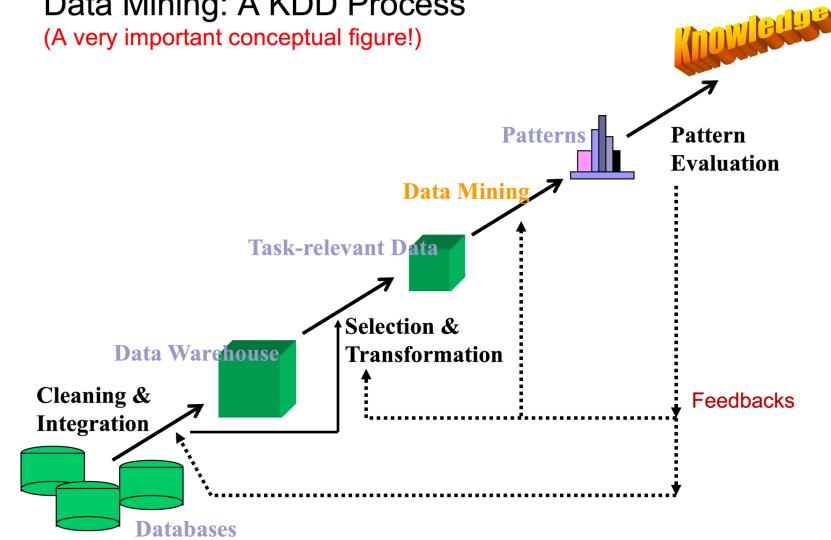


Data Preprocessing

(for Data Warehousing)

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

Why preprocess the data?



- Data in the real world is dirty
 - ❑ **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

Multi-Dimensional Measure of Data Quality

- A well-accepted multidimensional view:
 - ❑ Accuracy
 - ❑ Completeness
 - ❑ Consistency
 - ❑ Timeliness
 - ❑ Believability
 - ❑ Value added
 - ❑ Interpretability
 - ❑ Accessibility

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

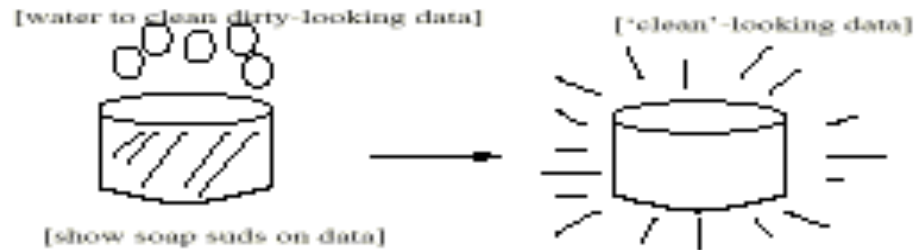
- ❑ Obtains reduced representation in volume but produces the same or similar analytical results

■ Data discretization

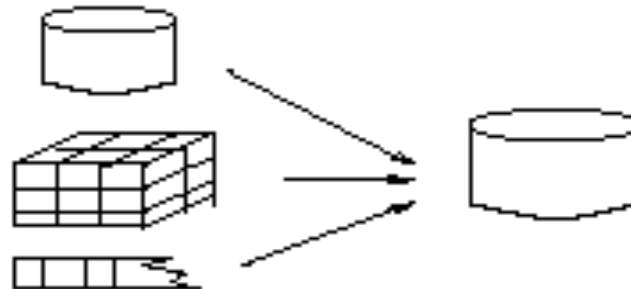
- ❑ Part of data reduction but with particular importance, especially for numerical data

Forms of data preprocessing

Data Cleaning



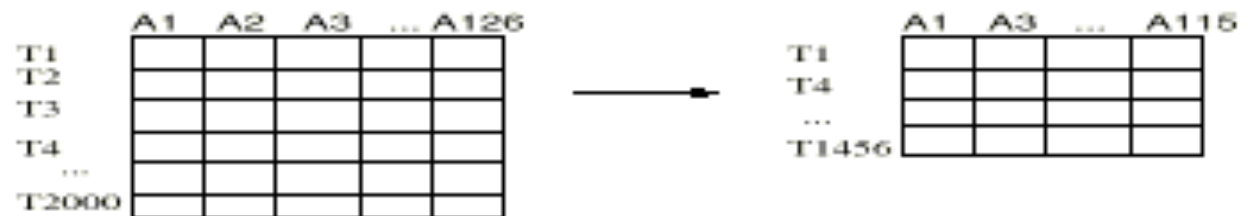
Data Integration



Data Transformation

-2, 32, 100, 59, 48 → -0.02, 0.32, 1.00, 0.59, 0.48

Data Reduction



Data Preprocessing

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

Data Cleaning

■ Data cleaning tasks

- ❑ Fill in missing values (**Missing Data Imputation**)
- ❑ Identify outliers (**cf. anomaly detection**) and smooth out noisy data (**noise filtering**)
- ❑ 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
 - data not entered due to misunderstanding
 - 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!! But how?*

Recall the slides of feature engineering:

How to Handle Missing Data?

