped to integers 0 to $n-1$, where n is the number of values)	$s = 1 - \frac{\lvert p-q \rvert}{n-1}$
Numerical	$d = \lvert p - q \rvert$	$s = -d$, $s = \frac{1}{1+d}$ or $s = 1 - \frac{d - min\_d}{max\_d - min\_d}$

Table 5.1. Similarity and dissimilarity for simple attributes

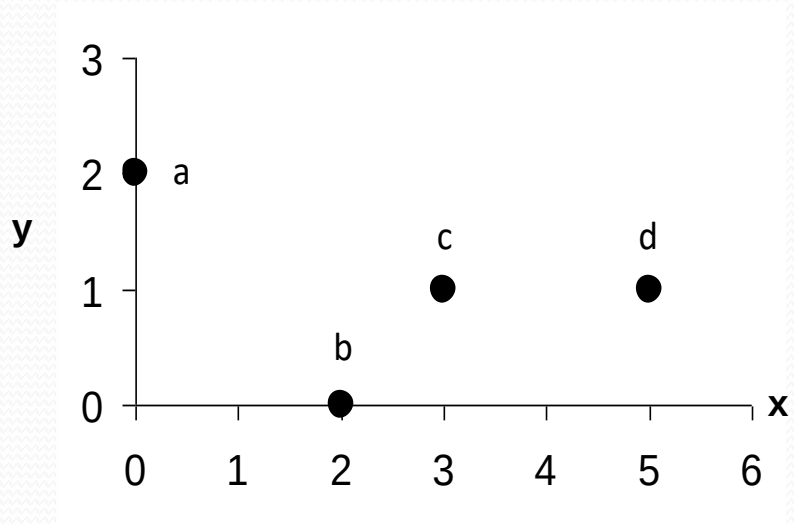
Distance measures for multi-dimensional objects

- Euclidean Distance

$$\boldsymbol{dist} = \sqrt{\sum_{\boldsymbol{k}=1}^{\boldsymbol{n}} (\boldsymbol{p_k} - \boldsymbol{q_k})^2}$$

  - Where n is the number of dimensions (attributes) and p_k and q_k are, respectively, the k-th attribute (component) of data objects p and q.
  - $p = (p_1, p_2, \ldots, p_n); q = (q_1, q_2, \ldots, q_n)$
- Normalization is necessary, if scales differ.

Euclidean Distance



point	x	y
a	0	2
b	2	0
c	3	1
d	5	1

	a	b	c	d
a	0	2.828	3.162	5.099
b	2.828	0	1.414	3.162
c	3.162	1.414	0	2
d	5.099	3.162	2	0

Distance Matrix

Minkowski Distance

- Minkowski Distance (a.k.a. L_p norm) is a generalization of Euclidean Distance

This p is actually this r

$$dist = \left(\sum_{k=1}^{n} | p_k - q_k |^r \right)^{\frac{1}{r}}$$

Minkowski Distance: Examples

- $r = 1$. City block (Manhattan, taxicab, L_1 norm) distance.
- r = 2. Euclidean distance
- $r \to \infty$. "supremum" (L_{max} norm, L_∞ norm) distance.
  - This is the maximum difference between any component of the vectors, i.e., $\max_k |p_k - q_k|$

Minkowski Distance

point	x	y
a	0	2
b	2	0
c	3	1
d	5	1

L1	a	b	c	d
a	0	4	4	6
b	4	0	2	4
c	4	2	0	2
d	6	4	2	0

L2	a	b	c	d
a	0	2.828	3.162	5.099
b	2.828	0	1.414	3.162
c	3.162	1.414	0	2
d	5.099	3.162	2	0

L_∞	a	b	c	d
a	0	2	3	5
b	2	0	1	3
c	3	1	0	2
d	5	3	2	0

Metric

- Some distance measures, such as the Euclidean distance, have the following well known properties.
  - $d(p, q) \geq 0$ for all p and q and $d(p, q) = 0$ only if p = q. (Positivity)
  - $d(p, q) = d(q, p)$ for all p and q. (Symmetry)
  - $d(p, r) \leq d(p, q) + d(q, r)$ for all points p, q, and r. (Triangle Inequality)

  where d(p, q) is the distance (dissimilarity) between points (data objects), p and q.
- A distance that satisfies these properties is a *metric*

Similarity Between Binary Vectors

- When objects p and q have only binary attributes
- Let

  M_{01} = the number of attributes where p is 0 and q is 1

  M_{10} = the number of attributes where p is 1 and q is 0

  M_{00} = the number of attributes where p is 0 and q is 0

  M_{11} = the number of attributes where p is 1 and q is 1

Simple Matching and Jaccard Coefficients

SMC = number of matches / number of attributes

$= (M_{11} + M_{00}) / (M_{01} + M_{10} + M_{11} + M_{00})$

for symmetric attributes

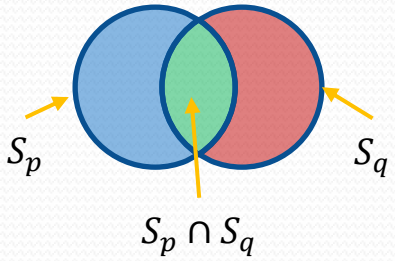
J = number of 1-matches / number of not-both-zero attributes

$= (M_{11}) / (M_{10} + M_{01} + M_{11})$

for asymmetric attributes

Symmetric Difference = $(M_{10} + M_{01}) / (M_{10} + M_{01} + M_{11})$

$= 1 - J$



SMC versus Jaccard: Example

p =	1	0	0	0	0	0	0	0	0	0
q =	0	0	0	0	0	0	1	0	0	1

$M_{01} = 2$ (the number of attributes where p was 0 and q was 1)
$M_{10} = 1$ (the number of attributes where p was 1 and q was 0)
$M_{00} = 7$ (the number of attributes where p was 0 and q was 0)
$M_{11} = 0$ (the number of attributes where p was 1 and q was 1)

$SMC = (M_{11} + M_{00})/(M_{01} + M_{10} + M_{11} + M_{00}) = (0+7) / (2+1+0+7) = 0.7$

$J = (M_{11}) / (M_{01} + M_{10} + M_{11}) = 0 / (2 + 1 + 0) = 0$

Document data: Cosine Similarity

If d1 and d2 are two document vectors, then

$$\cos(d1, d2) = (d1 \bullet d2) / ||d1|| \, ||d2||,$$

where • indicates vector dot product and ||d|| is the norm of vector d.

Example:

d1 =	3	2	0	5	0	0	0	2	0	0
d2 =	1	0	0	0	0	0	0	1	0	2

d1 • d2= 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5

||d1|| = (3*3+2*2+0*0+5*5+0*0+0*0+0*0+2*2+0*0+0*0)^0.5 = (42)^0.5 = 6.481

||d2|| = (1*1+0*0+0*0+0*0+0*0+0*0+0*0+1*1+0*0+2*2)^0.5 = (6)^0.5 = 2.45

cos(d1, d2) = .3150

Using Weights to Combine Similarities

- May not want to treat all attributes the same.
  - Use weights w_k's. $0 \leq w_k \leq 1; \sum_k w_k = 1$

$$\boldsymbol{similarity(p,q) = \sum_{k=1}^{n} w_k s_k}$$

Similarity between objects p and q w.r.t. attribute k