Introduction to Data Analytics

SS 2016

Prof. Dr. Xiaoming Fu

Acknowledgement: Bing Liu (UIC)

What is Data Analytics

- Numerous datasets available nowadays
  - Big vs. small, structured vs. non-structured, spatial vs. temporal, …
- Data Analytics: Apply a mechanical or algorithmic process to derive the insights, e.g. running through several datasets for correction
  - *data fusion*: analyzing data cross different domains
- When the data is excessive
  - Various big data methods necessary
  - Sometimes Hadoop/cloud servers are required

Related Concepts

- Big data
  - 4Vs – Volume, Variety, Velocity, Veracity (usefulness)
- Data science
  - A combination of math, statistics, programming, the domain knowledge, data collection & cleaning etc.
- Data Indexing, Searching, and Querying
  - Keyword based search
  - Pattern matching (XML/RDF)

Relationship with big data

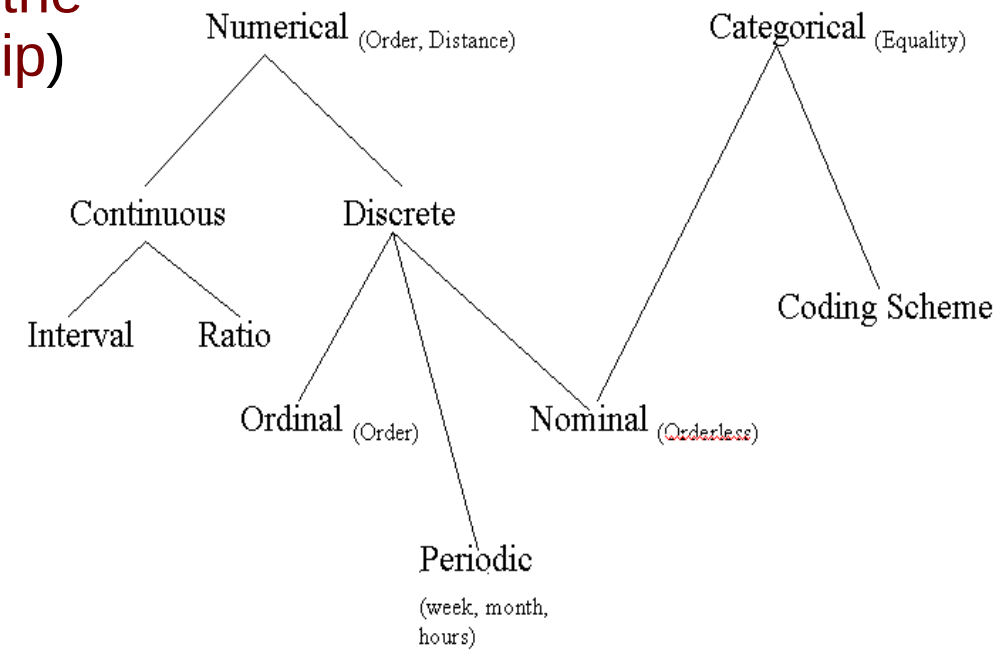
- Knowledge discovery
  - Data Preprocessing
  - Statistical Modeling
  - Data Mining: Discover *unknown* properties on the data
    - Unsupervised learning
    - Supervised learning
  - Machine learning
    - Use *known* properties learned from the *training data* to predict
- Social network analysis on (big) data
  - The study of social entities (people in an organization/social unit, called actors), and their interactions and relationships.

Data Preprocessing

Data Types and Forms

- Attribute-value data:
- Data types
  - numeric, categorical (see the hierarchy for its relationship)
  - static, dynamic (temporal)
- Other kinds of data
  - distributed data
  - text, Web, meta data
  - images, audio/video

A1	A2	...	An	C



Data Preprocessing

- Why preprocess the data?
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization
- Summary

Why Data Preprocessing?

- Data in the real world is dirty
  - incomplete: missing attribute values, lack of certain attributes of interest, or containing only aggregate data
    - e.g., occupation=“”
  - noisy: containing errors or outliers
    - e.g., Salary=“-10”
  - inconsistent: containing discrepancies in codes or names
    - e.g., Age=“42” Birthday=“03/07/1997”
    - e.g., Was rating “1,2,3”, now rating “A, B, C”
    - e.g., discrepancy between duplicate records

Why Is Data Preprocessing Important?

- No quality data, no quality mining results!
  - Quality decisions must be based on quality data
    - e.g., duplicate or missing data may cause incorrect or even misleading statistics.
- Data preparation, cleaning, and transformation comprises the majority of the work in a data mining application (90%).

Multi-Dimensional Measure of Data Quality

- A well-accepted multi-dimensional view:
  - Accuracy
  - Completeness
  - Consistency
  - Timeliness
  - Believability
  - Value added
  - Interpretability
  - Accessibility

Major Tasks in Data Preprocessing

- Data cleaning
  - Fill in missing values, smooth noisy data, identify or remove outliers and noisy data, and resolve inconsistencies
- Data integration
  - Integration of multiple databases, or files
- Data transformation
  - Normalization and aggregation
- Data reduction
  - Obtains reduced representation in volume but produces the same or similar analytical results
- Data discretization (for numerical data)

Data Preprocessing

- Why preprocess the data?
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization
- Summary

Data Cleaning

- Importance
  - “Data cleaning is the number one problem in data warehousing”
- Data cleaning tasks
  - Fill in missing values
  - Identify outliers and smooth out noisy data
  - Correct inconsistent data
  - Resolve redundancy caused by data integration

Missing Data

- Data is not always available
  - E.g., many tuples have no recorded values for several attributes, such as customer income in sales data
- Missing data may be due to
  - equipment malfunction
  - inconsistent with other recorded data and thus deleted
  - data not entered due to misunderstanding
  - certain data may not be considered important at the time of entry
  - not register history or changes of the data

How to Handle Missing Data?

- Ignore the tuple
- Fill in missing values manually: tedious + infeasible?
- Fill in it automatically with
  - a global constant : e.g., “unknown”, a new class?!
  - the attribute mean
  - the most probable value: inference-based such as Bayesian formula, decision tree, or EM algorithm

Noisy Data

- Noise: random error or variance in a measured variable.
- Incorrect attribute values may due to
  - faulty data collection instruments
  - data entry problems
  - data transmission problems
  - etc
- Other data problems which requires data cleaning
  - duplicate records, incomplete data, inconsistent data

How to Handle Noisy Data?

