Chapter 2: Data Preprocessing

Why Data Preprocessing?

- Data in the real world is dirty
  - incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
    - e.g., occupation=“ ”
  - noisy: containing errors or outliers
    - e.g., Salary=“-10”
  - inconsistent: containing discrepancies in codes or names
    - e.g., Age=“30” Birthday=“03/07/1980”
    - e.g., Was rating “1,2,3”, now rating “A, B, C”
    - e.g., discrepancy between duplicate records

Why Is Data Dirty?

- Incomplete data may come from
  - "Not applicable" data value when collected
  - Different considerations between the time when the data was collected and when it is analyzed.
  - Human/hardware/software problems
- Noisy data (incorrect values) may come from
  - Faulty data collection instruments
  - Human or computer error at data entry
  - Errors in data transmission
- Inconsistent data may come from
  - Different data sources
  - Functional dependency violation (e.g., modify some linked data)
- Duplicate records also need data cleaning

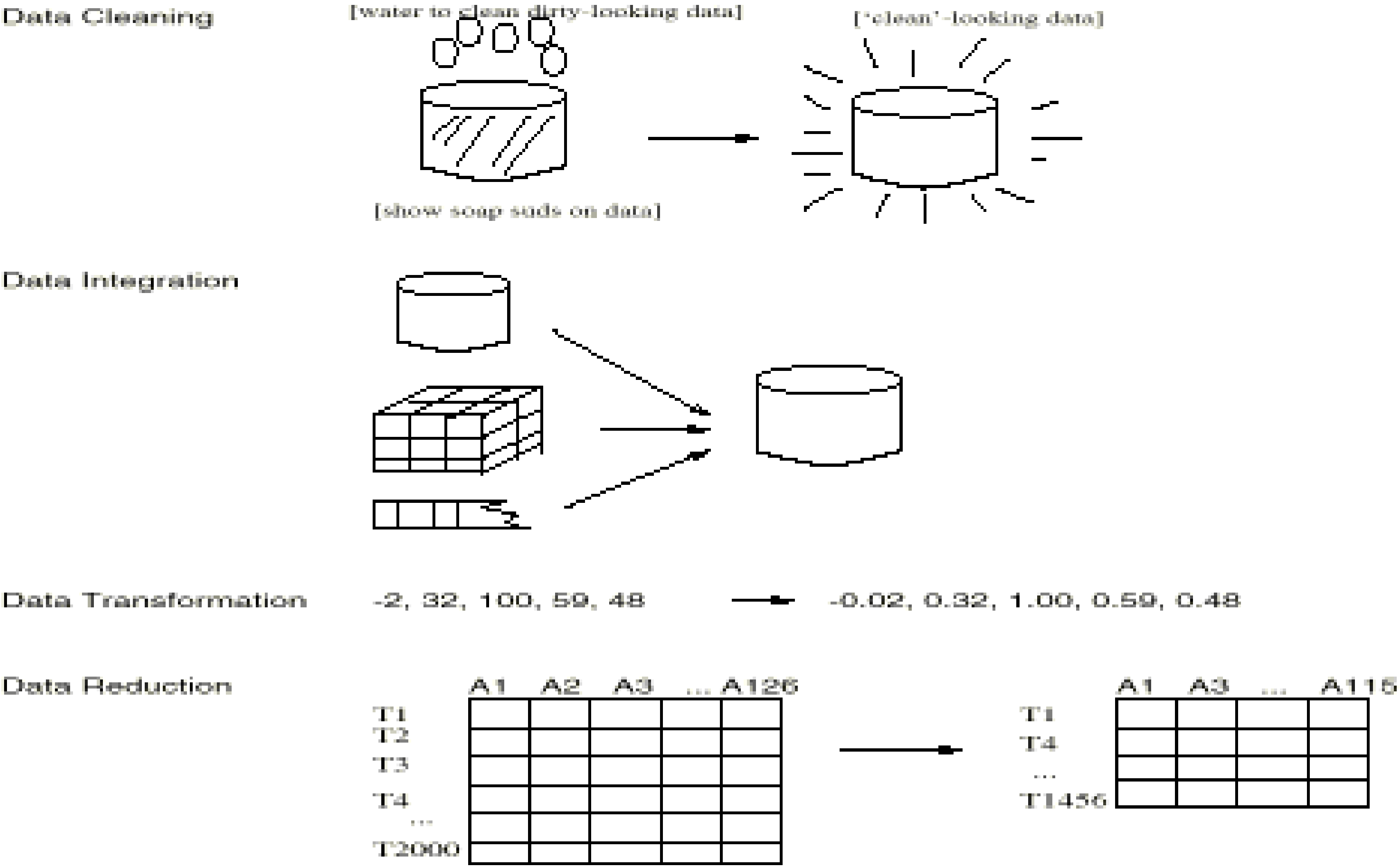
Why Is Data Preprocessing Important?

- No quality data, no quality mining results!
  - Quality decisions must be based on quality data
    - e.g., duplicate or missing data may cause incorrect or even misleading statistics.
  - Data warehouse needs consistent integration of quality data
- Data extraction, cleaning, and transformation comprises the majority of the work of building a data warehouse

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

- Importance
  - “Data cleaning is one of the three biggest problems in data warehousing”
- **Data Cleaning tasks**
  - Fill in missing values
  - Identify outliers and smooth out noisy data
  - Correct inconsistent data
  - Resolve redundancy caused by data integration

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.

How to Handle Missing Data?

- Ignore the tuple: usually done when class label is missing
- Fill in the missing value manually: tedious + infeasible?
- Fill in it automatically with
  - a global constant : e.g., “unknown”
  - the attribute mean
  - the attribute mean for all samples belonging to the same class: smarter
  - the most probable 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 due to..

  - faulty data collection instruments
  - data entry problems
  - data transmission problems
  - technology limitation
  - inconsistency in naming convention

How to Handle Noisy Data?

- Binning
  - first sort data and partition into (equal-frequency) bins
  - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
- Regression
  - smooth by fitting the data into regression functions
- Clustering
  - detect and remove outliers

Binning method

- Equal-depth (frequency) partitioning
  - Divides the range into *N* intervals, each containing approximately same number of samples
  - Good data scaling
  - Managing categorical attributes can be tricky
- Equal-width (distance) partitioning
  - Divides the range into *N* intervals of equal size
  - 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

Binning Methods for Data Smoothing (Example)

Sorted data for price (in dollars): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34

* Partition into equal-frequency (equi-depth) bins:

- Bin 1: 4, 8, 9, 15
- Bin 2: 21, 21, 24, 25
- Bin 3: 26, 28, 29, 34

1)Smoothing by bin means:

- Bin 1: 9, 9, 9, 9
- Bin 2: 23, 23, 23, 23
- Bin 3: 29, 29, 29, 29

2)Smoothing by bin boundaries:

- Bin 1: 4, 4, 4, 15
- Bin 2: 21, 21, 25, 25
- Bin 3: 26, 26, 26, 34

Q) Suppose a group of 12 *sales price* records has been sorted as follows:

5; 10; 11; 13; 15; 35; 50; 55; 72; 92; 204; 215;

Partition them into **three bins** by each of the following methods.

