UNIT-8 Mining Complex Types of Data

Lecture	Topic
**********	**********
Lecture-50	Multidimensional analysis and descriptive mining of complex data objects
Lecture-51	Mining spatial databases
Lecture-52	Mining multimedia databases
Lecture-53	Mining time-series and sequence data
Lecture-54	Mining text databases
Lecture-55	Mining the World-Wide Web

Lecture-50

Multidimensional analysis and descriptive mining of complex data objects

Mining Complex Data Objects: Generalization of Structured Data

- Set-valued attribute
  - Generalization of each value in the set into its corresponding higher-level concepts
  - Derivation of the general behavior of the set, such as the number of elements in the set, the types or value ranges in the set, or the weighted average for numerical data
  - *hobby* = {*tennis, hockey, chess, violin, nintendo_games*} generalizes to {*sports, music, video_games*}
- List-valued or a sequence-valued attribute
  - Same as set-valued attributes except that the order of the elements in the sequence should be observed in the generalization

Generalizing Spatial and Multimedia Data

- Spatial data:
  - Generalize detailed geographic points into clustered regions, such as business, residential, industrial, or agricultural areas, according to land usage
  - Require the merge of a set of geographic areas by spatial operations
- Image data:
  - Extracted by aggregation and/or approximation
  - Size, color, shape, texture, orientation, and relative positions and structures of the contained objects or regions in the image
- Music data:
  - Summarize its melody: based on the approximate patterns that repeatedly occur in the segment
  - Summarized its style: based on its tone, tempo, or the major musical instruments played

Generalizing Object Data

- Object identifier: generalize to the lowest level of class in the class/subclass hierarchies
- Class composition hierarchies
  - generalize nested structured data
  - generalize only objects closely related in semantics to the current one
- Construction and mining of object cubes
  - Extend the attribute-oriented induction method
    - Apply a sequence of class-based generalization operators on different attributes
    - Continue until getting a small number of generalized objects that can be summarized as a concise in high-level terms
  - For efficient implementation
    - Examine each attribute, generalize it to simple-valued data
    - Construct a multidimensional data cube (object cube)
    - Problem: it is not always desirable to generalize a set of values to single-valued data

An Example: Plan Mining by Divide and Conquer

- Plan: a variable sequence of actions
  - E.g., Travel (flight): <traveler, departure, arrival, d-time, a-time, airline, price, seat>
- Plan mining: extraction of important or significant generalized (sequential) patterns from a planbase (a large collection of plans)
  - E.g., Discover travel patterns in an air flight database, or
  - find significant patterns from the sequences of actions in the repair of automobiles
- Method
  - Attribute-oriented induction on sequence data
    - A generalized travel plan: <small-big*-small>
  - Divide & conquer:Mine characteristics for each subsequence
    - E.g., big*: same airline, small-big: nearby region

A Travel Database for Plan Mining

- Example: Mining a travel planbase

Travel plans table

plan#	action#	departure	depart_time	arrival	arrival_time	airline	…
1	1	ALB	800	JFK	900	TWA	…
1	2	JFK	1000	ORD	1230	UA	…
1	3	ORD	1300	LAX	1600	UA	…
1	4	LAX	1710	SAN	1800	DAL	…
2	1	SPI	900	ORD	950	AA	…
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.

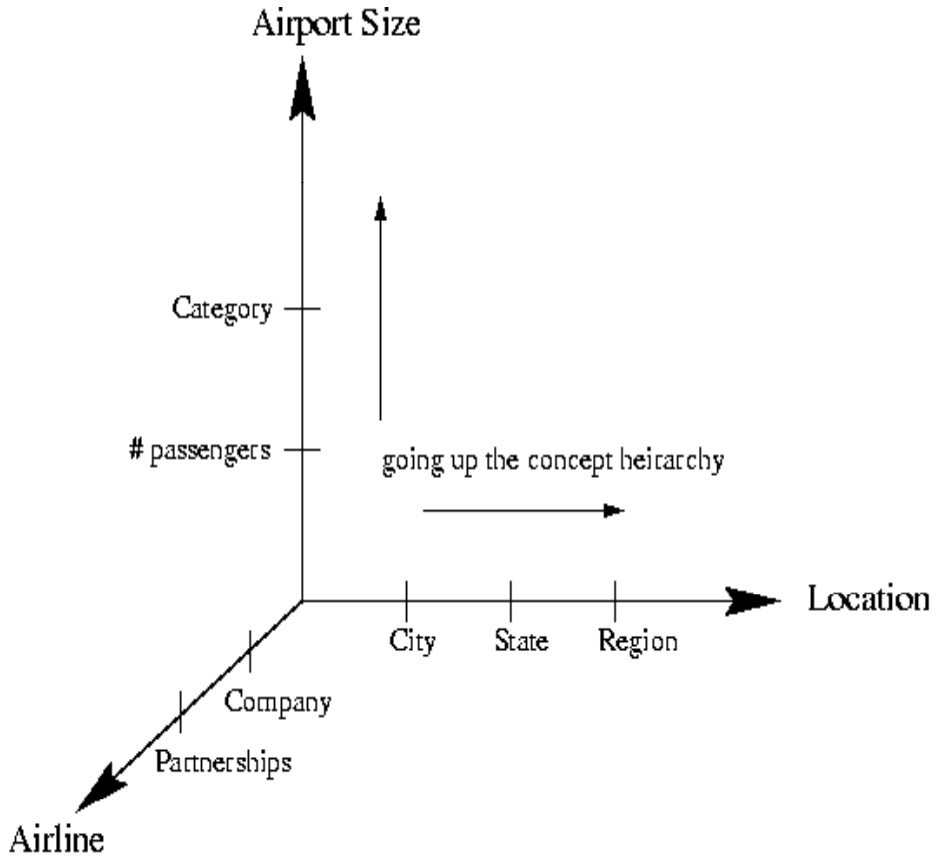
Airport info table

airport_code	city	state	region	airport_size	…
1	1	ALB		800	…
1	2	JFK		1000	…
1	3	ORD		1300	…
1	4	LAX		1710	…
2	1	SPI		900	…
.	.	.		.	.
.	.	.		.	.
.	.	.		.	.

Multidimensional Analysis

- Strategy
  - Generalize the planbase in different directions
  - Look for sequential patterns in the generalized plans
  - Derive high-level plans

A multi-D model for the planbase



Multidimensional Generalization

Multi-D generalization of the planbase

Plan#	Loc_Seq	Size_Seq	State_Seq
1	ALB - JFK - ORD - LAX - SAN	S - L - L - L - S	N - N - I - C - C
2	SPI - ORD - JFK - SYR	S - L - L - S	I - I - N - N
.	.		.
.	.		.
.	.		.

Merging consecutive, identical actions in plans

Plan#	Size_Seq	State_Seq	Region_Seq	…
1	S - L+ - S	N+ - I - C+	E+ - M - P+	…
2	S - L+ - S	I+ - N+	M+ - E+	…
.		.		.
.		.		.
.		.		.

$$flight(x, y,) \wedge airport\_size(x, S) \wedge airport\_size(y, L)$$
$$\Rightarrow region(x) = region(y) \quad [75\%]$$

Generalization-Based Sequence Mining

- Generalize planbase in multidimensional way using dimension tables
- Use no of distinct values (cardinality) at each level to determine the right level of generalization (level-"planning")
- Use operators *merge* "+", *option* "[]" to further generalize patterns
- Retain patterns with significant support

Generalized Sequence Patterns

- AirportSize-sequence survives the min threshold (after applying *merge* operator):

  S-L^+-**S** [35%], L^+-**S** [30%], **S**-L^+ [24.5%], L^+ [9%]

- After applying *option* operator:

  [S]-L^+-[**S**] [98.5%]

  – Most of the time, people fly via large airports to get to final destination

- Other plans: 1.5% of chances, there are other patterns: S-S, L-S-L

Lecture-51

Mining spatial databases

Spatial Data Warehousing

- Spatial data warehouse: Integrated, subject-oriented, time-variant, and nonvolatile spatial data repository for data analysis and decision making
- Spatial data integration: a big issue
  - Structure-specific formats (raster- vs. vector-based, OO vs. relational models, different storage and indexing)
  - Vendor-specific formats (ESRI, MapInfo, Integraph)
- Spatial data cube: multidimensional spatial database
  - Both dimensions and measures may contain spatial components

Dimensions and Measures in Spatial Data Warehouse

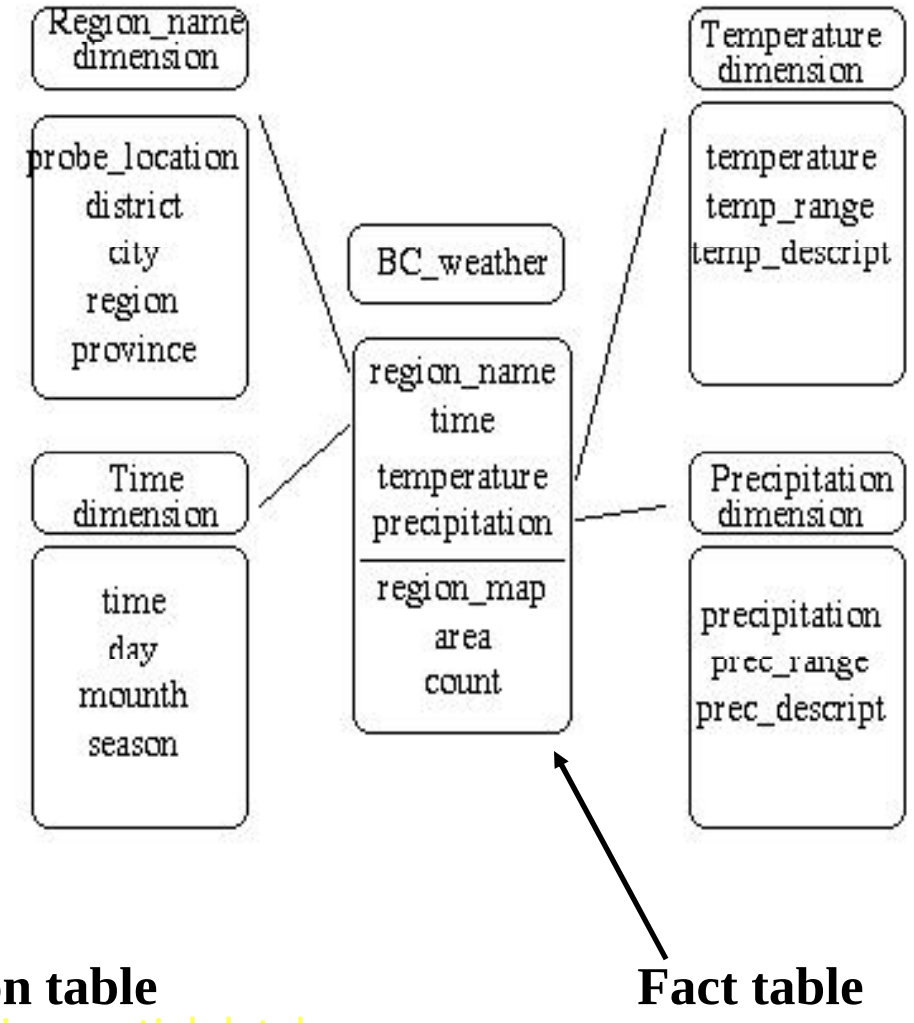
- Dimension modeling
  - nonspatial
    - e.g. temperature: 25-30 degrees generalizes to *hot*
  - spatial-to-nonspatial
    - e.g. region "B.C." generalizes to description "*western provinces*"
  - spatial-to-spatial
    - e.g. region "Burnaby" generalizes to region "Lower Mainland"
- Measures
  - numerical
    - distributive (count, sum)
    - algebraic (e.g. average)
    - holistic (e.g. median, rank)
  - spatial
    - collection of spatial pointers (e.g. pointers to all regions with 25-30 degrees in July)

Example: BC weather pattern analysis

- Input
  - A map with about 3,000 weather probes scattered in B.C.
  - Daily data for temperature, precipitation, 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)