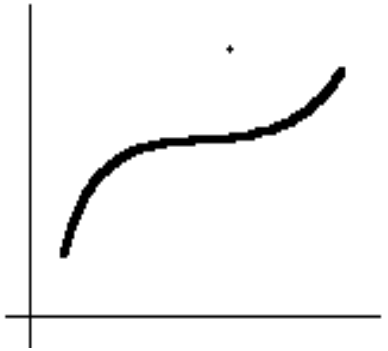
- Binning method:
  - first sort data and partition into (equi-depth) bins
  - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
- Clustering
  - detect and remove outliers
- Combined computer and human inspection
  - detect suspicious values and check by human (e.g., deal with possible outliers)

Binning Methods for Data Smoothing

- Sorted data for price (in dollars): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34
- Partition into (equi-depth) bins:
  - Bin 1: 4, 8, 9, 15
  - Bin 2: 21, 21, 24, 25
  - Bin 3: 26, 28, 29, 34
- Smoothing by bin means:
  - Bin 1: 9, 9, 9, 9
  - Bin 2: 23, 23, 23, 23
  - Bin 3: 29, 29, 29, 29
- Smoothing by bin boundaries:
  - Bin 1: 4, 4, 4, 15
  - Bin 2: 21, 21, 25, 25
  - Bin 3: 26, 26, 26, 34

Outlier Removal

- Data points inconsistent with the majority of data
- Different outliers
  - Valid: CEO's salary,
  - Noisy: One's age = 200, widely deviated points
- Removal methods
  - Clustering
  - Curve-fitting
  - Hypothesis-testing with a given model



Data Preprocessing

- Why preprocess the data?
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization
- Summary

Data Integration

- Data integration:
  - combines data from multiple sources
- Schema integration
  - integrate metadata from different sources
  - Entity identification problem: identify real world entities from multiple data sources, e.g., A.cust-id ≡ B.cust-#
- Detecting and resolving data value conflicts
  - for the same real world entity, attribute values from different sources are different, e.g., different scales, metric vs. British units
- Removing duplicates and redundant data

Data Transformation

- Smoothing: remove noise from data
- Normalization: scaled to fall within a small, specified range
- Attribute/feature construction
  - New attributes constructed from the given ones
- Aggregation: summarization
- Generalization: concept hierarchy climbing

Data Transformation: Normalization

- min-max normalization

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

- z-score normalization

$$v' = \frac{v - mean_A}{stand\_dev_A}$$

- normalization by decimal scaling

$$v' = \frac{v}{10^j}$$ Where j is the smallest integer such that Max($|v'|$)<1

Data Preprocessing

- Why preprocess the data?
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization
- Summary

Data Reduction Strategies

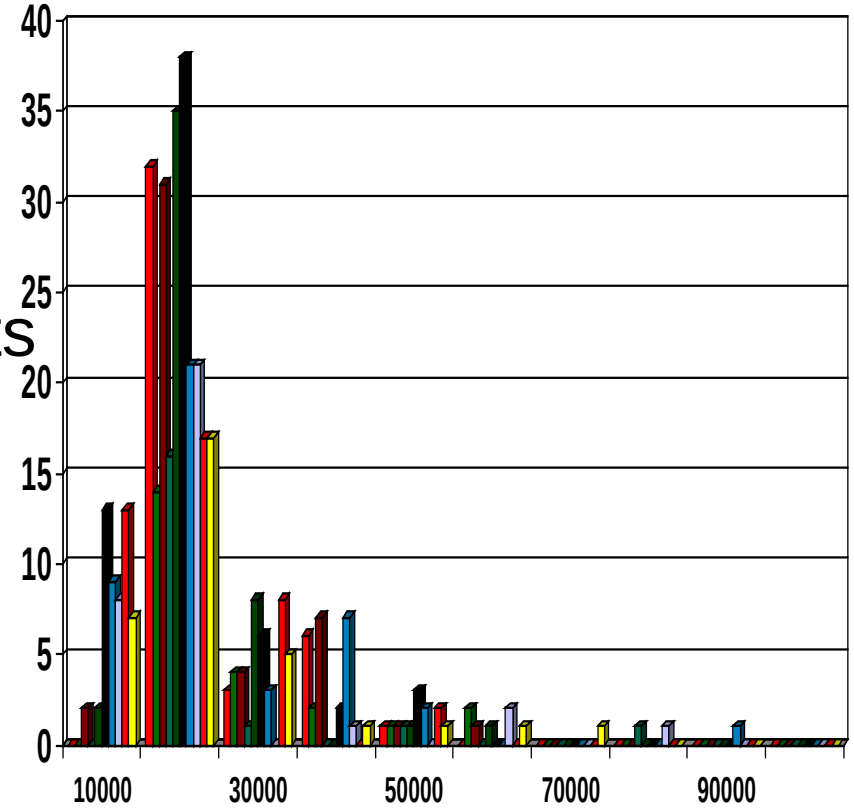
- Data is too big to work with
- Data reduction
  - Obtain a reduced representation of the data set that is much smaller in volume but yet produce the same (or almost the same) analytical results
- Data reduction strategies
  - Dimensionality reduction — remove unimportant attributes
  - Aggregation and clustering
  - Sampling

Dimensionality Reduction

- Feature selection (i.e., attribute subset selection):
  - Select a minimum set of attributes (features) that is sufficient for the data mining task.
- Heuristic methods (due to exponential # of choices):
  - step-wise forward selection
  - step-wise backward elimination
  - combining forward selection and backward elimination
  - etc

Histograms

- A popular data reduction technique
- Divide data into buckets and store average (sum) for each bucket



Clustering

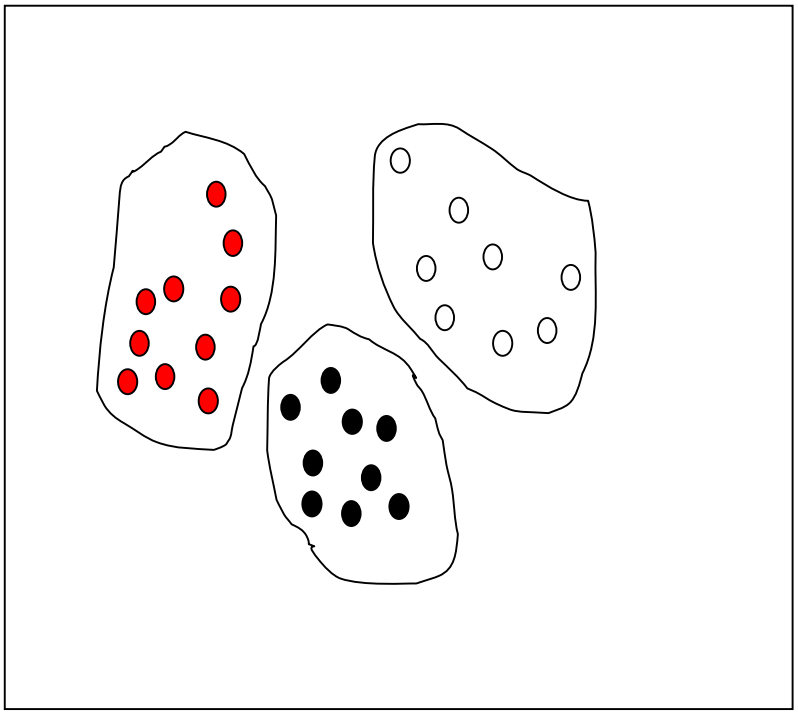
- Partition data set into clusters, and one can store cluster representation only
- Can be very effective if data is clustered but not if data is “smeared”
- There are many choices of clustering definitions and clustering algorithms. We will discuss them later.