(a) equal-frequency partitioning
(b) equal-width partitioning
(c) clustering

(a) equal-frequency partitioning

bin 1 -- 5,10,11,13

bin 2 -- 15,35,50,55

bin 3 -- 72,92,204,215

(b) equal-width partitioning

The width of each interval is (215 - 5)/3 = 70.

bin 1 -- 5,10,11,13,15,35,50,55,72 (5 to 75)

bin 2 -- 92 (76 to 146)

bin 3 -- 204,215 (147 to 217)

(c) clustering

We will use a simple clustering technique:

Partition the data along the 2 biggest gaps in the data.

bin 1	5,10,11,13,15
bin 2	35,50,55,72,92
bin 3	204,215

(ex. 5; 10; 11; 13; 15; 35; 50; 55; 72; 92; 204; 215;)

Data Integration

- Data integration:
  - Combines data from multiple sources into a coherent store
- Schema integration: e.g., A.cust-id ≡ B.cust-#
  - Integrate metadata from different sources
- Entity identification problem:
  - Identify real world entities from multiple data sources, e.g., Bill Clinton = William Clinton
- 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 Redundancy in Data Integration

- Redundant data occur often when integration of multiple databases
  - *Object identification*: The same attribute or object may have different names in different databases
  - *Derivable data:* One attribute may be a “derived” attribute in another table, e.g., annual revenue
- Redundant attributes 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

Redundancy and Correlation Analysis

- Some redundancy can be detected by correlation analysis.

- Such analysis can measure how strongly one attributes implies the other based on the available data

  - For nominal data, x^2 (chi-square) test
  - For numeric data correlation coefficient and covariance

Correlation Analysis (Categorical Data)

- X^2 (chi-square) test

$$\chi^2 = \sum \frac{(Observed - Expected)^2}{Expected}$$

- $Expected \ = \frac{(count\ A)*(count\ B)}{n}$

- The larger the X^2 value, the more likely the variables are related

Chi-Square Calculation: An Example

	Play chess	Not play chess	Sum (row): A
Like science fiction	250 (exp: 90)	200 (exp:360)	450
Not like science fiction	50 (exp:210)	1000(exp:840)	1050
Sum(col.): B	300	1200	1500 (n)

$$expected = \frac{(count\ A)*(count\ B)}{n} = \frac{450*300}{1500} = 90$$

$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840} = 507.93$$

- It shows that like_science_fiction and play_chess are correlated in the group

- We can get X^2 by:

$$\begin{aligned}\chi^2 &= \frac{(250-90)^2}{90}+\frac{(50-210)^2}{210}+\frac{(200-360)^2}{360}+\frac{(1000-840)^2}{840} \\ &= 284.44+121.90+71.11+30.48=507.93.\end{aligned}$$

- For this 2 x 2 table, the degrees of freedom are (2-1)(2-1) = 1.
- For 1 degree of freedom, the X^2 value needed to reject the hypothesis at the 0.001 significance level is 10.828 (taken from the table of upper percentage points of the X^2 distribution, typically available from any textbook on statistics).

507.93 >10.82

Calculated value is more than tabulated value of X^2

So,

Null hypothesis: play chess and preferred reading are independent (not related)

is rejected and conclude that the two attributes are strongly correlated for the given group of people

Correlation Analysis (Numeric Data)

- Correlation coefficient (also called Pearson's product moment coefficient)

$$r_{A,B} = \frac{Cov(A, B)}{\sigma_A \sigma_B}$$

- If $r_{A,B} > 0$,

  A and B are positively correlated (A's values increase as B's). The higher, the stronger correlation.

- $r_{A,B} = 0$: independent;

- $r_{A,B} < 0$: negatively correlated

Covariance (Numeric Data)

$$Cov(A,B) = E(A \cdot B) - \bar{A}\bar{B}$$

where $\overline{A}$and $\overline{B}$ are the respective mean or **expected values** of A and B

Positive covariance: If $\text{Cov}_{A,B} > 0$, then A and B both tend to be larger than their expected values

Negative covariance: If $\text{Cov}_{A,B} < 0$ then if A is larger than its expected value, B is likely to be smaller than its expected value

Independence: $\text{Cov}_{A,B} = 0$ but the converse is not true:

> Some pairs of random variables may have a covariance of 0 but are not independent. Only under some additional assumptions (e.g., the data follow multivariate normal distributions) does a covariance of 0 imply independence

Co-Variance: An Example

- It can be simplified in computation as

$$Cov(A, B) = E(A \cdot B) - \bar{A}\bar{B}$$

Example:

- Suppose two stocks A and B have the following values in one week: (2, 5), (3, 8), (5, 10), (4, 11), (6, 14).

- **Question: If the stocks are affected by the same industry trends, will their prices rise or fall together?**
  - $\bar{A}$ = (2 + 3 + 5 + 4 + 6)/ 5 = 20/5 = 4
  - $\bar{B}$ = (5 + 8 + 10 + 11 + 14) /5 = 48/5 = 9.6
  - Cov(A,B) = (2×5+3×8+5×10+4×11+6×14)/5 − 4 × 9.6 = 4

- Thus, A and B rise together since Cov(A, B) > 0.

Days	Stock A	Stock B
Monday	2	5
Tuesday	3	8
Wednesday	5	10
Thursday	4	11
Friday	6	14

Data Transformation

- A function that maps the entire set of values of a given attribute to a new set of replacement values so that each old value can be identified with one of the new values
- Methods
  - **Smoothing**: Remove noise from data
    - Techniques include binning, regression, clustering
  - **Attribute/feature construction**
    - New attributes constructed from the given ones
  - **Aggregation**: Summarization, data cube construction
    - Eg. Daily sales data aggregated to monthly and annual income

Data Transformation

- **Normalization**: attribute data are scaled to fall within a smaller range such as (-1.0 to 1.0) or (0.0 to 1.0)
  - min-max normalization
  - z-score normalization
  - normalization by decimal scaling

- **Discretization**: raw values of <u>numeric attributes </u>(e.g. age) are replaced by interval labels (0-10, 11-20) or conceptual labels(youth, adult, senior) resulting in Concept hierarchy for the numeric attributes

- **Concept hierarchy generation for <u>nominal data</u>:**
- Attribute such as street can be generalized to higher level concept, like city or country

