Text Mining

Dr. Yanjun Li

Associate Professor

Department of Computer and Information Sciences

Fordham University

Outline

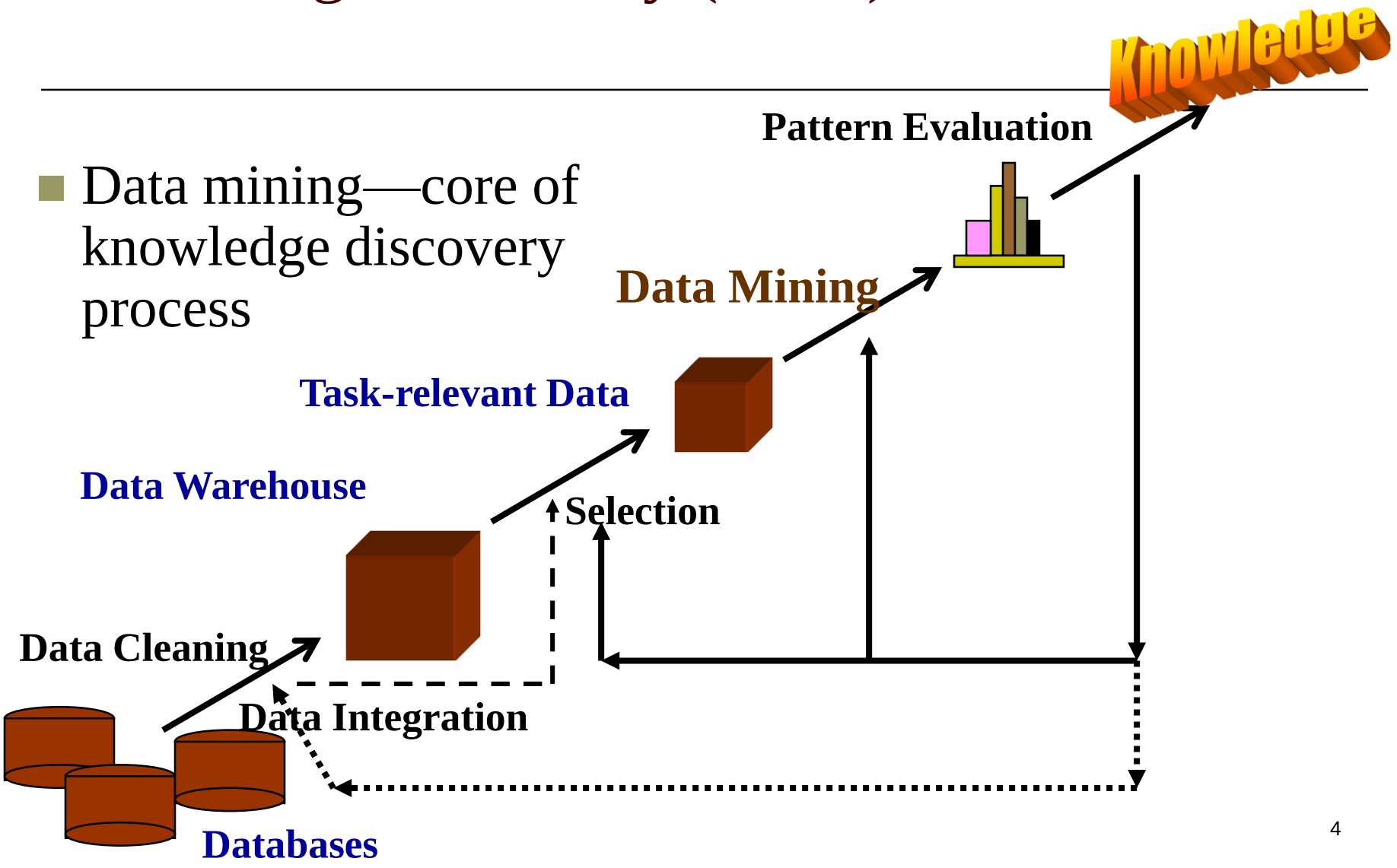
- Introduction: Data Mining
- Part One: Text Mining
- Part Two: Preprocessing Text Data
- Part Three: Feature Selection for Text Data
- Part Four: Text Mining Algorithms
- Part Five: Software Tools for Text Mining