Sampling

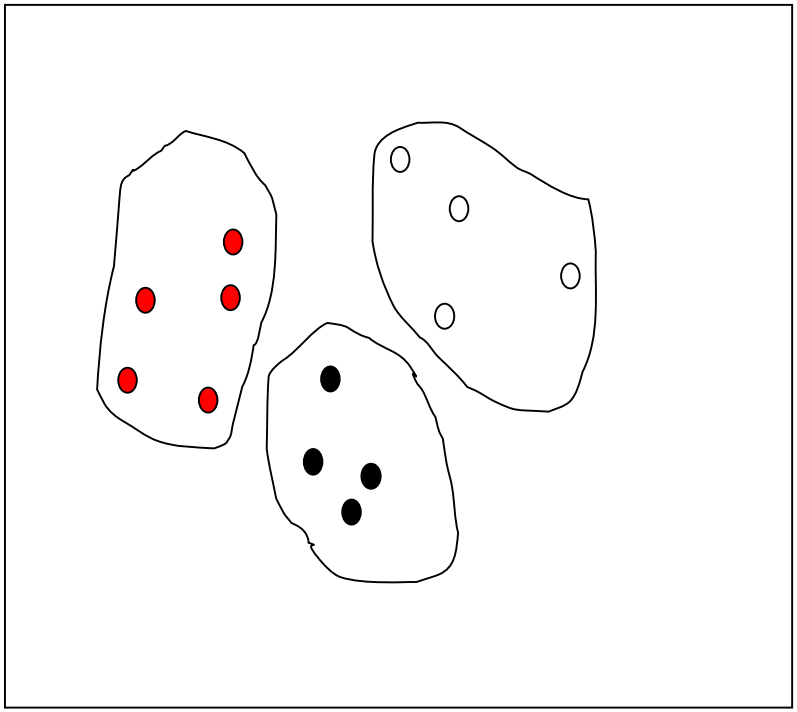
- Choose a representative subset of the data
  - Simple random sampling may have poor performance in the presence of skew.
- Develop adaptive sampling methods
  - Stratified sampling:
    - Approximate the percentage of each class (or subpopulation of interest) in the overall database
    - Used in conjunction with skewed data

Sampling

Raw Data



Cluster/Stratified Sample



Data Preprocessing

- Why preprocess the data?
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization
- Summary

Discretization

- Three types of attributes:
  - Nominal — values from an unordered set
  - Ordinal — values from an ordered set
  - Continuous — real numbers
- Discretization:
  - divide the range of a continuous attribute into intervals because some data mining algorithms only accept categorical attributes.
- Some techniques:
  - Binning methods – equal-width, equal-frequency
  - Entropy-based methods

Discretization and Concept Hierarchy

- Discretization
  - reduce the number of values for a given continuous attribute by dividing the range of the attribute into intervals. Interval labels can then be used to replace actual data values
- Concept hierarchies
  - reduce the data by collecting and replacing low level concepts (such as numeric values for the attribute age) by higher level concepts (such as young, middle-aged, or senior)

Binning

- Attribute values (for one attribute e.g., age):
  - 0, 4, 12, 16, 16, 18, 24, 26, 28
- Equi-width binning – for bin width of e.g., 10:
  - Bin 1: 0, 4 [-,10) bin
  - Bin 2: 12, 16, 16, 18 [10,20) bin
  - Bin 3: 24, 26, 28 [20,+) bin
  - – denote negative infinity, + positive infinity
- Equi-frequency binning – for bin density of e.g., 3:
  - Bin 1: 0, 4, 12 [-, 14) bin
  - Bin 2: 16, 16, 18 [14, 21) bin
  - Bin 3: 24, 26, 28 [21,+] bin

Summary

- Data preparation is a big issue for data mining
- Data preparation includes
  - Data cleaning and data integration
  - Data reduction and feature selection
  - Discretization
- Many methods have been proposed but still an active area of research

Data Mining

What is Data Mining?

- Discovery of useful, possibly unexpected, patterns in data
- Non-trivial extraction of implicit, previously unknown and potentially useful information from data
- Exploration & analysis, by automatic or semi-automatic means, of large quantities of data in order to discover meaningful patterns

Data Mining Process

- Understand the application domain
- Identify data sources and select target data
- Pre-processing: cleaning, attribute selection, etc
- Data mining to extract patterns or models
- Post-processing: identifying interesting or useful patterns/knowledge
- Incorporate patterns/knowledge in real world tasks

Data Mining Tasks

- Classification [Predictive]
- Clustering [Descriptive]
- Association Rule Discovery [Descriptive]
- Sequential Pattern Discovery [Descriptive]
- Regression [Predictive]
- Deviation Detection [Predictive]
- Collaborative Filter [Predictive]

Supervised learning (classification) vs. Unsupervised learning (clustering)

- Supervised learning: classification is seen as supervised learning from examples.
  - Supervision: The data (observations, measurements, etc.) are labeled with pre-defined classes. It is like that a “teacher” gives the classes (supervision).
  - Test data are classified into these classes too.
- Unsupervised learning (clustering)
  - Class labels of the data are unknown
  - Given a set of data, the task is to establish the existence of classes or clusters in the data

What do we mean by learning?

- Given
  - a data set *D*,
  - a task *T*, and
  - a performance measure *M*,

  a computer system is said to **learn** from *D* to perform the task *T* if after learning the system's performance on *T* improves as measured by *M*.
- In other words, the learned model helps the system to perform *T* better as compared to no learning.

An example

- Data: Loan application data
- Task: Predict whether a loan should be approved or not.
- Performance measure: accuracy.

No learning: classify all future applications (test data) to the majority class (i.e., Yes):

Accuracy = 9/15 = 60%.

- We can do better than 60% with learning.

Fundamental assumption of learning

Assumption: The distribution of training examples is identical to the distribution of test examples (including future unseen examples).

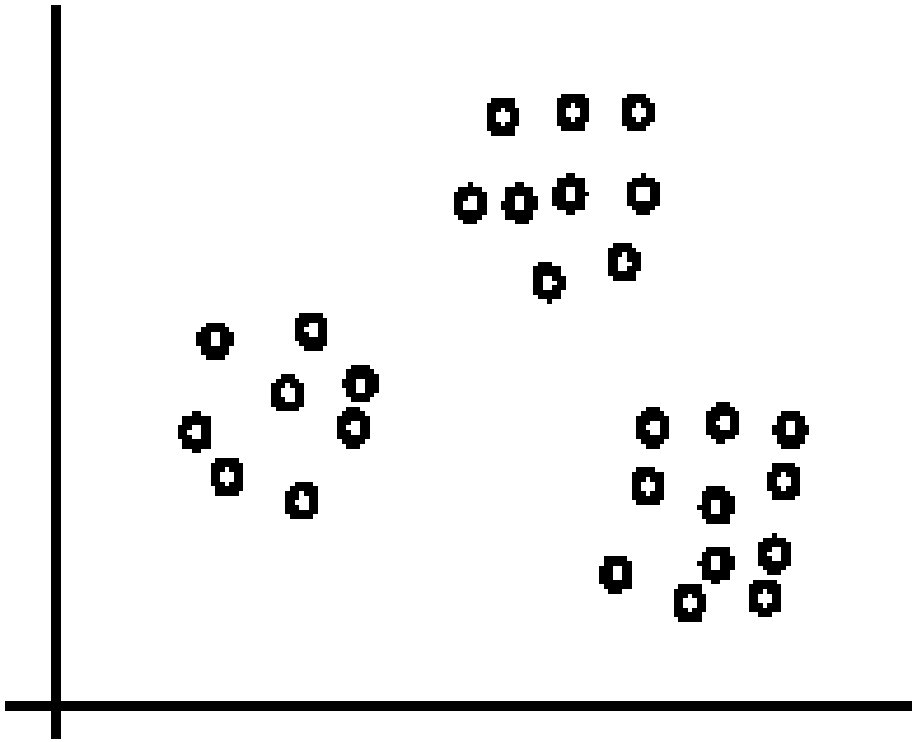
- In practice, this assumption is often violated to certain degree.
- Strong violations will clearly result in poor classification accuracy.
- To achieve good accuracy on the test data, training examples must be sufficiently representative of the test data.