Normalization

- **Min-max normalization**: it maps a value v of attribute A to new value v’ in the range [new_min$_A$, new_max$_A$] by computing,

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

  - Ex. Let min and max value for the attribute income are $12,000 and $98,000 resp. Now map income to the range [0.0, 1.0]. Then the value $73,600 for income is transformed as,

$$\frac{73{,}600 - 12{,}000}{98{,}000 - 12{,}000}(1.0 - 0) + 0 = 0.716$$

Normalization

- **Z-score normalization:** The values for attribute A are normalized based on mean and (µ) standard deviation(σ) of attribute A . Formula is given by,

$$v' = \frac{v - \mu_A}{\sigma_A}$$

  - Ex. Let µ = 54,000, σ = 16,000 for attribute income. With z-score normalization, a value $73600 for income is transformed to , $\frac{73,600 - 54,000}{16,000} = 1.225$

Normalization

- **Normalization by decimal scaling**: It transform the value by moving the decimal point of value of attribute A. The number of decimal points moved depends on the maximum absolute value of A.
- formula is given by, $v' = \frac{v}{10^j}$ Where j is the smallest integer such that Max(|v'|) < 1

- Eg. Let for attribute A range is -986 to 927. To normalize by decimal scaling, divide each value by 1000(i.e. j=3).

  Therefore -986 is normalized to -0.986 and 917 is normalized to 0.917.

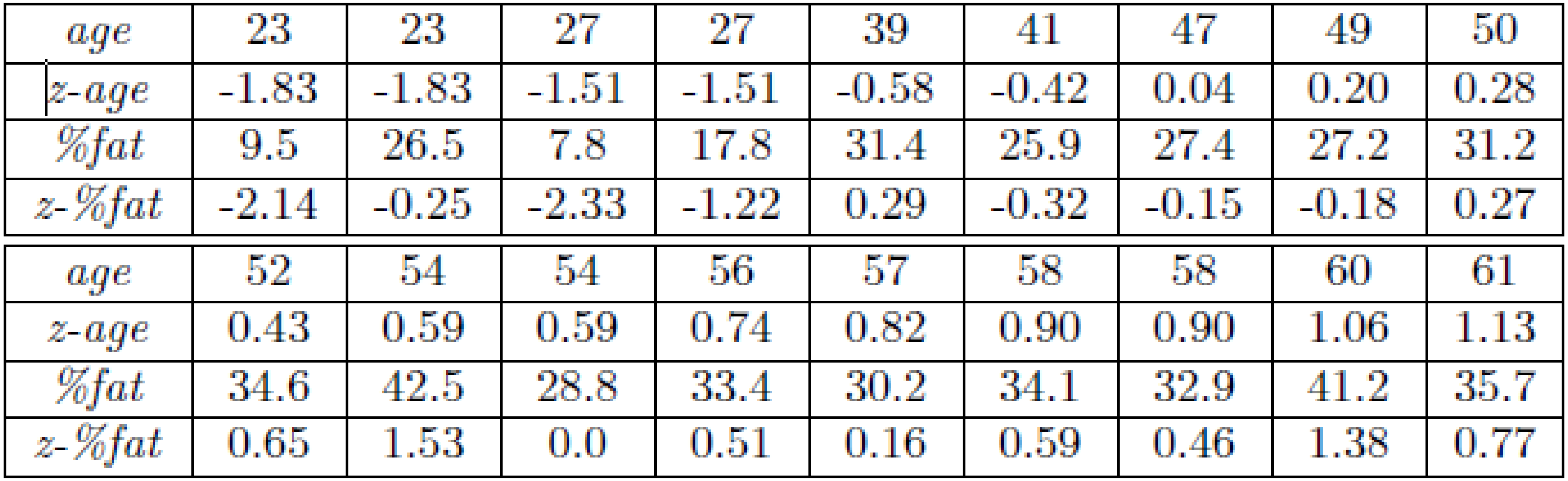
Example

Use the two methods below to *normalize* the following group of data:

200*;* 300*;* 400*;* 600*;* 1000

(a) min-max normalization by setting *min* = 0 and *max* = 1

(b) z-score normalization



age	23	23	27	27	39	41	47	49	50	52	54	54	56	57	58	58	60	61
z-age	-1.83	-1.83	-1.51	-1.51	-0.58	-0.42	0.04	0.20	0.28	0.43	0.59	0.59	0.74	0.82	0.90	0.90	1.06	1.13
%fat	9.5	26.5	7.8	17.8	31.4	25.9	27.4	27.2	31.2	34.6	42.5	28.8	33.4	30.2	34.1	32.9	41.2	35.7
z-%fat	-2.14	-0.25	-2.33	-1.22	0.29	-0.32	-0.15	-0.18	0.27	0.65	1.53	0.0	0.51	0.16	0.59	0.46	1.38	0.77

Using the data for *age* given in Exercise 2.4, answer the following:

(a) Use min-max normalization to transform the value 35 for *age* onto the range [0*:*0*;* 1*:*0]

(b) Use z-score normalization to transform the value 35 for *age*, where the standard deviation of *age* is 12.94 years.

(c) Use normalization by decimal scaling to transform the value 35 for *age*.

Concept Hierarchy Generation

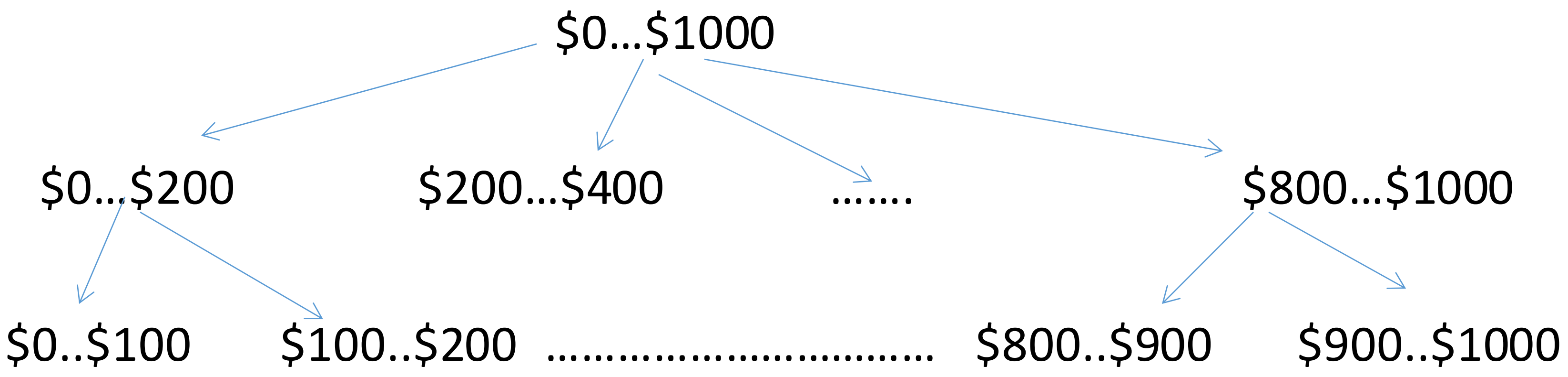
- **Concept hierarchy** organizes concepts (i.e., attribute values) hierarchically and is usually associated with each dimension in a data warehouse