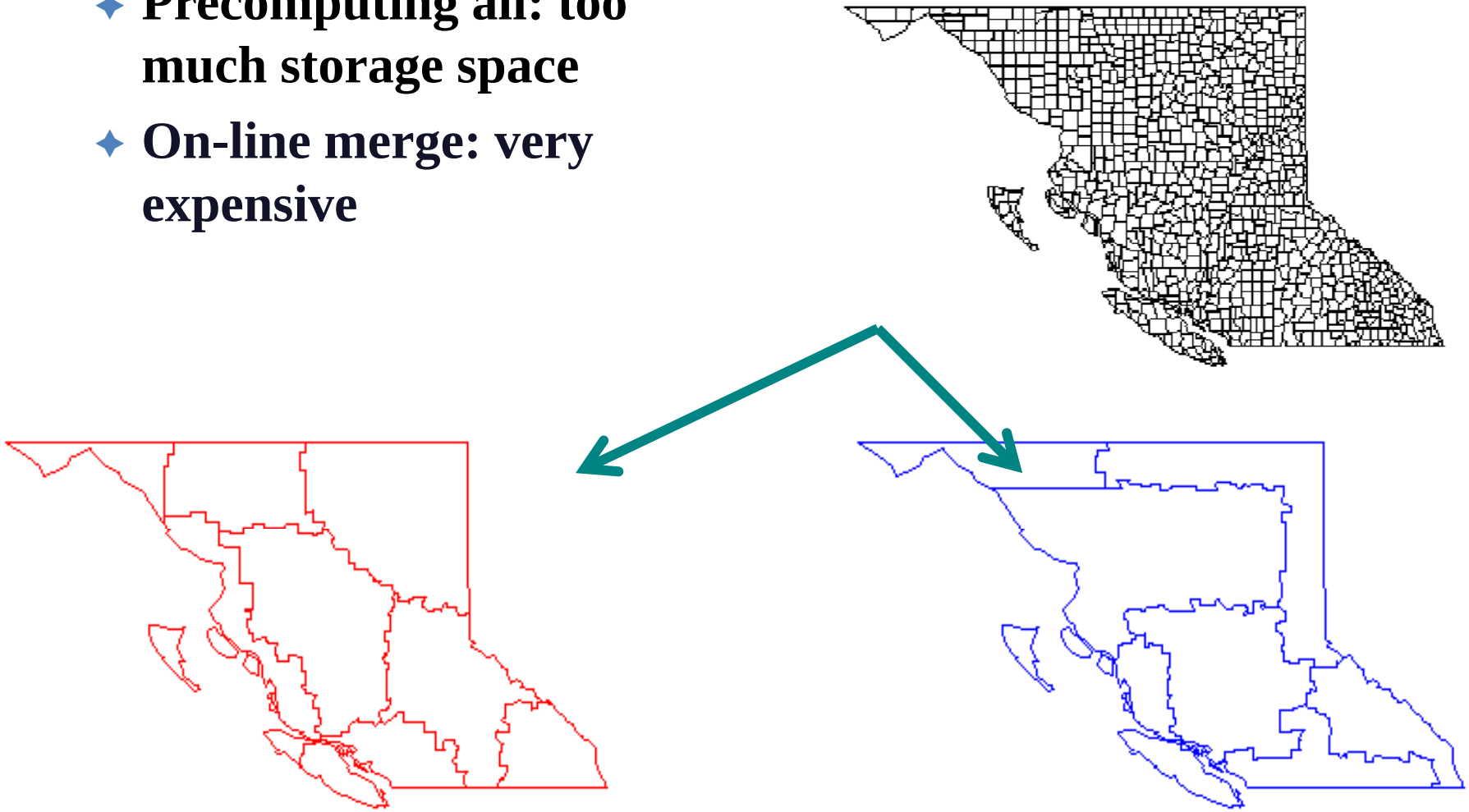
Star Schema of the BC Weather Warehouse

- Spatial data warehouse
  - Dimensions
    - region_name
    - time
    - temperature
    - precipitation
  - Measurements
    - region_map
    - area
    - count



Spatial Merge

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



Methods for Computation of Spatial Data Cube

- On-line aggregation: collect and store pointers to spatial objects in a spatial data cube
  - expensive and slow, need efficient aggregation techniques
- Precompute and store all the possible combinations
  - huge space overhead
- Precompute and store rough approximations in a spatial data cube
  - accuracy trade-off
- Selective computation: only materialize those which will be accessed frequently
  - a reasonable choice

Spatial Association Analysis

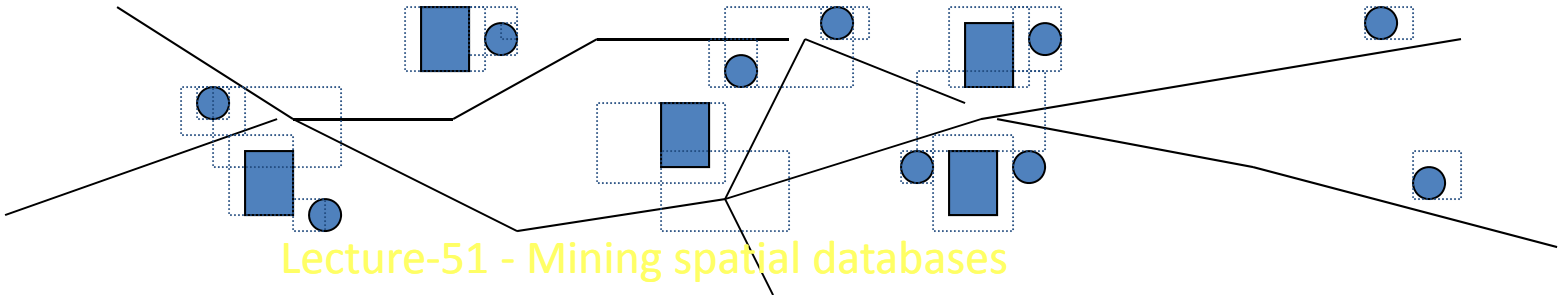
- Spatial association rule: $A \Rightarrow B$ [*s*%, *c*%]
  - A and B are sets of spatial or nonspatial predicates
    - Topological relations: *intersects, overlaps, disjoint,* etc.
    - Spatial orientations: *left_of, west_of, under,* etc.
    - Distance information: *close_to, within_distance,* etc.
  - *s*% is the support and *c*% is the confidence of the rule
- Examples

is_a(x, large_town) ^ intersect(x, highway) → adjacent_to(x, water) [7%, 85%]

is_a(x, large_town) ^adjacent_to(x, georgia_strait) → close_to(x, u.s.a.) [1%, 78%]

Progressive Refinement Mining of Spatial Association Rules

- Hierarchy of spatial relationship:
  - *g_close_to*: *near_by*, *touch*, *intersect*, *contain*, etc.
  - First search for rough relationship and then refine it
- Two-step mining of spatial association:
  - Step 1: Rough spatial computation (as a filter)
    - Using MBR or R-tree for rough estimation
  - Step2: Detailed spatial algorithm (as refinement)
    - Apply only to those objects which have passed the rough spatial association test (no less than *min_support*)



Spatial Classification and Spatial Trend Analysis

- Spatial classification
  - Analyze spatial objects to derive classification schemes, such as decision trees in relevance to certain spatial properties (district, highway, river, etc.)
  - Example: Classify regions in a province into *rich* vs. *poor* according to the average family income
- Spatial trend analysis
  - Detect changes and trends along a spatial dimension
  - Study the trend of nonspatial or spatial data changing with space
  - Example: Observe the trend of changes of the climate or vegetation with the increasing distance from an ocean

Lecture-52

Mining multimedia databases

Similarity Search in Multimedia Data

- Description-based retrieval systems
  - Build indices and perform object retrieval based on image descriptions, such as keywords, captions, size, and time of creation
  - Labor-intensive if performed manually
  - Results are typically of poor quality if automated
- Content-based retrieval systems
  - Support retrieval based on the image content, such as color histogram, texture, shape, objects, and wavelet transforms

Queries in Content-Based Retrieval Systems

- Image sample-based queries:
  - Find all of the images that are similar to the given image sample
  - Compare the feature vector (signature) extracted from the sample with the feature vectors of images that have already been extracted and indexed in the image database
- Image feature specification queries:
  - Specify or sketch image features like color, texture, or shape, which are translated into a feature vector
  - Match the feature vector with the feature vectors of the images in the database

Approaches Based on Image Signature

- Color histogram-based signature
  - The signature includes color histograms based on color composition of an image regardless of its scale or orientation
  - No information about shape, location, or texture
  - Two images with similar color composition may contain very different shapes or textures, and thus could be completely unrelated in semantics
- Multifeature composed signature
  - The signature includes a composition of multiple features: color histogram, shape, location, and texture
  - Can be used to search for similar images

Wavelet Analysis

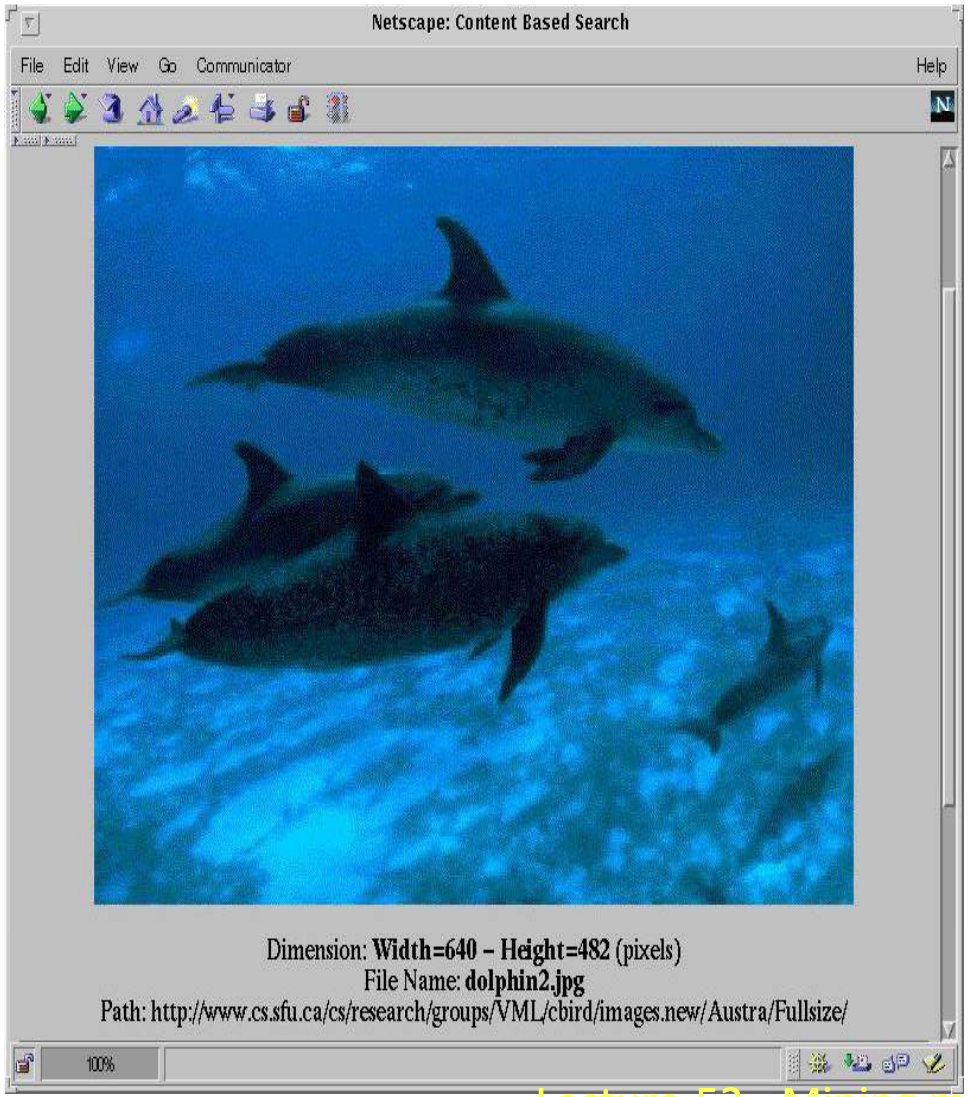
- Wavelet-based signature
  - Use the dominant wavelet coefficients of an image as its signature
  - Wavelets capture shape, texture, and location information in a single unified framework
  - Improved efficiency and reduced the need for providing multiple search primitives
  - May fail to identify images containing similar in location or size objects
- Wavelet-based signature with region-based granularity
  - Similar images may contain similar regions, but a region in one image could be a translation or scaling of a matching region in the other
  - Compute and compare signatures at the granularity of regions, not the entire image

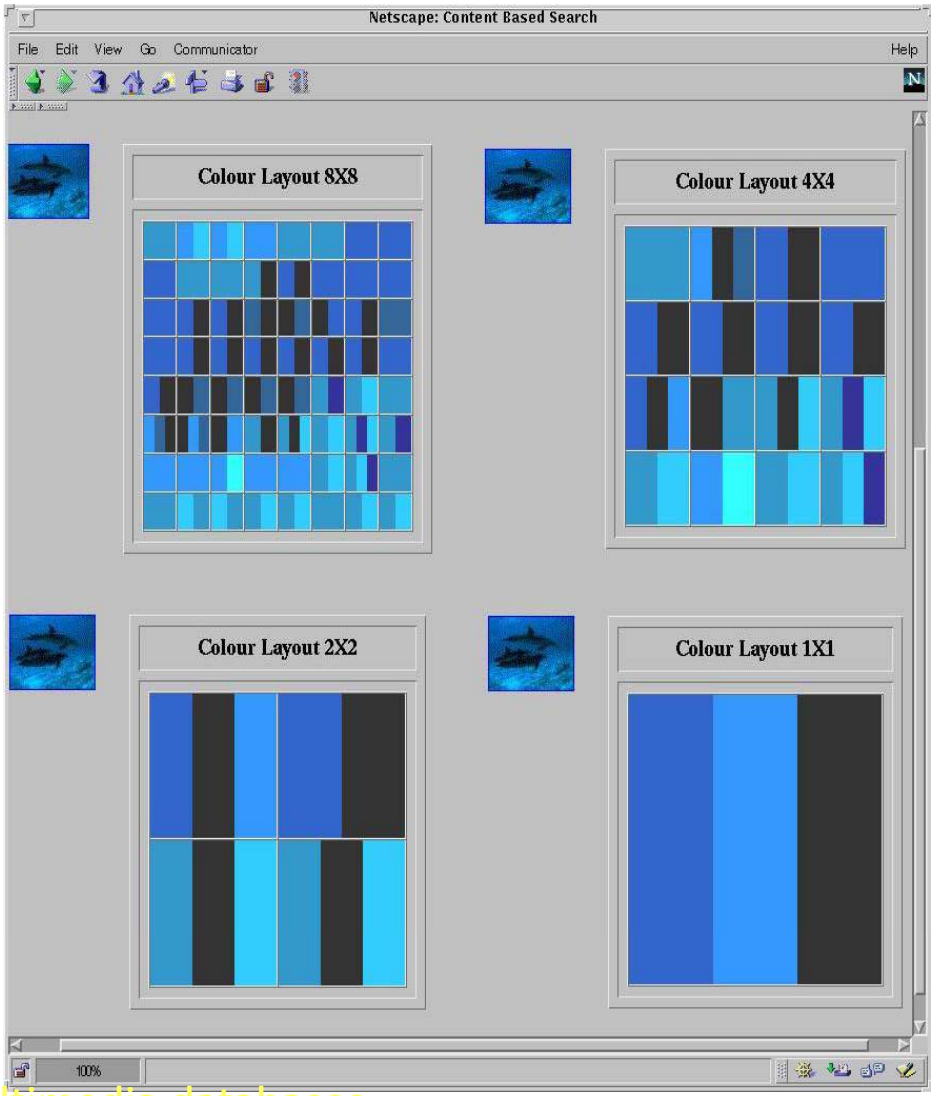
C-BIRD: Content-Based Image Retrieval from Digital libraries