Clustering

- Clustering is a technique for finding similarity groups in data, called **clusters**. I.e.,
  - it groups data instances that are similar to (near) each other in one cluster and data instances that are very different (far away) from each other into different clusters.
- Clustering is often called an **unsupervised learning** task as no class values denoting an *a priori* grouping of the data instances are given, which is the case in supervised learning.

An illustration

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



What is clustering for?

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

What is clustering for? (cont…)

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

Aspects of clustering

- A clustering algorithm
  - Partitional clustering, e.g., K-Means
  - Hierarchical clustering
  - …
- A distance (similarity, or dissimilarity) function
- Clustering quality
  - Inter-clusters distance $\Rightarrow$ maximized
  - Intra-clusters distance $\Rightarrow$ minimized
- The quality of a clustering result depends on the algorithm, the distance function, and the application.

K-means clustering

- K-means is a partitional clustering algorithm
- Let the set of data points (or instances) D be

  $$\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\},$$

  where $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ir})$ is a vector in a real-valued space $X \subseteq R^r$, and r is the number of attributes (dimensions) in the data.
- The *k*-means algorithm partitions the given data into *k* clusters.
  - Each cluster has a cluster **center**, called **centroid**.
  - *k* is specified by the user

K-means algorithm

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

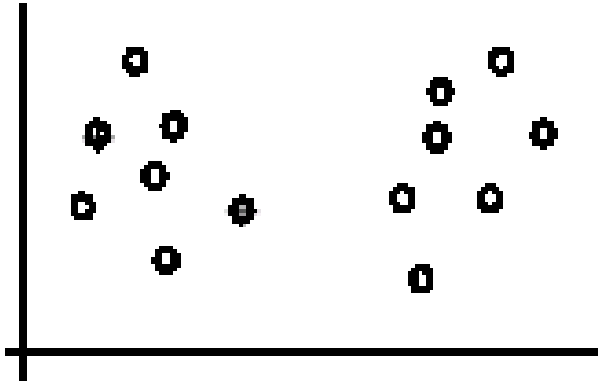
Stopping/convergence criterion

1. no (or minimum) re-assignments of data points to different clusters,
2. no (or minimum) change of centroids, or
3. minimum decrease in the **sum of squared error** (SSE),

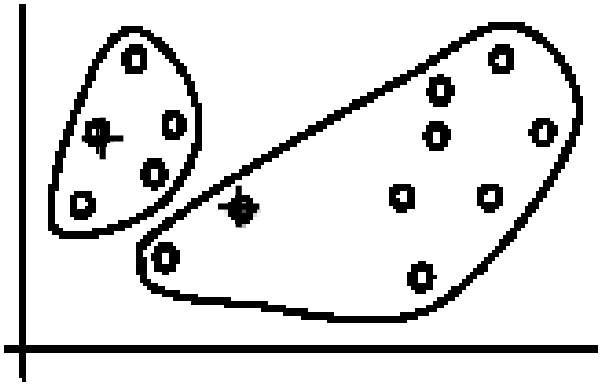
$$SSE = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} dist(\mathbf{x}, \mathbf{m}_j)^2 \quad (1)$$

- C_i is the jth cluster, $\mathbf{m}_j$ is the centroid of cluster C_j (the mean vector of all the data points in C_j), and $dist(\mathbf{x}, \mathbf{m}_j)$ is the distance between data point $\mathbf{x}$ and centroid $\mathbf{m}_j$.

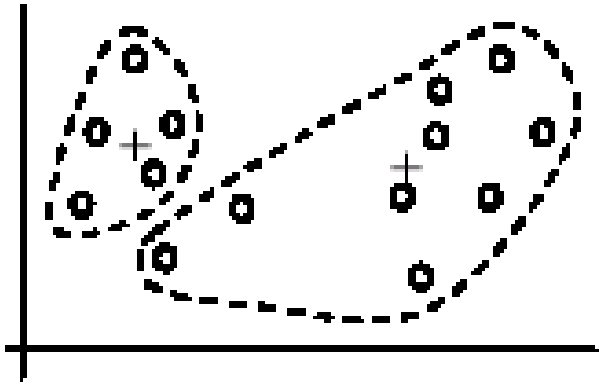
An example



(A). Random selection of k centers

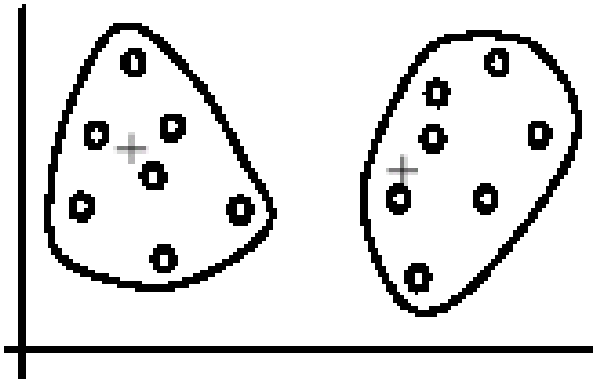


Iteration 1: (B). Cluster assignment

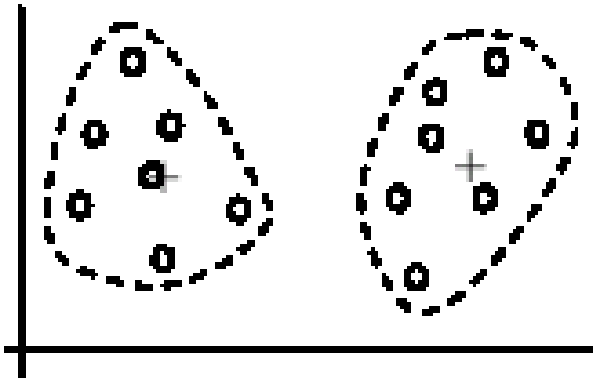


(C). Re-compute centroids

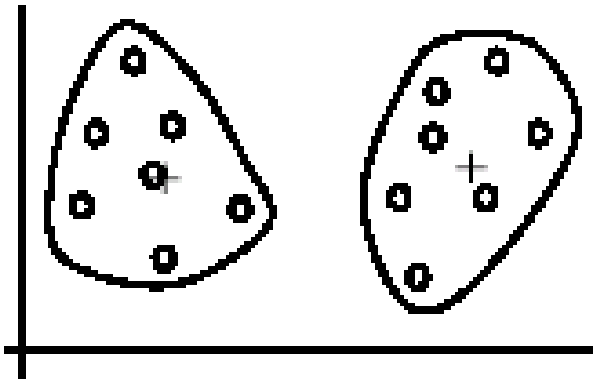
An example (cont …)