Introduction: Data Mining

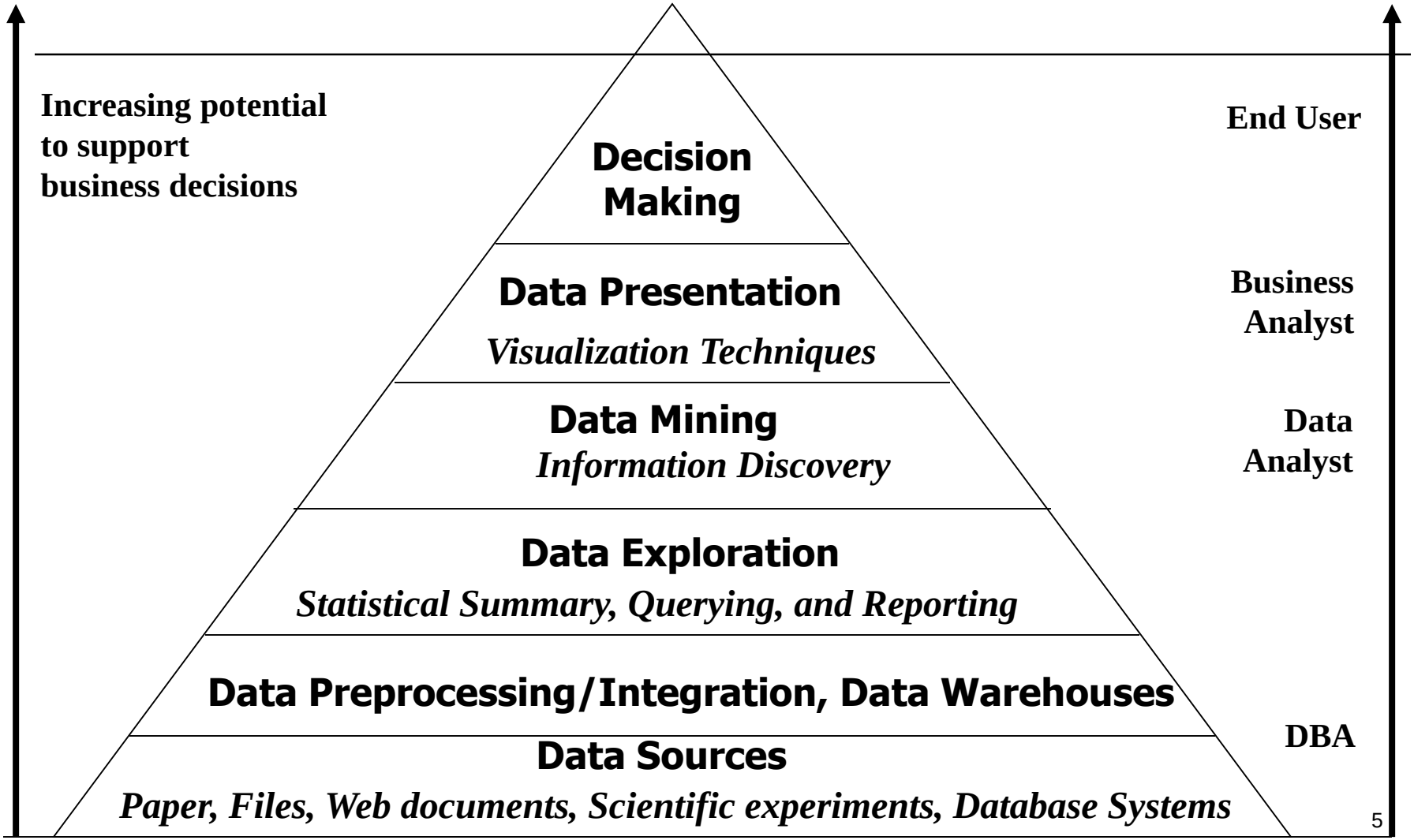
- Also known as **KDD** - ***K****nowledge* ***D****iscovery from* ***D****atabases*
  - Data mining : Knowledge mining from data.
  - Extraction of interesting (non-trivial, implicit, previously unknown and potentially useful) patterns or knowledge from huge amount of data.

Knowledge Discovery (KDD) Process

- Data mining—core of knowledge discovery process



Data Mining in Business Intelligence



Data Mining Functionalities

- Class Description
- Association Analysis
- Classification and Prediction
- Cluster Analysis
- Outlier Analysis
- Trend and Evolution Analysis

Part One: Text Mining

- The process of analyzing text to extract information that is useful for particular purposes.
- The information to be extracted is clearly and explicitly in the text.
- Automatic processing text data sets is a challenge for Text mining research.

Text Mining

- Text Classification
- Text Clustering
- Text Summarization
- Text Retrieval

Challenge of Text Mining Research

- Exploring features of text documents
  - Document representation
  - Documents similarity measurement
  - Reduction of high dimension
- Development of new high performance text mining algorithms which target the unique features of text documents.

Part Two: Preprocessing Text Data

- Text data is unstructured, amorphous, and difficult to deal with algorithmically.
- Document representation:
  - Vector space model: **a bag of words**.
  - Models with word sequence information
    - Generalized Suffix Tree

