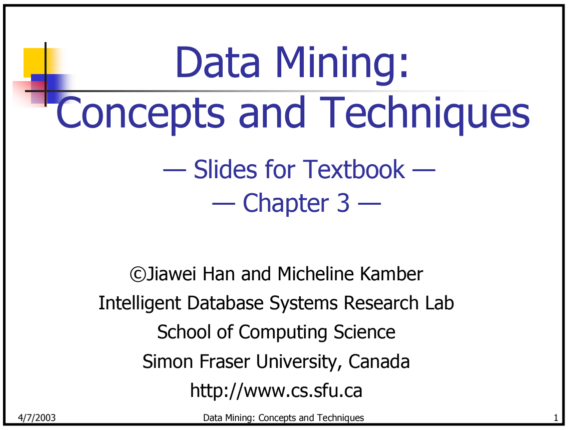
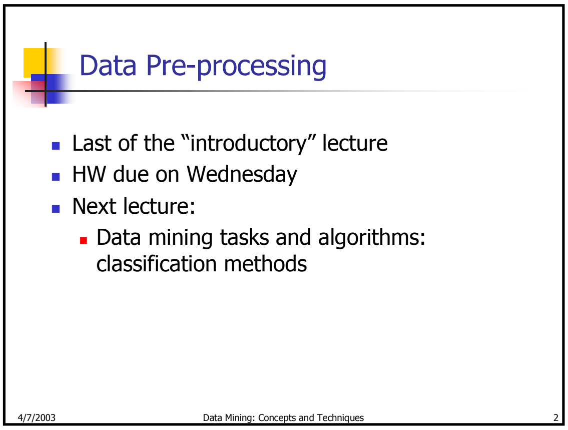


Data Mining: Concepts and Techniques

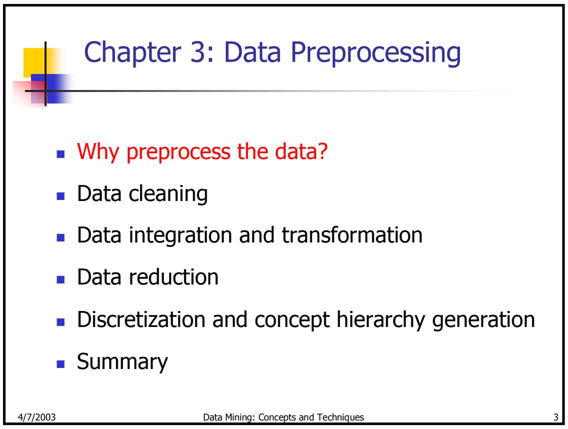
— Slides for Textbook —
— Chapter 3 —

©Jiawei Han and Micheline Kamber
Intelligent Database Systems Research Lab
School of Computing Science
Simon Fraser University, Canada
http://www.cs.sfu.ca



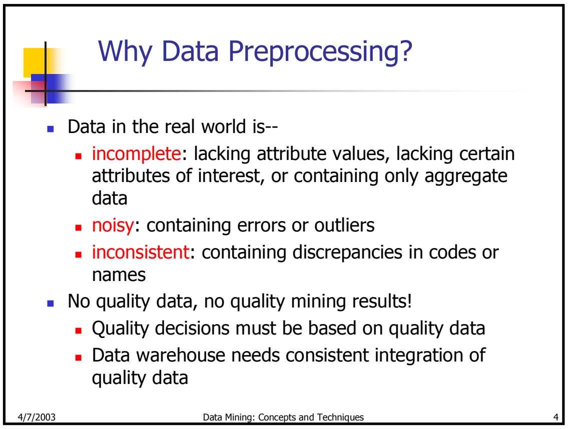
Data Pre-processing

- Last of the “introductory” lecture
- HW due on Wednesday
- Next lecture:
  - Data mining tasks and algorithms: classification methods



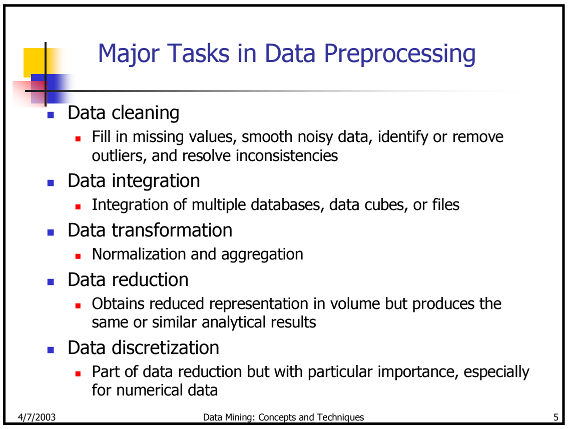
Chapter 3: Data Preprocessing

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



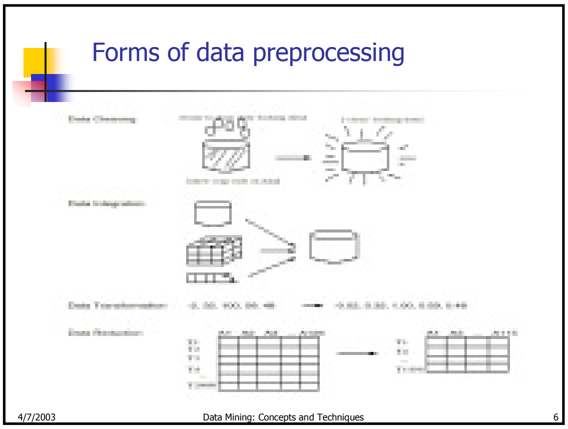
Why Data Preprocessing?

- Data in the real world is--
  - incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
  - noisy: containing errors or outliers
  - inconsistent: containing discrepancies in codes or names
- No quality data, no quality mining results!
  - Quality decisions must be based on quality data
  - Data warehouse needs consistent integration of quality data