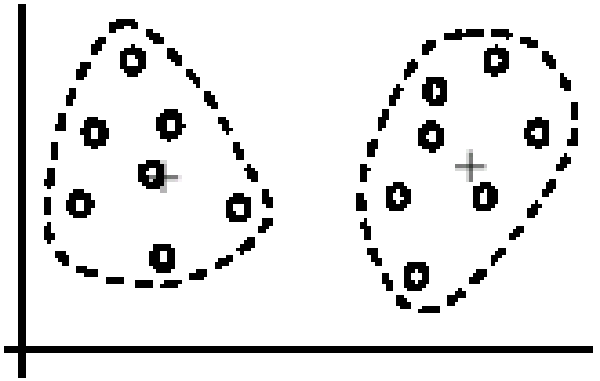
Iteration 2: (D). Cluster assignment



(E). Re-compute centroids



Iteration 3: (F). Cluster assignment



(G). Re-compute centroids

An example distance function

The *k*-means algorithm can be used for any application data set where the **mean** can be defined and computed. In the **Euclidean space**, the mean of a cluster is computed with:

$$\mathbf{m}_j = \frac{1}{|C_j|} \sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i \qquad (2)$$

where $|C_j|$ is the number of data points in cluster C_j. The distance from one data point $\mathbf{x}_i$ to a mean (centroid) $\mathbf{m}_j$ is computed with

$$dist(\mathbf{x}_i, \mathbf{m}_j) = \| \mathbf{x}_i - \mathbf{m}_j \| \qquad (3)$$

$$= \sqrt{(x_{i1} - m_{j1})^2 + (x_{i2} - m_{j2})^2 + \ldots + (x_{ir} - m_{jr})^2}$$

A disk version of *k*-means

- K-means can be implemented with data on disk
  - In each iteration, it scans the data once.
  - as the centroids can be computed incrementally
- It can be used to cluster large datasets that do not fit in main memory
- We need to control the number of iterations
  - In practice, a limited is set (< 50).
- Not the best method. There are other scale-up algorithms, e.g., BIRCH.

Strengths of k-means

- Strengths:
  - Simple: easy to understand and to implement
  - Efficient: Time complexity: $O(tkn)$,

    where n is the number of data points,

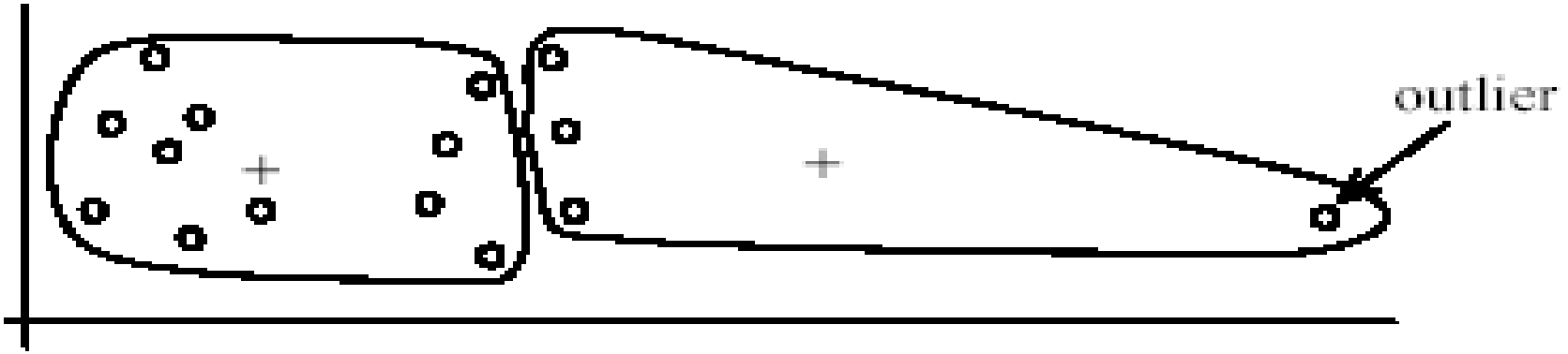
    k is the number of clusters, and

    t is the number of iterations.
  - Since both k and t are small. k-means is considered a linear algorithm.
- K-means is the most popular clustering algorithm.
- Note that: it terminates at a local optimum if SSE is used. The global optimum is hard to find due to complexity.

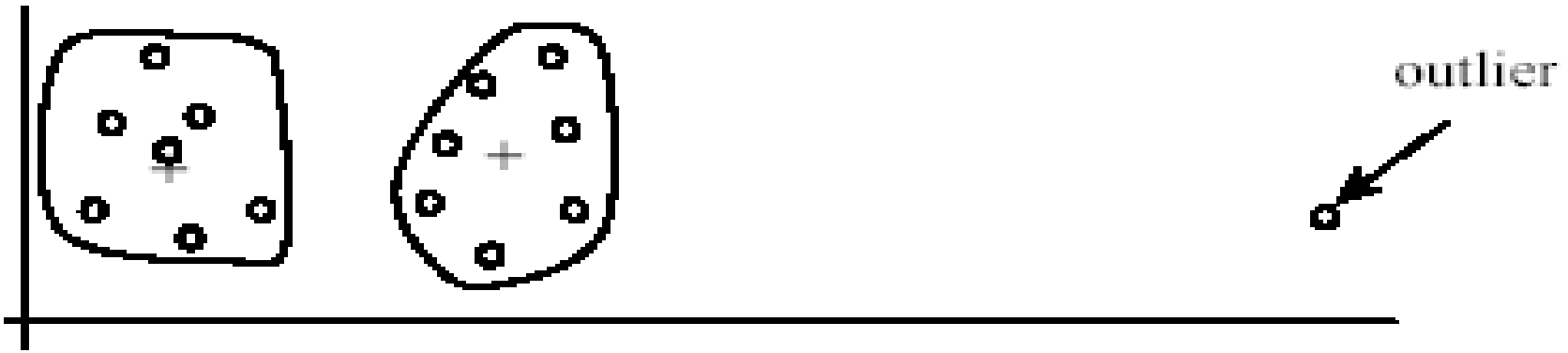
Weaknesses of k-means

- The algorithm is only applicable if the mean is defined.
  - For categorical data, *k*-mode - the centroid is represented by most frequent values.
- The user needs to specify *k*.
- The algorithm is sensitive to **outliers**
  - Outliers are data points that are very far away from other data points.
  - Outliers could be errors in the data recording or some special data points with very different values.

Weaknesses of k-means: Problems with outliers



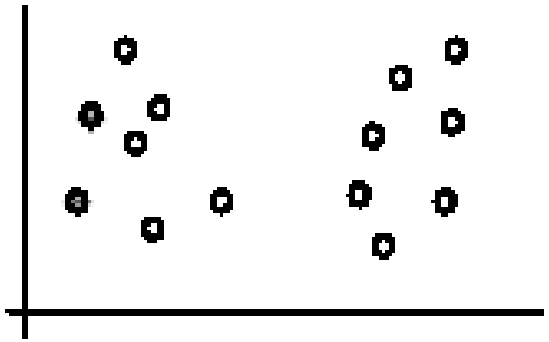
(A): Undesirable clusters



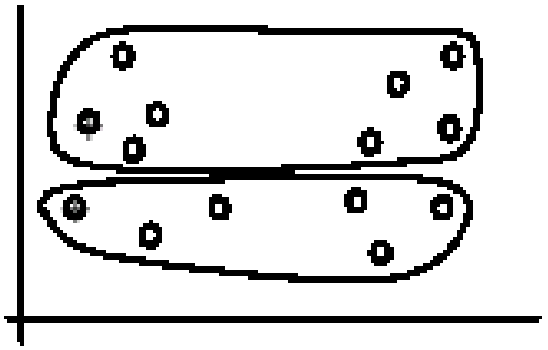
(B): Ideal clusters

Weaknesses of k-means (cont …)

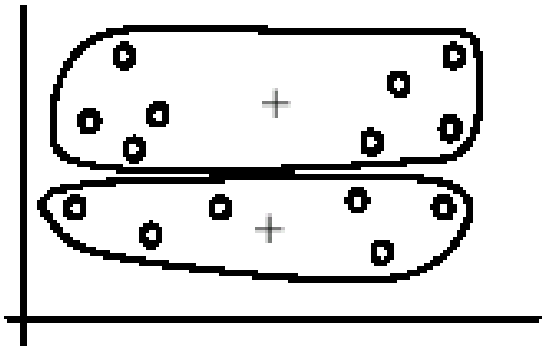
- The algorithm is sensitive to initial seeds.



