

# Unit-V

## Mining Spatial, Multimedia, Text, and Web Data



**Dr. K.RAGHAVA RAO**

Professor of CSE

Dept. of MCA

KL University

[krroacse@gmail.com](mailto:krroacse@gmail.com)

<http://datamining.blog.com>



# Mining Object, Spatial, Multimedia, Text, and Web Data

---

Data Mining



# Mining Complex Types of Data

---

- Mining spatial data
- Mining image data
- Mining text data
- Mining the Web



# Mining Spatial Databases

---

- Spatial database
  - Space related data: maps, VLSI layouts, ...
  - Topological, distance information organized by spatial indexing structures
- Spatial data warehousing
  - Issue: different representations & structures
  - Dimensions
    - Nonspatial: 25-30 degree → hot
    - Spatial-to-nonspatial: "New York" → "western provinces"
    - Spatial-to-spatial: equi. temp region → 0-5 degree region
  - Measures
    - numerical
    - Spatial: collection of spatial pointers (0-5 degree region)



# Example: BC Weather Pattern Analysis

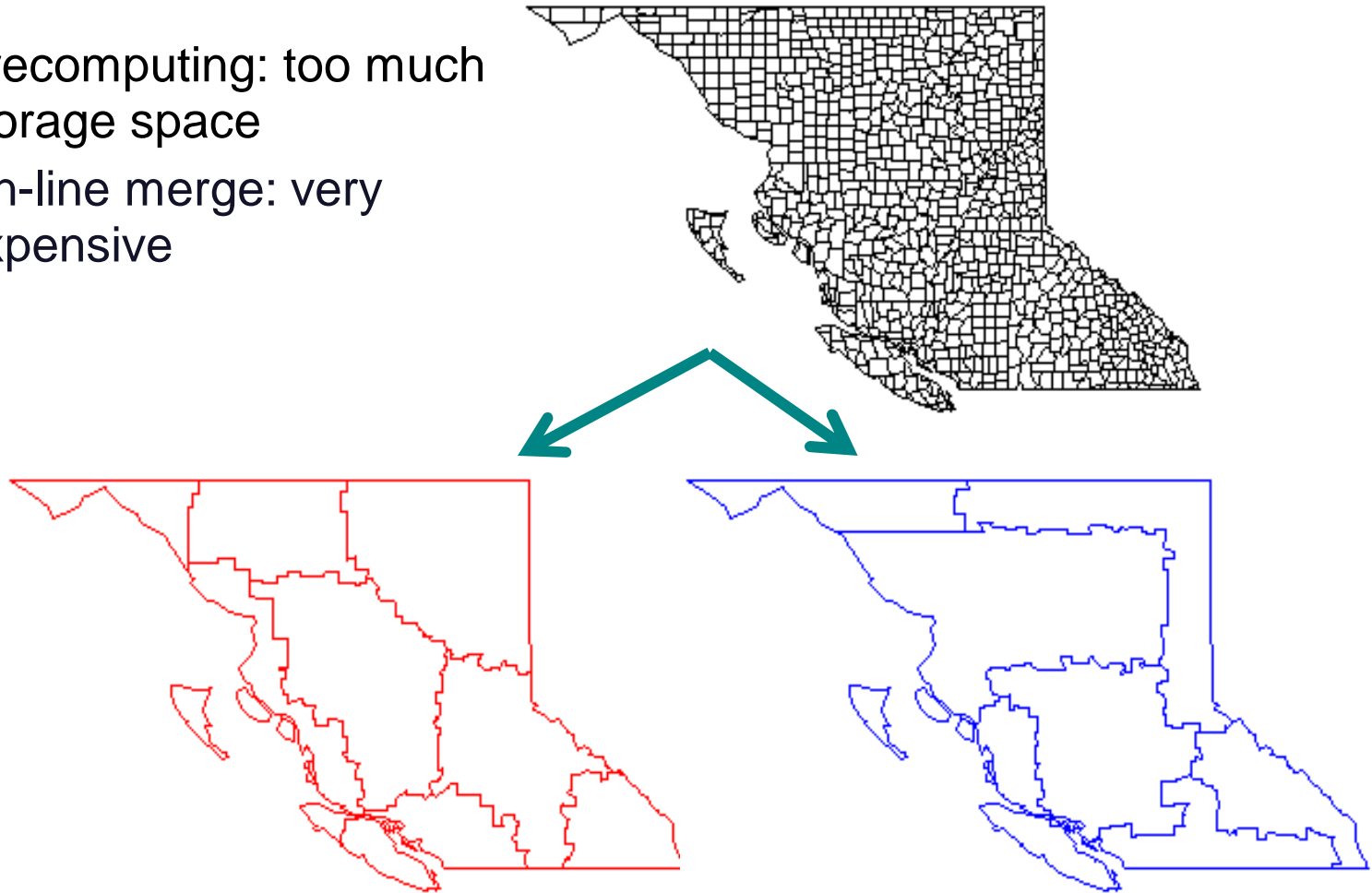
---

- Input
  - A map with about 3,000 weather probes scattered in B.C.
  - Daily data for temperature, wind velocity, etc.
  - Concept hierarchies for all attributes
- Output
  - A map that reveals patterns: merged (similar) regions
- Goals
  - Interactive analysis (drill-down, slice, dice, pivot, roll-up)
  - Fast response time, Minimizing storage space used
- Challenge
  - A merged region may contain hundreds of “primitive” regions (polygons)



# Spatial Merge

- Precomputing: too much storage space
- On-line merge: very expensive





# Spatial Association Analysis

---

- Spatial association rule:  $A \Rightarrow B [s\%, c\%]$ 
  - A and B are sets of spatial or nonspatial predicates
    - Topological relations: *intersects, overlaps, disjoint*, etc.