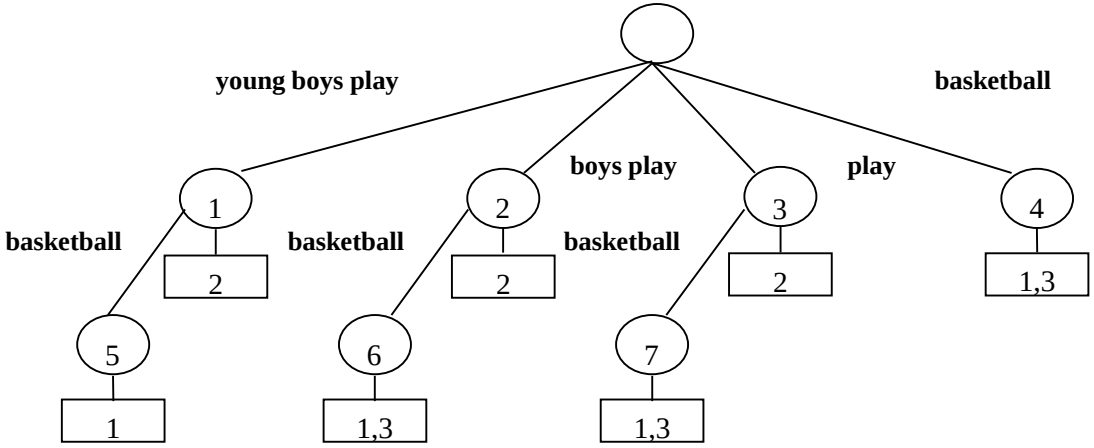
Basic Preprocessing Techniques

- Tokenization
- Stopwords removal
- Stemmer
- Weight of words: TF/IDF
- Normalization

Alternative Document Representation

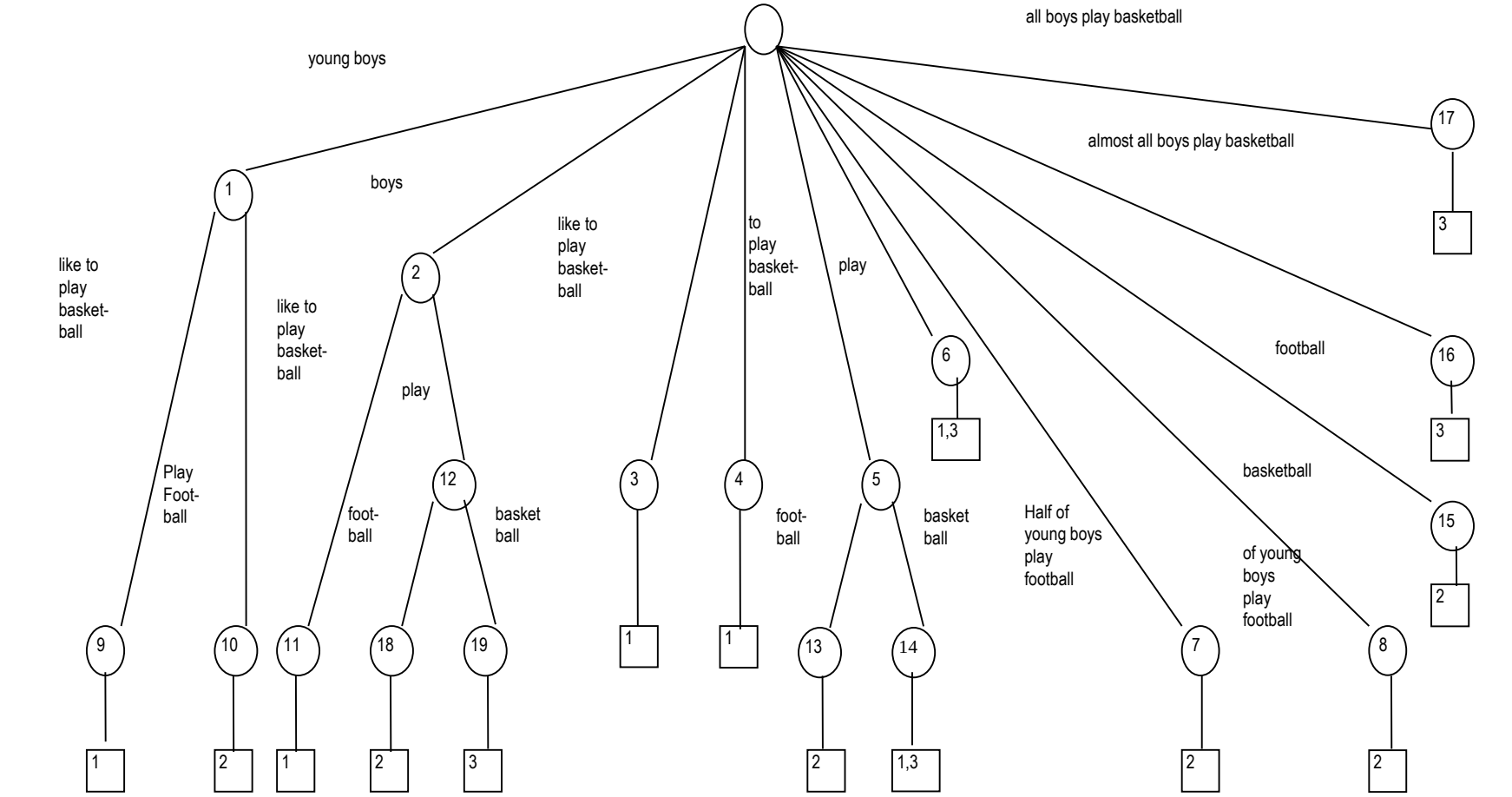
- The document model better preserves the sequential relationship between words in the document since the accurate meaning of a sentence has close relationship with it.
- Word meanings are better than word forms to represent the topics of documents. Ontology should be used.