- Ignore the tuple: usually done when class label is missing; but is not good for attribute values (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
- Use the most probable value to fill in the missing value: inference-based such as Bayesian formula or decision tree and associative-based

Guessing the missing data (Aggarwal et al., KDD2006)

Name	Title	Gender	M.Status	Education	Salary Level
Amy	Assistant	F	Unmarried	HD	SL-3
Bobby	Assistant	M	Married	HD	SL-3
Catherine	Assistant	F	Married	University	SL-3
Don	Manager	F	Unmarried	University	SL-5
Elaine	Manager	F	Married	University	SL-5
Franky	Manager	M	Married	University	SL-5
Grace	Manager	F	Married	M.B.A.	SL-7
Helen	Manager	F	Married	Ph.D	SL-5
Ivan	Accountant	F	Unmarried	M.B.A.	SL-5
Jenny	Accountant	M	Married	University	SL-4

- Catherine wants to hide her salary level
- Franky wants to hide his education background
- Grace wants to hide her marriage status
- Ivan wants to hide his gender
- Jenny wants to hide her gender as well

So, the following is resulted.

Name	Title	Gender	M.Status	Education	Salary Level
Amy	Assistant	F	Unmarried	HD	SL-3
Bobby	Assistant	M	Married	HD	SL-3
Catherine	Assistant	F	Married	University	
Don	Manager	F	Unmarried	University	SL-5
Elaine	Manager	F	Married	University	SL-5
Franky	Manager	M	Married		SL-5
Grace	Manager	F		M.B.A.	SL-7
Helen	Manager	F	Married	Ph.D	SL-5
Ivan	Accountant		Unmarried	M.B.A.	SL-5
Jenny	Accountant		Married	University	SL-4

Guessing the missing data (Aggarwal et al., KDD2006)

By apply association analysis to the dataset with missing data, we can obtain

- ❑ R1: Assistant \rightarrow SL-3 (support:2, confidence=100%)
- ❑ R2: Manager \wedge SL-5 \rightarrow University (support:2, confidence=66.7%)
- ❑ R3: Manager \wedge Female \rightarrow Married (support:2, confidence=66.7%)

So, we can guess the missing value as follows.

- ❑ Catherine's salary level is **SL-3**
- ❑ Franky's education is **university**
- ❑ Grace's marriage status is **married**

Noisy Data

- Noise: random error or variance in a measured variable
- Incorrect attribute values may 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 the data in bins by bin's means, by bin's median, by bin's 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

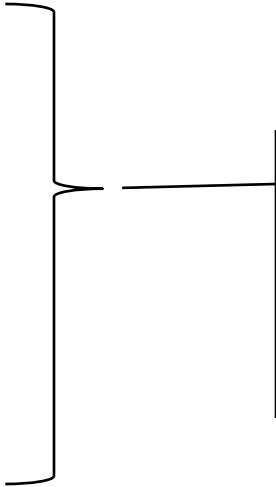
Binning:

A simple discretization methods

- **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: $W = (B-A)/N$.
 - ❑ The most straightforward
 - ❑ 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.