- Concept hierarchies facilitate drilling and rolling in data warehouses to view data in multiple granularity

- Concept hierarchy formation: Recursively reduce the data by collecting and replacing low level concepts (such as numeric values for *age*) by higher level concepts (such as *youth, adult*, or *senior*)

- Concept hierarchies can be explicitly specified by domain experts and/or data warehouse designers

Concept hierarchy for the numeric attributes (Discretization)

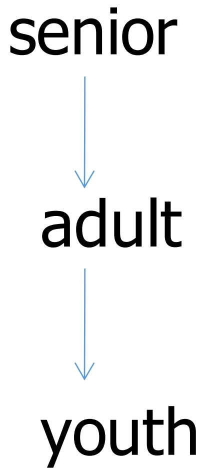


$0…$1000

$0…$200 $200…$400 ……. $800…$1000

$0..$100 $100..$200 …………………………… $800..$900 $900..$1000

Concept hierarchy for attribute price (interval labels)



senior

adult

youth

Concept hierarchy for attribute Age (conceptual labels)

Concept Hierarchy Generation for Nominal Data

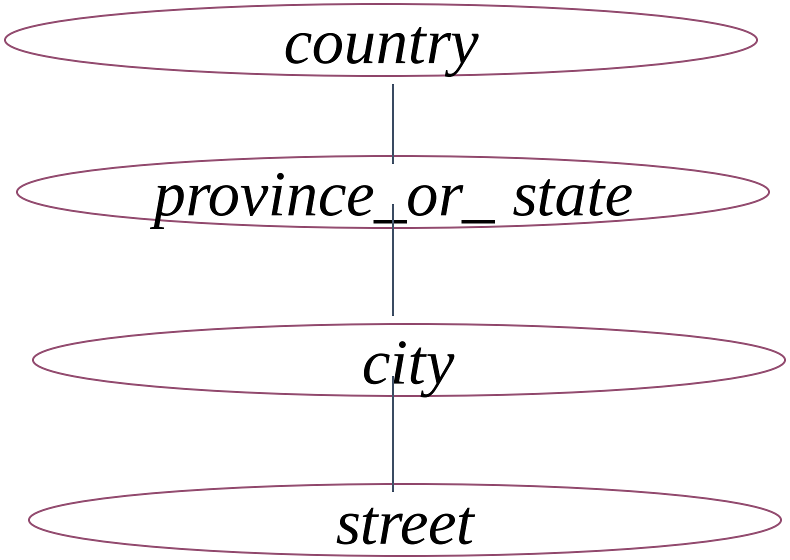
1. **Specification of a partial/total ordering of attributes explicitly at the schema level by users or experts.**
   - *street* < *city* < *state* < *country*

2. **Specification of a hierarchy for a set of values by explicit data grouping**
   - {Punjab, Haryana, Delhi} < North_India
   - {Karnataka, Tamilnadu, kerala} < South_India

3. **Automatic generation of hierarchies (or attribute levels) by the analysis of the number of distinct values**

Automatic Concept Hierarchy Generation

- Some hierarchies can be automatically generated based on the analysis of the number of distinct values per attribute in the data set
  - The attribute with the most distinct values is placed at the lowest level of the hierarchy



country	15 distinct values
province_or_ state	365 distinct values
city	3567 distinct values
street	674,339 distinct values

Data Discretization Methods

- Typical methods: All the methods can be applied recursively
  - Binning
    - Top-down split, unsupervised
  - Histogram analysis
    - Top-down split, unsupervised
  - Clustering analysis (unsupervised, top-down split or bottom-up merge)
  - Decision-tree analysis (supervised, top-down split)
  - Correlation (e.g., χ^2) analysis (unsupervised, bottom-up merge)

Data Reduction

- Why data reduction?
  - A database/data 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
  - Obtain a reduced representation of the data set that is much smaller in volume but yet produce the same (or almost the same) analytical results

Data reduction strategies

- Data cube aggregation:
- Eg. Total sales per year instead of per quarter

- Dimensionality reduction: original data is transform into smaller space. Encoding methods are used.
- Eg. Wavelet transform and principal component analysis

- Attribute subset selection: remove unimportant(irrelevant, redundant, weakly relevant) attributes
- Eg. For new CD purchase, customers phone number is irrelevant

Data reduction strategies

- Numerosity reduction : replace original data by alternatives smaller form of data representation
  - Parametric method: model is used to estimate the data
    - Eg. Regression, log-linear models
  - Non Parametric method
    - Eg. Histograms, clustering, sampling and Data cube aggregation

Data Cube Aggregation

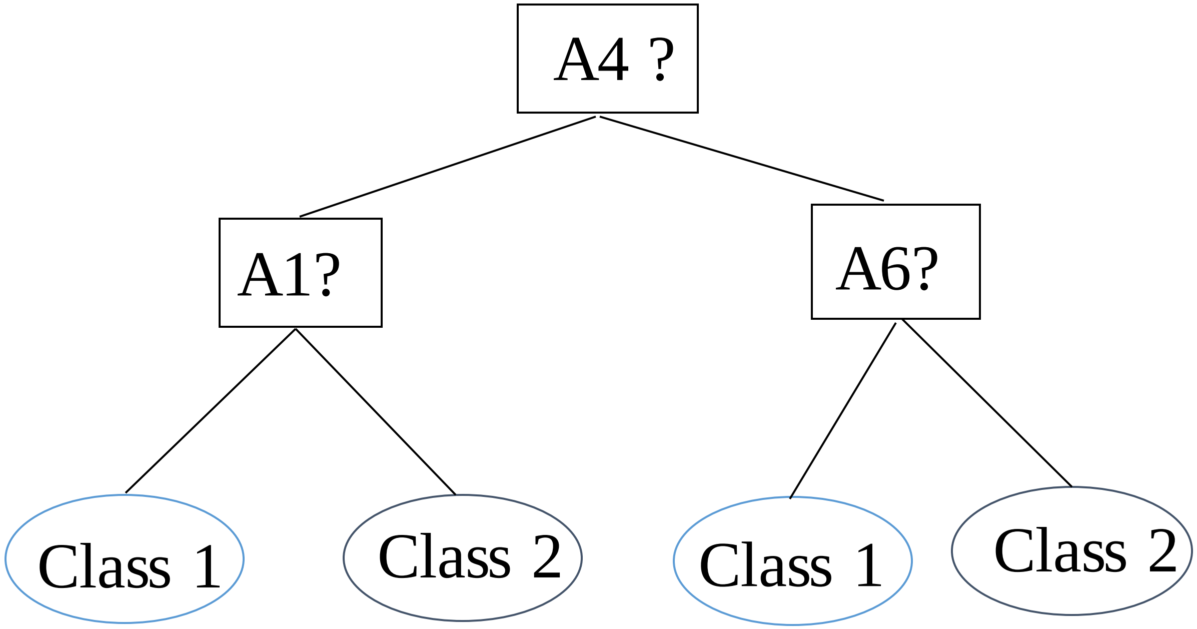
- The lowest level of a data cube (base cuboid)
  - The aggregated data for an individual entity of interest
  - E.g., a customer in a phone calling data warehouse
- Multiple levels of aggregation in data cubes
  - Further reduce the size of data to deal with
- Reference appropriate levels
  - Use the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