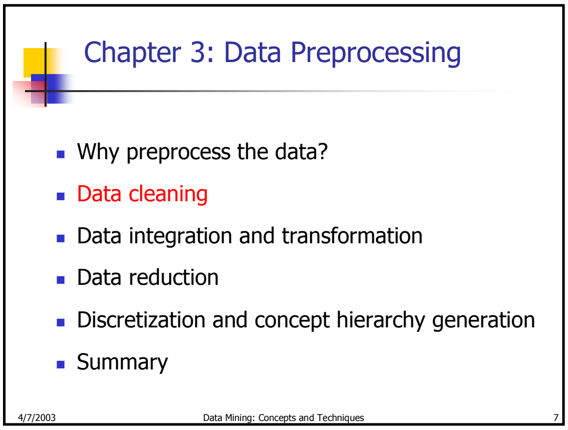


Major Tasks in Data Preprocessing

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

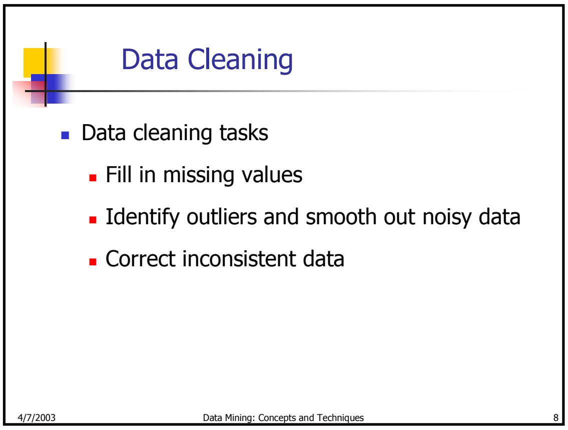


Forms of data preprocessing



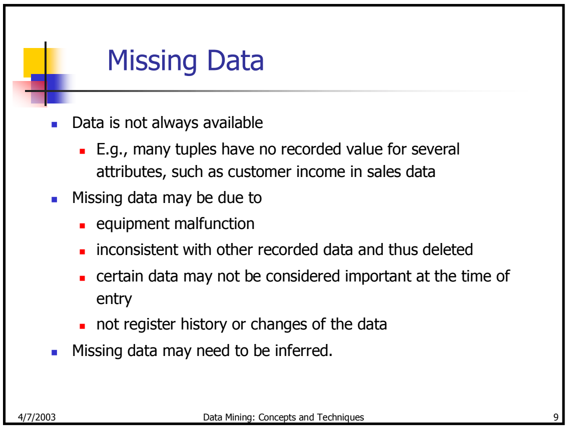
Chapter 3: Data Preprocessing

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



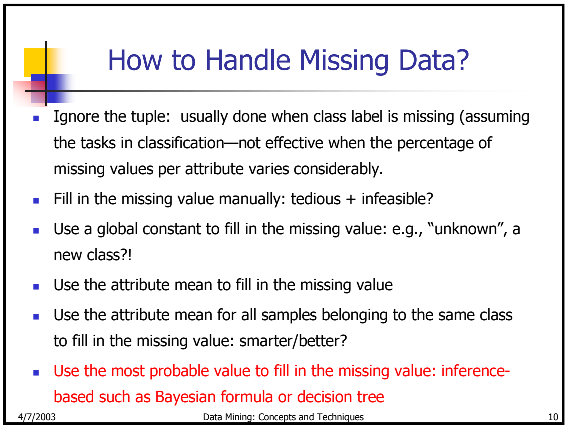
Data Cleaning

- Data cleaning tasks
  - Fill in missing values
  - Identify outliers and smooth out noisy data
  - Correct inconsistent data



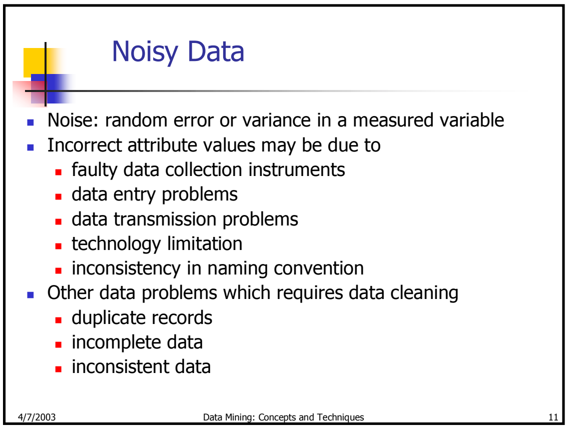
Missing Data

- Data is not always available
  - E.g., many tuples have no recorded value 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
  - certain data may not be considered important at the time of entry
  - not register history or changes of the data
- Missing data may need to be inferred.



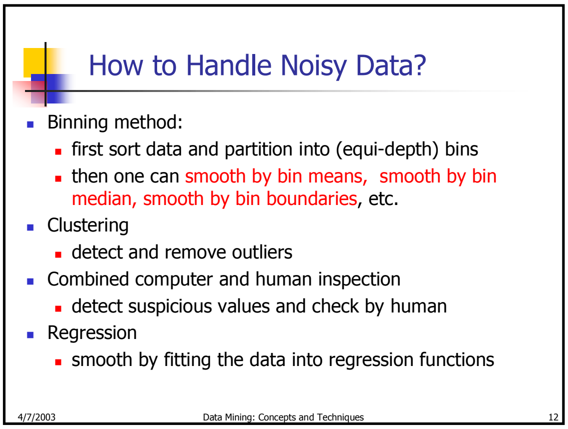
How to Handle Missing Data?

- Ignore the tuple: usually done when class label is missing (assuming the tasks in classification—not effective when the percentage of missing values per attribute varies considerably.
- Fill in the missing value manually: tedious + infeasible?
- Use a global constant to fill in the missing value: e.g., "unknown", a new class?!
- Use the attribute mean to fill in the missing value
- Use the attribute mean for all samples belonging to the same class to fill in the missing value: smarter/better?
- Use the most probable value to fill in the missing value: inference-based such as Bayesian formula or decision tree



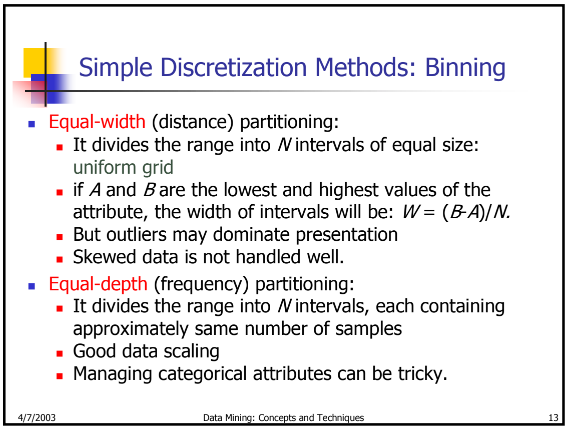
Noisy Data

- Noise: random error or variance in a measured variable
- Incorrect attribute values may be due to
  - faulty data collection instruments
  - data entry problems
  - data transmission problems
  - technology limitation
  - inconsistency in naming convention
- Other data problems which requires data cleaning
  - duplicate records
  - incomplete data
  - inconsistent data



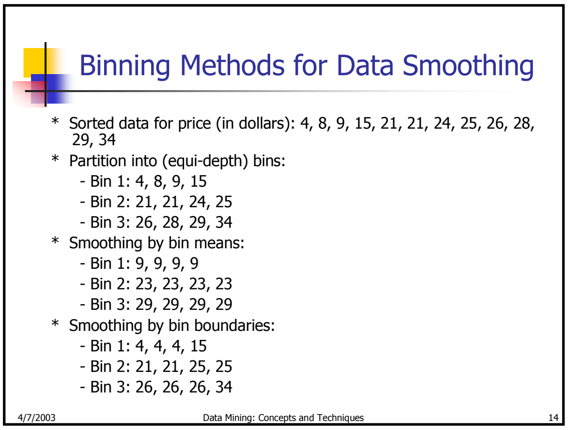
How to Handle Noisy Data?

- 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
- Regression
  - smooth by fitting the data into regression functions



Simple Discretization Methods: Binning

- Equal-width (distance) partitioning:
  - It divides the range into *N* intervals of equal size: uniform grid
  - if *A* and *B* are the lowest and highest values of the attribute, the width of intervals will be: *W* = (*B*-*A*)/*N.*
  - But outliers may dominate presentation
  - Skewed data is not handled well.
- Equal-depth (frequency) partitioning:
  - It divides the range into *N* intervals, each containing approximately same number of samples
  - Good data scaling
  - Managing categorical attributes can be tricky.