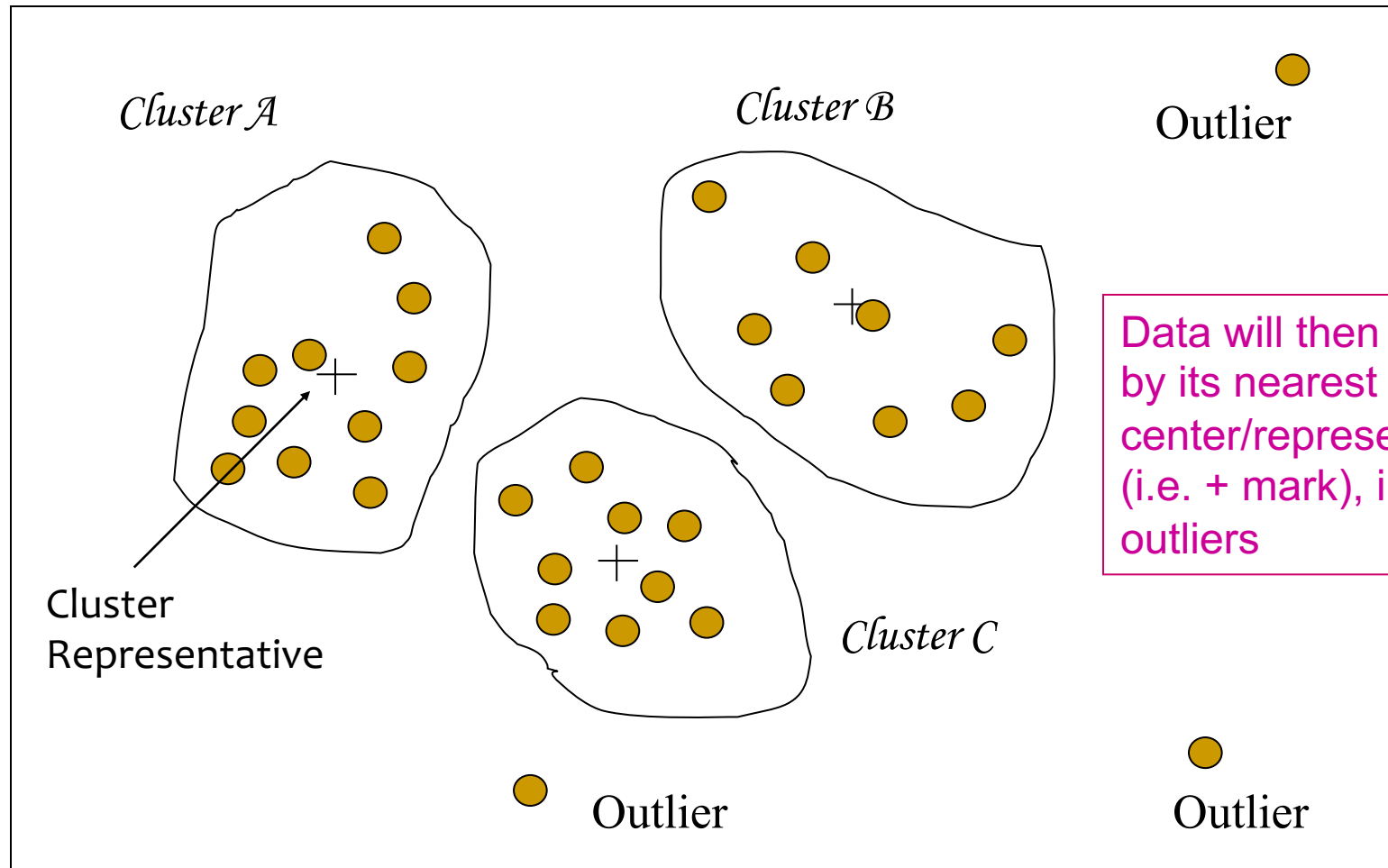
Binning Methods for Data Smoothing

- * 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

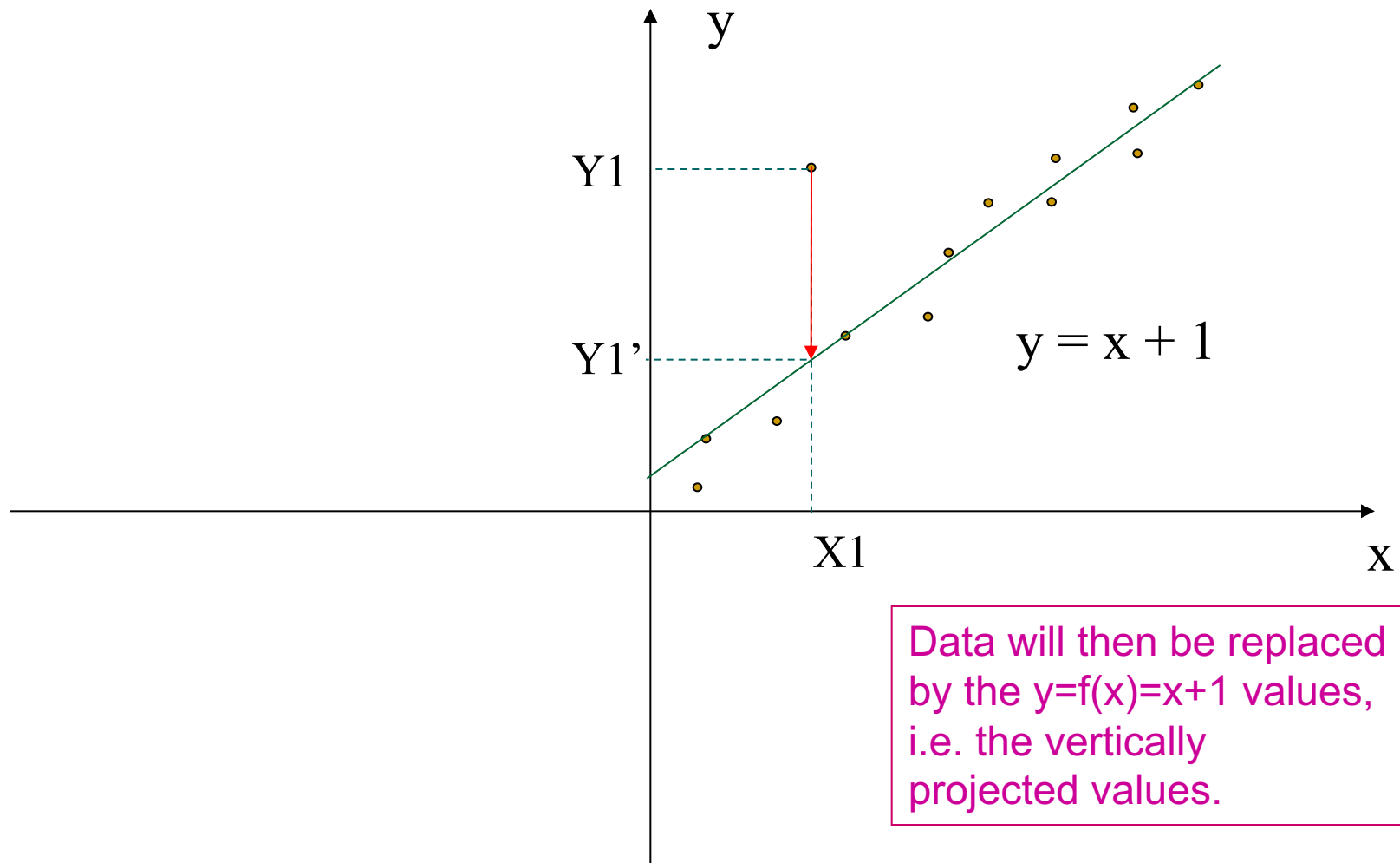


The original data will then be replaced by the smoothed data

Smoothing by Cluster Analysis



Smoothing by Regression



Data Preprocessing

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

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 \equiv 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 when integrating 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 correlation analysis
- Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality

