

COMP8400: Algorithms and Techniques for Data Mining

Text data mining

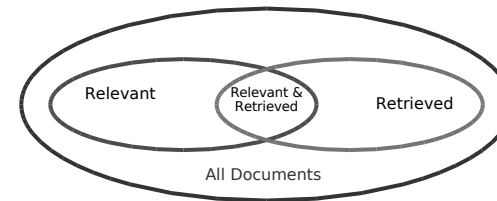
Text data analysis and information retrieval

- Typical information retrieval systems
 - Online library catalogs
 - Online document management systems
 - Internet search engines
- Information retrieval (IR) versus database (DB) systems
 - Some DB problems are not present in IR, such as: updates, transaction management, complex structured objects
 - Some IR problems are not addressed well in DBMS, for example: unstructured documents, approximate search using keywords and relevance

Lecture outline

- Text data analysis and text/information retrieval
 - Basic measures for text/information retrieval
 - Information retrieval techniques
 - Boolean and vector space model
 - Similarity-based retrieval in text data
 - TF-IDF weighting
 - Vector space model
- Types of text data mining
 - Keyword based association analysis
 - Text classification and categorisation
 - Document clustering

Basic measures for text retrieval



- Precision: the percentage of retrieved documents that are in fact relevant to the query (i.e., the “correct” responses)

$$precision = \frac{|{\textit{Relevant}} \cap {\textit{Retrieved}}|}{|{\textit{Retrieved}}|}$$

- Recall: the percentage of documents that are relevant to the query and were, in fact, retrieved

$$recall = \frac{|{\textit{Relevant}} \cap {\textit{Retrieved}}|}{|{\textit{Relevant}}|}$$

Information retrieval techniques

- Basic concepts

- A document can be described by a set of representative keywords called *index terms*
- Different index terms have varying relevance when used to describe document contents
- This effect is captured through the *assignment of numerical weights* to each index term of a document (for example, frequency, or TF-IDF)

- DBMS analogy

- Index terms → Attributes
- Weights → Attribute values

5

Information retrieval techniques (2)

- Index terms (attribute) selection

- Stop word list
- Word stem
- Index terms weighting methods

- Term and document frequency matrices

- Information retrieval models

- Boolean model
- Vector model
- Probabilistic model (categories modeled by probability distributions, find likelihood a document belongs to a certain category, similar to Bayesian classification)

6

Boolean model

- Consider that index terms are either present or absent in a document

- For example: $1=$ present, $0=$ absent
- As a result, the index term weights are assumed to be all binaries

- A query is composed of index terms linked by three connectives: *not*, *and*, and *or*

- For example: “*car and repair*”, “*plane or airplane*”

- The Boolean model predicts that each document is either relevant or non-relevant based on the match of a document to the query

7

Similarity-based retrieval in text data

- Finds similar documents based on a set of common keywords

- Answer should be based on the degree of relevance based on the nearness of the keywords, relative frequency of the keywords, etc.

- Use of stop lists

- Set of words that are deemed “irrelevant”, even though they may appear frequently
- For example: *a*, *the*, *of*, *for*, *to*, *with*, etc.
- Stop lists may vary when document set varies

8

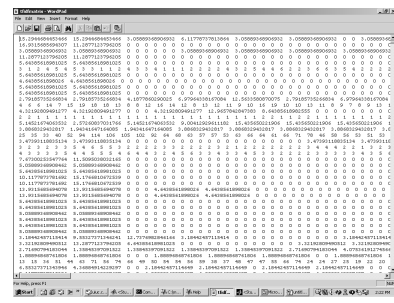
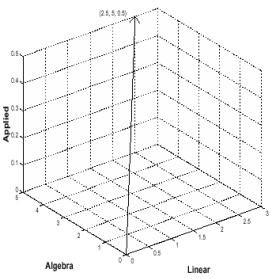
Similarity-based retrieval in text data (2)

- Apply word stemming
 - Several words are small syntactic variants of each other since they share a common word stem
 - For example, *drug, drugs, drugged* → *drug*
- A term and document frequency matrix (or table)
 - Each entry $frequent_table(i, j) =$ number of occurrences of the word t_i in document d_j
 - Usually, the *ratio* instead of the absolute number of occurrences is used
- Similarity metrics: measure the closeness of a document to a query (a set of keywords)
 - Relative term occurrences
 - Cosine distance:

$$sim(v_1, v_2) = \frac{v_1 \cdot v_2}{\|v_1\| \|v_2\|}$$

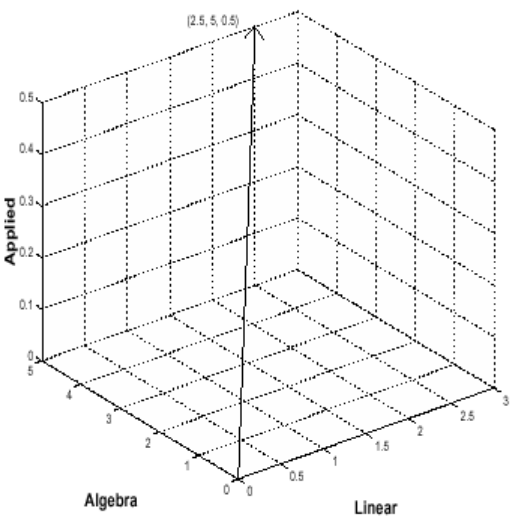
Vector space model

- Documents and user queries are represented as m -dimensional vectors, where m is the total number of index terms in the document collection
- The degree of similarity of the document d with regard to the query q is calculated as the correlation between the vectors that represent them, using measures such as the *Euclidean distance* or the *cosine* of the angle between these two vectors



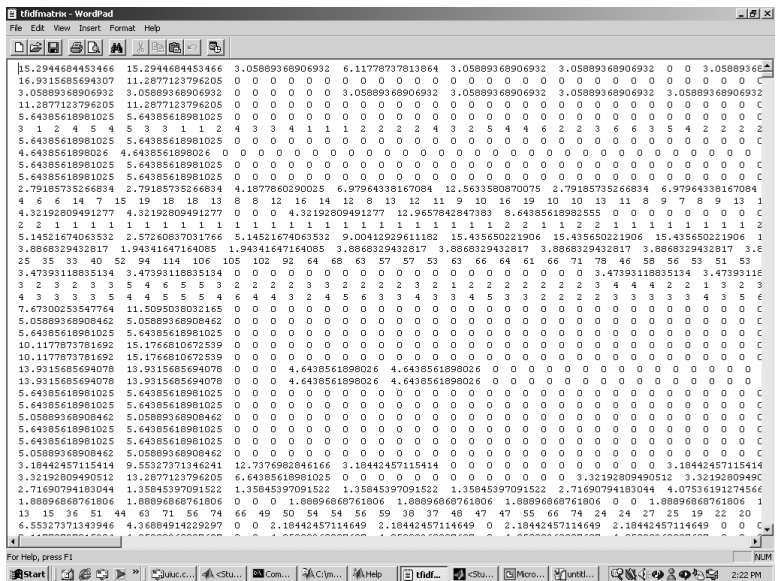
Source: Han and Kamber, DM Book, 2nd Ed. (Copyright © 2006 Elsevier Inc.)

Vector space model (2)



Source: Han and Kamber, DM Book, 2nd Ed. (Copyright © 2006 Elsevier Inc.)

Vector space model (3)



Vector space model (4)

- Represent a document by a *term* or *feature vector*
 - Term: basic concept, for example, *word* or *phrase* (like “data mining”)
 - Each term defines one dimension (large number of dimensions!)
 - N terms define a N -dimensional space
 - Element of vector corresponds to term weight
 - For example, $d = (x_1, \dots, x_N)$, x_i is “importance” of term i
 - These term vectors are *sparse* (most weights are 0)
- New document is assigned to the most likely category based on vector similarity

13

How to assign weights

- Two-fold heuristics based on frequency
- TF (Term Frequency)
 - More frequent *within* a document \rightarrow more relevant to semantics
 - For example, “classification” versus “SVM”
 - Raw TF = $f(t, d)$ (how many times term t appears in doc d)
 - Document length varies \Rightarrow relative frequency preferred
 - Perform normalisation (for example, *maximum frequency normalisation*)
$$TF(t, d) = 0.5 + \frac{0.5 * f(t, d)}{MaxFreq(d)}$$
- IDF (Inverse Document Frequency)
 - Less frequent *among* documents \rightarrow more discriminative
 - For example “algebra” versus “science”
$$IDF(t) = 1 + \log\left(\frac{n}{k}\right)$$
 - Formula:
 - n = total number of documents
 - k = number of documents with term t appearing

14

TF-IDF weighting

- TF-IDF weighting: $weight(t, d) = TF(t, d) * IDF(t)$
 - Frequent within doc \rightarrow high TF \rightarrow high weight
 - Selective among docs \rightarrow high IDF \rightarrow high weight
- Recall vector space model
 - Each selected term represents one dimension
 - Each document is represented by a *term* or *feature vector*
 - Its t -term coordinate of document d is the TF-IDF weight
- Just for illustration ...
 - Many complex and more effective weighting variants exist in practice

15

How to measure similarity?