Generalized Suffix Tree



Node No.	Word Sequence	Length of Word Seq.	Document Id Set	Number of Document Ids
1	**young boys play**	**3**	**1,2**	**2**
2	**boys play**	**2**	**1,2,3**	**3**
3	*play*	*1*	*1,2,3*	*3*
4	*basketball*	*1*	*1,3*	*2*
5	*young boys play basketball*	*4*	*1*	*1*
6	**boys play basketball**	**3**	**1,3**	**2**
7	**play basketball**	**2**	**1,3**	**2**

GST of the Original Documents



Investigating Word Meaning with WordNet

WordNet is an on-line lexical reference system hosted by Princeton University.

- English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept.
- Illustration of the concept of a Lexical Matrix:

Word Meanings	Word Forms			
	F1	F2	…..	Fn
M1	E1,1	E1,2		
M2		E2,2		
M3				
:			…..	
:				
Mm				Em,n

Preprocessing by Using the WordNet

- For each word form, retrieve multiple synsets (synonym sets) and hypernyms from the WordNet to build ***meaning unions*** (MU) with ***synset links*** (SLs).
  - For a word form “box”:
    - MU = {SS_1-> SS_2, SS_3->SS_4};
    - SS_1 = {box}, SS_2 = {container};
    - SS_3 = {box, loge}, SS_4 = {compartment};
- A document becomes d'=<MU_1, MU_2, MU_3, …>, which is treated as **a string of word meanings**.

Part Three: Feature Selection for Text Data

- Feature selection removes **irrelevant** or **redundant** features. The selected feature subset contains **more sufficient** or **reliable** information about the original data set. At the same time, it **reduces the dimensions** of the database.
- Widely used methods
  - **Supervised** : Information Gain (IG), $\chi 2$ Statistic (CHI).
  - **Unsupervised** : Document Frequency (DF), Term Strength (TS).

Attribute Selection Method: Information Gain

- Select the attribute with the ***highest*** information gain
  - This attribute minimizes the information needed to classify the tuples in the resulting partitions.
- Let p_i be the probability that an arbitrary tuple in D belongs to class C_i, estimated by $|C_{i,D}|/|D|$
- Expected information (entropy) needed to classify a tuple in D:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

  - Entropy represents the average amount of information needed to identify the class label of a tuple in D.

Supervised Feature Selection Method - CHI

- χ^2 term-category independency test:

$$E(i,j) = \frac{\sum_{a\in\{w,\neg w\}} O(a,j) \times \sum_{b\in\{c,\neg c\}} O(i,b)}{N}$$

$$\chi^2_{w,c} = \sum_{i\in\{w,\neg w\}} \sum_{j\in\{c,\neg c\}} \frac{(O(i,j) - E(i,j))^2}{E(i,j)}$$

	c	$\neg c$	Σ
w	40	80	120
$\neg w$	60	320	380
Σ	100	400	500

A 2-way contingency table

$\chi^2_{w,c} = 17.61$

- By ranking $\chi 2$ statistic, CHI method selects features that have strong dependency on the categories.

For m-ary classifier : $\chi^2_{\max}(w) = \max_{k=1}^{m} \{\chi^2(w, c_k)\}$

Part Four: Text Mining Algorithms

- Text Classification
  - Naïve Bayes: Benchmark
  - Support Vector Machine
- Text Clustering
  - K-means

Text Classification

- Supervised learning
  - Supervision: The training data (observations, measurements, etc.) for model construction are accompanied by labels indicating the class of the observations
  - The model is represented as classification rules, decision trees, or mathematical formulae
  - New data is classified based on the model

Bayesian Theorem: Basics

- Let **X** be a data sample (*evidence*) with class label unknown
- Let H be a *hypothesis* that X belongs to class C
- Classification is to determine P(H|**X**), (*posteriori probability*), the probability that the hypothesis holds given the observed data sample **X**
- P(H) (*prior probability*), the initial probability, which is independent of **X**.
  - E.g., **A customer** will buy computer, regardless of age, income, …
- P(**X**): probability that sample data is observed
- P(**X**|H) (*likelihood*), the probability of observing the sample **X**, given that the hypothesis holds
  - E.g., Given that **X** will buy computer, the prob. that X' age is 31..40, medium income

Bayesian Theorem

- Given training data **X**, *posteriori probability of a hypothesis* H, P(H|**X**), follows the **Bayes' theorem**

$$P(H \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid H) P(H)}{P(\mathbf{X})}$$

- Informally, this can be written as

  posteriori = likelihood x prior/evidence

- Predicts X belongs to Ci iff the probability P(Ci|X) is the highest among all the $P(C_k|X)$ for all the k classes