Search

- by image colors
- by color percentage
- by color layout
- by texture density
- by texture Layout
- by object model
- by illumination invariance
- by keywords

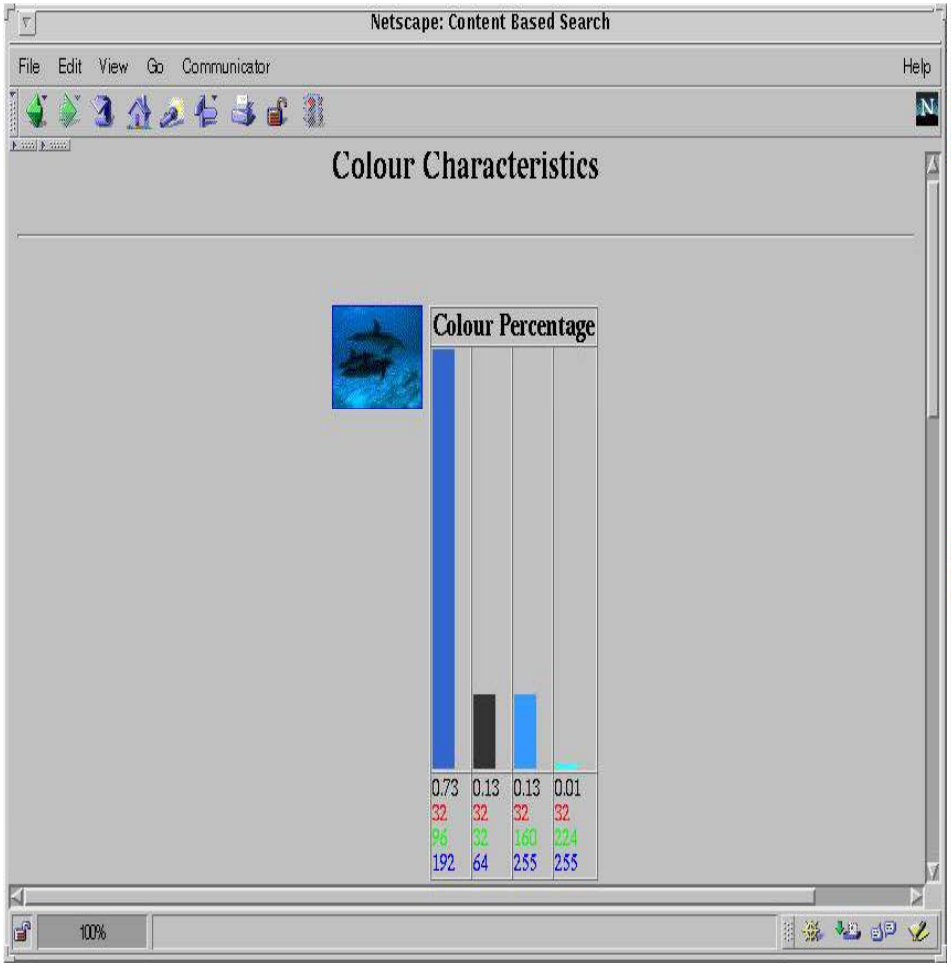
Multi-Dimensional Search in Multimedia Databases



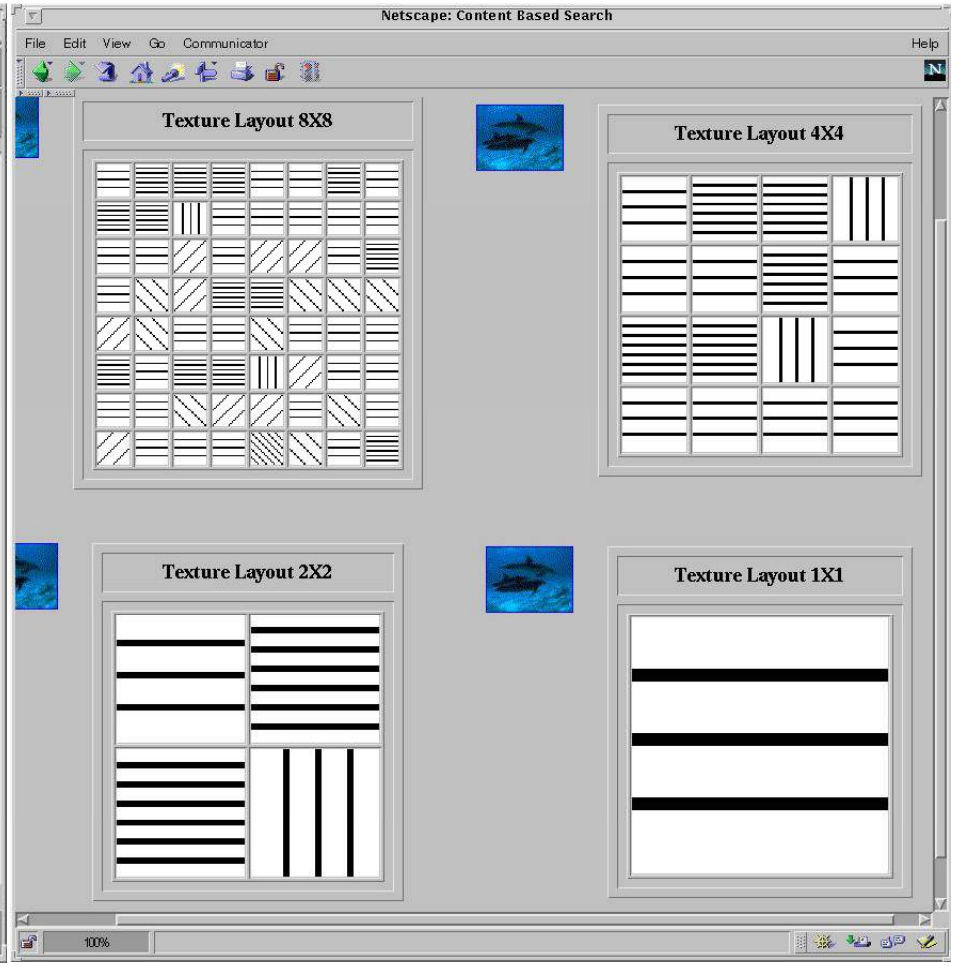


Multi-Dimensional Analysis in Multimedia Databases

Color histogram



Texture layout



Mining Multimedia Databases

Refining or combining searches

Search for “blue sky”
(top layout grid is blue)

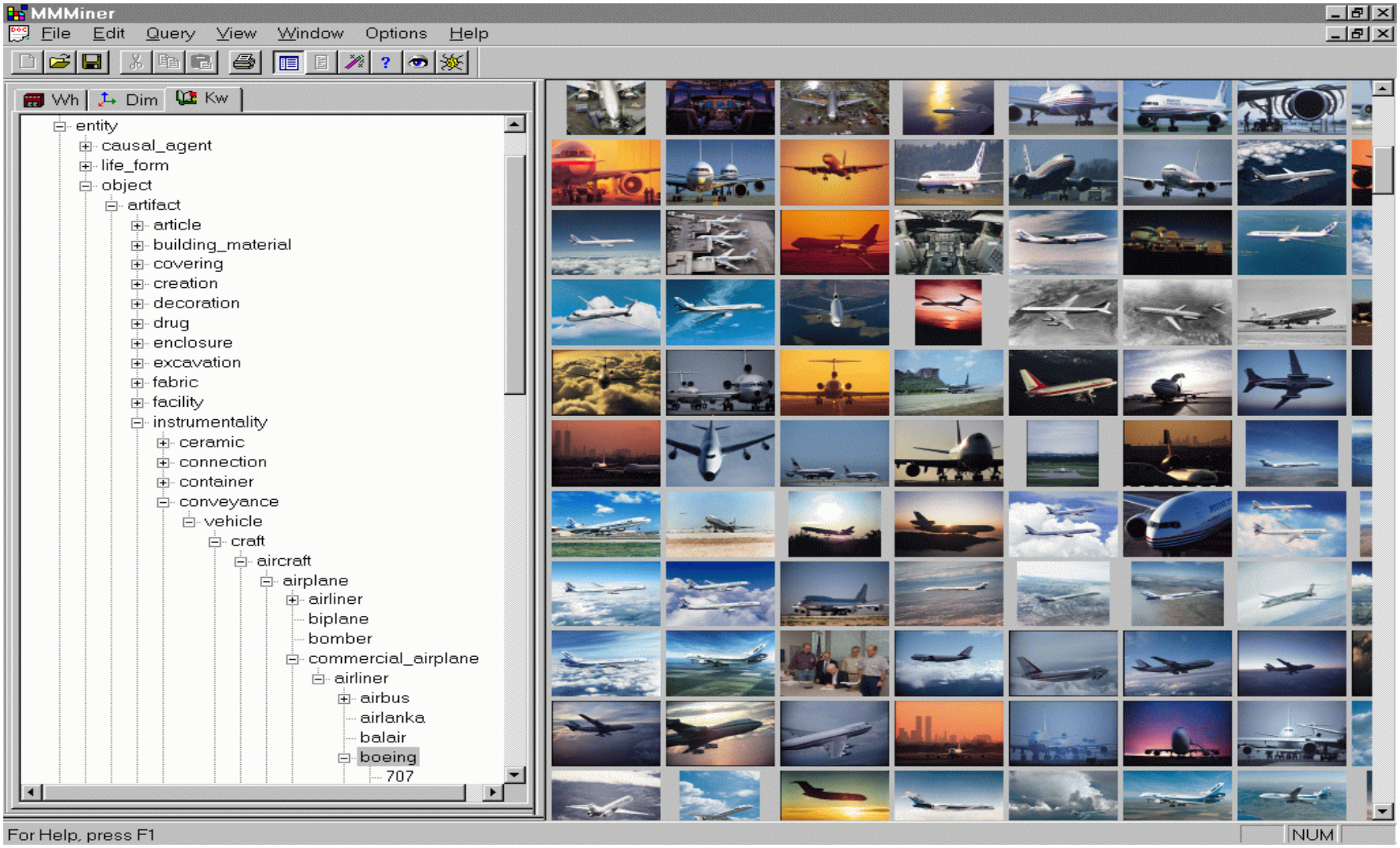
Search for “airplane in blue sky”
(top layout grid is blue and
keyword = “airplane”)

Search for “blue sky and green meadows”
(top layout grid is blue
and bottom is green)