Data Transformation

Different Forms of Transformation

- Smoothing (is a form of transformation): 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

$$v' = \frac{v - \min_A}{\max_A - \min_A} (\text{new_max}_A - \text{new_min}_A) + \text{new_min}_A$$

- z-score normalization

$$v' = \frac{v - \text{mean}_A}{\text{std_dev}_A}$$

- normalization by decimal scaling

$$v' = \frac{v}{10^j} \quad \text{where } j \text{ is the smallest integer such that } \max(|v'|) \leq 1$$

Data Preprocessing

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

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
 - ❑ **Dimensionality reduction**
 - ❑ Numerosity reduction
 - ❑ Discretization and concept hierarchy generation

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

Data Reduction



- **Feature extraction**/transformation - Combining (mapping) existing features into smaller number of new/alternative features
 - Linear combination (projection)
 - Nonlinear combination

Feature Selection vs Extraction

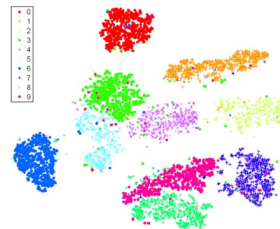
- **Feature selection:** Choosing $k < d$ important features, ignoring the remaining $d - k$
 - Subset selection algorithms
- **Feature extraction:** Project the original x_i , $i = 1, \dots, d$ dimensions to new $k < d$ dimensions, z_j , $j = 1, \dots, k$
 - Principal component analysis (PCA), linear discriminant analysis (LDA), factor analysis (FA)

Feature Selection: Subset Selection

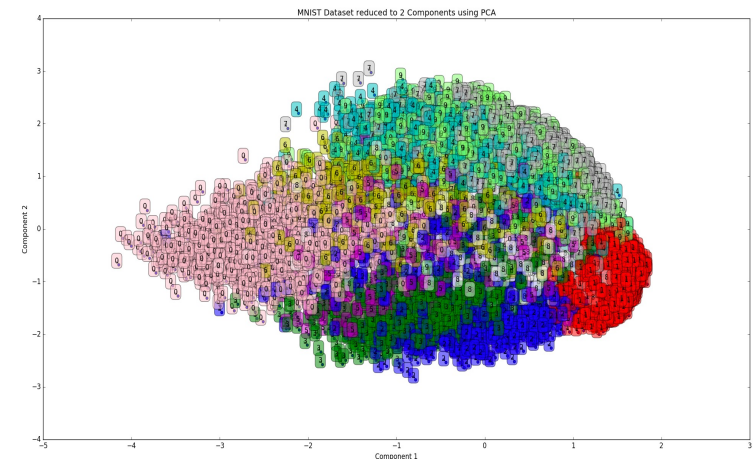
- There are 2^d subsets of d features
- *Forward search methods*: Add the best feature at each step
 - Set of features F initially \emptyset .
 - At each iteration, find the best new feature
$$j = \operatorname{argmin}_i E (F \cup x_i)$$
 - Add x_j to F if $E (F \cup x_j) < E (F)$
 - Greedy hill climbing approach
- *Backward search methods*: Start with all features and remove one at a time, if possible.
- *Floating search methods*: (Add k , remove l)

Linear dimensionality reduction

- Linearly project n -dimensional data onto a k -dimensional space
 - $k < n$, often $k \ll n$
 - Example: project 10^4 -D space of words into 3 dimensions
 - Example: project MNIST 28x28 (784-D) image pixels into 2 dimensions.
- There are infinitely many k -dimensional subspaces we can project the data onto.
- Which one should we choose?