- Given two document

$$D_i = (w_{i1}, w_{i2}, \dots, w_{iN}) \quad D_j = (w_{j1}, w_{j2}, \dots, w_{jN})$$

- Similarity definition

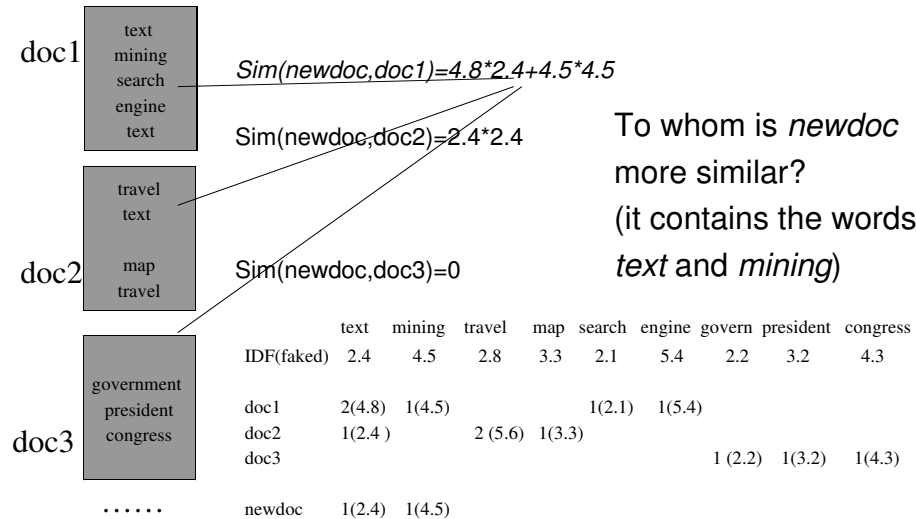
- Dot product
$$Sim(D_i, D_j) = \sum_{t=1}^N w_{it} * w_{jt}$$

Normalised dot product (or *cosine similarity*)

$$Sim(D_i, D_j) = \frac{\sum_{t=1}^N w_{it} * w_{jt}}{\sqrt{\sum_{t=1}^N (w_{it})^2 * \sum_{t=1}^N (w_{jt})^2}}$$

16

Illustrative example



17

Vector space model-based classifiers

- What do we have so far?
 - A feature space with similarity measure
 - This is a classic supervised learning problem
 - Search for an approximation to classification hyper plane
- Vector space model based classifiers
 - Decision tree based
 - Neural networks
 - Support vector machine
 - ...

18

Types of text data mining

- Keyword-based association analysis
- Automatic document classification
- Similarity detection
 - Cluster documents by a common author
 - Cluster documents containing information from a common source
- Link analysis: unusual correlation between entities
- Sequence analysis: predicting a recurring event
- Anomaly detection: find information that violates usual patterns
- Hypertext analysis
 - Patterns in anchors/links (for example, anchor text correlations with linked objects)
- Applications: news article classification, automatic e-mail filtering, Web page classification, etc.

19

Keyword-based association analysis

- Motivation
 - Collect sets of keywords or terms that occur frequently together and then find the *association* or *correlation* relationships among them
- Association analysis process
 - Pre-process the text data by parsing, stemming, removing stop words, etc.
 - Evoke association mining algorithms
 - Consider each document as a transaction
 - View a set of keywords in the document as a set of items in the transaction
 - Term level association mining
 - No need for human effort in tagging documents
 - The number of meaningless results and the execution time is greatly reduced

20

Text classification

- Motivation
 - Automatic classification for the large number of on-line text documents (Web pages, e-mails, corporate intranet documents, etc.)
- Classification process
 - Data pre-processing
 - Definition of training set and test sets
 - Creation of the classification model using the selected classification algorithm
 - Classification model validation
 - Classification of new/unknown text documents
- Text document classification differs from the classification of relational data
 - Document databases are not structured according to attribute-value pairs

Text classification (2)

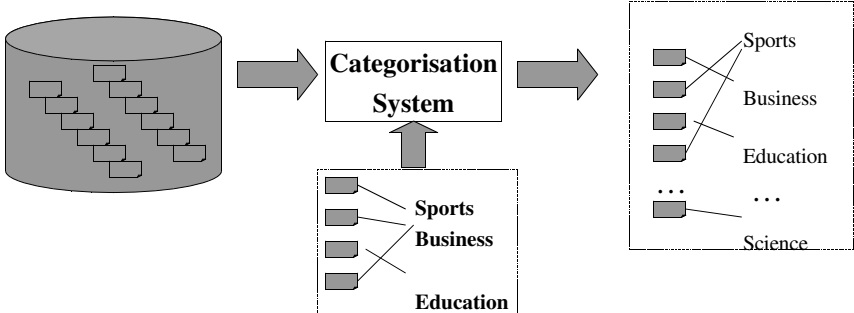
Classification Algorithms

- Support vector machines
- K-nearest neighbors
- Naïve Bayes
- Neural networks
- Decision trees
- Association rule-based
- Boosting
- ...