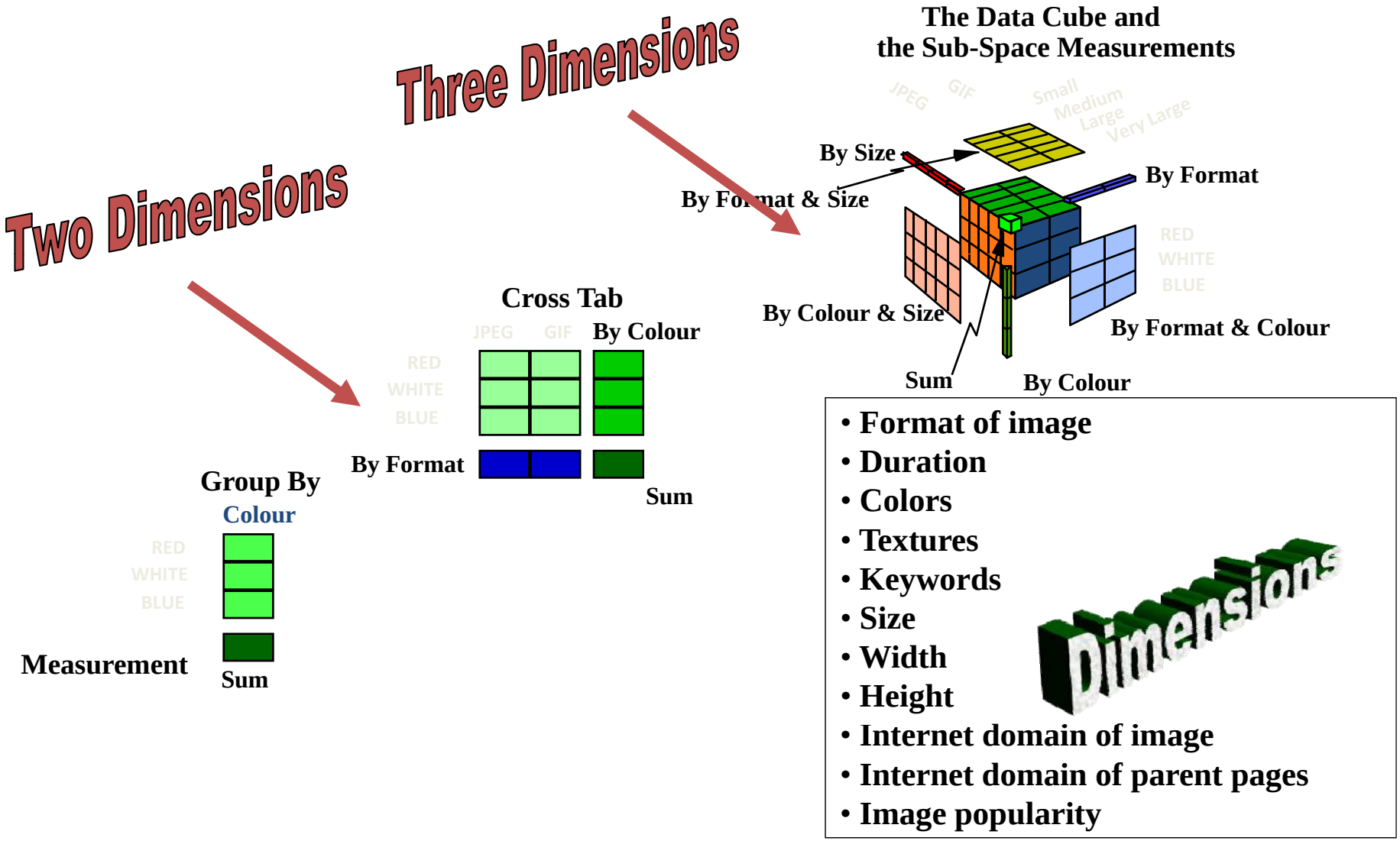
Multidimensional Analysis of Multimedia Data

- Multimedia data cube
  - Design and construction similar to that of traditional data cubes from relational data
  - Contain additional dimensions and measures for multimedia information, such as color, texture, and shape
- The database does not store images but their descriptors
  - Feature descriptor: a set of vectors for each visual characteristic
    - Color vector: contains the color histogram
    - MFC (Most Frequent Color) vector: five color centroids
    - MFO (Most Frequent Orientation) vector: five edge orientation centroids
  - Layout descriptor: contains a color layout vector and an edge layout vector

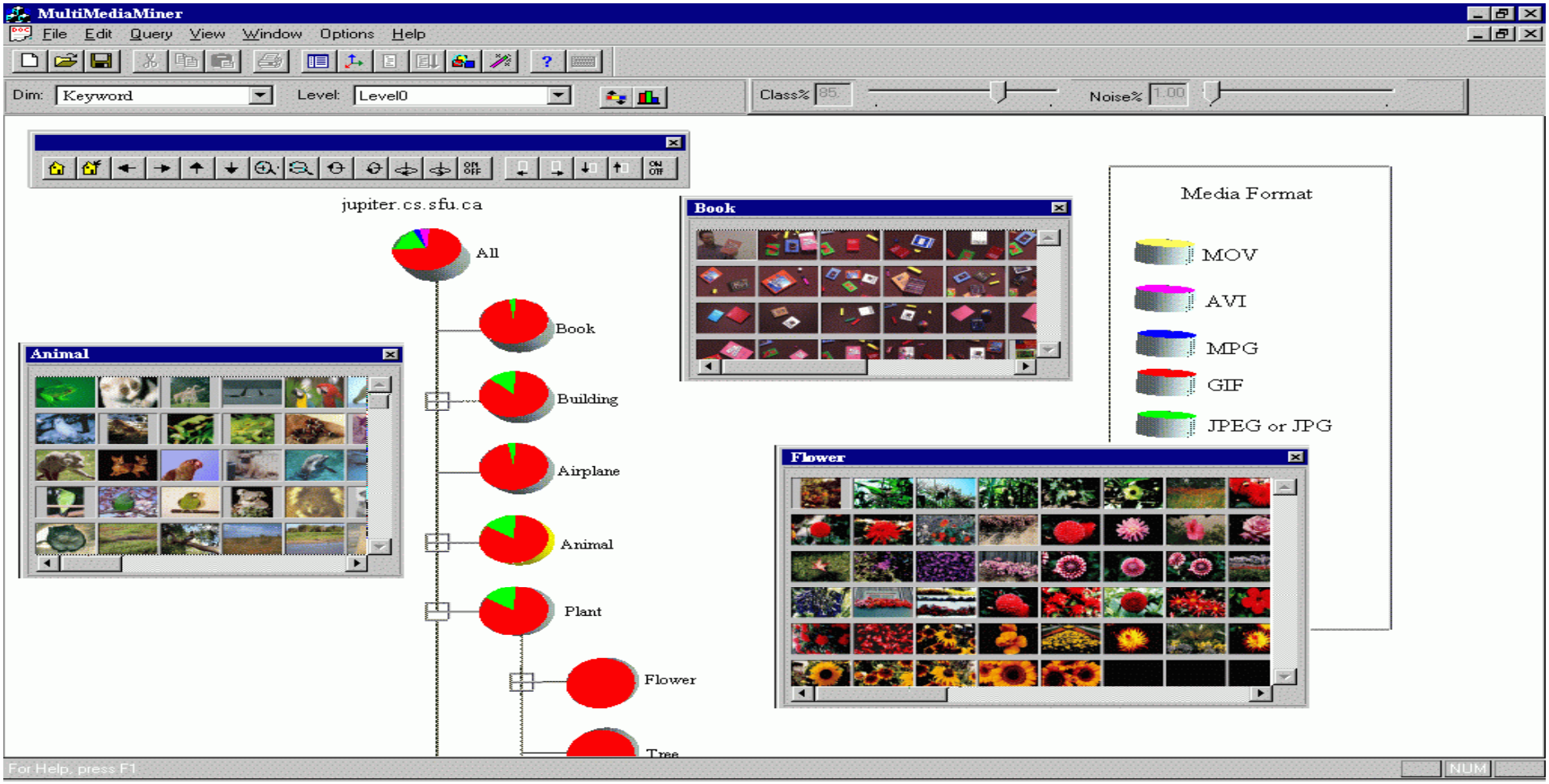
Mining Multimedia Databases in MultiMediaMiner



Mining Multimedia Databases



Classification in MultiMediaMiner



Mining Associations in Multimedia Data

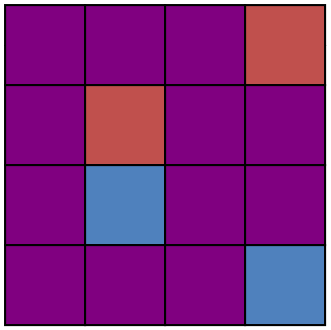
- Special features:
  - Need # of occurrences besides Boolean existence, e.g.,
    - "Two red square and one blue circle" implies theme "air-show"
  - Need spatial relationships
    - Blue on top of white squared object is associated with brown bottom
  - Need multi-resolution and progressive refinement mining
    - It is expensive to explore detailed associations among objects at high resolution
    - It is crucial to ensure the completeness of search at multi-resolution space

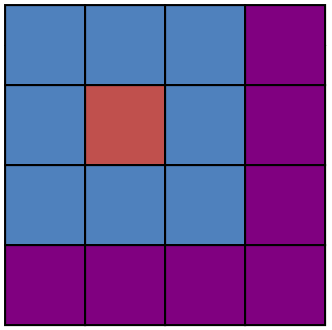
Mining Multimedia Databases

Spatial Relationships from Layout

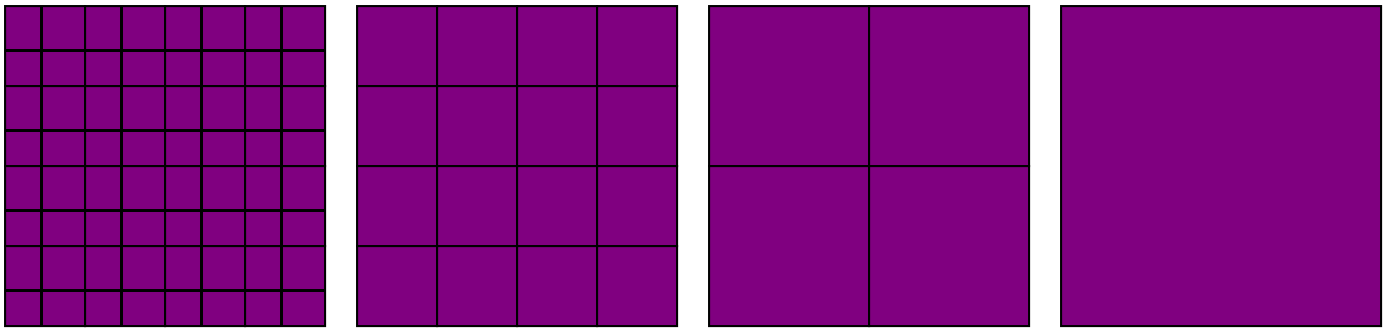
property **P1** ***on-top-of*** property **P2**

property **P1** ***next-to*** property **P2**



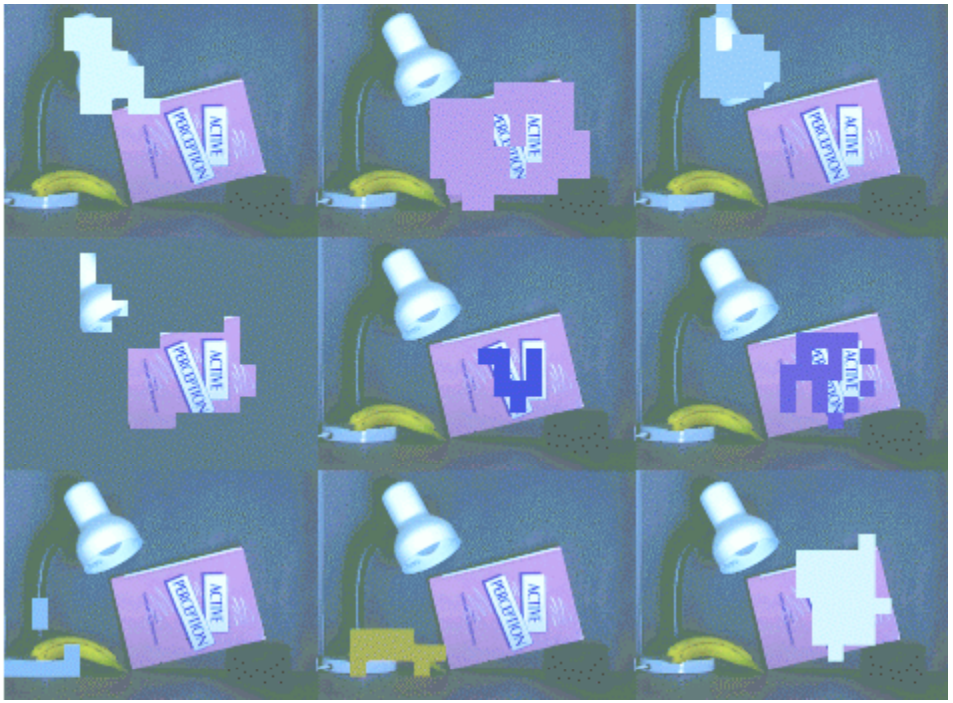


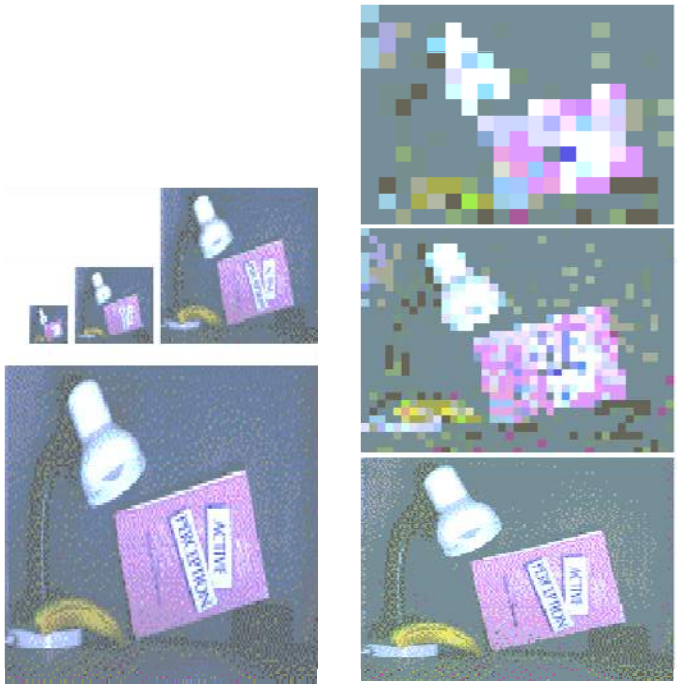
Different Resolution Hierarchy



Mining Multimedia Databases

From Coarse to Fine Resolution Mining





Challenge: Curse of Dimensionality

- Difficult to implement a data cube efficiently given a large number of dimensions, especially serious in the case of multimedia data cubes
- Many of these attributes are set-oriented instead of single-valued
- Restricting number of dimensions may lead to the modeling of an image at a rather rough, limited, and imprecise scale
- More research is needed to strike a balance between efficiency and power of representation