Attribute Subset Selection

- 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}

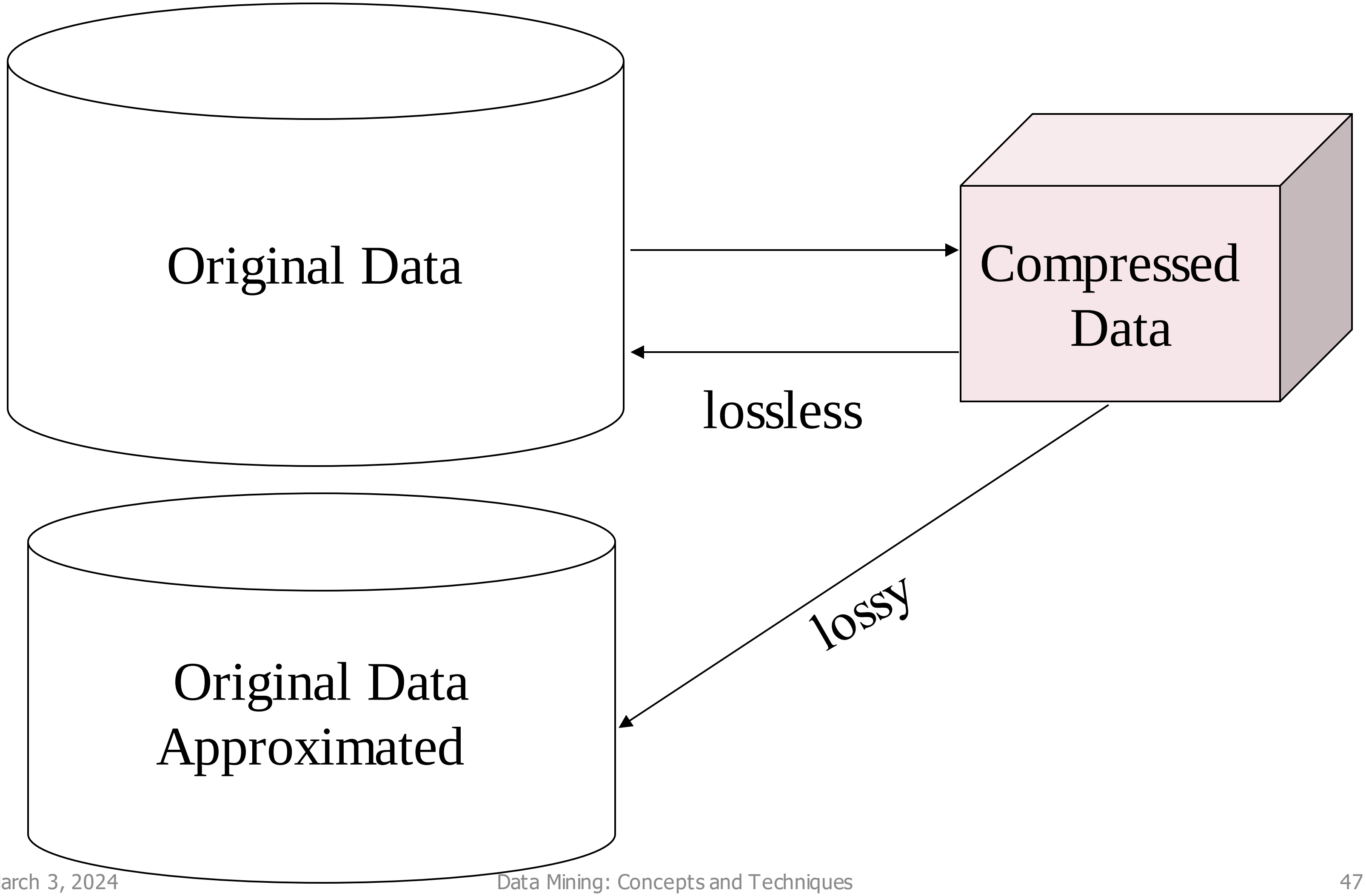
Heuristic Feature Selection Methods

- 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

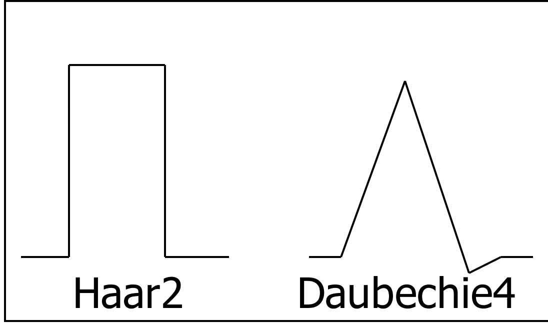
Data Compression

- 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

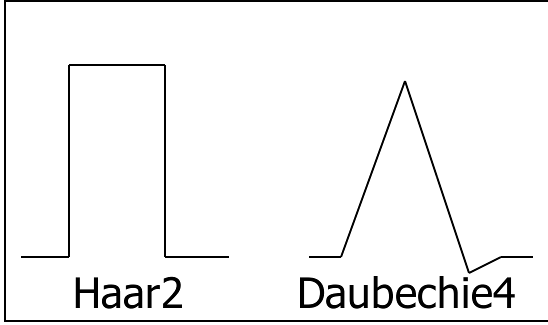


Dimensionality Reduction: Wavelet Transformation



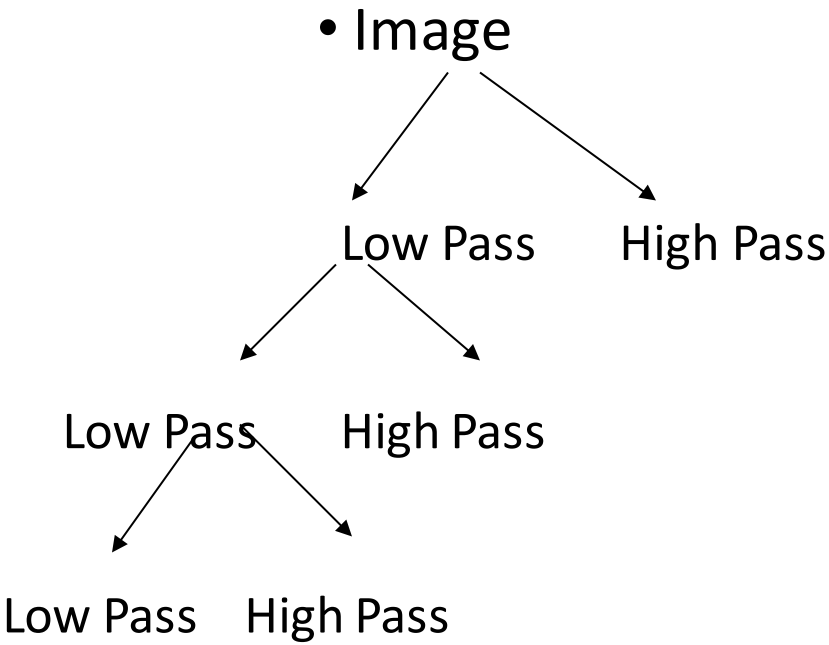
- Data encoding techniques or transformation are applied on the original data to obtain a reduced or compressed representation of the data.
- Reconstructed data can be lossy or lossless
- Lossy dimensionality reduction methods:
    - Wavelet transforms(DWT)
        - Haar tranform
    - Principle component analysis(PCA)
        - K-L method

Dimensionality Reduction: Wavelet Transformation



- Discrete wavelet transform (DWT): linear signal processing, multi-resolutional analysis
- 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 0’s, 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