Towards Naïve Bayesian Classifier

- Let D be a training set of tuples and their associated class labels, and each tuple is represented by an n-D attribute vector $\mathbf{X} = (x_1, x_2, \ldots, x_n)$
- Suppose there are *m* classes $C_1, C_2, \ldots, C_m$.
- Classification is to derive the maximum posteriori, i.e., the maximal $P(C_i|\mathbf{X})$
- This can be derived from Bayes' theorem

$$P(C_i|\mathbf{X}) = \frac{P(\mathbf{X}|C_i)P(C_i)}{P(\mathbf{X})}$$

- Since P(X) is constant for all classes, only needs to be maximized

$$P(C_i|\mathbf{X}) = P(\mathbf{X}|C_i)P(C_i)$$

Derivation of Naïve Bayes Classifier

- **Naïve**:
  - A simplified assumption: attributes are conditionally independent (i.e., no dependence relation between attributes):

$$P(\mathbf{X}|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times ... \times P(x_n|C_i)$$

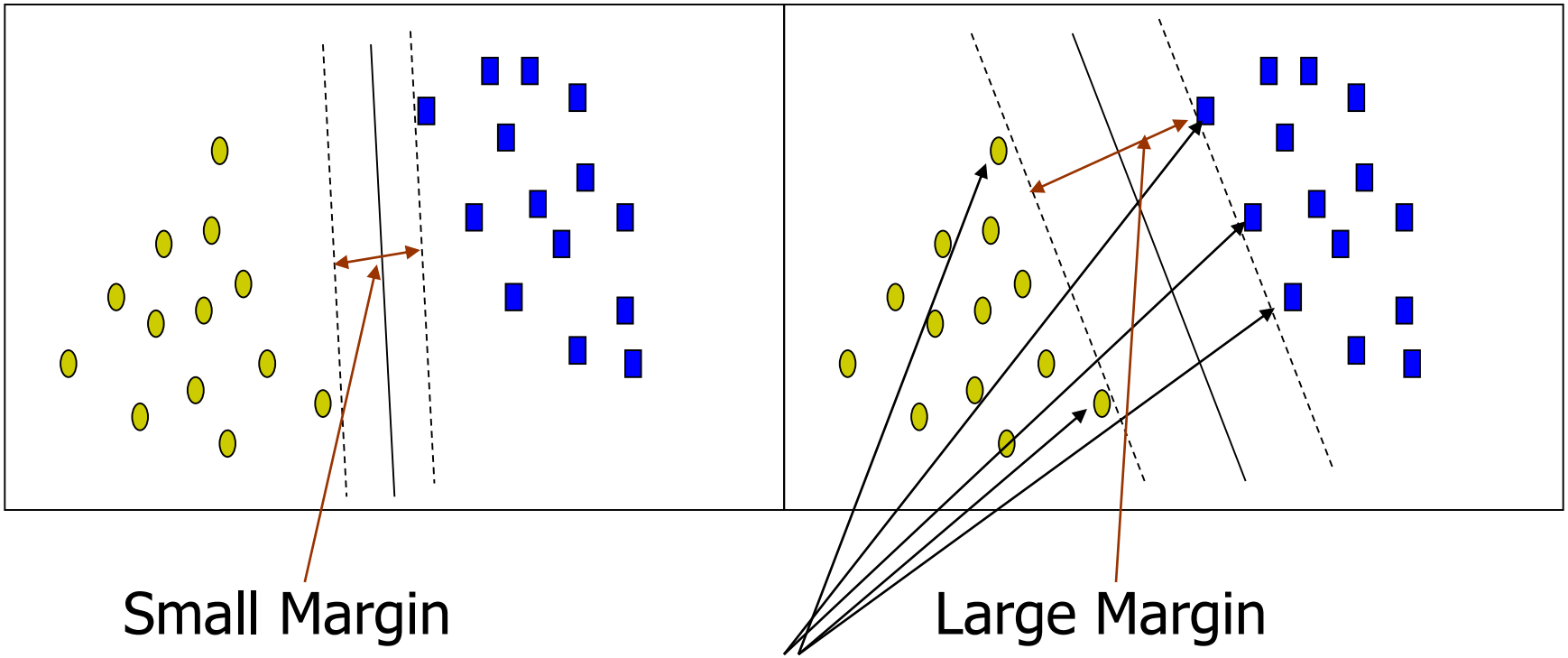
- This greatly reduces the computation cost: Only counts the class distribution
- If Attribute A_k is categorical, $P(x_k|C_i)$ is the # of tuples in C_i having value x_k for A_k divided by $|C_{i,D}|$ (# of tuples of C_i in D)
- If A_k is continous-valued, $P(x_k|C_i)$ is usually computed based on Gaussian distribution with a mean μ and standard deviation σ: probability density function $g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$

and $P(x_k|C_i)$ is $P(\mathbf{X}|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})$