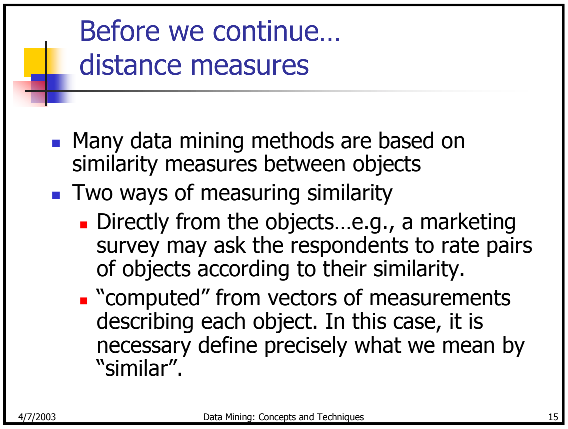
Binning Methods for Data Smoothing

* 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

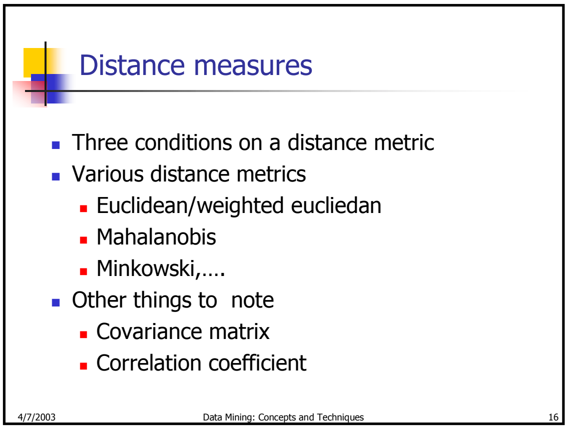
* Smoothing by bin means:
- Bin 1: 9, 9, 9, 9
- Bin 2: 23, 23, 23, 23
- Bin 3: 29, 29, 29, 29

* Smoothing by bin boundaries:
- Bin 1: 4, 4, 4, 15
- Bin 2: 21, 21, 25, 25
- Bin 3: 26, 26, 26, 34



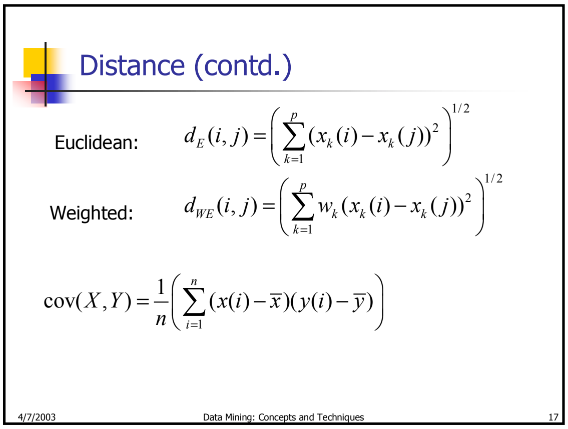
Before we continue… distance measures

- Many data mining methods are based on similarity measures between objects
- Two ways of measuring similarity
  - Directly from the objects…e.g., a marketing survey may ask the respondents to rate pairs of objects according to their similarity.
  - "computed" from vectors of measurements describing each object. In this case, it is necessary define precisely what we mean by "similar".



Distance measures

- Three conditions on a distance metric
- Various distance metrics
  - Euclidean/weighted eucliedan
  - Mahalanobis
  - Minkowski,….
- Other things to note
  - Covariance matrix
  - Correlation coefficient

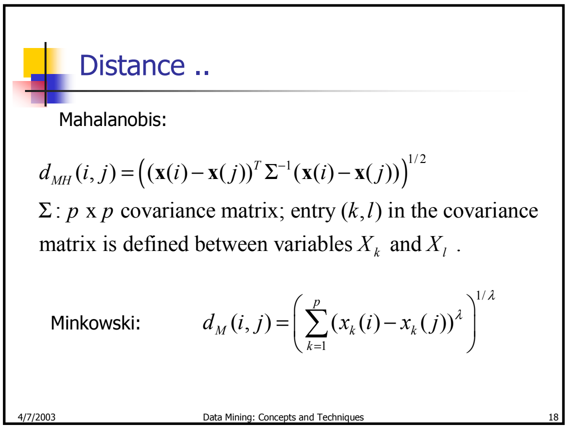


Distance (contd.)

Euclidean: $$d_E(i,j) = \left(\sum_{k=1}^{p} (x_k(i) - x_k(j))^2 \right)^{1/2}$$

Weighted: $$d_{WE}(i,j) = \left(\sum_{k=1}^{p} w_k (x_k(i) - x_k(j))^2 \right)^{1/2}$$

$$\mathrm{cov}(X,Y) = \frac{1}{n} \left(\sum_{i=1}^{n} (x(i) - \overline{x})(y(i) - \overline{y}) \right)$$



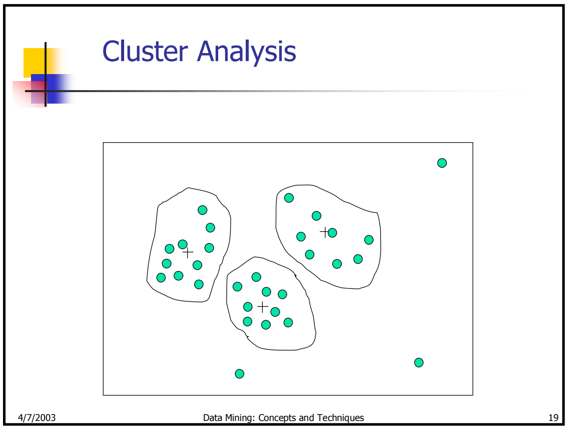
Distance ..

Mahalanobis:

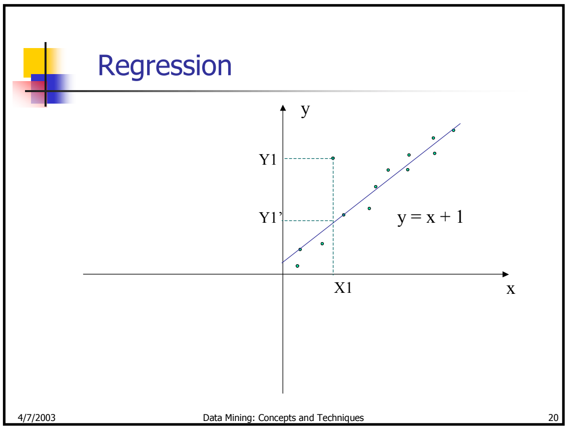
$$d_{MH}(i,j) = \left((\mathbf{x}(i) - \mathbf{x}(j))^T \Sigma^{-1} (\mathbf{x}(i) - \mathbf{x}(j)) \right)^{1/2}$$

Σ: p x p covariance matrix; entry (k,l) in the covariance matrix is defined between variables X_k and X_l .