    - Spatial orientations: *left\_of, west\_of, under*, etc.
    - Distance information: *close\_to, within\_distance*, etc.
  - Example
    - $is\_a(x, "school") \wedge close\_to(x, "sports\_center")$   
 $\Rightarrow close\_to(x, "park")$  [7%, 85%]
- Progressive Refinement
  - First search for rough relationship (e.g. *g\_close\_to* for *close\_to, touch, intersect*) using rough evaluation (e.g. MBR)
  - Then apply only to those objects which have passed the rough test



# Spatial Classification

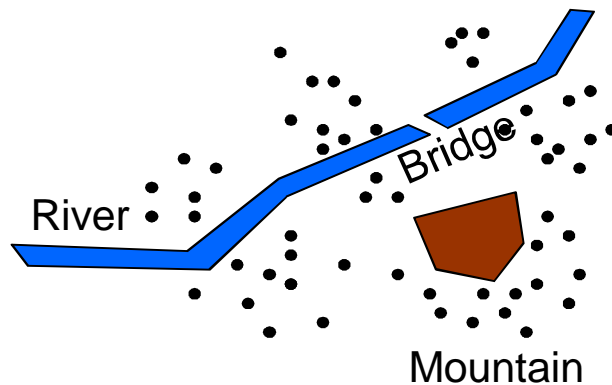
---

- Spatial classification
  - Analyze spatial objects to derive classification schemes, such as decision trees in relevance to spatial properties
  - Example
    - Classify regions into *rich* vs. *poor*
    - Properties: containing university, containing highway, near ocean, etc.

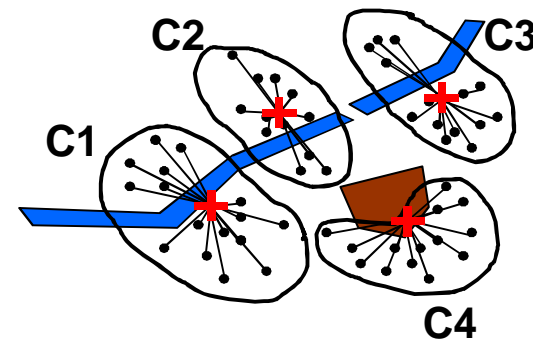


# Spatial Cluster Analysis

- Constraints-based clustering
  - Selection of relevant objects before clustering
  - Parameters as constraints
    - K-means, density-based: radius, min points
  - Clustering with obstructed distance



Spatial data with obstacles



Clustering *without* taking obstacles into consideration



# Mining Text Databases

---

- Text databases (document databases)
  - Large collections of documents from various sources
    - News articles, research papers, books, e-mail messages, and Web pages
  - Data stored is usually ***semi-structured***
  - Traditional information retrieval techniques become inadequate for the increasingly vast amounts of text data
- Information retrieval
  - Information is organized into documents
  - Information retrieval problem
    - Locating *relevant documents* based on user input, such as keywords or example documents