(A). Random selection of seeds (centroids)



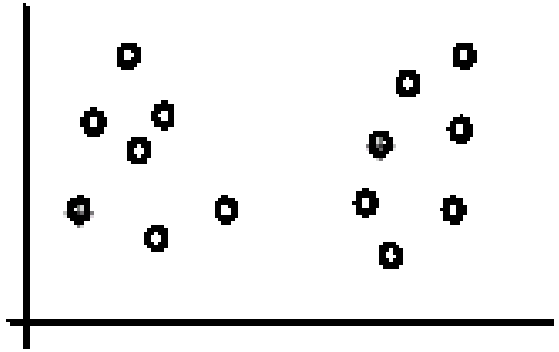
(B). Iteration 1



(C). Iteration 2

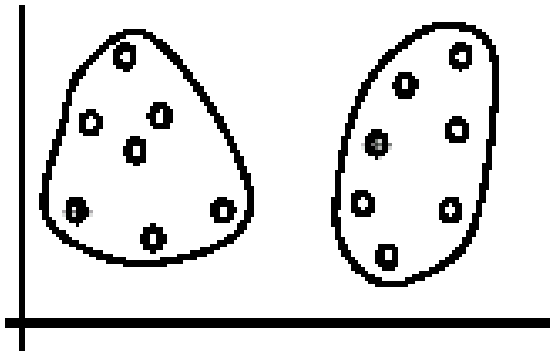
Weaknesses of k-means (cont …)

- If we use different seeds: good results

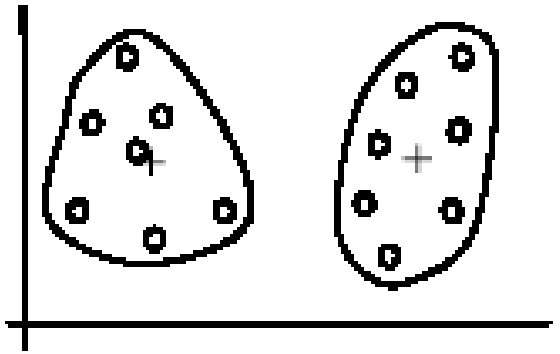


There are some methods to help choose good seeds

(A). Random selection of k seeds (centroids)



(B). Iteration 1



(C). Iteration 2

Classification / Supervised Learning

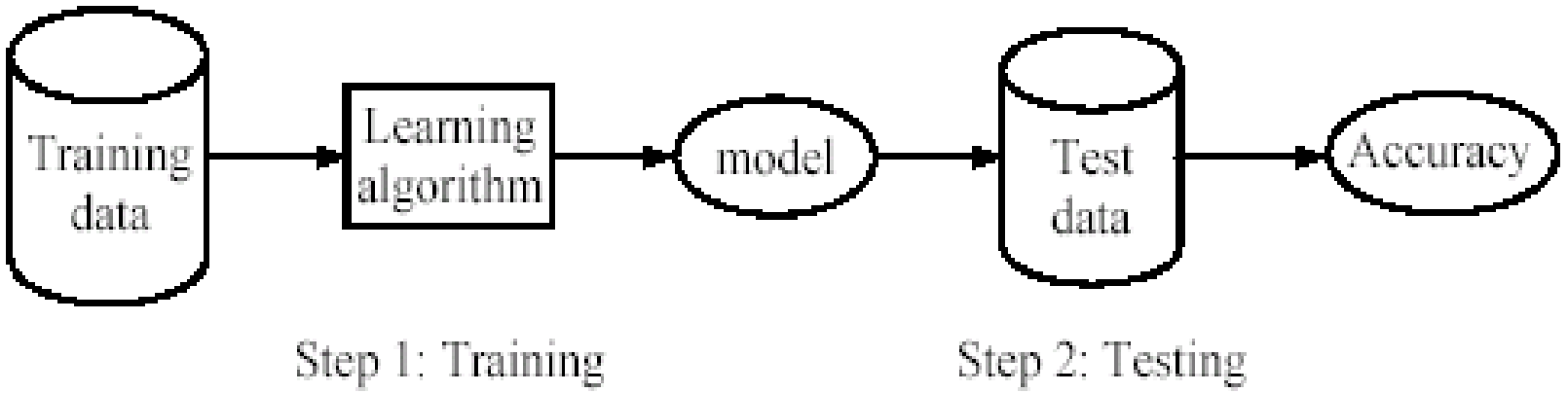
Classification: Definition

- Given a collection of records (*training set*)
  - Each record contains a set of *attributes*, one of the attributes is the *class*.
- Find a *model* for class attribute as a function of the values of other attributes.
- Goal: previously unseen records should be assigned a class as accurately as possible.
  - A *test set* is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

Supervised learning process: two steps

- Learning (training): Learn a model using the training data
- Testing: Test the model using unseen test data to assess the model accuracy

$$Accuracy = \frac{\text{Number of correct classifications}}{\text{Total number of test cases}},$$



An example application

- An emergency room in a hospital measures 17 variables (e.g., blood pressure, age, etc) of newly admitted patients.
- A decision is needed: whether to put a new patient in an intensive-care unit.
- Due to the high cost of ICU (Intensive Care Unit), those patients who may survive less than a month are given higher priority.
- Problem: to predict high-risk patients and discriminate them from low-risk patients.

Another application

- A credit card company receives thousands of applications for new cards. Each application contains information about an applicant,
  - age
  - Marital status
  - annual salary
  - outstanding debts
  - credit rating
  - etc.
- Problem: to decide whether an application should approved, or to classify applications into two categories, approved and not approved.

Machine learning and our focus

- Like human learning from past experiences.
- A computer does not have “experiences”.
- A computer system learns from data, which represent some “past experiences” of an application domain.
- Our focus: learn a target function that can be used to predict the values of a discrete class attribute, e.g., approve or not-approved, and high-risk or low risk.
- The task is commonly called: Supervised learning, classification, or inductive learning.

The data and the goal

- Data: A set of data records (also called examples, instances or cases) described by
  - *k* attributes: A_1, A_2, … A_k.
  - a class: Each example is labelled with a pre-defined class.
- Goal: To learn a classification model from the data that can be used to predict the classes of new (future, or test) cases/instances.

An example: data (loan application)

Approved or not

ID	Age	Has_Job	Own_House	Credit_Rating	Class
1	young	false	false	fair	**No**
2	young	false	false	good	**No**
3	young	true	false	good	**Yes**
4	young	true	true	fair	**Yes**
5	young	false	false	fair	**No**
6	middle	false	false	fair	**No**
7	middle	false	false	good	**No**
8	middle	true	true	good	**Yes**
9	middle	false	true	excellent	**Yes**
10	middle	false	true	excellent	**Yes**
11	old	false	true	excellent	**Yes**
12	old	false	true	good	**Yes**
13	old	true	false	good	**Yes**
14	old	true	false	excellent	**Yes**
15	old	false	false	fair	**No**

An example: the learning task

- Learn a classification model from the data
- Use the model to classify future loan applications into
  - Yes (approved) and
  - No (not approved)
- What is the class for following case/instance?