Lecture-53

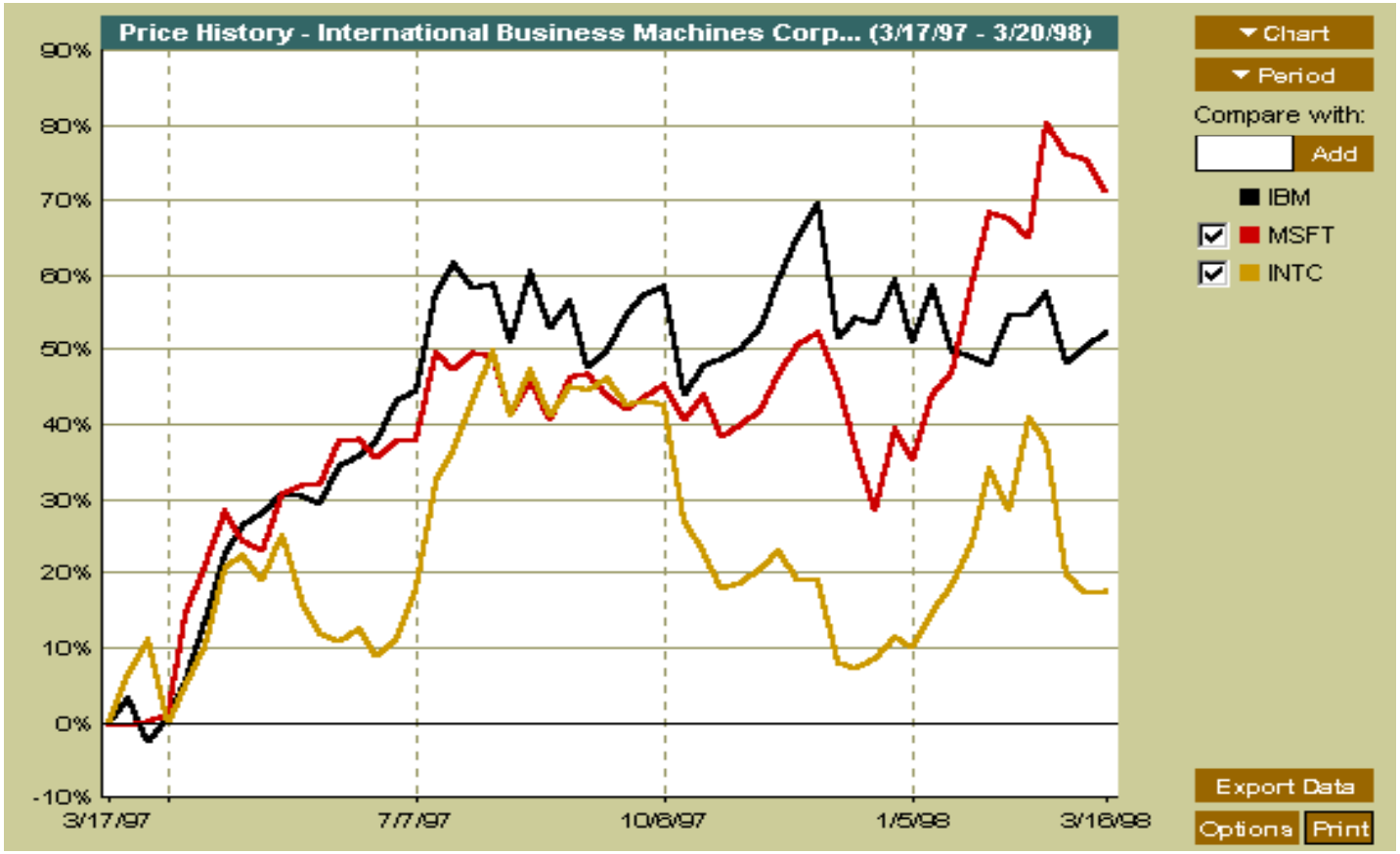
Mining time-series and sequence data

Mining Time-Series and Sequence Data

- Time-series database
  - Consists of sequences of values or events changing with time
  - Data is recorded at regular intervals
  - Characteristic time-series components
    - Trend, cycle, seasonal, irregular
- Applications
  - Financial: stock price, inflation
  - Biomedical: blood pressure
  - Meteorological: precipitation

Mining Time-Series and Sequence Data

Time-series plot



Mining Time-Series and Sequence Data: Trend analysis

- A time series can be illustrated as a time-series graph which describes a point moving with the passage of time
- Categories of Time-Series Movements
  - Long-term or trend movements (trend curve)
  - Cyclic movements or cycle variations, e.g., business cycles
  - Seasonal movements or seasonal variations
    - i.e, almost identical patterns that a time series appears to follow during corresponding months of successive years.
  - Irregular or random movements

Estimation of Trend Curve

- The freehand method
  - Fit the curve by looking at the graph
  - Costly and barely reliable for large-scaled data mining
- The least-square method
  - Find the curve minimizing the sum of the squares of the deviation of points on the curve from the corresponding data points
- The moving-average method
  - Eliminate cyclic, seasonal and irregular patterns
  - Loss of end data
  - Sensitive to outliers

Discovery of Trend in Time-Series

- Estimation of seasonal variations
  - Seasonal index
    - Set of numbers showing the relative values of a variable during the months of the year
    - E.g., if the sales during October, November, and December are 80%, 120%, and 140% of the average monthly sales for the whole year, respectively, then 80, 120, and 140 are seasonal index numbers for these months
  - Deseasonalized data
    - Data adjusted for seasonal variations
    - E.g., divide the original monthly data by the seasonal index numbers for the corresponding months

Discovery of Trend in Time-Series

- Estimation of cyclic variations
  - If (approximate) periodicity of cycles occurs, cyclic index can be constructed in much the same manner as seasonal indexes
- Estimation of irregular variations
  - By adjusting the data for trend, seasonal and cyclic variations
- With the systematic analysis of the trend, cyclic, seasonal, and irregular components, it is possible to make long- or short-term predictions with reasonable quality

Similarity Search in Time-Series Analysis

- Normal database query finds exact match
- Similarity search finds data sequences that differ only slightly from the given query sequence
- Two categories of similarity queries
  - Whole matching: find a sequence that is similar to the query sequence
  - Subsequence matching: find all pairs of similar sequences
- Typical Applications
  - Financial market
  - Market basket data analysis
  - Scientific databases
  - Medical diagnosis

Data transformation

- Many techniques for signal analysis require the data to be in the frequency domain
- Usually data-independent transformations are used
  - The transformation matrix is determined a priori
    - E.g., discrete Fourier transform (DFT), discrete wavelet transform (DWT)
  - The distance between two signals in the time domain is the same as their Euclidean distance in the frequency domain
  - DFT does a good job of concentrating energy in the first few coefficients
  - If we keep only first a few coefficients in DFT, we can compute the lower bounds of the actual distance

Multidimensional Indexing

- Multidimensional index
  - Constructed for efficient accessing using the first few Fourier coefficients
- Use the index can to retrieve the sequences that are at most a certain small distance away from the query sequence
- Perform postprocessing by computing the actual distance between sequences in the time domain and discard any false matches

Subsequence Matching

- Break each sequence into a set of pieces of window with length *w*
- Extract the features of the subsequence inside the window
- Map each sequence to a “trail” in the feature space
- Divide the trail of each sequence into “subtrails” and represent each of them with minimum bounding rectangle
- Use a multipiece assembly algorithm to search for longer sequence matches

Enhanced similarity search methods

- Allow for gaps within a sequence or differences in offsets or amplitudes
- Normalize sequences with amplitude scaling and offset translation
- Two subsequences are considered similar if one lies within an envelope of ε width around the other, ignoring outliers
- Two sequences are said to be similar if they have enough non-overlapping time-ordered pairs of similar subsequences
- Parameters specified by a user or expert: sliding window size, width of an envelope for similarity, maximum gap, and matching fraction

Steps for performing a similarity search

- Atomic matching
  - Find all pairs of gap-free windows of a small length that are similar
- Window stitching
  - Stitch similar windows to form pairs of large similar subsequences allowing gaps between atomic matches
- Subsequence Ordering
  - Linearly order the subsequence matches to determine whether enough similar pieces exist

Query Languages for Time Sequences

- Time-sequence query language
  - Should be able to specify sophisticated queries like

  Find all of the sequences that are similar to some sequence in class *A*, but not similar to any sequence in class *B*
  - Should be able to support various kinds of queries: range queries, all-pair queries, and nearest neighbor queries
- Shape definition language
  - Allows users to define and query the overall shape of time sequences
  - Uses human readable series of sequence transitions or macros
  - Ignores the specific details
    - E.g., the pattern up, Up, UP can be used to describe increasing degrees of rising slopes
    - Macros: spike, valley, etc.

Sequential Pattern Mining

- Mining of frequently occurring patterns related to time or other sequences
- Sequential pattern mining usually concentrate on symbolic patterns
- Examples
  - Renting “Star Wars”, then “Empire Strikes Back”, then “Return of the Jedi” in that order
  - Collection of ordered events within an interval
- Applications
  - Targeted marketing
  - Customer retention
  - Weather prediction

Mining Sequences (cont.)

Customer-sequence

CustId	Video sequence
1	{(C), (H)}
2	{(AB), (C), (DFG)}
3	{(CEG)}
4	{(C), (DG), (H)}
5	{(H)}

Map Large Itemsets

Large Itemsets	MappedID
(C)	1
(D)	2
(G)	3
(DG)	4
(H)	5

Sequential patterns with support > 0.25

{(C), (H)}

{(C), (DG)}

Sequential pattern mining: Cases and Parameters

- Duration of a time sequence *T*
  - Sequential pattern mining can then be confined to the data within a specified duration
  - Ex. Subsequence corresponding to the year of 1999
  - Ex. Partitioned sequences, such as every year, or every week after stock crashes, or every two weeks before and after a volcano eruption
- Event folding window *w*
  - If *w = T*, time-insensitive frequent patterns are found
  - If *w = 0* (no event sequence folding), sequential patterns are found where each event occurs at a distinct time instant
  - If *0 < w < T*, sequences occurring within the same period *w* are folded in the analysis

Sequential pattern mining: Cases and Parameters

- Time interval, *int*, between events in the discovered pattern
  - *int* = 0: no interval gap is allowed, i.e., only strictly consecutive sequences are found
    - Ex. “Find frequent patterns occurring in consecutive weeks”
  - *min_int* $\leq$ *int* $\leq$ *max_int*: find patterns that are separated by at least *min_int* but at most *max_int*
    - Ex. “If a person rents movie A, it is likely she will rent movie B within 30 days” (*int* $\leq$ 30)
  - *int* = c $\neq$ 0: find patterns carrying an exact interval
    - Ex. “Every time when Dow Jones drops more than 5%, what will happen exactly two days later?” (*int* = 2)

Episodes and Sequential Pattern Mining Methods

- Other methods for specifying the kinds of patterns
  - Serial episodes: A $\rightarrow$ B
  - Parallel episodes: A & B
  - Regular expressions: (A | B)C*(D $\rightarrow$ E)
- Methods for sequential pattern mining
  - Variations of Apriori-like algorithms, e.g., GSP
  - Database projection-based pattern growth
    - Similar to the frequent pattern growth without candidate generation

Periodicity Analysis

- Periodicity is everywhere: tides, seasons, daily power consumption, etc.
- Full periodicity
  - Every point in time contributes (precisely or approximately) to the periodicity
- Partial periodicit: A more general notion
  - Only some segments contribute to the periodicity
    - Jim reads NY Times 7:00-7:30 am every week day
- Cyclic association rules
  - Associations which form cycles
- Methods
  - Full periodicity: FFT, other statistical analysis methods
  - Partial and cyclic periodicity: Variations of Apriori-like mining methods

