



CS 412 Intro. to Data Mining


Chapter 3. Data Preprocessing

Jiawei Han, Computer Science, Univ. Illinois at Urbana-Champaign, 2017





Chapter 3: Data Preprocessing

- ❑ Data Preprocessing: An Overview 
- ❑ Data Cleaning
- ❑ Data Integration
- ❑ Data Reduction and Transformation
- ❑ Dimensionality Reduction
- ❑ Summary

What is Data Preprocessing? — Major Tasks

- ❑ **Data cleaning**

- ❑ Handle missing data, smooth noisy data, identify or remove outliers, and resolve inconsistencies

- ❑ **Data integration**

- ❑ Integration of multiple databases, data cubes, or files

- ❑ **Data reduction**

- ❑ Dimensionality reduction
 - ❑ Numerosity reduction
 - ❑ Data compression


- ❑ **Data transformation and data discretization**

- ❑ Normalization
 - ❑ Concept hierarchy generation

Why Preprocess the Data? — Data Quality Issues

- ❑ Measures for data quality: A multidimensional view
 - ❑ Accuracy: correct or wrong, accurate or not
 - ❑ Completeness: not recorded, unavailable, ...
 - ❑ Consistency: some modified but some not, dangling, ...
 - ❑ Timeliness: timely update?
 - ❑ Believability: how trustable the data are correct?
 - ❑ Interpretability: how easily the data can be understood?

Chapter 3: Data Preprocessing

- ❑ Data Preprocessing: An Overview
- ❑ Data Cleaning 
- ❑ Data Integration
- ❑ Data Reduction and Transformation
- ❑ Dimensionality Reduction
- ❑ Summary

Data Cleaning

- ❑ Data in the Real World Is Dirty: Lots of potentially incorrect data, e.g., instrument faulty, human or computer error, and transmission error
- ❑ Incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
 - ❑ e.g., *Occupation* = “ ” (missing data)
- ❑ Noisy: containing noise, errors, or outliers
 - ❑ e.g., *Salary* = “-10” (an error)
- ❑ Inconsistent: containing discrepancies in codes or names, e.g.,
 - ❑ *Age* = “42”, *Birthday* = “03/07/2010”
 - ❑ Was rating “1, 2, 3”, now rating “A, B, C”
 - ❑ discrepancy between duplicate records
- ❑ Intentional (e.g., *disguised missing* data)
 - ❑ Jan. 1 as everyone’s birthday?

Incomplete (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 were not entered due to misunderstanding
 - ❑ Certain data may not be considered important at the time of entry
 - ❑ Did 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 (when doing classification)—not effective when the % of missing values per attribute varies considerably
- ❑ Fill in the missing value manually: tedious + infeasible?
- ❑ Fill in it automatically with
 - ❑ a global constant : e.g., “unknown”, a new class?!
 - ❑ 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 be due to
 - ❑ Faulty data collection instruments
 - ❑ Data entry problems
 - ❑ Data transmission problems
 - ❑ Technology limitation
 - ❑ Inconsistency in naming convention
- ❑ **Other data problems**
 - ❑ Duplicate records
 - ❑ Incomplete data
 - ❑ Inconsistent data

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
- ❑ Semi-supervised: Combined computer and human inspection
 - ❑ Detect suspicious values and check by human (e.g., deal with possible outliers)

Data Cleaning as a Process


❑ Data discrepancy detection

- ❑ Use metadata (e.g., domain, range, dependency, distribution)
- ❑ Check field overloading
- ❑ Check uniqueness rule, consecutive rule and null rule
- ❑ Use commercial tools
 - ▢ Data scrubbing: use simple domain knowledge (e.g., postal code, spell-check) to detect errors and make corrections
 - ▢ Data auditing: by analyzing data to discover rules and relationship to detect violators (e.g., correlation and clustering to find outliers)

❑ Data migration and integration

- ❑ Data migration tools: allow transformations to be specified
- ❑ ETL (Extraction/Transformation/Loading) tools: allow users to specify transformations through a graphical user interface
- ❑ Integration of the two processes
 - ❑ Iterative and interactive (e.g., Potter's Wheels)

Chapter 3: Data Preprocessing

- ❑ Data Preprocessing: An Overview
- ❑ Data Cleaning
- ❑ Data Integration 
- ❑ Data Reduction and Transformation
- ❑ Dimensionality Reduction
- ❑ Summary