t-SNE

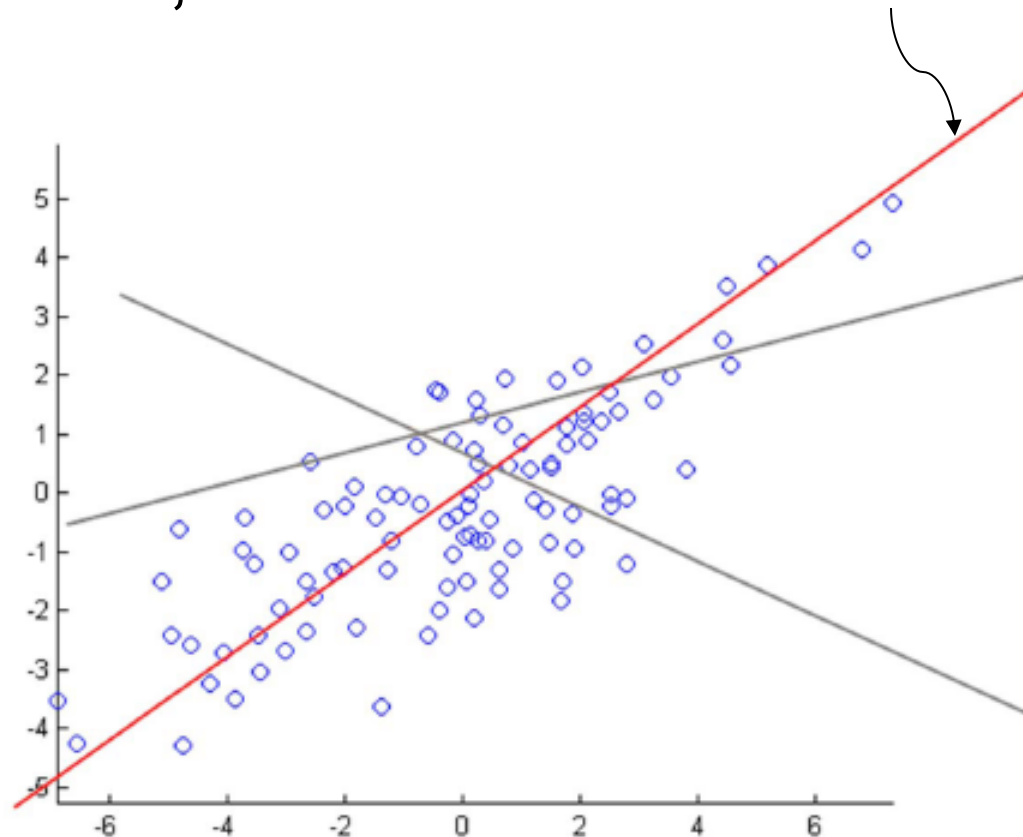


Linear dimensionality reduction

- Best k -dimensional subspace for projection depends on task
 - Classification: maximize separation among classes (like SVM)
 - Example: linear discriminant analysis (LDA)
 - DR is not limited to unsupervised learning!
 - Regression: maximize correlation between projected data and response variable
 - Example: partial least squares (PLS)
 - Unsupervised: retain as much data variance as possible
 - Example: principal component analysis (PCA)

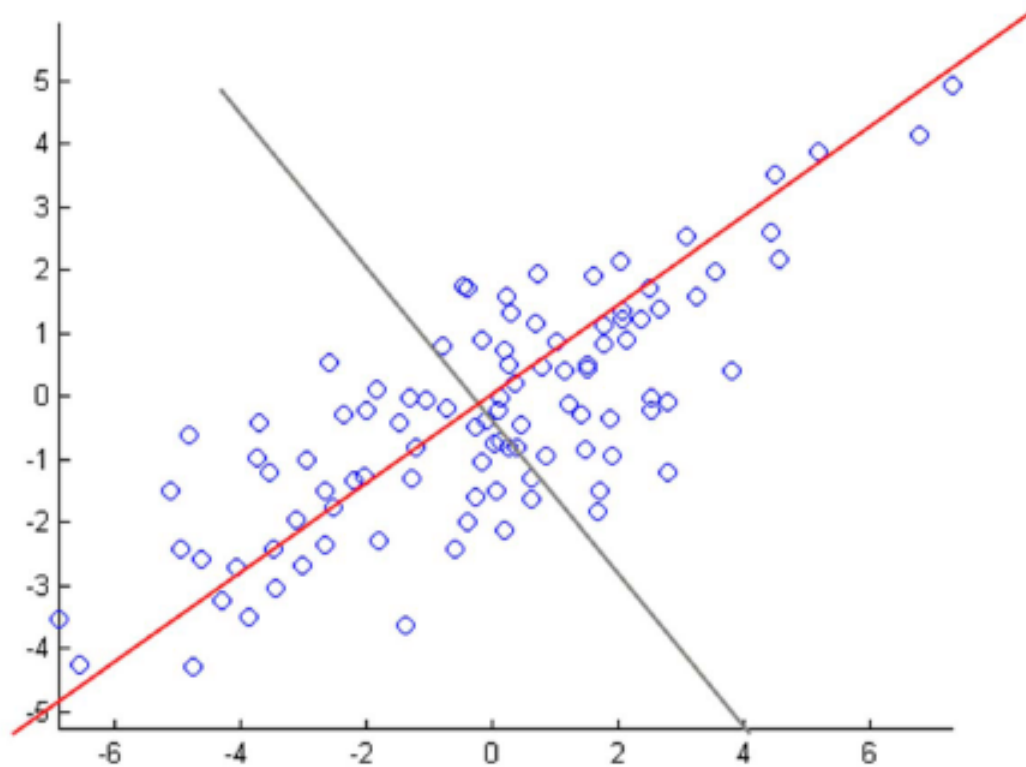
PCA: Conceptual algorithm

- Find a line, such that when the data is projected onto that line, and it has the maximum variance.