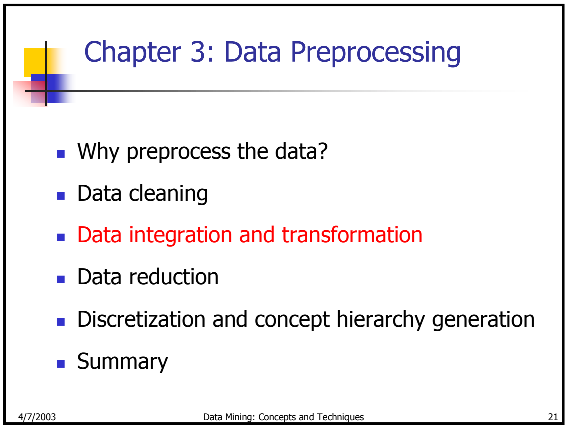
Minkowski: $$d_M(i,j) = \left(\sum_{k=1}^{p} (x_k(i) - x_k(j))^{\lambda} \right)^{1/\lambda}$$



Cluster Analysis

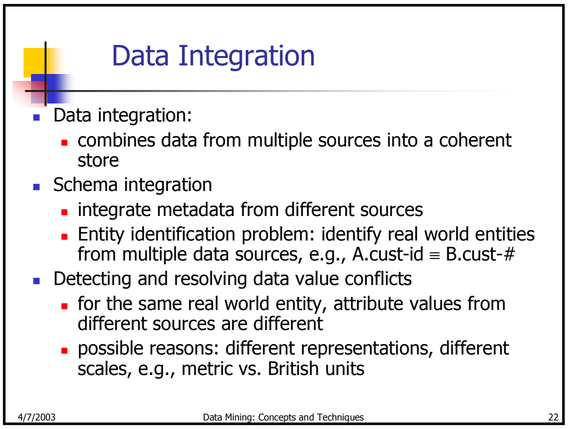


Regression



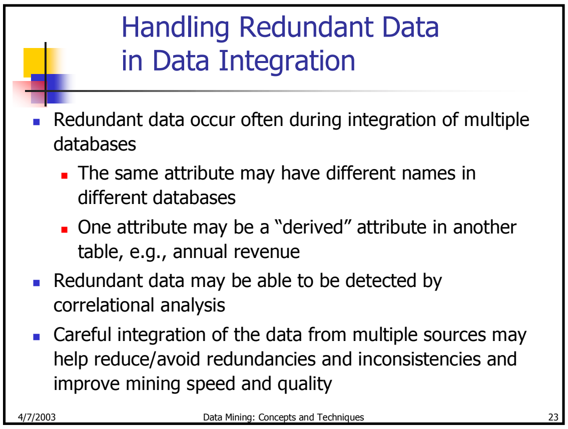
Chapter 3: Data Preprocessing

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



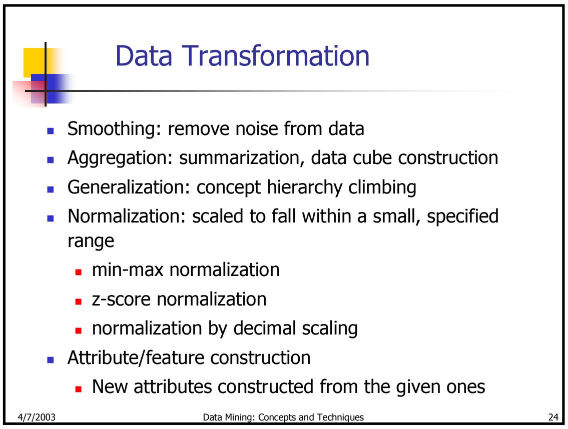
Data Integration

- Data integration:
  - combines data from multiple sources into a coherent store
- 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
  - possible reasons: different representations, different scales, e.g., metric vs. British units



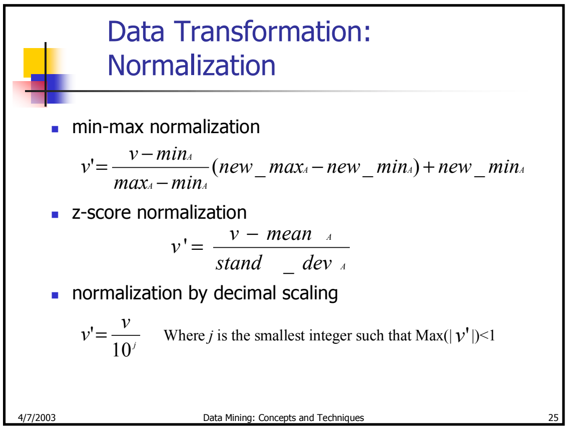
Handling Redundant Data in Data Integration

- Redundant data occur often during integration of multiple databases
  - The same attribute may have different names in different databases
  - One attribute may be a "derived" attribute in another table, e.g., annual revenue
- Redundant data may be able to be detected by correlational analysis
- Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality



Data Transformation

- Smoothing: remove noise from data
- Aggregation: summarization, data cube construction
- Generalization: concept hierarchy climbing
- Normalization: scaled to fall within a small, specified range
  - min-max normalization
  - z-score normalization
  - normalization by decimal scaling
- Attribute/feature construction
  - New attributes constructed from the given ones



Data Transformation: Normalization

- min-max normalization

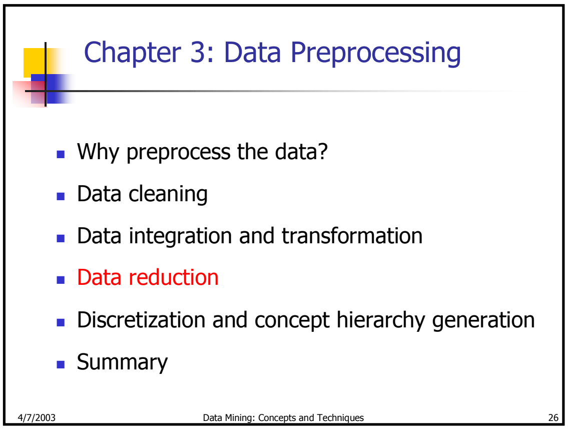
$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