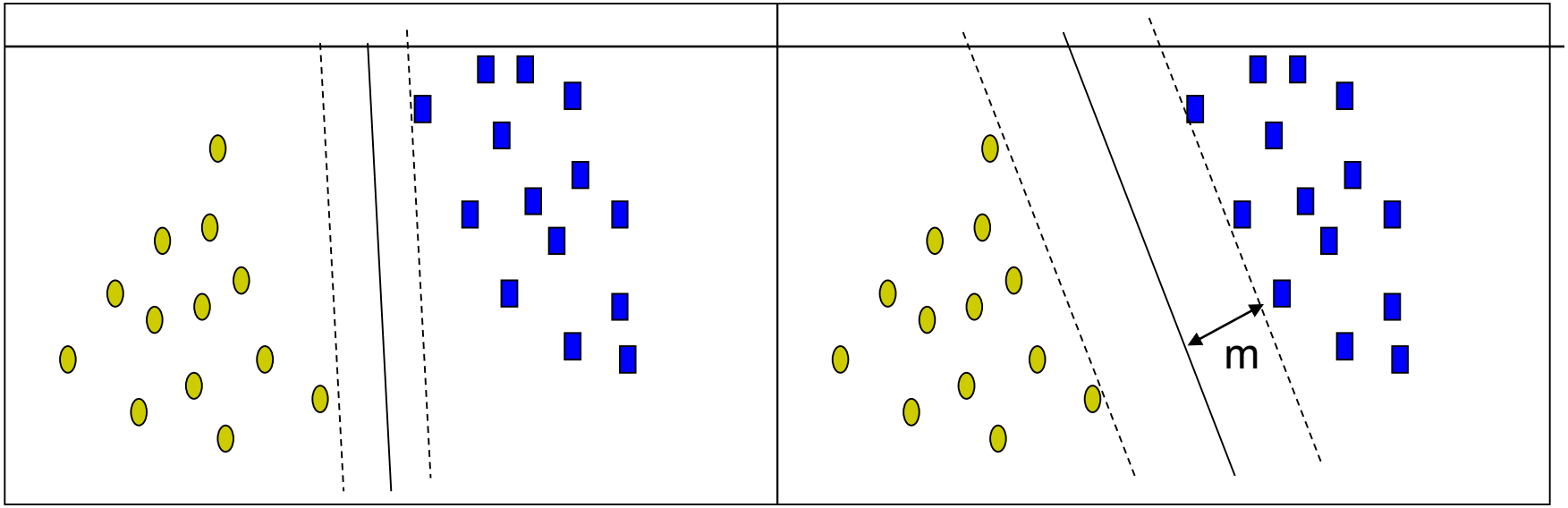
SVM—Support Vector Machines

- A new classification method for both linear and nonlinear data
- It uses a nonlinear mapping to transform the original training data into a higher dimension
- With the new dimension, it searches for the linear optimal separating hyperplane (i.e., “decision boundary”)
- With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated by a hyperplane
- SVM finds this hyperplane using support vectors (“essential” training tuples) and margins (defined by the support vectors)

SVM—General Philosophy



SVM—When Data Is Linearly Separable



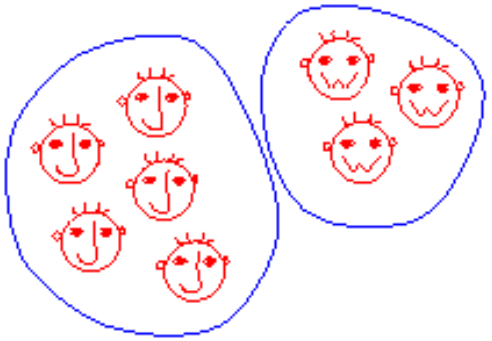
Let data D be ($\mathbf{X}_1$, y_1), …, ($\mathbf{X}_{|D|}$, $y_{|D|}$), where $\mathbf{X}_i$ is the set of training tuples associated with the class labels y_i

There are infinite lines (hyperplanes) separating the two classes but we want to find the best one (the one that minimizes classification error on unseen data)

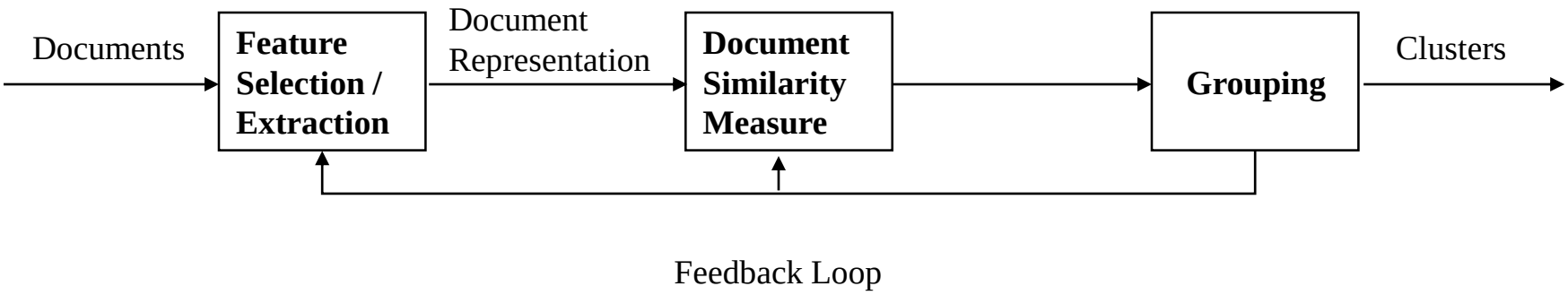
SVM searches for the hyperplane with the largest margin, i.e., **maximum marginal hyperplane** (MMH)

Text Clustering

- Text clustering is known as **unsupervised**, **automatic grouping** of text documents into conceptually **meaningful** clusters, so that documents within a cluster have a **high similarity** among them, but they are **dissimilar** to documents in other clusters.
- It is different from text classification because of the lack of labeled documents for training.