DWT for Image Compression

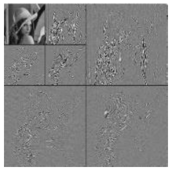


- Image

Low Pass High Pass

Low Pass High Pass

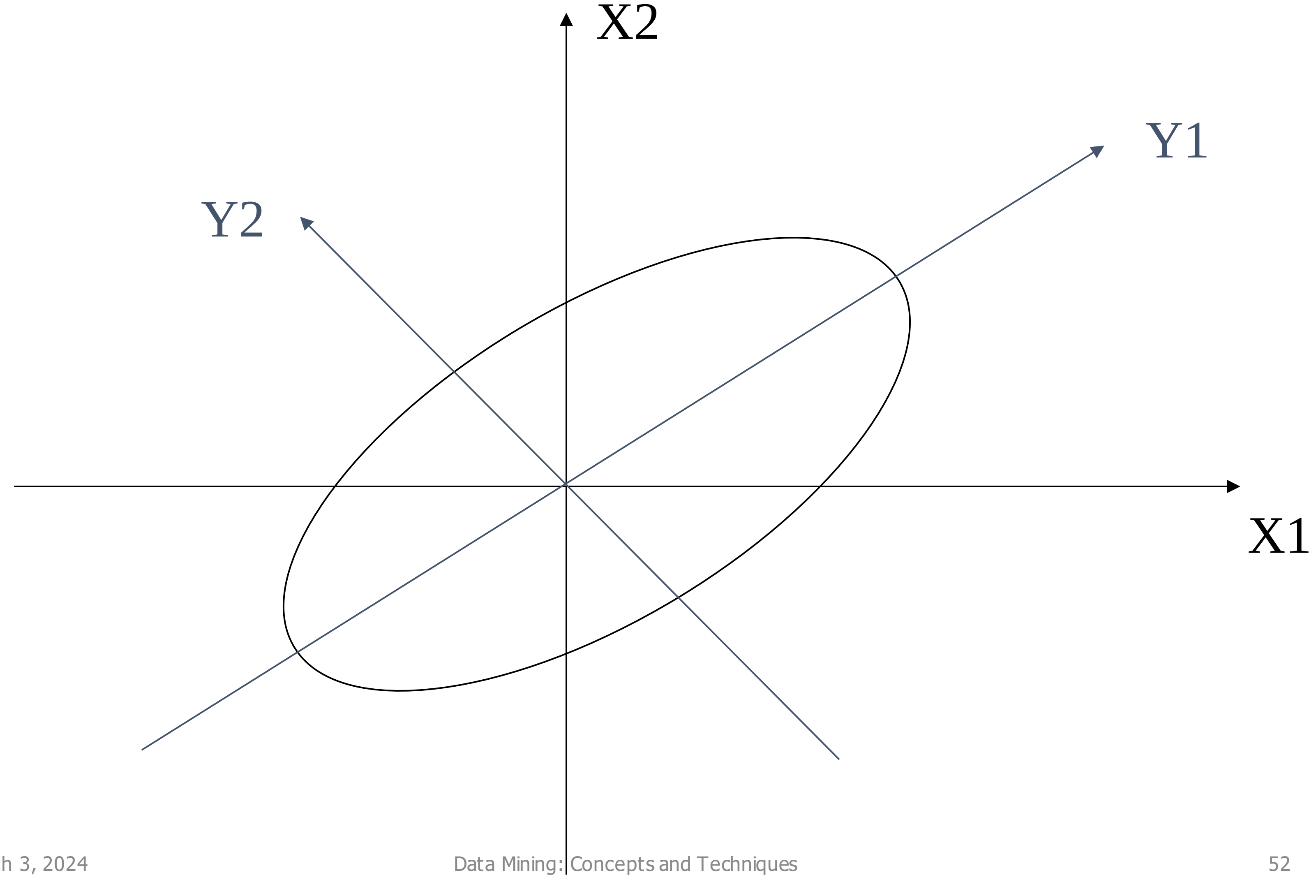
Low Pass High Pass



Dimensionality Reduction: Principal Component Analysis (PCA)

- Given *N* data vectors from *n*-dimensions, find *k* ≤ *n* orthogonal vectors (*principal components*) that can be best used to represent data
- Steps
  - Normalize input data: Each attribute falls within the same range
  - Compute *k* orthonormal (unit) vectors, i.e., *principal components*
  - Each input data (vector) is a linear combination of the *k* principal component vectors
  - The principal components are sorted in order of decreasing “significance” or strength
  - Since the components are sorted, the size of the data can be reduced by eliminating the weak components, i.e., those with low variance. (i.e., using the strongest principal components, it is possible to reconstruct a good approximation of the original data
- Works for numeric data only
- Used when the number of dimensions is large