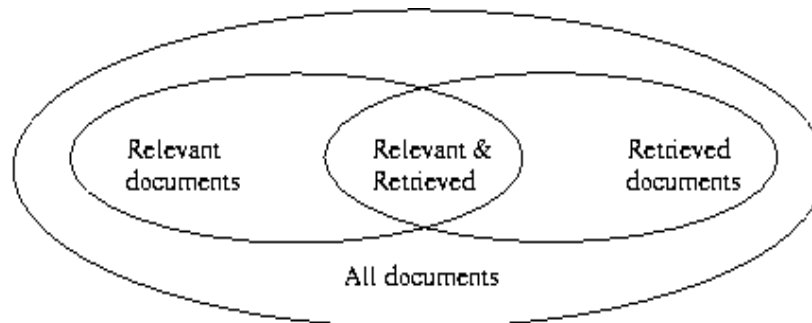
# Basic Measures for IR

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

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

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

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





# Keyword-Based Retrieval

---

- A document is represented by a set of keywords
  - Retrieval by keyword matching
- Queries may use expressions of keywords
  - (Car *and* accessory), (C++ *or* Java)
- Major difficulties
  - **Synonymy**: same meaning but different word
    - Ex> Q: "software" → Doc: about programming, do not have the keyword
  - **Polysemy**: same word but different meaning
    - Ex> Q: "mining" → Doc: about gold mining, have the keyword



# Similarity-Based Retrieval

---

- A document is represented as a *keyword vector*
  - Retrieval by similarity computing
- Basic techniques
  - Stop list – set of words that are frequent but irrelevant
    - Ex> *a, the, of, for, with, ...*
  - Stemming – use a common word stem
    - Ex> *drug, drugs, drugged* → *drug*
  - Weighting – count frequency
    - Term frequency, inverse document frequency, ...
- Similarity metrics
  - Measure the closeness of a document to a query
  - **Cosine similarity:** 
$$\text{sim}(v_1, v_2) = \frac{v_1 \cdot v_2}{|v_1| |v_2|}$$



# Latent Semantic Indexing

- Reduce the dimension of keyword matrix
  - To resolve the synonym problem and the size problem
  - Use a **singular value decomposition** (SVD) techniques
- Example

	<i>universe</i>	<i>rocket</i>	<i>moon</i>	<i>car</i>	<i>truck</i>
<i>D1</i>	1	0	1	1	0
<i>D2</i>	0	1	1	0	0
<i>D3</i>	1	0	0	0	0
<i>D4</i>	0	0	0	1	1
<i>D5</i>	0	0	0	1	0
<i>D6</i>	0	0	0	0	1



# SVD

---

- Singular Value Decomposition

- Decompose the matrix  $A_{mn}$

$$A_{mn} = U_{mm} S_{mn} (V_{nn})^T$$

- Reduce dimension

- Select largest k singular values

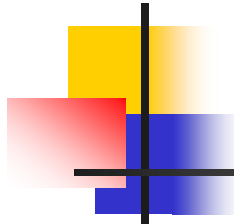
$$A'_{mn} = U_{mk} S_{kk} (V_{nk})^T$$

- Projection of A into k dimension

$$A'_{mn} V_{nk} = U_{mk} S_{kk}$$

- Computing similarity

$$\begin{aligned} AA^T &= USV^T(USV^T)^T \\ &= USV^T V S^T U^T \\ &= (US)(US)^T \end{aligned}$$



# SVD

$$U = \begin{bmatrix} -0.75 & -0.29 & 0.28 & 0.00 & -0.53 \\ -0.28 & -0.53 & -0.75 & 0.00 & 0.29 \\ -0.20 & -0.19 & 0.45 & 0.58 & 0.63 \\ -0.45 & 0.63 & -0.20 & 0.00 & 0.19 \\ -0.33 & 0.22 & 0.12 & -0.58 & 0.41 \\ -0.12 & 0.41 & -0.33 & 0.58 & -0.22 \end{bmatrix} \quad S = \begin{bmatrix} 2.16 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 1.59 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 1.28 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 1.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.39 \end{bmatrix} \quad V^T = \dots$$