- z-score normalization

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

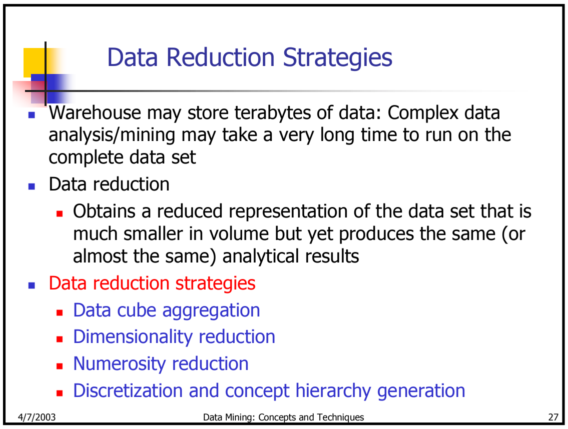
- normalization by decimal scaling

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



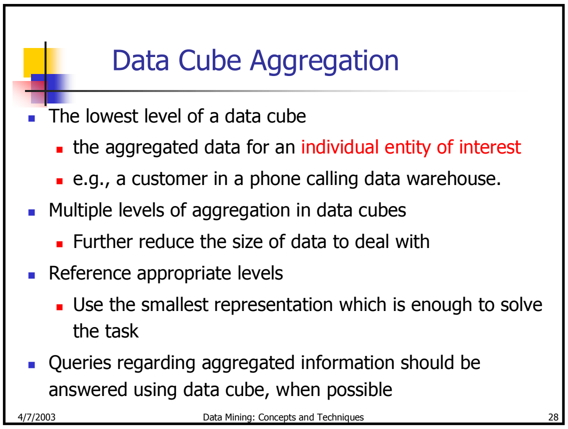
Chapter 3: Data Preprocessing

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



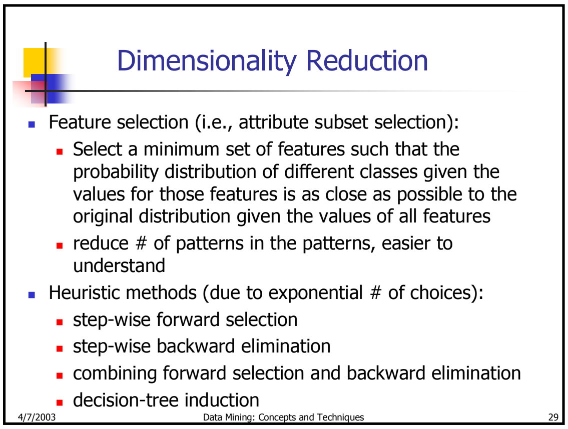
Data Reduction Strategies

- Warehouse may store terabytes of data: Complex data analysis/mining may take a very long time to run on the complete data set
- Data reduction
  - Obtains a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results
- Data reduction strategies
  - Data cube aggregation
  - Dimensionality reduction
  - Numerosity reduction
  - Discretization and concept hierarchy generation



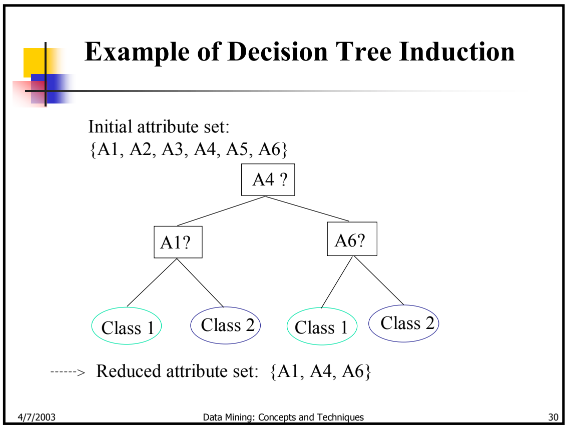
Data Cube Aggregation

- The lowest level of a data cube
  - the aggregated data for an individual entity of interest
  - e.g., a customer in a phone calling data warehouse.
- Multiple levels of aggregation in data cubes
  - Further reduce the size of data to deal with
- Reference appropriate levels
  - Use the smallest representation which is enough to solve the task
- Queries regarding aggregated information should be answered using data cube, when possible



Dimensionality Reduction

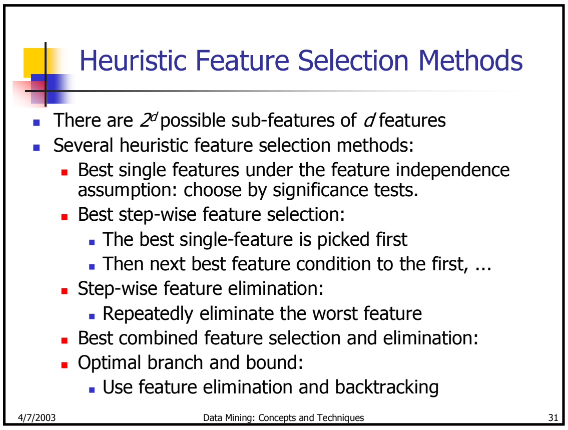
- Feature selection (i.e., attribute subset selection):
  - Select a minimum set of features such that the probability distribution of different classes given the values for those features is as close as possible to the original distribution given the values of all features
  - reduce # of patterns in the patterns, easier to understand
- Heuristic methods (due to exponential # of choices):
  - step-wise forward selection
  - step-wise backward elimination
  - combining forward selection and backward elimination
  - decision-tree induction



Example of Decision Tree Induction

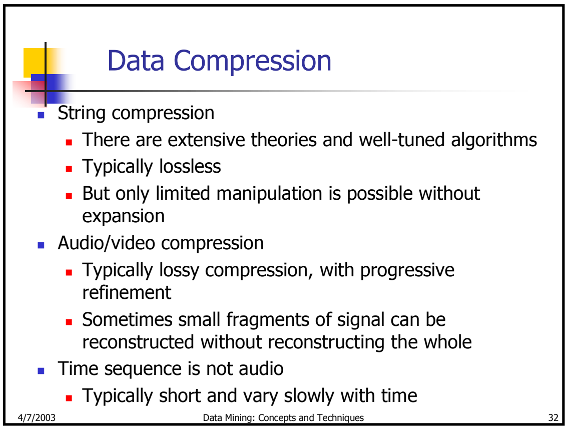
Initial attribute set:
{A1, A2, A3, A4, A5, A6}

-----> Reduced attribute set: {A1, A4, A6}



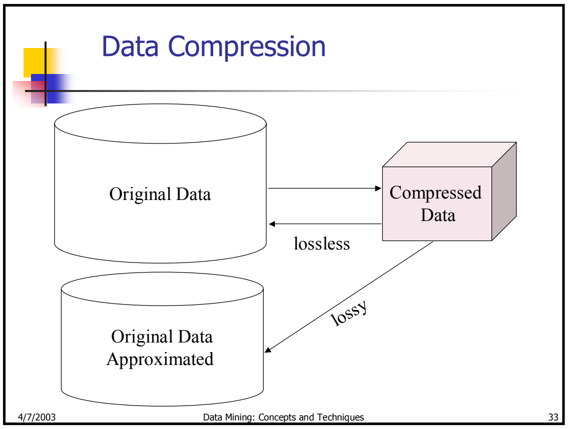
Heuristic Feature Selection Methods

- There are 2^d possible sub-features of *d* features
- Several heuristic feature selection methods:
  - Best single features under the feature independence assumption: choose by significance tests.
  - Best step-wise feature selection:
    - The best single-feature is picked first
    - Then next best feature condition to the first, ...
  - Step-wise feature elimination:
    - Repeatedly eliminate the worst feature
  - Best combined feature selection and elimination:
  - Optimal branch and bound:
    - Use feature elimination and backtracking