Data Integration

- Data integration
 - Combining data from multiple sources into a coherent store
- Schema integration: e.g., $A.cust-id \equiv B.cust-\#$
 - Integrate metadata from different sources
- **Entity identification:**
 - 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* and *covariance analysis***
- ❑ Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality

Correlation Analysis (for Categorical Data)

□ χ^2 (chi-square) test:

$$\chi^2 = \sum_i^n \frac{\overset{\text{observed}}{\downarrow} (O_i - E_i)^2}{\underset{\text{expected}}{E_i}}$$

- Null hypothesis: The two distributions are independent
- The cells that contribute the most to the χ^2 value are those whose actual count is very different from the expected count
 - The larger the χ^2 value, the more likely the variables are related
- Note: Correlation does not imply causality
 - # of hospitals and # of car-theft in a city are correlated
 - Both are causally linked to the third variable: population


Chi-Square Calculation: An Example

| | Play chess | Not play chess | Sum (row) |
|--------------------------|------------|----------------|-----------|
| Like science fiction | 250 (90) | 200 (360) | 450 |
| Not like science fiction | 50 (210) | 1000 (840) | 1050 |
| Sum(col.) | 300 | 1200 | 1500 |

How to derive 90?
 $450/1500 * 300 = 90$

We can reject the null hypothesis of independence at a confidence level of 0.001

- χ^2 (chi-square) calculation (numbers in parenthesis are expected counts calculated based on the data distribution in the two categories)

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


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

Variance for Single Variable (Numerical Data)

- The variance of a random variable X provides a measure of how much the value of X deviates from the mean or expected value of X :

$$\sigma^2 = \text{var}(X) = E[(X - \mu)^2] = \begin{cases} \sum_x (x - \mu)^2 f(x) & \text{if } X \text{ is discrete} \\ \int_{-\infty}^{\infty} (x - \mu)^2 f(x) dx & \text{if } X \text{ is continuous} \end{cases}$$

- where σ^2 is the variance of X , σ is called *standard deviation*
 μ is the mean, and $\mu = E[X]$ is the expected value of X
- That is, variance is the expected value of the square deviation from the mean
- It can also be written as: $\sigma^2 = \text{var}(X) = E[(X - \mu)^2] = E[X^2] - \mu^2 = E[X^2] - [E(x)]^2$
- Sample variance is the average squared deviation of the data value x_i from the sample mean $\hat{\mu}$
$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{\mu})^2$$

Covariance for Two Variables

- Covariance between two variables X_1 and X_2

$$\sigma_{12} = E[(X_1 - \mu_1)(X_2 - \mu_2)] = E[X_1 X_2] - \mu_1 \mu_2 = E[X_1 X_2] - E[X_1]E[X_2]$$

where $\mu_1 = E[X_1]$ is the respective mean or **expected value** of X_1 ; similarly for μ_2

- Sample covariance between X_1 and X_2 : $\hat{\sigma}_{12} = \frac{1}{n} \sum_{i=1}^n (x_{i1} - \hat{\mu}_1)(x_{i2} - \hat{\mu}_2)$

- Sample covariance is a generalization of the sample variance:

$$\hat{\sigma}_{11} = \frac{1}{n} \sum_{i=1}^n (x_{i1} - \hat{\mu}_1)(x_{i1} - \hat{\mu}_1) = \frac{1}{n} \sum_{i=1}^n (x_{i1} - \hat{\mu}_1)^2 = \hat{\sigma}_1^2$$

- **Positive covariance:** If $\sigma_{12} > 0$

- **Negative covariance:** If $\sigma_{12} < 0$

- **Independence:** If X_1 and X_2 are independent, $\sigma_{12} = 0$ but the reverse is not true

- Some pairs of random variables may have a covariance 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

Example: Calculation of Covariance

- Suppose two stocks X_1 and X_2 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?

- Covariance formula

$$\sigma_{12} = E[(X_1 - \mu_1)(X_2 - \mu_2)] = E[X_1X_2] - \mu_1\mu_2 = E[X_1X_2] - E[X_1]E[X_2]$$

- Its computation can be simplified as: $\sigma_{12} = E[X_1X_2] - E[X_1]E[X_2]$

- $E(X_1) = (2 + 3 + 5 + 4 + 6) / 5 = 20/5 = 4$

- $E(X_2) = (5 + 8 + 10 + 11 + 14) / 5 = 48/5 = 9.6$

- $\sigma_{12} = (2 \times 5 + 3 \times 8 + 5 \times 10 + 4 \times 11 + 6 \times 14) / 5 - 4 \times 9.6 = 4$