Principal Component Analysis



Numerosity Reduction

- Reduce data volume by choosing alternative, smaller forms of data representation
- Parametric methods
  - Assume the data fits some model, estimate model parameters, store only the parameters, and discard the data (except possible outliers)
  - Example: Regression and 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

Data Reduction Method (1): Regression and Log-Linear Models

- Linear regression: Data are modeled to fit a straight line

$$y=wx + b$$

Where, y and x--→numerical database attributes

w and b-→ regression coefficient

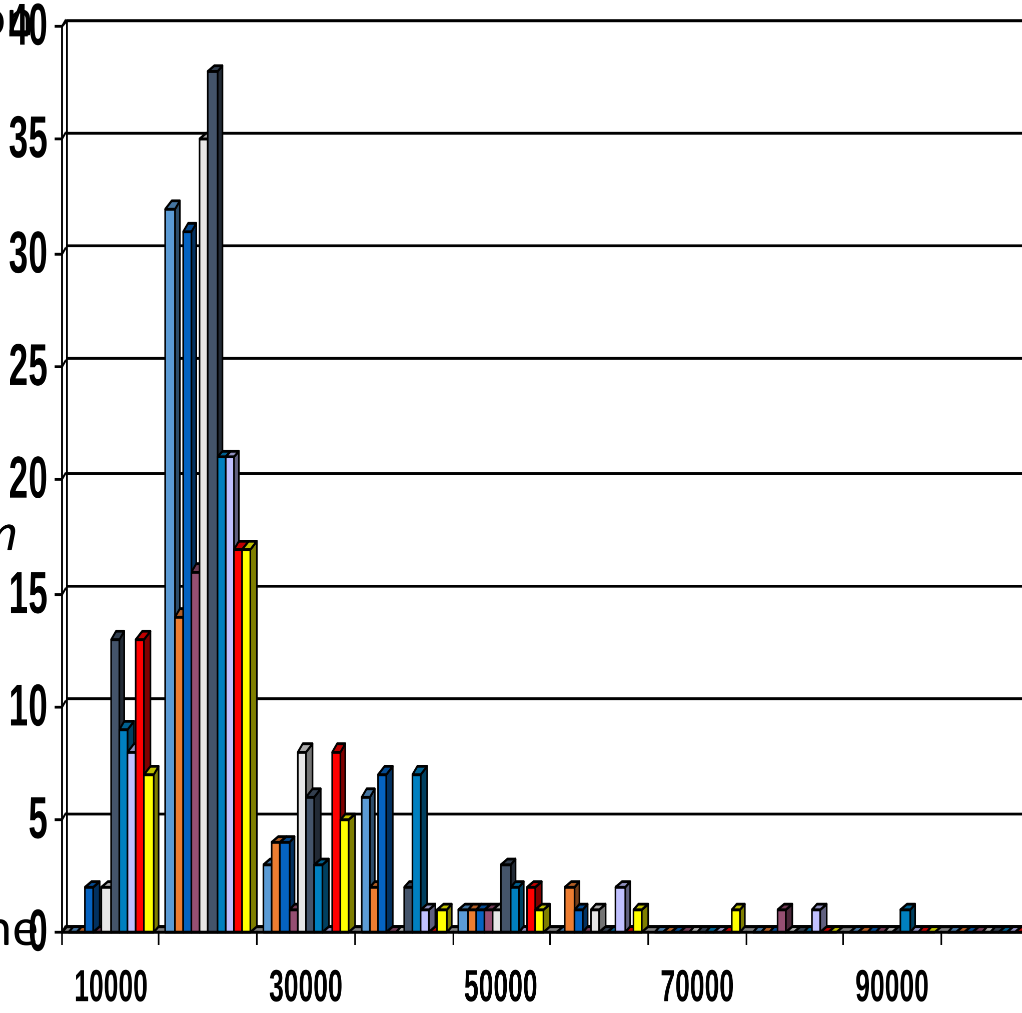
- Multiple regression: extension of linear regression method where y is modeled as a linear function of two or more predictor variable
- Log-linear model: estimate the probability of each point in a multidimentional space for a set of discretized attributes, based on the smaller subset of dimensional combinations

Regress Analysis and Log-Linear Models

- Linear regression: $Y = w X + b$
  - Two regression coefficients, *w* and *b,* 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: $p(a, b, c, d) = \alpha ab\ \beta ac \chi ad\ \delta bcd$

Data Reduction Method (2): Histograms

- Histogram for an attribute A→ partition data distribution of A into disjoint subsets or buckets.
- Partitioning rules:
  - Equal-width: equal bucket range
  - Equal-frequency (or equal-depth)
  - V-optimal: with the least *histogram variance* (weighted sum of the original values that each bucket represents)
  - MaxDiff: set bucket boundary between each pair for pairs have the β–1 largest differences



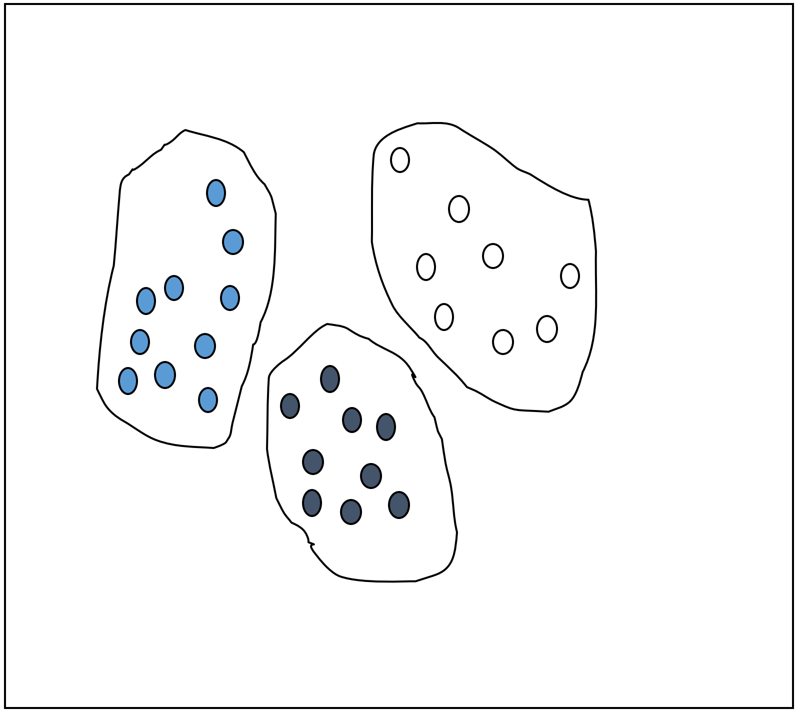
- List of prices of commonly sold items at AllElectronics(rounded to nearest dollar)
- 1,1,5,5,5,5,8,8,10,10,10,10,10,12,14,14,14,14,15,15,15,15,15,15