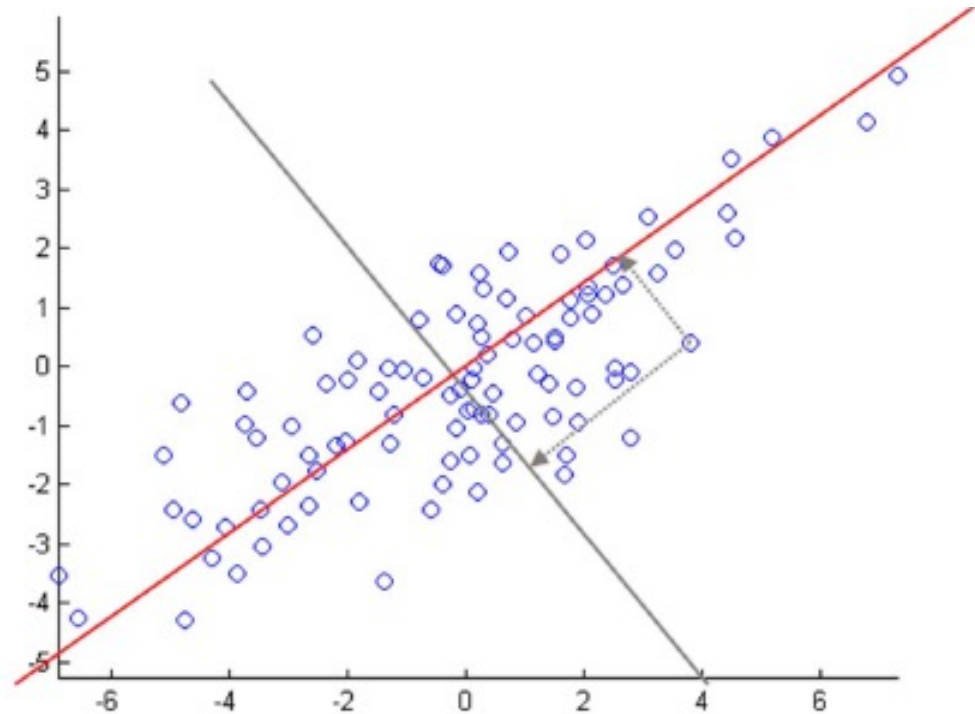
PCA: Conceptual algorithm

- Find a second line, orthogonal to the first, that has maximum projected variance.



PCA: Conceptual algorithm

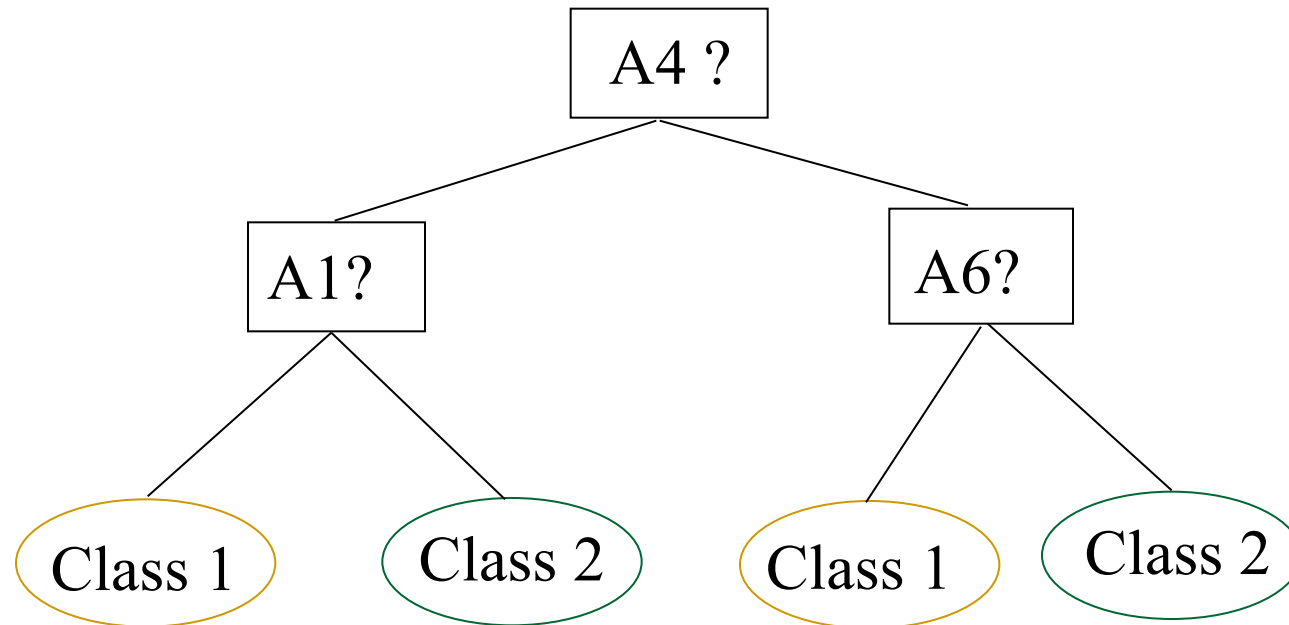
- Repeat until having k orthogonal lines
- The projected position of a point on these lines gives the coordinates in the k -dimensional reduced space.