Age	Has_Job	Own_house	Credit-Rating	Class
young	false	false	good	?

Naïve Bayesian classification

- Probabilistic view: Supervised learning can naturally be studied from a probabilistic point of view.
- Let A_1 through A_k be attributes with discrete values. The class is C.
- Given a test example *d* with observed attribute values a_1 through a_k.
- Classification is basically to compute the following posteriori probability. The prediction is the class c_j such that

$$\Pr(C = c_j \mid A_1 = a_1, \ldots, A_{|A|} = a_{|A|})$$

is maximal

Apply Bayes' Rule

$$\Pr(C = c_j \mid A_1 = a_1, \ldots, A_{|A|} = a_{|A|})$$

$$= \frac{\Pr(A_1 = a_1, \ldots, A_{|A|} = a_{|A|} \mid C = c_j)\Pr(C = c_j)}{\Pr(A_1 = a_1, \ldots, A_{|A|} = a_{|A|})}$$

$$= \frac{\Pr(A_1 = a_1, \ldots, A_{|A|} = a_{|A|} \mid C = c_j)\Pr(C = c_j)}{\sum_{r=1}^{|C|} \Pr(A_1 = a_1, \ldots, A_{|A|} = a_{|A|} \mid C = c_r)\Pr(C = c_r)}$$

- $\Pr(C=c_j)$ is the class *prior* probability: easy to estimate from the training data.

Computing probabilities

- The denominator $P(A_1=a_1,...,A_k=a_k)$ is irrelevant for decision making since it is the same for every class.
- We only need $P(A_1=a_1,...,A_k=a_k \mid C=c_i)$, which can be written as

  $Pr(A_1=a_1|A_2=a_2,...,A_k=a_k, C=c_j)* Pr(A_2=a_2,...,A_k=a_k |C=c_j)$
- Recursively, the second factor above can be written in the same way, and so on.
- Now an assumption is needed.

Conditional independence assumption

- All attributes are conditionally independent given the class $C = c_j$.
- Formally, we assume,

  $\Pr(A_1=a_1 \mid A_2=a_2, ..., A_{|A|}=a_{|A|}, C=c_j) = \Pr(A_1=a_1 \mid C=c_j)$

  and so on for A_2 through $A_{|A|}$. I.e.,

$$\Pr(A_1 = a_1,..., A_{|A|} = a_{|A|} \mid C = c_i) = \prod_{i=1}^{|A|} \Pr(A_i = a_i \mid C = c_j)$$

Final naïve Bayesian classifier

$$\Pr(C = c_j \mid A_1 = a_1, \ldots, A_{|A|} = a_{|A|})$$

$$= \frac{\Pr(C = c_j) \prod_{i=1}^{|A|} \Pr(A_i = a_i \mid C = c_j)}{\sum_{r=1}^{|C|} \Pr(C = c_r) \prod_{i=1}^{|A|} \Pr(A_i = a_i \mid C = c_r)}$$

- We are done!
- How do we estimate $P(A_i = a_i | C=c_j)$? Easy!.

Classify a test instance

- If we only need a decision on the most probable class for the test instance, we only need the numerator as its denominator is the same for every class.
- Thus, given a test example, we compute the following to decide the most probable class for the test instance

$$c = \arg\max_{c_j} \Pr(c_j) \prod_{i=1}^{|A|} \Pr(A_i = a_i \mid C = c_j)$$

An example

- Compute all probabilities required for classification

A	B	C
m	b	t
m	s	t
g	q	t
h	s	t
g	q	t
g	q	f
g	s	f
h	b	f
h	q	f
m	b	f

$Pr(C = t) = 1/2$, $Pr(C = f) = 1/2$

$Pr(A=m \mid C=t) = 2/5$
$Pr(A=m \mid C=f) = 1/5$
$Pr(B=b \mid C=t) = 1/5$
$Pr(B=b \mid C=f) = 2/5$

$Pr(A=g \mid C=t) = 2/5$
$Pr(A=g \mid C=f) = 2/5$
$Pr(B=s \mid C=t) = 2/5$
$Pr(B=s \mid C=f) = 1/5$

$Pr(A=h \mid C=t) = 1/5$
$Pr(A=h \mid C=n) = 2/5$
$Pr(B=q \mid C=t) = 2/5$
$Pr(B=q \mid C=f) = 2/5$

Now we have a test example:

$A = m \quad B = q \quad C = ?$

An Example (cont …)

- For C = t, we have

$$\Pr(C=t)\prod_{j=1}^{2}\Pr(A_j=a_j\,|\,C=t)=\frac{1}{2}\times\frac{2}{5}\times\frac{2}{5}=\frac{2}{25}$$

- For class C = f, we have

$$\Pr(C=f)\prod_{j=1}^{2}\Pr(A_j=a_j\,|\,C=f)=\frac{1}{2}\times\frac{1}{5}\times\frac{2}{5}=\frac{1}{25}$$

- C = t is more probable. t is the final class.

On naïve Bayesian classifier

- Advantages:
  - Easy to implement
  - Very efficient
  - Good results obtained in many applications
- Disadvantages
  - Assumption: class conditional independence, therefore loss of accuracy when the assumption is seriously violated (those highly correlated data sets)

SVM

- Decision tree
- Support vector machine
  - SVMs are linear classifiers that find a hyperplane to separate two class of data, positive and negative.
  - perhaps the best classifier for text classification.
- k-Nearest Neighbor Classification (kNN)
- Factor graph

- Ensemble methods:
  - Combining multiple classifiers to produce a better one

Text Mining

- Data mining on text
  - Due to online texts on the Web and other sources
  - Text contains a huge amount of information of almost any imaginable type!
  - A major direction and tremendous opportunity!
- Main topics
  - Text classification and clustering
    - cluster Web pages to find related pages
    - cluster pages a user has visited to organize their visit history
    - classify Web pages automatically into a Web directory
  - Information retrieval
  - Information extraction
  - Opinion mining

Social network analysis

- Social network is the study of social entities (people in an organization, called **actors**), and their interactions and relationships.
- The interactions and relationships can be represented with a network or graph,
  - each vertex (or node) represents an actor and
  - each link represents a relationship.
- From the network, we can study the properties of its structure, and the role, position and prestige of each social actor.
- We can also find various kinds of sub-graphs, e.g., **communities** formed by groups of actors.

SNA & one SNA application: Web

- Social network analysis is useful for the Web because the Web is essentially a virtual society, and thus a virtual social network,
  - Each page: a social actor and
  - each hyperlink: a relationship.
- Many results from social network can be adapted and extended for use in the Web context.
- We study two types of social network analysis, **centrality** and **prestige**, which are closely related to hyperlink analysis and search on the Web.

Centrality

- Important or prominent actors are those that are linked or involved with other actors extensively.
- A person with extensive contacts (links) or communications with many other people in the organization is considered more important than a person with relatively fewer contacts.
- The links can also be called **ties**. A central actor is one involved in many ties.

Degree Centrality

Central actors are the most active actors that have most links or ties with other actors. Let the total number of actors in the network be n.