- Thus, X_1 and X_2 rise together since $\sigma_{12} > 0$

Correlation between Two Numerical Variables

- ❑ **Correlation** between two variables X_1 and X_2 is the standard covariance, obtained by normalizing the covariance with the standard deviation of each variable

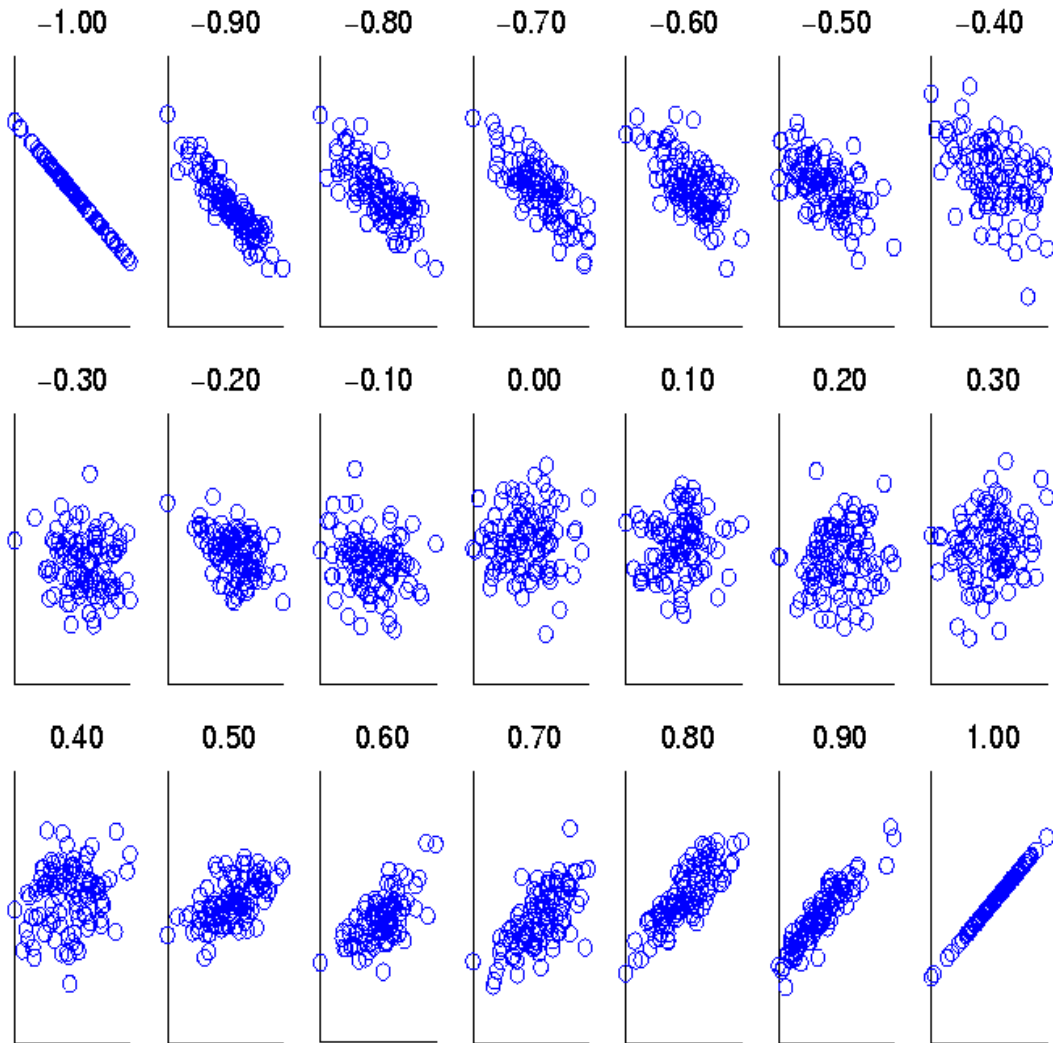
$$\rho_{12} = \frac{\sigma_{12}}{\sigma_1 \sigma_2} = \frac{\sigma_{12}}{\sqrt{\sigma_1^2 \sigma_2^2}}$$

- ❑ **Sample correlation** for two attributes X_1 and X_2 : $\hat{\rho}_{12} = \frac{\hat{\sigma}_{12}}{\hat{\sigma}_1 \hat{\sigma}_2} = \frac{\sum_{i=1}^n (x_{i1} - \hat{\mu}_1)(x_{i2} - \hat{\mu}_2)}{\sqrt{\sum_{i=1}^n (x_{i1} - \hat{\mu}_1)^2 \sum_{i=1}^n (x_{i2} - \hat{\mu}_2)^2}}$

where n is the number of tuples, μ_1 and μ_2 are the respective means of X_1 and X_2 , σ_1 and σ_2 are the respective standard deviation of X_1 and X_2