Data Reduction Method (3): Clustering

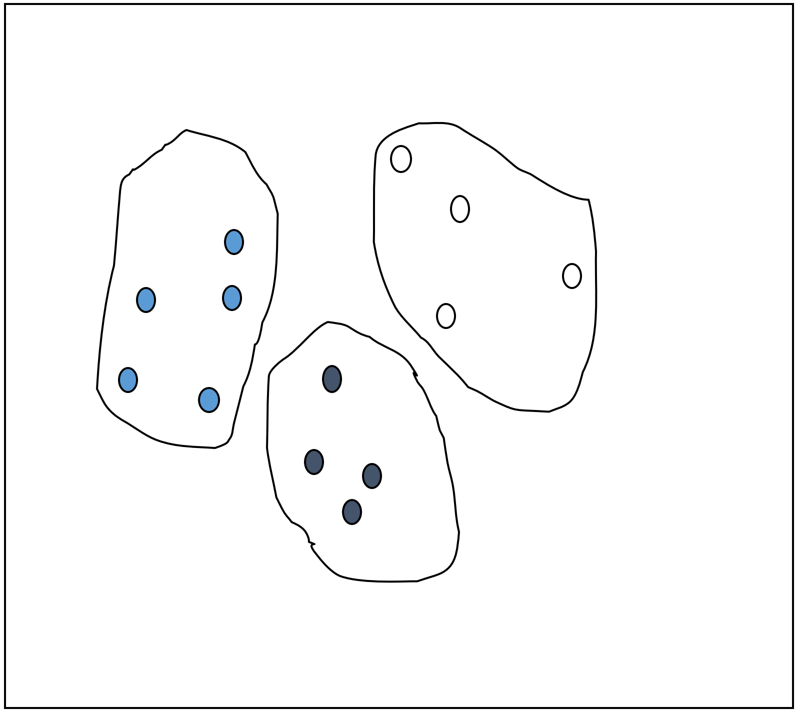
- Partition data set into clusters based on similarity, and store cluster representation (e.g., centroid and diameter) only
- Can be very effective if data is clustered but not if data is “smeared”
- Can have hierarchical clustering and be stored in multi-dimensional index tree structures
- There are many choices of clustering definitions and clustering algorithms
- Cluster analysis will be studied in depth in Chapter 7

Sampling: Cluster or Stratified Sampling

Raw Data



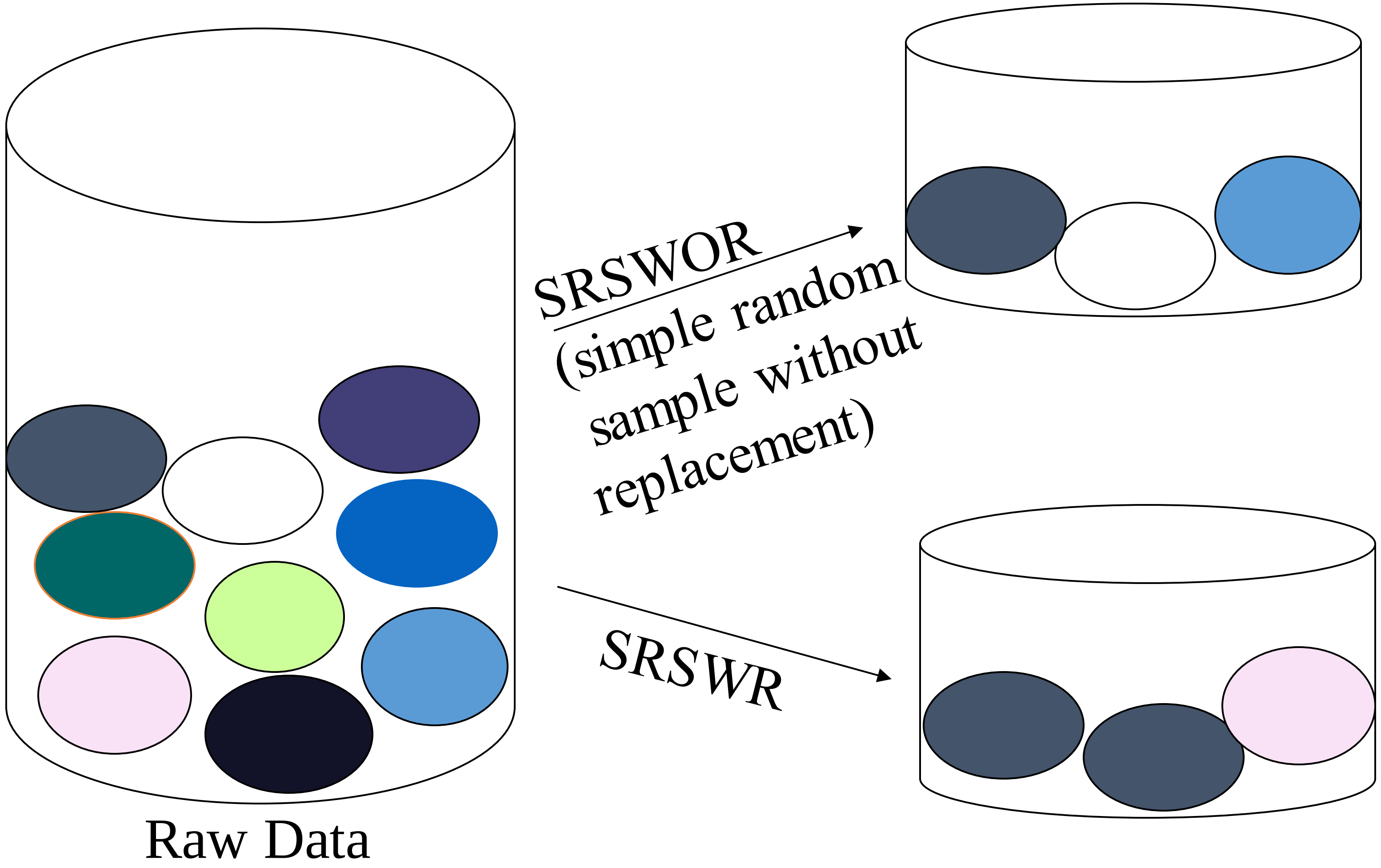
Cluster/Stratified Sample



Data Reduction Method (4): Sampling

- Sampling: obtaining a small sample *s* to represent the whole data set *N*
- 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
- Note: Sampling may not reduce database I/Os (page at a time)

Sampling: with or without Replacement


SRSWOR
(simple random
sample without
replacement)
SRSWR
Raw Data