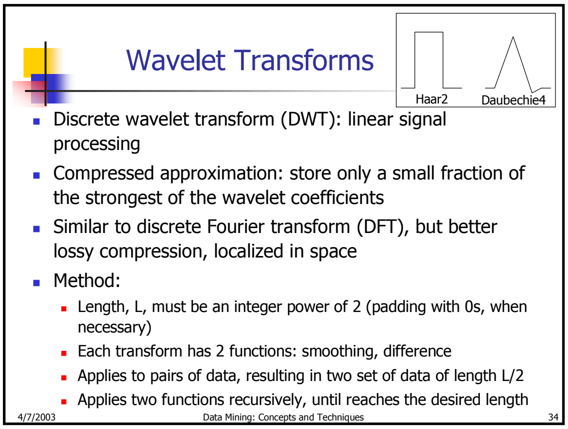


Data Compression

- String compression
  - There are extensive theories and well-tuned algorithms
  - Typically lossless
  - But only limited manipulation is possible without expansion
- Audio/video compression
  - Typically lossy compression, with progressive refinement
  - Sometimes small fragments of signal can be reconstructed without reconstructing the whole
- Time sequence is not audio
  - Typically short and vary slowly with time



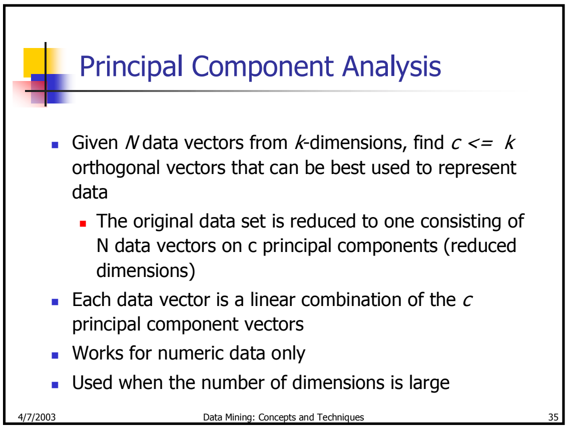
Data Compression



Wavelet Transforms

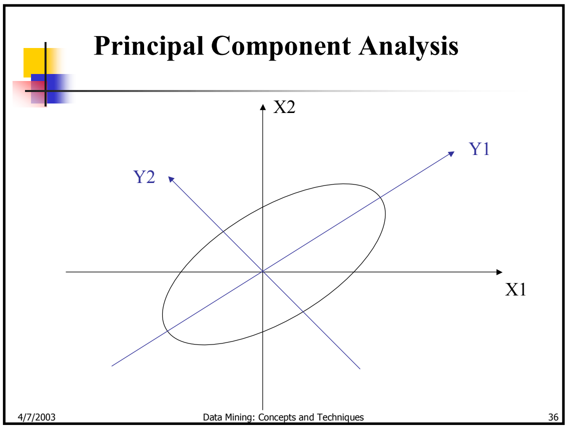
Haar2
Daubechie4

- Discrete wavelet transform (DWT): linear signal processing
- Compressed approximation: store only a small fraction of the strongest of the wavelet coefficients
- Similar to discrete Fourier transform (DFT), but better lossy compression, localized in space
- Method:
  - Length, L, must be an integer power of 2 (padding with 0s, when necessary)
  - Each transform has 2 functions: smoothing, difference
  - Applies to pairs of data, resulting in two set of data of length L/2
  - Applies two functions recursively, until reaches the desired length

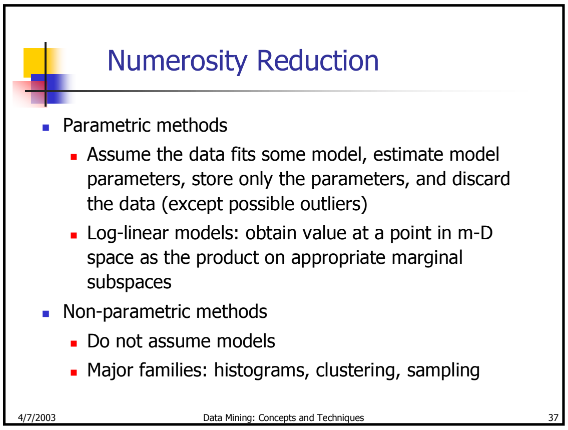


Principal Component Analysis

- Given *N* data vectors from *k*-dimensions, find *c* <= *k* orthogonal vectors that can be best used to represent data
  - The original data set is reduced to one consisting of N data vectors on c principal components (reduced dimensions)
- Each data vector is a linear combination of the *c* principal component vectors
- Works for numeric data only
- Used when the number of dimensions is large

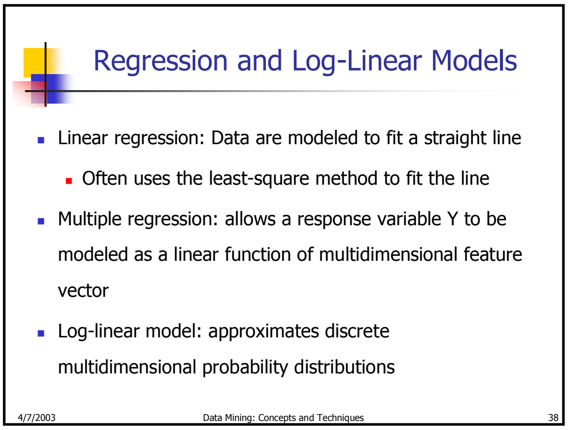


Principal Component Analysis



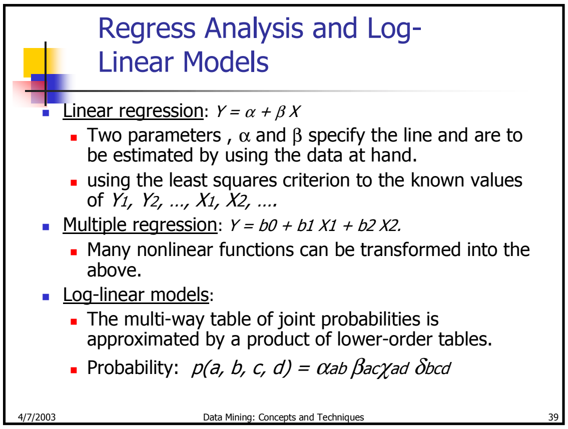
Numerosity Reduction

- Parametric methods
  - Assume the data fits some model, estimate model parameters, store only the parameters, and discard the data (except possible outliers)
  - Log-linear models: obtain value at a point in m-D space as the product on appropriate marginal subspaces
- Non-parametric methods
  - Do not assume models
  - Major families: histograms, clustering, sampling



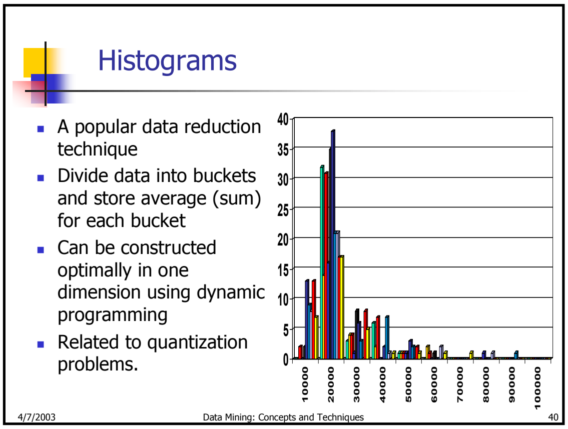
Regression and Log-Linear Models

- Linear regression: Data are modeled to fit a straight line
  - Often uses the least-square method to fit the line
- Multiple regression: allows a response variable Y to be modeled as a linear function of multidimensional feature vector
- Log-linear model: approximates discrete multidimensional probability distributions