- ❑ If $\rho_{12} > 0$: A and B are positively correlated (X_1 's values increase as X_2 's)
 - ▢ The higher, the stronger correlation
- ❑ If $\rho_{12} = 0$: independent (under the same assumption as discussed in co-variance)
- ❑ If $\rho_{12} < 0$: negatively correlated

Visualizing Changes of Correlation Coefficient



- Correlation coefficient value range: $[-1, 1]$
- A set of scatter plots shows sets of points and their correlation coefficients changing from -1 to 1

Covariance Matrix

- The variance and covariance information for the two variables X_1 and X_2 can be summarized as 2 X 2 covariance matrix as

$$\begin{aligned}\Sigma &= E[(\mathbf{X} - \mu)(\mathbf{X} - \mu)^T] = E\left[\begin{pmatrix} X_1 - \mu_1 \\ X_2 - \mu_2 \end{pmatrix} \begin{pmatrix} X_1 - \mu_1 & X_2 - \mu_2 \end{pmatrix}\right] \\ &= \begin{pmatrix} E[(X_1 - \mu_1)(X_1 - \mu_1)] & E[(X_1 - \mu_1)(X_2 - \mu_2)] \\ E[(X_2 - \mu_2)(X_1 - \mu_1)] & E[(X_2 - \mu_2)(X_2 - \mu_2)] \end{pmatrix} \\ &= \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{pmatrix}\end{aligned}$$

- Generalizing it to d dimensions, we have,

$$D = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{d1} & x_{d2} & \cdots & x_{dd} \end{pmatrix} \quad \Sigma = E[(\mathbf{X} - \mu)(\mathbf{X} - \mu)^T] = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1d} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{d1} & \sigma_{d2} & \cdots & \sigma_d^2 \end{pmatrix}$$

Chapter 3: Data Preprocessing

- ❑ Data Preprocessing: An Overview
- ❑ Data Cleaning
- ❑ Data Integration
- ❑ Data Reduction and Transformation
- ❑ Dimensionality Reduction
- ❑ Summary

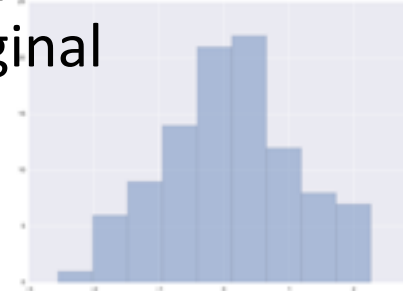
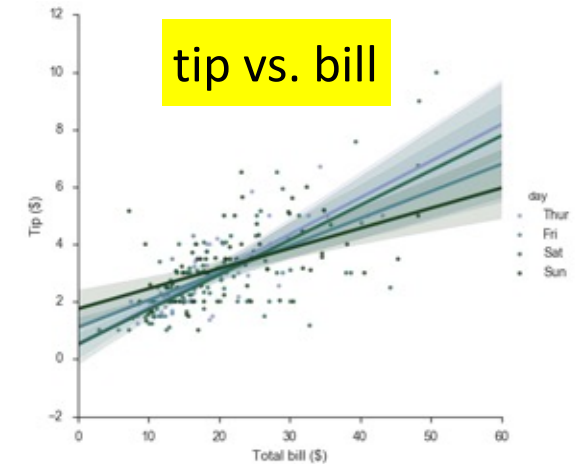


Data Reduction

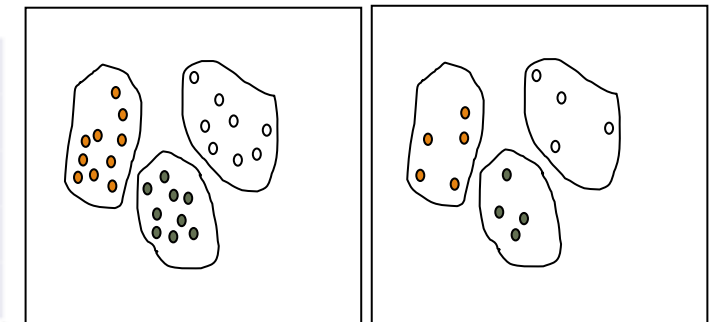
- ❑ **Data reduction:**
 - ❑ Obtain a reduced representation of the data set
 - ❑ much smaller in volume but yet produces *almost* the same analytical results
- ❑ Why data reduction?—A database/data warehouse may store terabytes of data
 - ❑ Complex analysis may take a very long time to run on the complete data set
- ❑ **Methods for data reduction** (also *data size reduction* or *numerosity reduction*)
 - ❑ Regression and Log-Linear Models
 - ❑ Histograms, clustering, sampling
 - ❑ Data cube aggregation
 - ❑ Data compression