Decision Tree Induction for Dimensionality Reduction

Initial attribute set:

$\{A1, A2, A3, A4, A5, A6\}$



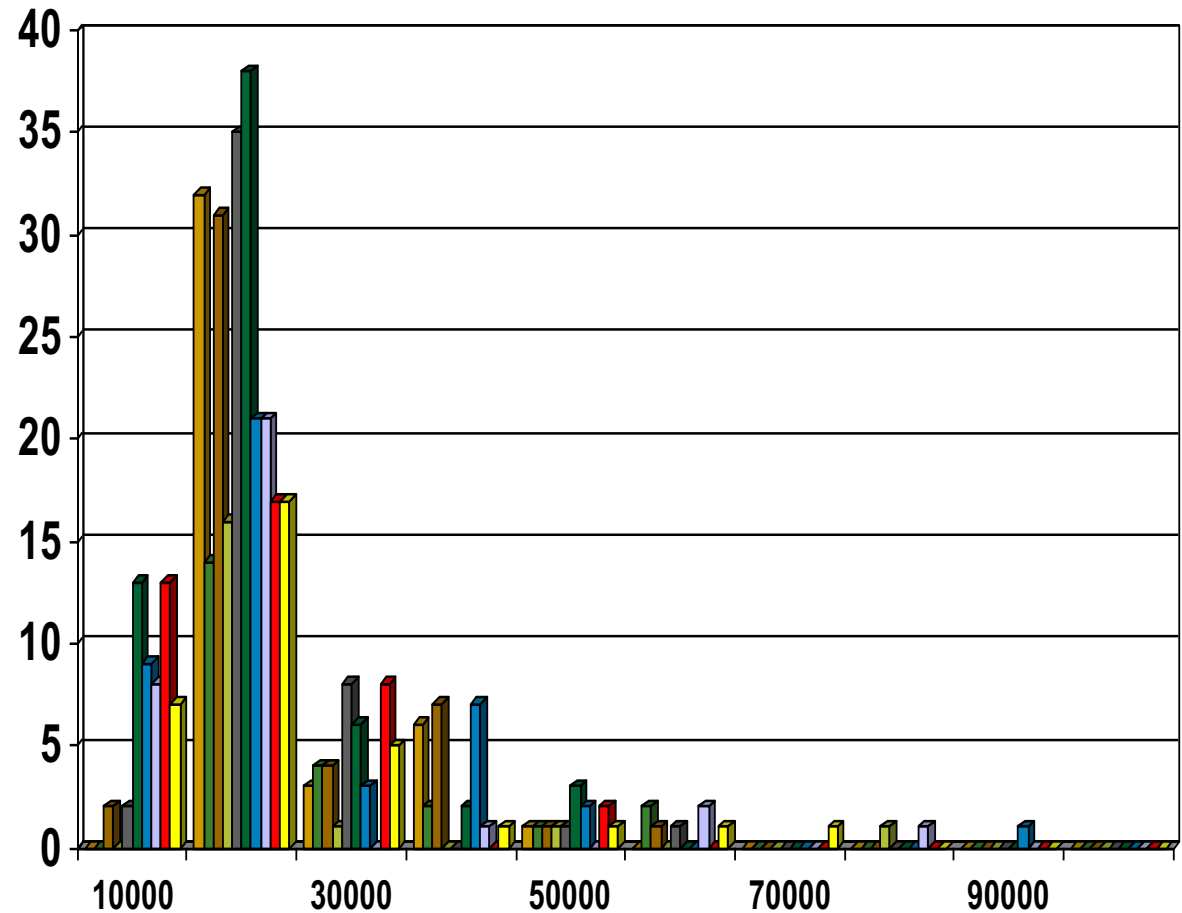
➡ Reduced attribute set: $\{A1, A4, A6\} \rightarrow$ saved 50%!!

Data Reduction via Numerosity Reduction

- *“Can we reduce the data volume by choosing alternative, ‘smaller’ forms of data representation?”*
- Parametric methods
 - Assume the data fits some (regression) model, estimate model parameters, store only the parameters, and discard the data (except possible outliers)
 - E.g. Linear regression where data are modeled to fit a straight line
 - Often uses the least-square method to fit the line $Y = \alpha + \beta X$
- Non-parametric methods
 - Do not assume models
 - Major families: histograms, clustering, sampling

Non-parametric Numerosity Reduction: Histograms

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



Data Reduction By Clustering

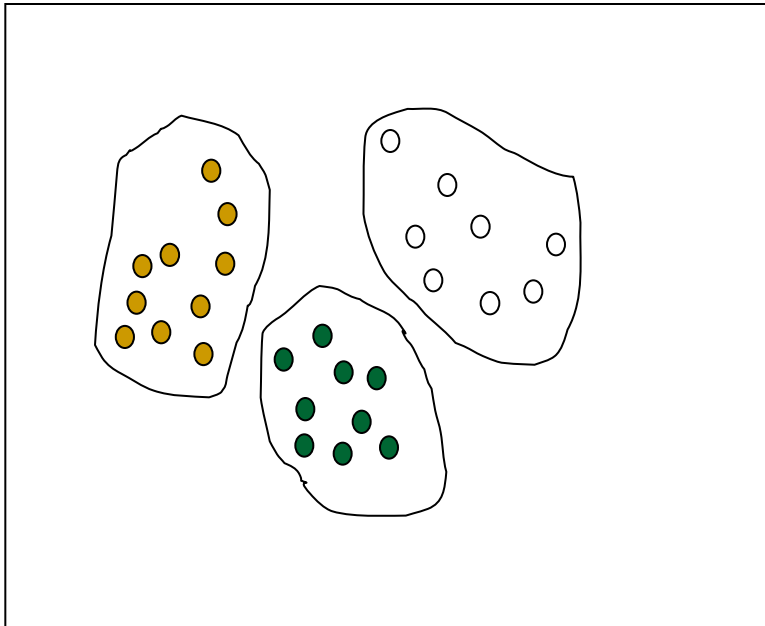
- Partition data set into clusters, and one can store cluster representation only
- Can be very effective if data is clustered but not if data is “smeared”
- Can have hierarchical clustering and be stored in multi-dimensional index tree structures
- There are many choices of clustering definitions and clustering algorithms