Lecture-54

Mining text databases

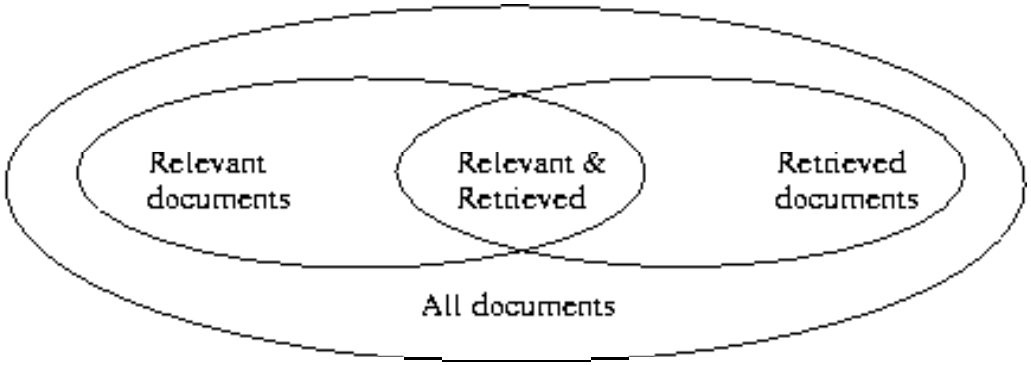
Text Databases and IR

- Text databases (document databases)
  - Large collections of documents from various sources: news articles, research papers, books, digital libraries, e-mail messages, and Web pages, library database, etc.
  - Data stored is usually *semi-structured*
  - Traditional information retrieval techniques become inadequate for the increasingly vast amounts of text data
- Information retrieval
  - A field developed in parallel with database systems
  - Information is organized into (a large number of) documents
  - Information retrieval problem: locating relevant documents based on user input, such as keywords or example documents

Information Retrieval

- Typical IR systems
  - Online library catalogs
  - Online document management systems
- Information retrieval vs. database systems
  - Some DB problems are not present in IR, e.g., update, transaction management, complex objects
  - Some IR problems are not addressed well in DBMS, e.g., unstructured documents, approximate search using keywords and relevance

Basic Measures for Text Retrieval



- 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

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

Keyword-Based Retrieval

- A document is represented by a string, which can be identified by a set of keywords
- Queries may use expressions of keywords
  - E.g., car *and* repair shop, tea *or* coffee, DBMS *but not* Oracle
  - Queries and retrieval should consider synonyms, e.g., repair and maintenance
- Major difficulties of the model
  - Synonymy: A keyword *T* does not appear anywhere in the document, even though the document is closely related to *T*, e.g., data mining
  - Polysemy: The same keyword may mean different things in different contexts, e.g., mining

Similarity-Based Retrieval in Text Databases

- 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.
- Basic techniques
- Stop list
    - Set of words that are deemed “irrelevant”, even though they may appear frequently
    - E.g., *a, the, of, for, with*, etc.
    - Stop lists may vary when document set varies

Similarity-Based Retrieval in Text Databases

- Word stem
  - Several words are small syntactic variants of each other since they share a common word stem
  - E.g., *drug*, *drugs, drugged*
- A term frequency table
  - Each entry *frequent_table(i, j)* = # of occurrences of the word t_i in document d_i
  - 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:

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

Latent Semantic Indexing

- Basic idea
  - Similar documents have similar word frequencies
  - Difficulty: the size of the term frequency matrix is very large
  - Use a singular value decomposition (SVD) techniques to reduce the size of frequency table
  - Retain the *K* most significant rows of the frequency table
- Method
  - Create a term frequency matrix, *freq_matrix*
  - SVD construction: Compute the singular valued decomposition of *freq_matrix* by splitting it into 3 matrices, *U*, *S*, *V*
  - Vector identification: For each document *d*, replace its original document vector by a new excluding the eliminated terms
  - Index creation: Store the set of all vectors, indexed by one of a number of techniques (such as TV-tree)

Other Text Retrieval Indexing Techniques

- Inverted index
  - Maintains two hash- or B+-tree indexed tables:
    - document_table: a set of document records <doc_id, postings_list>
    - term_table: a set of term records, <term, postings_list>
  - Answer query: Find all docs associated with one or a set of terms
  - Advantage: easy to implement
  - Disadvantage: do not handle well synonymy and polysemy, and posting lists could be too long (storage could be very large)
- Signature file
  - Associate a signature with each document
  - A signature is a representation of an ordered list of terms that describe the document
  - Order is obtained by frequency analysis, stemming and stop lists

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
    - Anchor text correlations with linked objects

Keyword-based association analysis

- Collect sets of keywords or terms that occur frequently together and then find the association or correlation relationships among them
- First preprocess the text data by parsing, stemming, removing stop words, etc.
- Then 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

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: The computer system discovers the classification rules
  - Application: The discovered rules can be applied to classify new/unknown documents
- Text document classification differs from the classification of relational data
  - Document databases are not structured according to attribute-value pairs

Association-Based Document Classification

- Extract keywords and terms by information retrieval and simple association analysis techniques
- Obtain concept hierarchies of keywords and terms using
  - Available term classes, such as WordNet
  - Expert knowledge
  - Some keyword classification systems
- Classify documents in the training set into class hierarchies
- Apply term association mining method to discover sets of associated terms
- Use the terms to maximally distinguish one class of documents from others
- Derive a set of association rules associated with each document class
- Order the classification rules based on their occurrence frequency and discriminative power
- Used the rules to classify new documents

Document Clustering

- Automatically group related documents based on their contents
- Require no training sets or predetermined taxonomies, generate a taxonomy at runtime
- Major steps
  - Preprocessing
    - Remove stop words, stem, feature extraction, lexical analysis, …
  - Hierarchical clustering
    - Compute similarities applying clustering algorithms, …
  - Slicing
    - Fan out controls, flatten the tree to configurable number of levels, …

Lecture-55

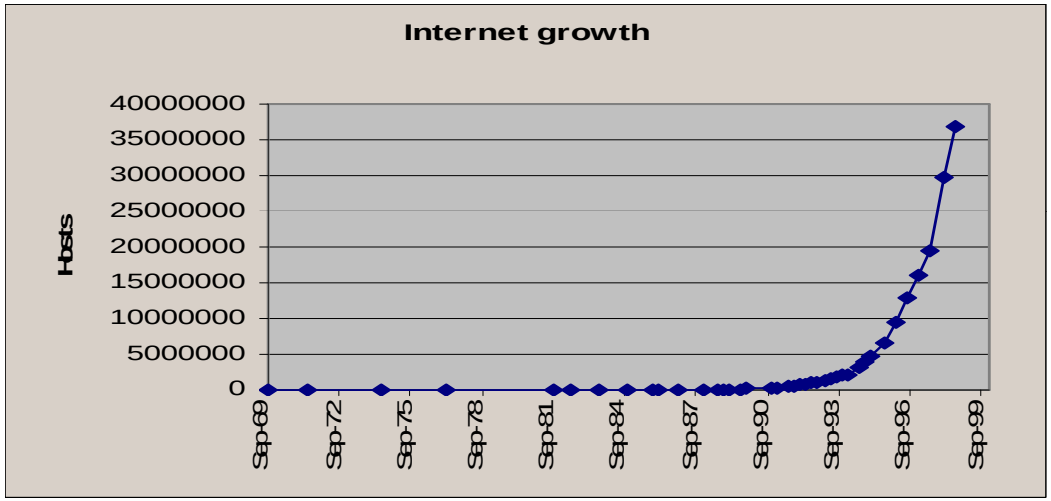
Mining the World-Wide Web

Mining the World-Wide Web

- The WWW is huge, widely distributed, global information service center for
  - Information services: news, advertisements, consumer information, financial management, education, government, e-commerce, etc.
  - Hyper-link information
  - Access and usage information
- WWW provides rich sources for data mining
- Challenges
  - Too huge for effective data warehousing and data mining
  - Too complex and heterogeneous: no standards and structure

Mining the World-Wide Web

- Growing and changing very rapidly



- Broad diversity of user communities
- Only a small portion of the information on the Web is truly relevant or useful
  - 99% of the Web information is useless to 99% of Web users
  - How can we find high-quality Web pages on a specified topic?

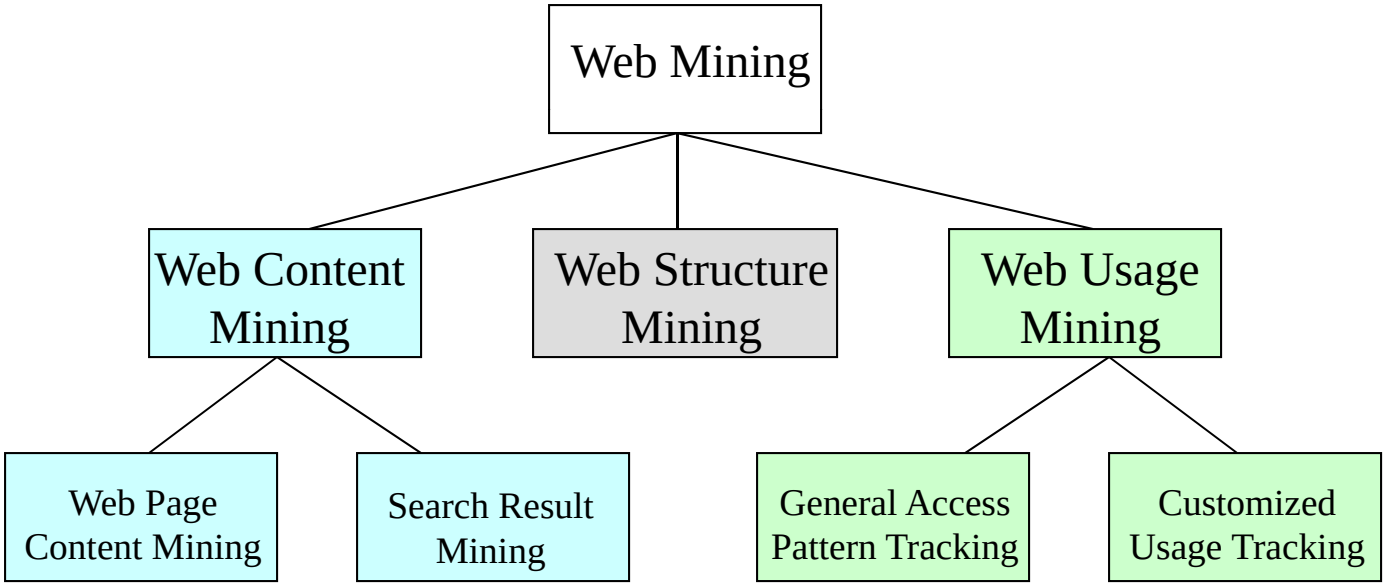
Web search engines

- Index-based: search the Web, index Web pages, and build and store huge keyword-based indices
- Help 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 (polysemy)

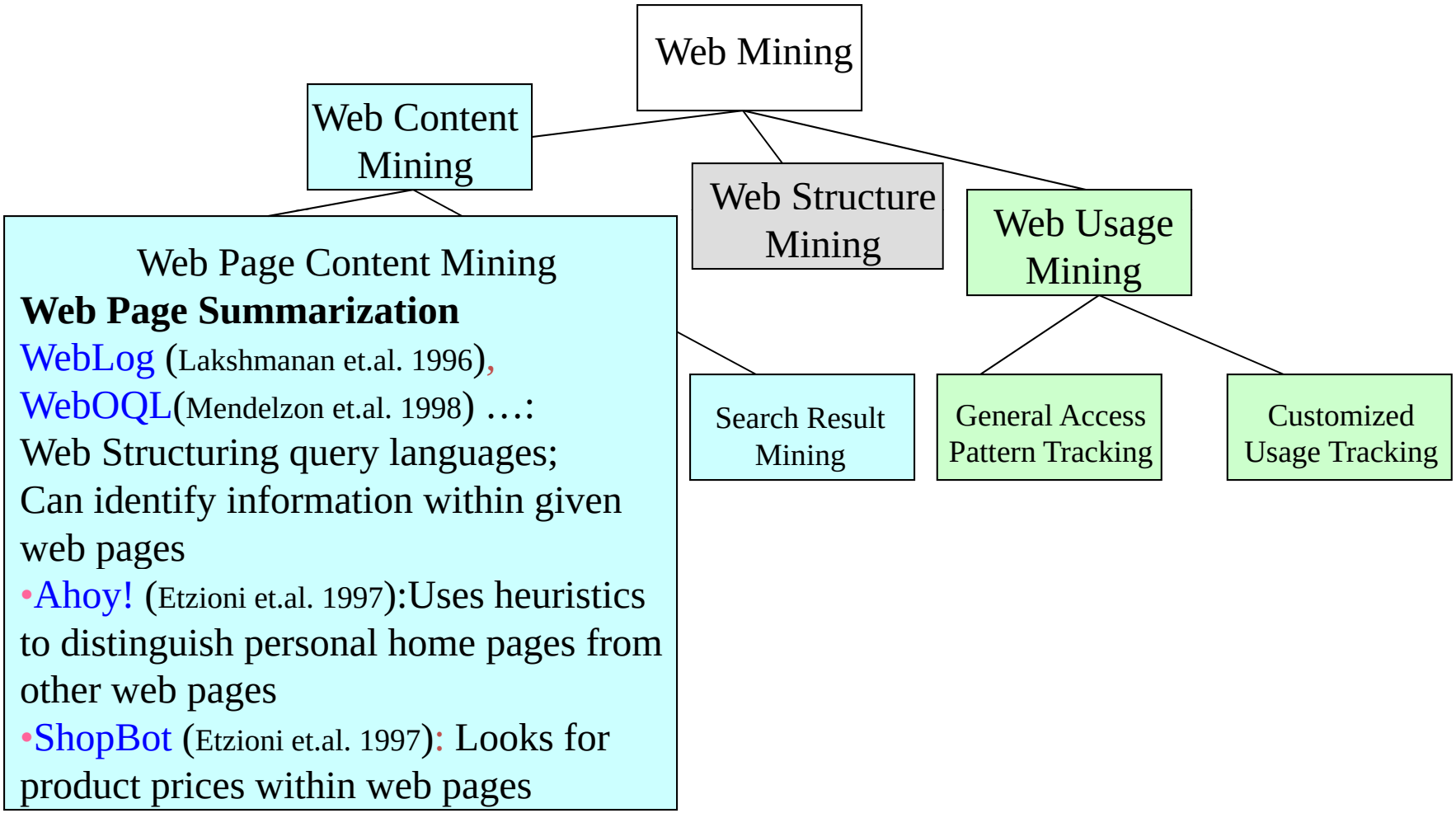
Web Mining: A more challenging task

- Searches for
  - Web access patterns
  - Web structures
  - Regularity and dynamics of Web contents
- Problems
  - The “abundance” problem
  - Limited coverage of the Web: hidden Web sources, majority of data in DBMS
  - Limited query interface based on keyword-oriented search
  - Limited customization to individual users