Text Clustering



- Typical text clustering activity involves these three steps.
- Cluster validity analysis
  - **Internal** Quality Measure : Overall Similarity
  - **External** Quality Measure : F-measure, Entropy, Purity

The *K-Means* Clustering Method

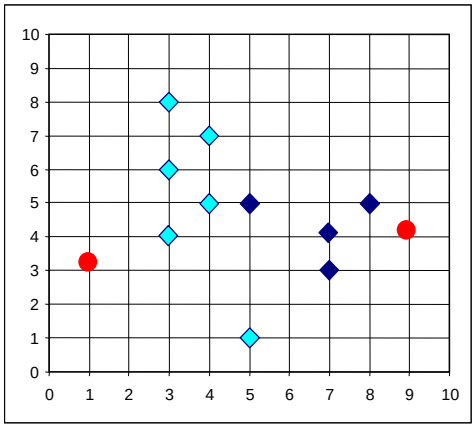
- Centroid of a cluster for numerical values: the mean value of all objects in a cluster

$$C_m = \frac{\sum_{i=1}^{N} (t_{ip})}{N}$$

- Given k, the k-means algorithm is implemented in four steps:
  1. Select *k* seed points from D as the initial centroids.
  2. Assigning:
     - Assign each object of D to the cluster with the nearest centroid.
  3. Updating:
     - Compute centroids of the clusters of the current partition.
  4. Go back to Step 2 and continue, stop when no more new assignment.

The *K-Means* Clustering Method

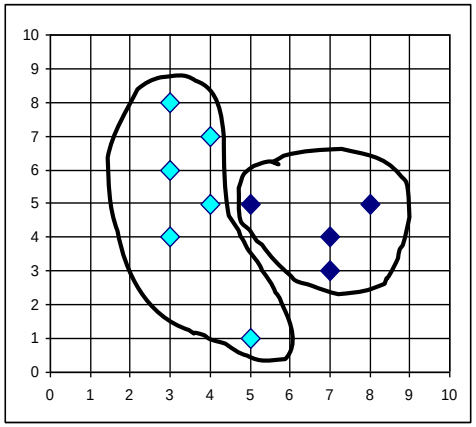
- Example



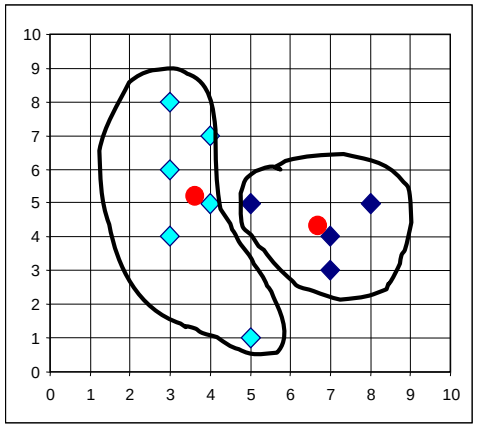
K=2

Arbitrarily choose K object as initial cluster center

Assign each objects to most similar center

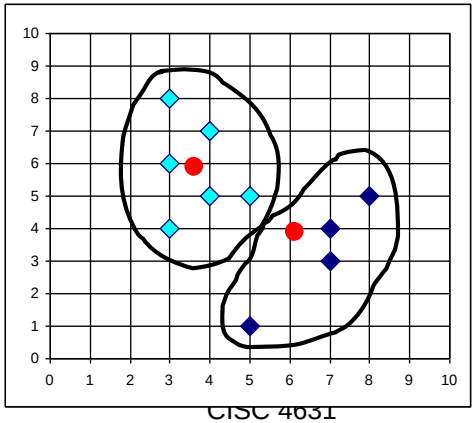


Update the cluster means

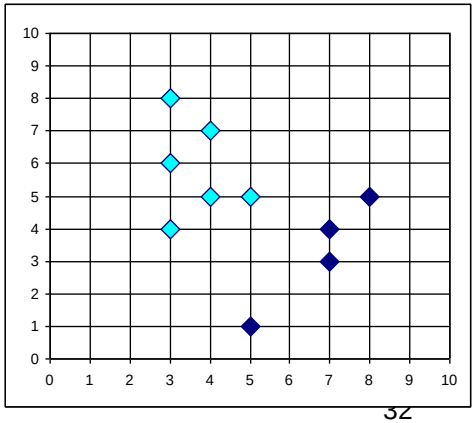


reassign

reassign



Update the cluster means



Cosine Similarity

- Terms in document are represented as Vector objects.
- Cosine measure: If d_1 and d_2 are two 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:

  d_1 = 3 2 0 5 0 0 0 2 0 0

  d_2 = 1 0 0 0 0 0 0 1 0 2

  $d_1 \bullet d_2 = 3*1+2*0+0*0+5*0+0*0+0*0+0*0+2*1+0*0+0*2 = 5$

  $\|d_1\| = (3*3+2*2+0*0+5*5+0*0+0*0+0*0+2*2+0*0+0*0)^{0.5} = (42)^{0.5} = 6.481$

  $\|d_2\| = (1*1+0*0+0*0+0*0+0*0+0*0+0*0+1*1+0*0+2*2)^{0.5} = (6)^{0.5} = 2.245$

  $\cos(d_1, d_2) = .3150$

Cosine Function May Not Work

- A document may have a small subset of terms from a large vocabulary of the topic. For example, topic and subtopic; Synonyms usage.
- The vocabulary sizes for different topics are different. A cluster with a large vocabulary will be forced to split to increase the value of criterion function.

$$\cos(d_i, d_j) = \frac{d_i^t d_j}{\| d_i \| \| d_j \|} = d_i^t d_j$$

Neighbor Matrix

- The ***neighbors*** of a document *d* are those considered similar to it.

$$sim(d_i, d_j) \geq \theta, \theta \in [0,1]$$

- Neighbor Matrix:
  - *n* x *n* adjacency matrix *M* →
- ***Link*** is the number of *common neighbors* between two document d_i and d_j.

$$link(d_i, d_j) = \sum_{m=1}^{n} (M[i,m] * M[m,j])$$

	d_1	d_2	d_3	d_4	d_5	d_6
d_1	1	1	0	1	0	0
d_2	1	1	0	1	0	0
d_3	0	0	1	1	1	0
d_4	1	1	1	1	0	0
d_5	0	0	1	0	1	0
d_6	0	0	0	0	0	1

Similarity Measurement Function – CL (*Cosine* and *link*)

- Global information given by the neighbor matrix can be combined with the pair-wise similarity measurement (*cosine* function) to solve those problems.

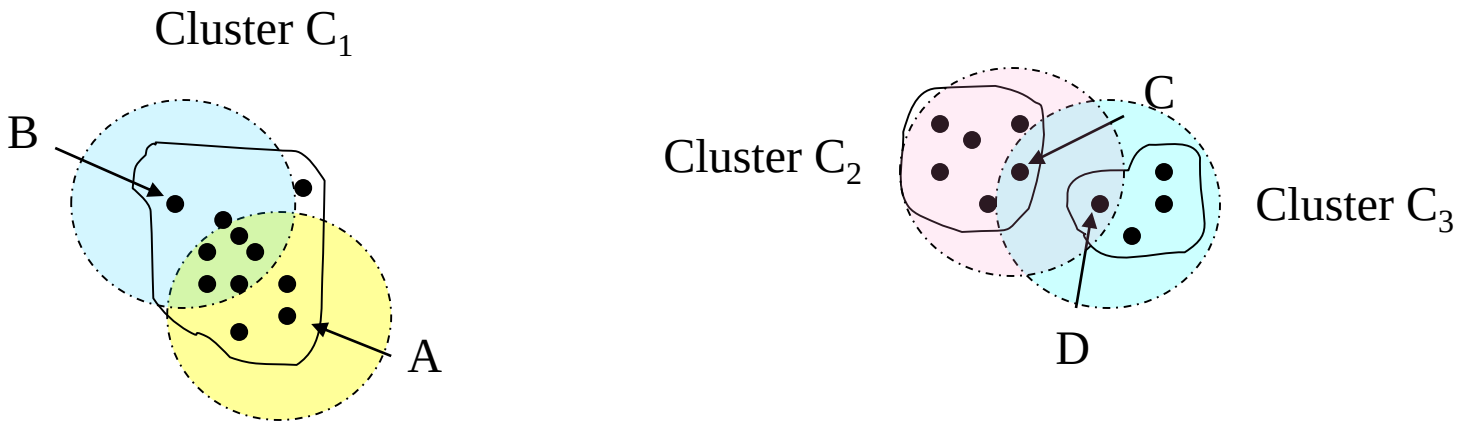
$$link(d_i, c_j) = \sum_{m=1}^{n} M'[i,m] * M^{'}[m, n+j]$$

	d_1	d…	d_n	c_1	c…	c_k
d_1	1	1	0	1	0	0
d_2	1	1	0	1	0	0
d_3	0	0	1	1	1	0
d_4	1	1	1	1	0	0
d…	0	0	1	0	1	0
d_n	0	0	0	0	0	1

- Experimental results show that the range of coefficient ***a*** is 0.8 ~ 0.95.

$$f(d_i, c_j) = a * \frac{link(d_i, c_j)}{N_{\max}} + (1-a) * \cos(d_i, c_j), a \in [0,1]$$

Neighbors and Links Concepts



Part Five: Software Tools for Text Mining

- Lemur in C++
  - https://www.lemurproject.org/
  - University of Massachusetts at Amherst and Carnegie Mellon University,
- Weka in Java
  - http://www.cs.waikato.ac.nz/ml/weka/
  - The university of Waikato

Questions?