Data Reduction by Sampling

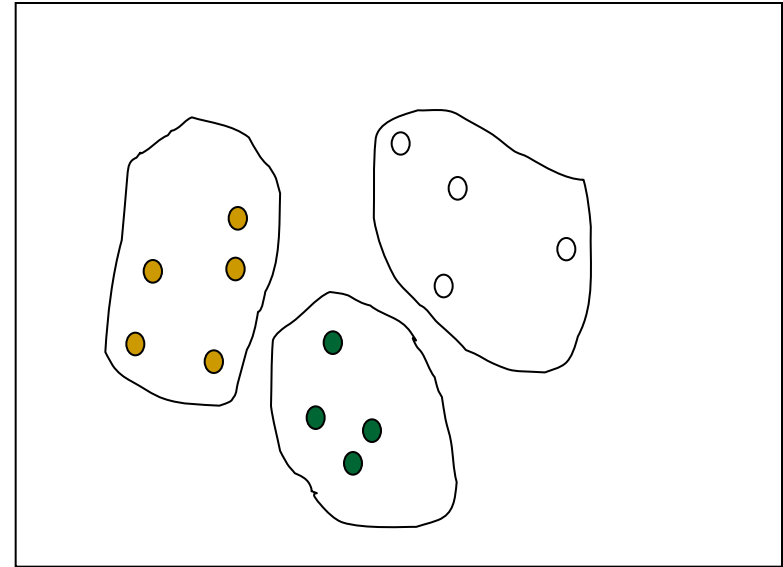
- Allow a mining algorithm to run in complexity that is potentially sub-linear to the size of the data
- Choose a **representative** subset of the data
 - ❑ Simple random sampling may have very poor performance in the presence of skew
- Develop adaptive sampling methods
 - ❑ Stratified sampling:
 - Approximate the percentage of each class (or subpopulation of interest) in the overall database
 - Used in conjunction with skewed data

Sampling

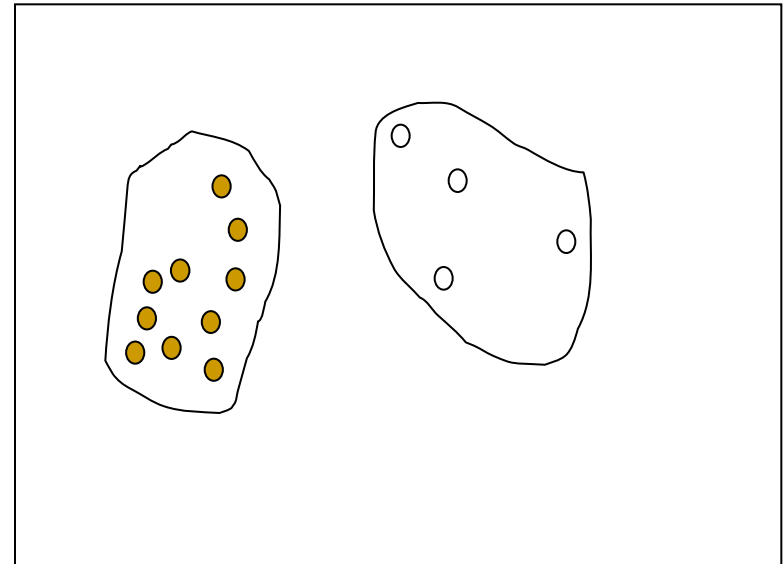
Raw Data



Cluster/Stratified Sample



Random Sample (extreme case)



Data Preprocessing

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

Discretization

- Three types of attributes:
 - ❑ Nominal/Discrete/**Categorical** — values from an unordered set (e.g. color set)
 - ❑ **Ordinal** — values from an ordered set (1,2,3,4,5 of a 5-pt system as in movie ratings)
 - ❑ **Continuous** — real numbers
- Discretization:
 - ❑ divide the range of a continuous attribute into interval, e.g. using the binning method
 - ❑ Some classification algorithms only accept categorical attributes.
 - ❑ Reduce data size by discretization
 - ❑ Prepare for further analysis

Discretization and Concept Hierarchy

■ 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).
- ❑ Other examples include building \Rightarrow street \Rightarrow district \Rightarrow city \Rightarrow country and those hierarchies for generalized association rule mining.

Summary

- Data preparation is a big issue for both warehousing and mining
- Data preparation includes
 - Data cleaning and data integration
 - Data reduction and feature selection
 - Discretization
- A lot of methods have been developed but still an active area of research. Many studies are about privacy preserving issues.

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.
- C.C. Aggarwal, et al. "On Privacy Preservation Against Adversarial Data Mining," *KDD 2006*, pp.510-516.