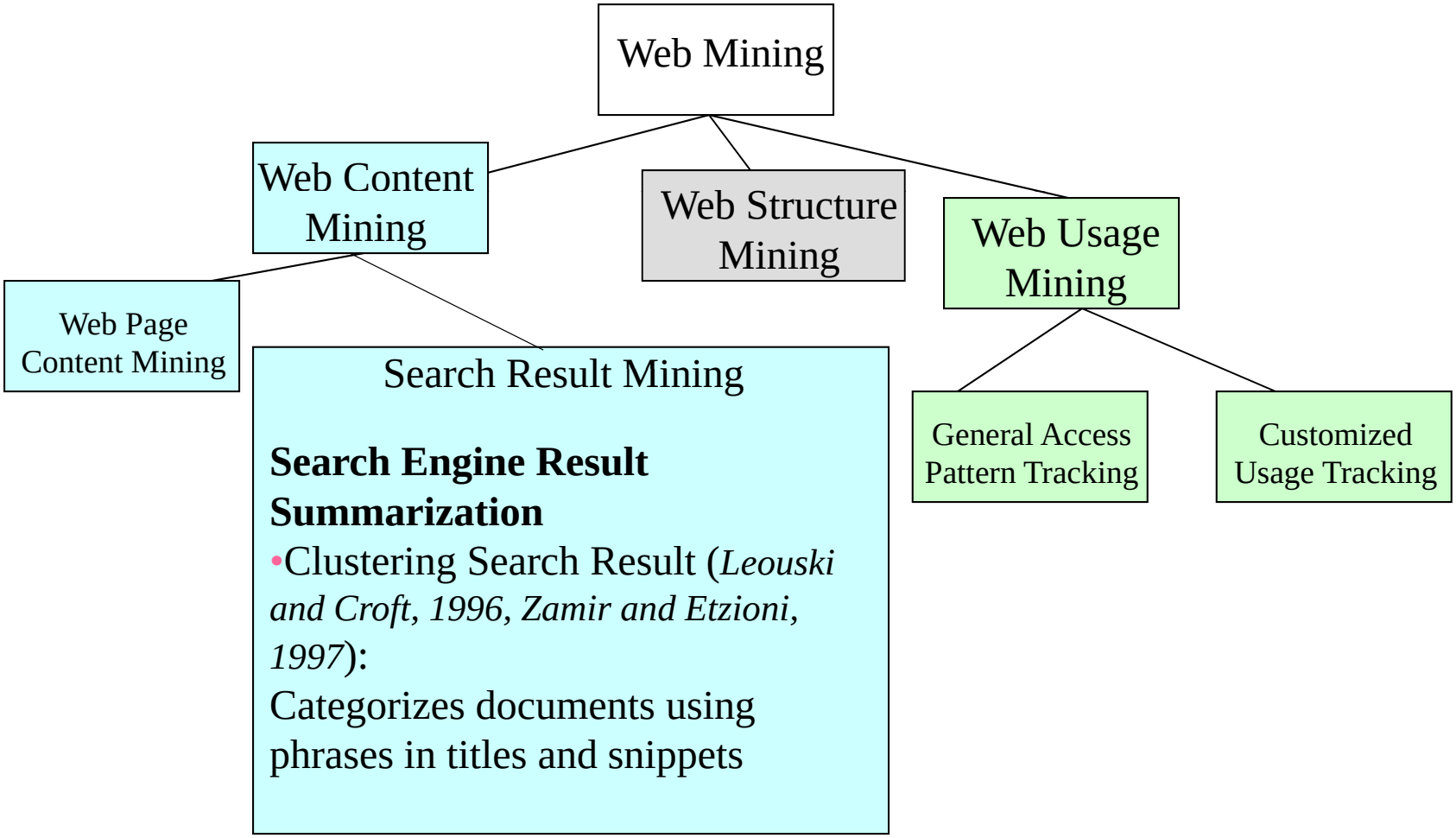
Web Mining Taxonomy



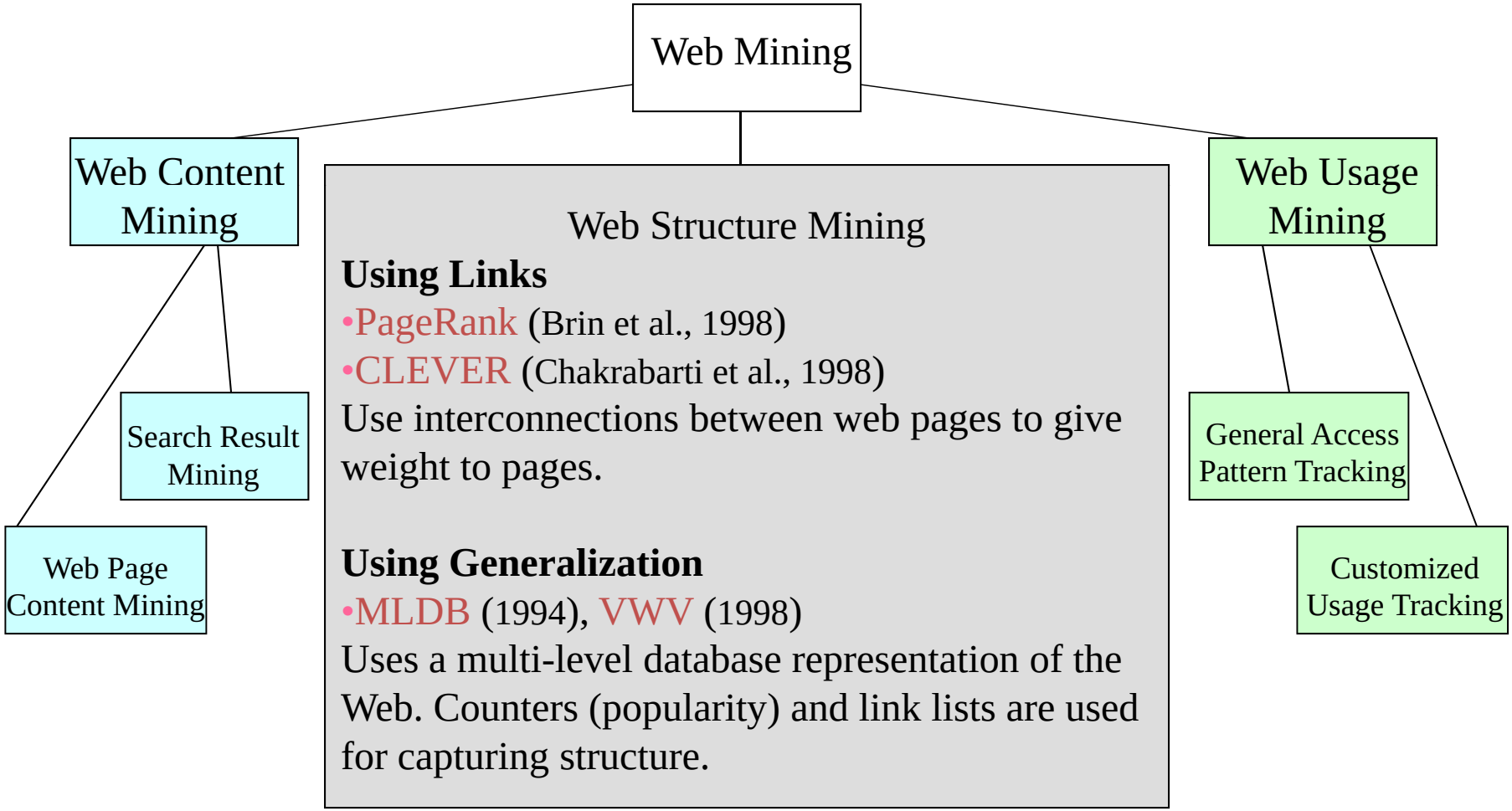
Mining the World-Wide Web



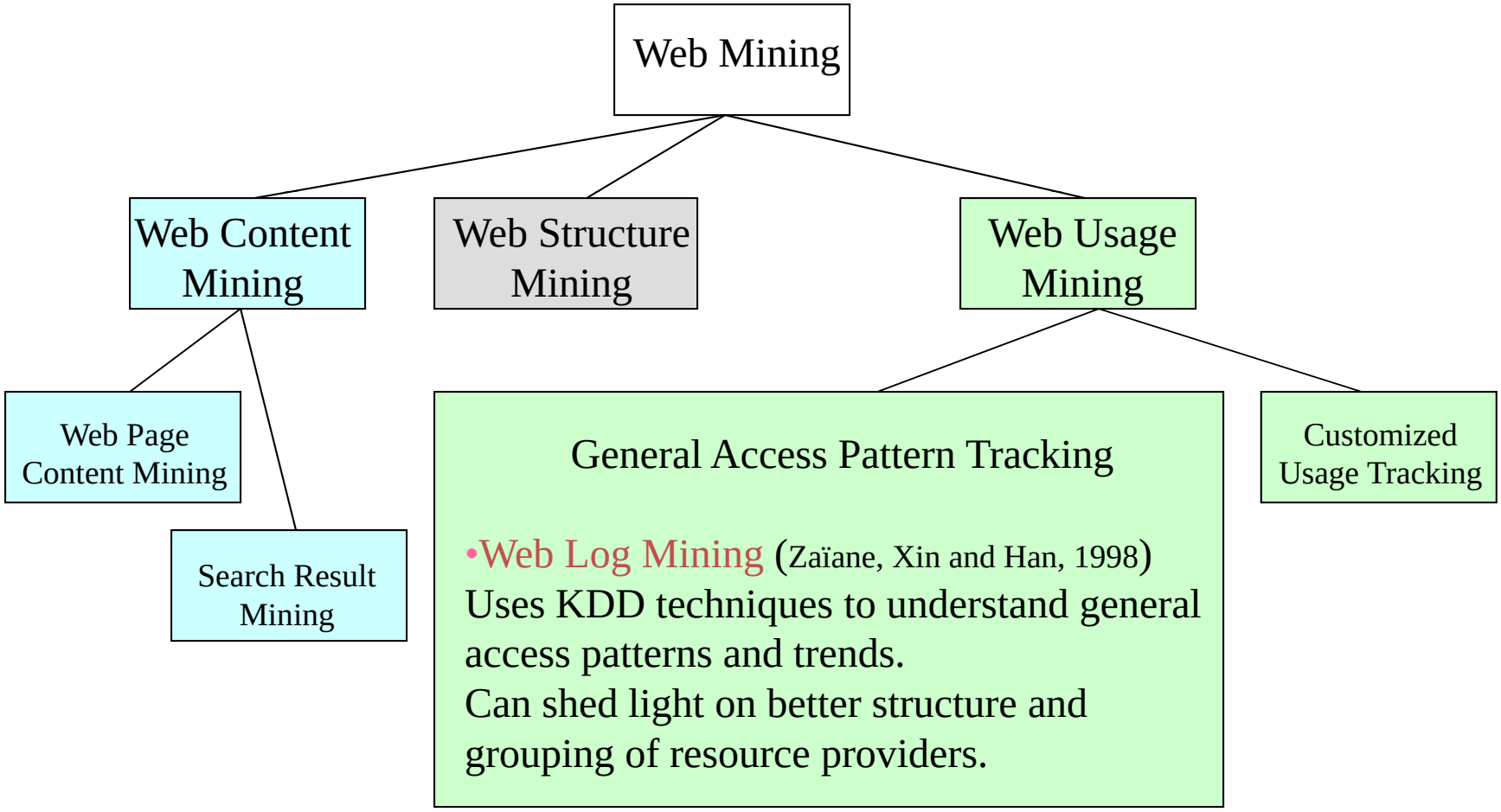
Mining the World-Wide Web



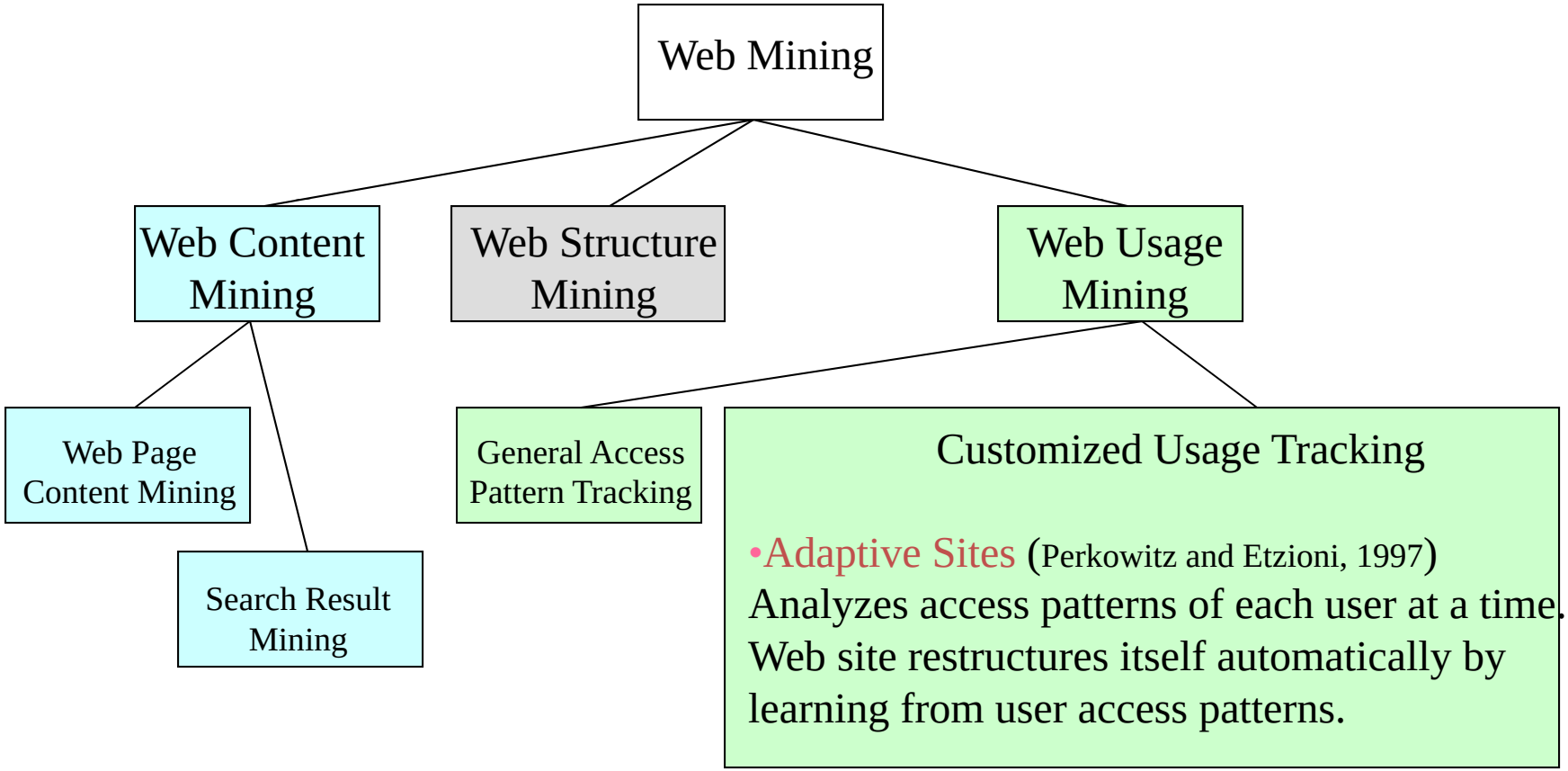
Mining the World-Wide Web



Mining the World-Wide Web



Mining the World-Wide Web



Mining the Web's Link Structures

- 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
  - The Web consists not only of pages, but also of hyperlinks pointing from one page to another
  - These hyperlinks contain an enormous amount of latent human annotation
  - A hyperlink pointing to another Web page, this can be considered as the author's endorsement of the other page

Mining the Web's Link Structures

- Problems with the Web linkage structure
  - Not every hyperlink represents an endorsement
    - Other purposes are for navigation or for paid advertisements
    - If the majority of hyperlinks are for endorsement, the collective opinion will still dominate
  - One authority will seldom have its Web page point to its rival authorities in the same field
  - Authoritative pages are seldom particularly descriptive
- Hub
  - Set of Web pages that provides collections of links to authorities

HITS (Hyperlink-Induced Topic Search)

- Explore interactions between hubs and authoritative pages
- Use an index-based search engine to form the root set
  - Many of these pages are presumably relevant to the search topic
  - Some of them should contain links to most of the prominent authorities
- 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, up to a designated size cutoff
- Apply weight-propagation
  - An iterative process that determines numerical estimates of hub and authority weights

Systems Based on HITS

  - Output a short list of the pages with large hub weights, and the pages with large authority weights for the given search topic

- Systems based on the HITS algorithm
  - Clever, Google: achieve better quality search results than those generated by term-index engines such as AltaVista and those created by human ontologists such as Yahoo!
- Difficulties from ignoring textual contexts
  - Drifting: when hubs contain multiple topics
  - Topic hijacking: when many pages from a single Web site point to the same single popular site

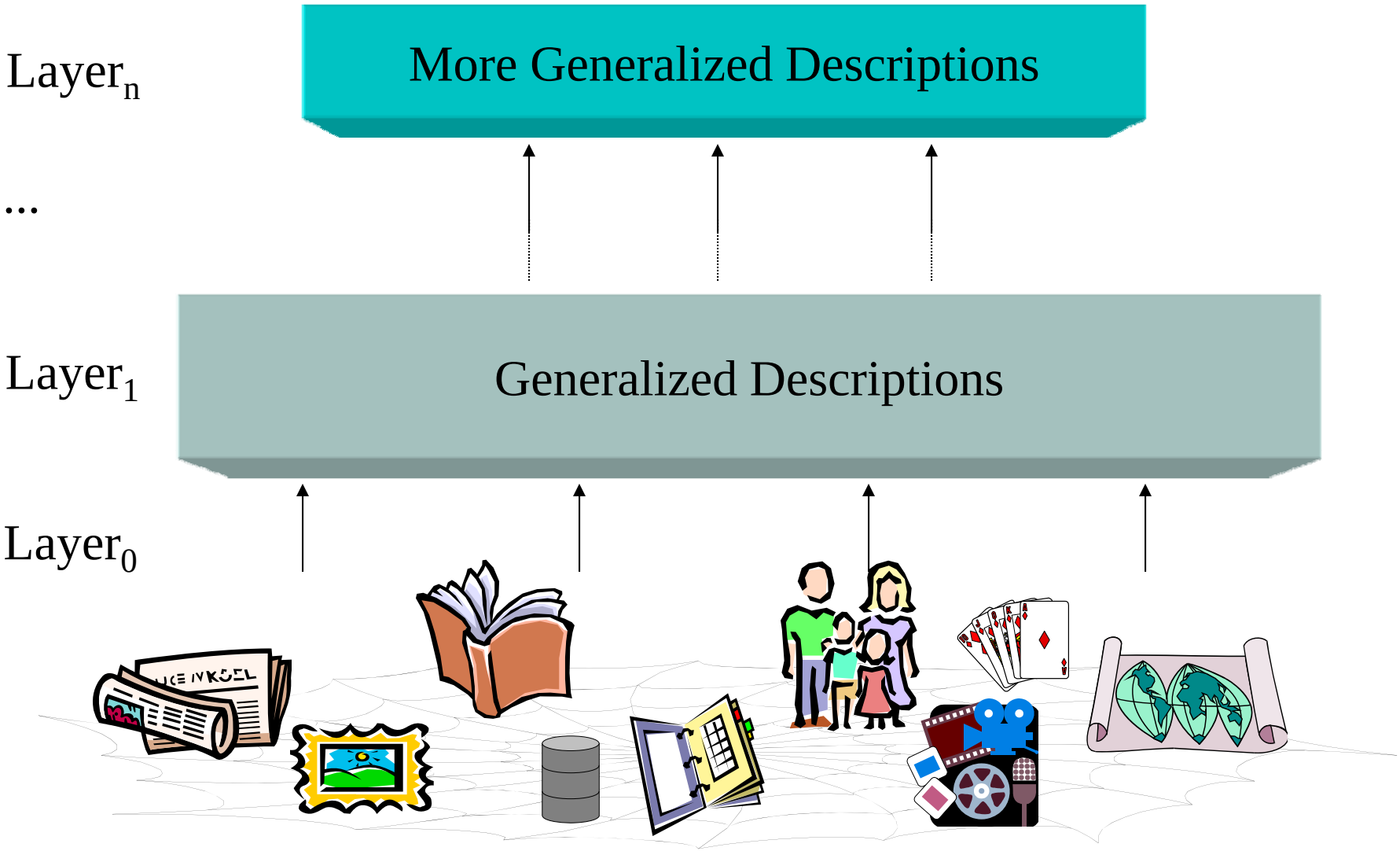
Automatic Classification of Web Documents

- Assign a class label to each document 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 scheme
  - Use the scheme classify new Web documents by assigning categories from the same taxonomy
- Keyword-based document classification methods
- Statistical models

Multilayered Web Information Base

- $Layer_0$: the Web itself
- $Layer_1$: the Web page descriptor layer
  - Contains descriptive information for pages on the Web
  - An abstraction of $Layer_0$: substantially smaller but still rich enough to preserve most of the interesting, general information
  - Organized into dozens of semistructured classes
    - *document, person, organization, ads, directory, sales, software, game, stocks, library_catalog, geographic_data, scientific_data*, etc.
- $Layer_2$ and up: various Web directory services constructed on top of $Layer_1$
  - provide multidimensional, application-specific services

Multiple Layered Web Architecture



Mining the World-Wide Web

Layer-0: Primitive data

Layer-1: dozen database relations representing types of objects (metadata)

document, organization, person, software, game, map, image,…

• **document**(file_addr, authors, title, publication, publication_date, abstract, language, table_of_contents, category_description, keywords, index, multimedia_attached, num_pages, format, first_paragraphs, size_doc, timestamp, access_frequency, links_out,...)