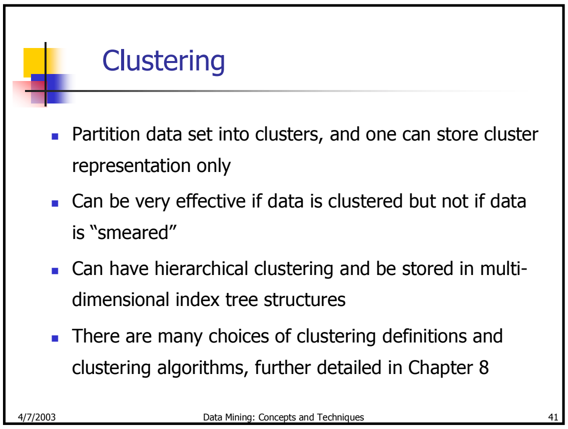
Regress Analysis and Log-Linear Models

- Linear regression: $Y = \alpha + \beta X$
  - Two parameters , α and β specify the line and are to be estimated by using the data at hand.
  - using the least squares criterion to the known values of $Y_1, Y_2, \ldots, X_1, X_2, \ldots$.
- Multiple regression: $Y = b0 + b1\ X1 + b2\ X2$.
  - Many nonlinear functions can be transformed into the above.
- Log-linear models:
  - The multi-way table of joint probabilities is approximated by a product of lower-order tables.
  - Probability: $p(a, b, c, d) = \alpha ab\ \beta ac \chi ad\ \delta bcd$



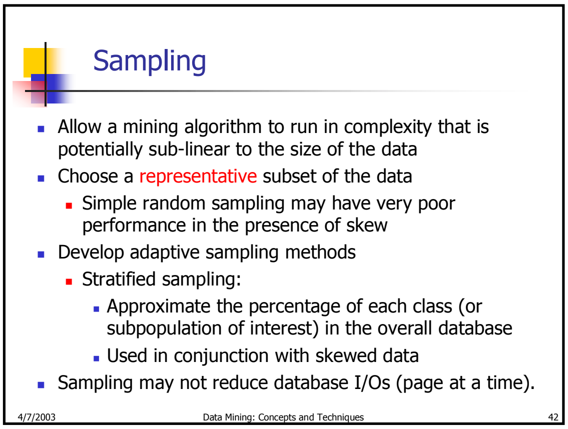
Histograms

- A popular data reduction technique
- Divide data into buckets and store average (sum) for each bucket
- Can be constructed optimally in one dimension using dynamic programming
- Related to quantization problems.



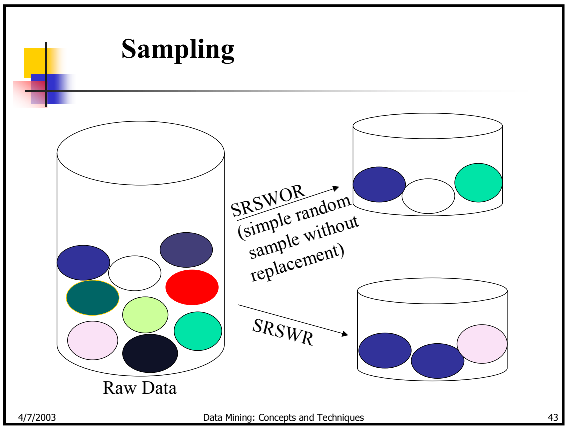
Clustering

- 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"
- Can have hierarchical clustering and be stored in multi-dimensional index tree structures
- There are many choices of clustering definitions and clustering algorithms, further detailed in Chapter 8

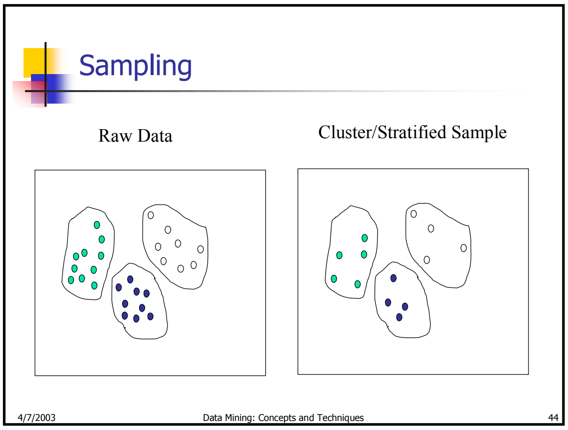


Sampling

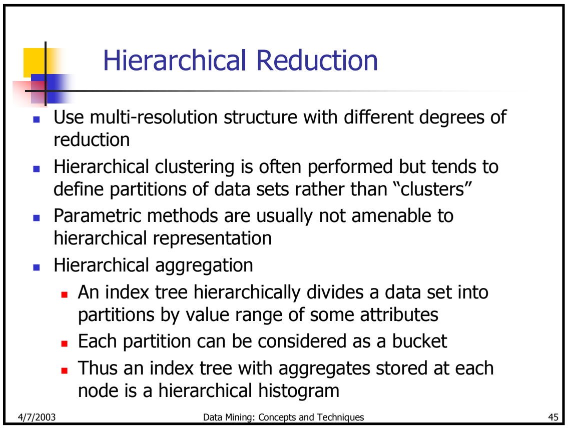
- Allow a mining algorithm to run in complexity that is potentially sub-linear to the size of the data
- Choose a representative subset of the data
  - Simple random sampling may have very poor performance in the presence of skew
- Develop adaptive sampling methods
  - Stratified sampling:
    - Approximate the percentage of each class (or subpopulation of interest) in the overall database
    - Used in conjunction with skewed data
- Sampling may not reduce database I/Os (page at a time).



Sampling

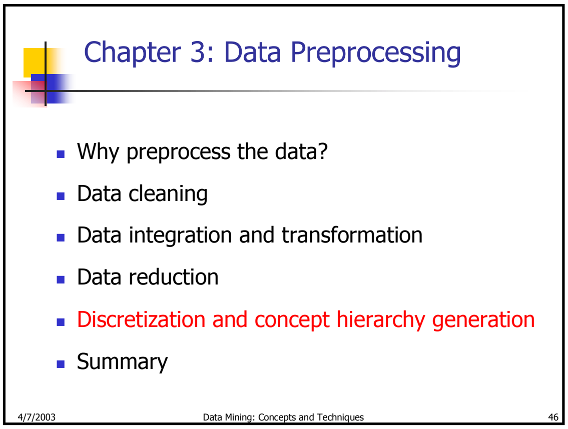


Sampling



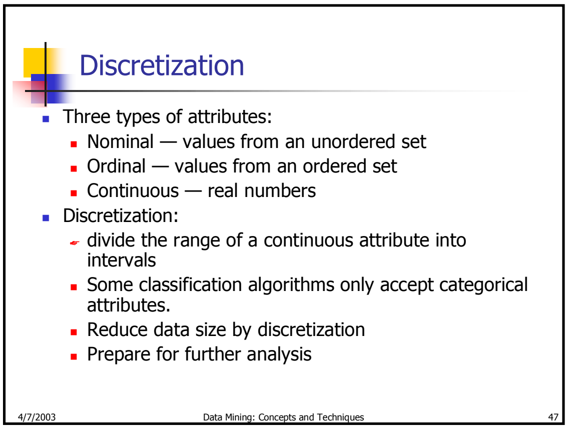
Hierarchical Reduction

- Use multi-resolution structure with different degrees of reduction
- Hierarchical clustering is often performed but tends to define partitions of data sets rather than "clusters"
- Parametric methods are usually not amenable to hierarchical representation
- Hierarchical aggregation
  - An index tree hierarchically divides a data set into partitions by value range of some attributes
  - Each partition can be considered as a bucket
  - Thus an index tree with aggregates stored at each node is a hierarchical histogram