		#1	#2	#3	#4	#5
		# of documents	21,450	14,347	13,272	12,502
		# of training documents	14,704	10,607	9,610	9,603
		# of test documents	6,746	3,740	3,662	2,899
		# of categories	135	63	62	60
System	Type	Results reported by				
SVML	(non-linear)	Yang 1999	.150	.310	.490	
ProbBayes	probabilistic	[Dumais et al. 1998]			.762	.816
Bu	probabilistic	[Joachims 1998]				.720
Nu	probabilistic	[Lam et al. 1997]				
C4.5	decision trees	[Lewis 1992a]				
DL	decision trees	[Li and Yamashita 1999]			.747	
DL	decision trees	[Yang and Liu 1999]			.755	
SMPL	decision rules	[Dumais et al. 1998]				.881
SUPERDECISION	decision rules	[Joachims 1998]				.794
DL	decision rules	[Lewis and Hingstetter 1994]	.670			
DL	decision rules	[Pate et al. 1994]		.805		
DL	decision rules	[Cohen and Singer 1999]	.683	.811		.820
DL	decision rules	[Cohen and Singer 1999]		.750		.827
DL	decision rules	[Li and Yamashita 1999]				.820
DL	decision rules	[Moulinier and Charoat 1996]			.738	
DL	decision rules	[Moulinier et al. 1996]			.783 (F ₁)	
DL	decision rules	Yang 1999		.855	.810	
DL	decision rules	[Yang and Liu 1999]				.849
BALANCEWINDOW	on-line linear	[Dagan et al. 1997]	.747 (M)	.833 (M)		
WINDOW	on-line linear	[Lam and Ho 1998]				.822
ROCCHIO	batch linear	[Cohen and Singer 1999]	.660	.718		.779
ROCCHIO	batch linear	[Dumais et al. 1998]				.617
ROCCHIO	batch linear	[Joachims 1998]				.781
ROCCHIO	batch linear	[Lam and Ho 1998]				.625
ROCCHIO	batch linear	[Li and Yamashita 1999]				
NET	neural network	[Ng et al. 1997]		.602		.838
NET	neural network	[Yang and Liu 1999]				.860
NET	neural network	[Winer et al. 1995]				.864
GRM	example-based	[Lam and Ho 1998]			.820	
k-NN	example-based	[Joachims 1998]				.823
k-NN	example-based	[Lam and Ho 1998]				.820
k-NN	example-based	[Yang 1999]	.690	.852	.820	.856
k-NN	example-based	[Yang and Liu 1999]				
SVM	SVM	[Dumais et al. 1998]				.870
SVM	SVM	[Joachims 1998]				.841
SVM	SVM	[Li and Yamashita 1999]				.859
SVM	SVM	[Yang and Liu 1999]				
COMMITTEE	committee	[Schapire and Singer 2000]			.860	
COMMITTEE	committee	[Winer et al. 1995]				.878
Bayesian net	Bayesian net	[Dumais et al. 1998]				.850
Bayesian net	Bayesian net	[Lam et al. 1997]	.542 (M _{F1})			.850

Text classification (3)

- Pre-given classes (categories) and labeled document examples (categories may form hierarchy)
- Classify new documents
- A standard classification (supervised learning) problem



Categorisation methods

- Manual: Typically rule-based
 - Does not scale up (labor-intensive, rule inconsistency)
 - May be appropriate for special data on a particular domain
- Automatic: Typically exploiting machine learning techniques
 - Vector space model based
 - Prototype-based (Rocchio)
 - K-nearest neighbor (KNN)
 - Decision-tree (learn rules)
 - Neural networks (learn non-linear classifier)
 - Support vector machines (SVM)
- Probabilistic or generative model based
 - Naïve Bayes classifier

Document clustering

- **Motivation**

- Automatically group related documents based on their contents
- No predetermined training sets or taxonomies
- Generate a taxonomy at runtime

- **Clustering process**

- Data preprocessing: remove stop words, stem, feature extraction, lexical analysis, etc.
- Hierarchical clustering: compute similarities applying clustering algorithms
- Model-based clustering (neural network approach): clusters are represented by “exemplars” (for example Self-Organising Maps, SOM)