• **person**(last_name, first_name, home_page_addr, position, picture_attached, phone, e-mail, office_address, education, research_interests, publications, size_of_home_page, timestamp, access_frequency, ...)

• **image**(image_addr, author, title, publication_date, category_description, keywords, size, width, height, duration, format, parent_pages, colour_histogram, Colour_layout, Texture_layout, Movement_vector, localisation_vector, timestamp, access_frequency, ...)

Mining the World-Wide Web

Layer-2: simplification of layer-1

•**doc_brief**(file_addr, authors, title, publication, publication_date, abstract, language, category_description, key_words, major_index, num_pages, format, size_doc, access_frequency, links_out)

•**person_brief** (last_name, first_name, publications,affiliation, e-mail, research_interests, size_home_page, access_frequency)

Layer-3: generalization of layer-2

•**cs_doc**(file_addr, authors, title, publication, publication_date, abstract, language, category_description, keywords, num_pages, form, size_doc, links_out)

•**doc_summary**(affiliation, field, publication_year, count, first_author_list, file_addr_list)

•**doc_author_brief**(file_addr, authors, affiliation, title, publication, pub_date, category_description, keywords, num_pages, format, size_doc, links_out)

•**person_summary**(affiliation, research_interest, year, num_publications, count)

XML and Web Mining

- XML can help to extract the correct descriptors
  - Standardization would greatly facilitate information extraction

```
<NAME> eXtensible Markup Language</NAME>
<RECOM>World-Wide Web Consortium</RECOM>
<SINCE>1998</SINCE>
<VERSION>1.0</VERSION>
<DESC>Meta language that facilitates more meaningful and
precise declarations of document content</DESC>
<HOW>Definition of new tags and DTDs</HOW>
```

  - Potential problem
    - XML can help solve heterogeneity for vertical applications, but the freedom to define tags can make horizontal applications on the Web more heterogeneous

Benefits of Multi-Layer Meta-Web

- Benefits:
  - Multi-dimensional Web info summary analysis
  - Approximate and intelligent query answering
  - Web high-level query answering (WebSQL, WebML)
  - Web content and structure mining
  - Observing the dynamics/evolution of the Web
- Is it realistic to construct such a meta-Web?
  - Benefits even if it is partially constructed
  - Benefits may justify the cost of tool development, standardization and partial restructuring

Web Usage Mining

- Mining Web log records to discover user access patterns of Web pages
- Applications
  - Target potential customers for electronic commerce
  - Enhance the quality and delivery of Internet information services to the end user
  - Improve Web server system performance
  - Identify potential prime advertisement locations
- Web logs provide rich information about Web dynamics
  - Typical Web log entry includes the URL requested, the IP address from which the request originated, and a timestamp

Techniques for Web usage mining

- Construct multidimensional view on the Weblog database
  - Perform multidimensional OLAP analysis to find the top *N* users, top *N* accessed Web pages, most frequently accessed time periods, etc.
- Perform data mining on Weblog records
  - Find association patterns, sequential patterns, and trends of Web accessing
  - May need additional information,e.g., user browsing sequences of the Web pages in the Web server buffer
- Conduct studies to
  - Analyze system performance, improve system design by Web caching, Web page prefetching, and Web page swapping

Mining the World-Wide Web

- Design of a Web Log Miner
  - Web log is filtered to generate a relational database
  - A data cube is generated form database
  - OLAP is used to drill-down and roll-up in the cube
  - OLAM is used for mining interesting knowledge