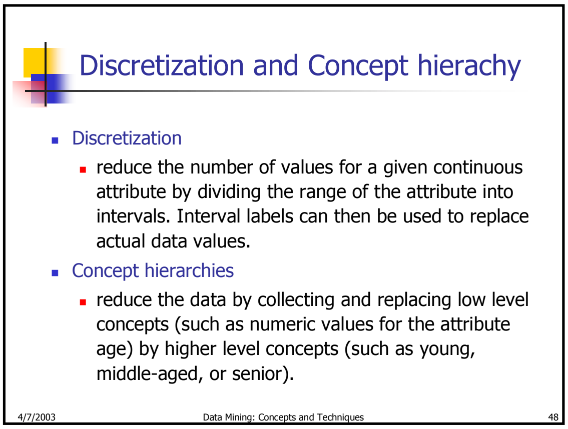
Chapter 3: Data Preprocessing

- Why preprocess the data?
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization and concept hierarchy generation
- 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
  - Some classification algorithms only accept categorical attributes.
  - Reduce data size by discretization
  - Prepare for further analysis



Discretization and Concept hierachy

- 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).

Discretization and concept hierarchy generation for numeric data

- Binning (see sections before)
- Histogram analysis (see sections before)
- Clustering analysis (see sections before)
- Entropy-based discretization
- Segmentation by natural partitioning

Entropy-Based Discretization

- Given a set of samples S, if S is partitioned into two intervals S1 and S2 using boundary T, the entropy after partitioning is $E(S,T)=\frac{|S_1|}{|S|}Ent(S_1)+\frac{|S_2|}{|S|}Ent(S_2)$
- The boundary that minimizes the entropy function over all possible boundaries is selected as a binary discretization.
- The process is recursively applied to partitions obtained until some stopping criterion is met, e.g.,

$$Ent(S)-E(T,S)>\delta$$

- Experiments show that it may reduce data size and improve classification accuracy

Entropy

$$Ent(S)=-\sum_{i=1}^{m} p_i \log_2(p_i)$$

Where p_i is the probability of class *i* in S.

Segmentation by natural partitioning

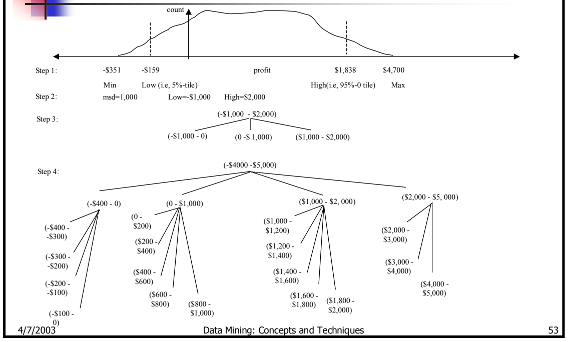
3-4-5 rule can be used to segment numeric data into relatively uniform, "natural" intervals.

* If an interval covers 3, 6, 7 or 9 distinct values at the most significant digit, partition the range into 3 equi-width intervals

* If it covers 2, 4, or 8 distinct values at the most significant digit, partition the range into 4 intervals

* If it covers 1, 5, or 10 distinct values at the most significant digit, partition the range into 5 intervals

Example of 3-4-5 rule



count

Step 1: -$351 -$159 profit $1,838 $4,700

Min Low (i.e, 5%-tile) High(i.e, 95%-0 tile) Max

Step 2: msd=1,000 Low=-$1,000 High=$2,000

Step 3: (-$1,000 - $2,000)

(-$1,000 - 0) (0 -$ 1,000) ($1,000 - $2,000)

Step 4: (-$4000 -$5,000)

(-$400 - 0) (0 - $1,000) ($1,000 - $2, 000) ($2,000 - $5, 000)

(-$400 - -$300) (-$300 - -$200) (-$200 - -$100) (-$100 - 0)

(0 - $200) ($200 - $400) ($400 - $600) ($600 - $800) ($800 - $1,000)

($1,000 - $1,200) ($1,200 - $1,400) ($1,400 - $1,600) ($1,600 - $1,800) ($1,800 - $2,000)

($2,000 - $3,000) ($3,000 - $4,000) ($4,000 - $5,000)

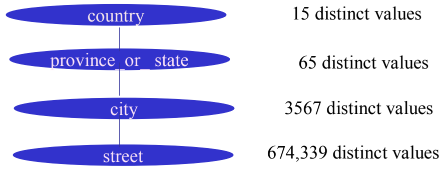
Concept hierarchy generation for categorical data

Categorical data are discrete data, with no ordering among the values; e.g., geographic location, item type, etc.

- Specification of a partial ordering of attributes explicitly at the schema level by users or experts (*city, state*,..)
- Specification of a portion of a hierarchy by explicit data grouping (los angeles, ventura are "in" california)
- Specification of a set of attributes, but not of their partial ordering
- Specification of only a partial set of attributes

Specification of a set of attributes

Concept hierarchy can be automatically generated based on the number of distinct values per attribute in the given attribute set. The attribute with the most distinct values is placed at the lowest level of the hierarchy.



Chapter 3: Data Preprocessing

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

Summary

- Data preparation is an important issue for both warehousing and mining
- Data preparation includes
  - Data cleaning and data integration
  - Data reduction and feature selection
  - Discretization
- Many methods have been developed but still an active area of research

References

- D. P. Ballou and G. K. Tayi. Enhancing data quality in data warehouse environments. Communications of ACM, 42:73-78, 1999.
- Jagadish et al., Special Issue on Data Reduction Techniques. Bulletin of the Technical Committee on Data Engineering, 20(4), December 1997.
- D. Pyle. Data Preparation for Data Mining. Morgan Kaufmann, 1999.
- T. Redman. Data Quality: Management and Technology. Bantam Books, New York, 1992.
- Y. Wand and R. Wang. Anchoring data quality dimensions ontological foundations. Communications of ACM, 39:86-95, 1996.
- R. Wang, V. Storey, and C. Firth. A framework for analysis of data quality research. IEEE Trans. Knowledge and Data Engineering, 7:623-640, 1995.