Undirected graph: In an undirected graph, the **degree centrality** of an actor i (denoted by $C_D(i)$) is simply the node degree (the number of edges) of the actor node, denoted by $d(i)$, normalized with the maximum degree, n-1.

$$C_D(i) = \frac{d(i)}{n-1} \qquad (1)$$

Directed graph: In this case, we need to distinguish **in-links** of actor i (links pointing to i), and **out-links** (links pointing out from i). The degree centrality is defined based on only the out-degree (the number of out-links or edges), $d_o(i)$.

$$C_D'(i) = \frac{d_o(i)}{n-1} \qquad (2)$$

Closeness Centrality

This view of centrality is based on the closeness or distance. The basic idea is that an actor x_i is central if it can easily interact with all other actors. That is, its distance to all other actors is short. Thus, we can use the shortest distance to compute this measure. Let the shortest distance from actor i to actor j be $d(i, j)$.

Undirected graph: The closeness centrality $C_C(i)$ of actor i is defined as

$$C_C(i) = \frac{n-1}{\sum_{j=1}^{n} d(i,j)} \qquad (3)$$

The value of this measure also ranges between 0 and 1 as n-1 is the minimum value of the denominator, which is the sum of shortest distances from i to all other actors. Note that this equation is only meaningful for a connected graph.

Directed graph: The same equation can be used for a directed graph. The distance computation needs to consider directions of links or edges.

Betweenness Centrality

- If two non-adjacent actors *j* and *k* want to interact and actor *i* is on the path between *j* and *k*, then *i* may have some control over the interactions between *j* and *k*.
- Betweenness measures this control of *i* over other pairs of actors. Thus,
  - if *i* is on the paths of many such interactions, then *i* is an important actor.

Betweenness Centrality (cont …)

- **Undirected graph**: Let p_{jk} be the number of shortest paths between actor *j* and actor *k*.
- The betweenness of an actor *i* is defined as the number of shortest paths that pass *i* ($p_{jk}(i)$) normalized by the total number of shortest paths.

$$\sum_{j<k} \frac{p_{jk}(i)}{p_{jk}} \qquad (4)$$

Betweenness Centrality (cont …)

Note that there may be multiple shortest paths between j and k. Some passes i and some do not. If we are to ensure the value range is between 0 and 1, we can normalize it with $(n\text{-}1)(n\text{-}2)/2$, which is the maximum value of the above quantity, i.e., the number of pairs of actors not including i. The final betweenness of i is defined as

$$C_B(i) = \frac{2\sum_{j<k} \frac{p_{jk}(i)}{p_{jk}}}{(n-1)(n-2)} \qquad (5)$$

Unlike the closeness measure, the betweenness can be computed even if the graph is not connected.

Directed graph: The same equation can be used but must be multiplied by 2 because there are now $(n\text{-}1)(n\text{-}2)$ pairs considering a path from j to k is different from a path from k to j. Likewise, p_{jk} must consider paths from both directions.

Prestige

- Prestige is a more refined measure of prominence of an actor than centrality.
  - Distinguish: ties sent (out-links) and ties received (in-links).
- A prestigious actor is one who is object of extensive ties as a recipient.
  - To compute the prestige: we use only in-links.
- Difference between centrality and prestige:
  - centrality focuses on out-links
  - prestige focuses on in-links.
- We study three prestige measures. **Rank prestige** forms the basis of most Web page link analysis algorithms, including PageRank and HITS.

Degree prestige

Based on the definition of the prestige, it is clear that an actor is prestigious if it receives many in-links or nominations. Thus, the simplest measure of prestige of an actor i (denoted by $P_D(i)$) is its in-degree.

$$P_D(i) = \frac{d_I(i)}{n-1}, \tag{6}$$

where $d_I(i)$ is in-degree of i (the number of in-links of actor i_i) and n is the total number of actors in the network. As in the degree centrality, dividing $n-1$ standardizes the prestige value to the range from 0 and 1. The maximum prestige value is 1 when every other actor links to or chooses actor i.

Proximity prestige

- The degree index of prestige of an actor *i* only considers the actors that are adjacent to *i*.
- The proximity prestige generalizes it by considering both the actors directly and indirectly linked to actor *i*.
  - We consider every actor *j* that can reach *i*.
- Let I_i be the set of actors that can reach actor *i*.
- The **proximity** is defined as closeness or distance of other actors to *i*.
- Let *d*(*j*, *i*) denote the distance from actor *j* to actor *i*.

Proximity prestige (cont …)

$$\frac{\sum_{j \in I_i} d(j,i)}{|I_i|}, \tag{7}$$

where $|I_i|$ is the size of the set I_i. If we look at the ratio or proportion of actors who can reach i to the average distance that these actors are from i, we obtain the following, which has the value range of [0, 1]:

$$P_P(i) = \frac{|I_i|/(n-1)}{\sum_{j \in I_i} d(j,i) / |I_i|}, \tag{8}$$

where $|I_i|/(n\text{-}1)$ is the proportion of actors that can reach actor i. In one extreme, every actor can reach actor i, which gives $|I_i|/(n\text{-}1) = 1$. The denominator is 1 if every actor is adjacent to i. Thus, $P_P(i) = 1$. On the other extreme, no actor can reach actor i. Then $|I_i| = 0$, and $P_P(i) = 0$. Each link has the unit distance.

Rank prestige

- In the previous two prestige measures, an important factor is considered,
  - the **prominence** of individual actors who do the “voting”
- In the real world, a person *i* chosen by an important person is more prestigious than chosen by a less important person.
  - For example, if a company CEO votes for a person is much more important than a worker votes for the person.
- If one’s circle of influence is full of prestigious actors, then one’s own prestige is also high.
  - Thus one’s prestige is affected by the ranks or statuses of the involved actors.

Rank prestige (cont …)

- Based on this intuition, the rank prestige $P_R(i)$ is define as a linear combination of links that point to i:

$$P_R(i) = A_{1i}P_R(1) + A_{2i}P_R(2) + \ldots + A_{ni}P_R(n), \quad (9)$$

where $A_{ji} = 1$ if j points to i, and 0 otherwise. This equation says that an actor's rank prestige is a function of the ranks of the actors who vote or choose the actor, which makes perfect sense.

Since we have n equations for n actors, mathematically we can write them in the matrix notation. We use $\boldsymbol{P}$ to represent the vector that contains all the rank prestige values, i.e., $\boldsymbol{P} = (P_R(1), P_R(2), \ldots, P_R(n))^T$ (T means **matrix transpose**). $\boldsymbol{P}$ is represented as a column vector. We use matrix $\boldsymbol{A}$ (where $A_{ij} = 1$ if i points to j, and 0 otherwise) to represent the adjacency matrix of the network or graph. As a notational convention, we use bold italic letters to represent matrices. We then have

$$\boldsymbol{P} = \boldsymbol{A}^T \boldsymbol{P} \quad (10)$$

This equation is precisely the characteristic equation used for finding the **eigensystem** of the matrix $\boldsymbol{A}^T$. $\boldsymbol{P}$ is an **eigenvector** of $\boldsymbol{A}^T$.

Concluding Remarks

- Data analytics basis
- Mining knowledge from (big) data
- Mining social structure from the actors embedded in the data
- Often, highly computational (algorithmic)-intensive and memory intensive
  - May require high-capacity cloud/server platform (MapReduce, Hadoop etc.)