$$A'V = US_2 = \begin{bmatrix} -0.62 & -0.46 \\ -0.60 & -0.84 \\ -0.04 & -0.30 \\ -0.97 & 1.00 \\ -0.71 & 0.35 \\ -0.26 & 0.65 \end{bmatrix} \quad (US)(US)^T = \begin{bmatrix} 1.00 & 0.78 & 0.40 & 0.47 & 0.74 & 0.10 \\ & 1.00 & 0.88 & -0.18 & 0.16 & -0.54 \\ & & 1.00 & -0.62 & -0.32 & -0.87 \\ & & & 1.00 & 0.94 & 0.93 \\ & & & & 1.00 & 0.74 \\ & & & & & 1.00 \end{bmatrix}$$





# Automatic Document Classification

---

- Motivation
  - Automatic classification for the tremendous number of on-line text documents (Web pages, e-mails, etc.)
- A classification problem
  - Training set: Human experts generate a training data set
  - Classification(learning): The system discovers the classification rules
- Methods
  - Extract keywords and weights from documents
    - Documents are represented as (keyword, weight) pairs
  - Classify training documents into classes
  - Apply classification algorithm
    - Decision tree, Bayesian, neural network, etc.



# Mining the World-Wide Web

---

- WWW provides rich sources for data mining
  - **Contents** information
  - **Hyperlink** information
  - **Usage** information
- Challenges
  - Too huge for effective data warehousing and data mining
  - Too complex and heterogeneous
  - Growing and changing very rapidly



# Web Search Engines

---

- Index-based
  - Search the Web, collect Web pages, index Web pages, and build and store huge keyword-based indices
  - Locate sets of Web pages containing certain keywords
- Deficiencies
  - A topic of any breadth may easily contain hundreds of thousands of documents
  - Many documents that are highly relevant to a topic may not contain keywords defining them (synonymy, polysemy)



# Web Contents Mining - Classification

---

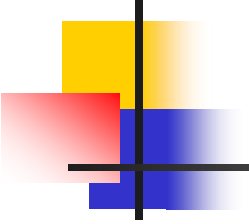
- Web page/site classification
  - Assign a ***class label to each web page*** from a set of predefined topic categories
  - Based on a set of examples of preclassified documents
- Example
  - Use Yahoo!'s taxonomy and its associated documents as training and test sets
  - Derive a Web document classification model
  - Use the model to classify new Web documents by assigning categories from the same taxonomy
- Methods
  - Keyword-based classification, use of hyperlink information, statistical models, ...



# Web Structure Mining

---

- Finding authoritative Web pages
  - Retrieving pages that are not only relevant, but also of high quality, or *authoritative* on the topic
- *Hyperlinks* can infer the notion of authority
  - A hyperlink pointing to another Web page, this can be considered as the author's endorsement of the other page
- Problems
  - Not every hyperlink represents an endorsement
  - One authority will seldom point to its rival authority
  - Authoritative pages are seldom particularly descriptive
- Hub
  - Set of Web pages that provides collections of links to authorities



# HITS (Hyperlink-Induced Topic Search)

---

- Method
  1. Use an index-based search engine to form the **root set**
  2. Expand the root set into a **base set**
    - Include all of the pages that the root-set pages link to, and all of the pages that link to a page in the root set
  3. Apply weight-propagation
    - Determines numerical estimates of **hub and authority** weights
  4. Output a list of the pages
    - Large hub weights, large authority weights for the given search topic
- Systems based on the HITS algorithm
  - Clever, Google
    - Achieve better quality search results than AltaVista, Yahoo!



# Web Usage Mining

---

- Mining *Web log* records
  - Discover *user access patterns*
  - Typical Web log entry - URL requested, the IP address from which the request originated, timestamp, etc.
- OLAP on the Weblog database
  - Find the top  $N$  users, top  $N$  accessed Web pages, most frequently accessed time periods, etc.
- Data mining on Weblog records
  - Find association patterns, sequential patterns, and trends of Web accessing



# Web Usage Mining

---

- Applications
  - Target potential customers for electronic commerce
  - Identify potential prime advertisement locations
  - Enhance the quality and delivery of Internet information services to the end user
  - Improve Web server system performance
    - Web caching, Web page prefetching, and Web page swapping