Data Reduction: Parametric vs. Non-Parametric Methods

- ❑ Reduce data volume by choosing alternative, *smaller forms* of data representation
- ❑ **Parametric methods** (e.g., regression)
 - ❑ Assume the data fits some model, estimate model parameters, store only the parameters, and discard the data (except possible outliers)
 - ❑ Ex.: 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, ...



Histogram

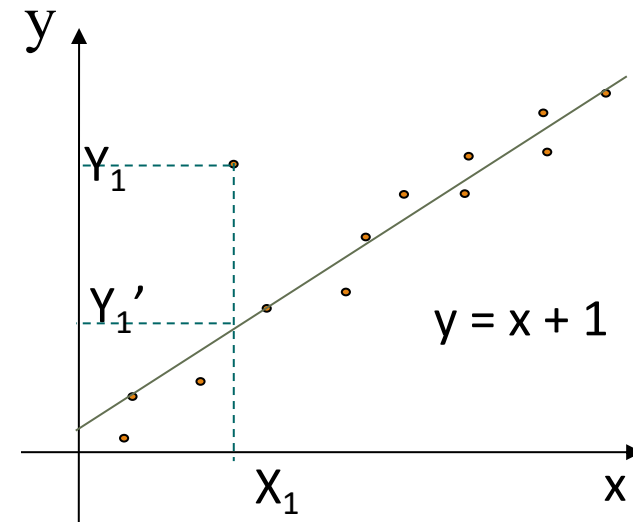


Clustering on the Raw Data

Stratified Sampling

Parametric Data Reduction: Regression Analysis

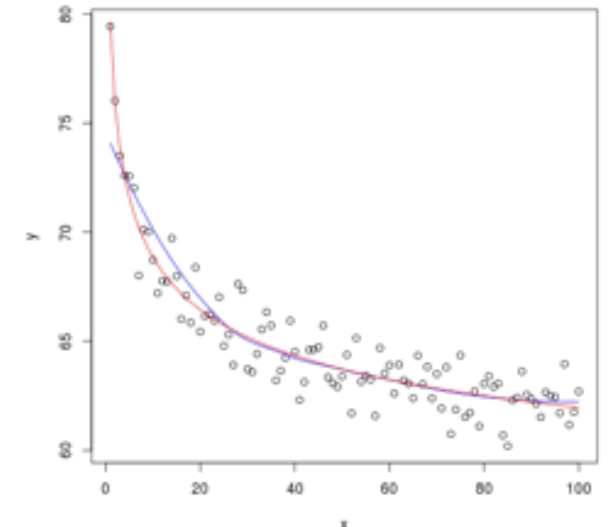
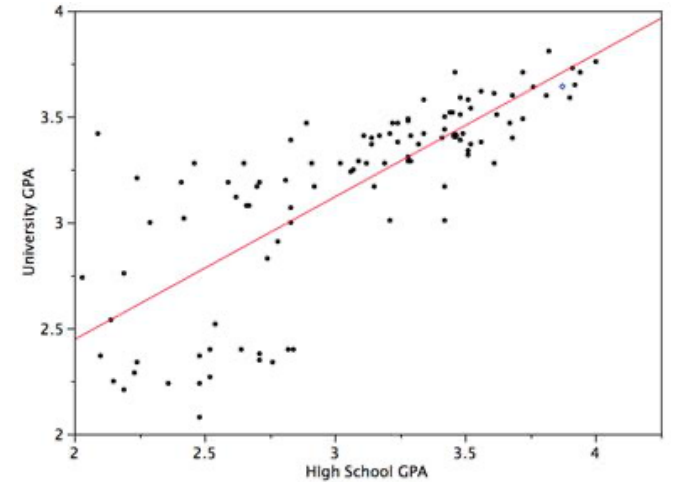
- ❑ Regression analysis: A collective name for techniques for the modeling and analysis of numerical data consisting of values of a **dependent variable** (also called **response variable** or *measurement*) and of one or more **independent variables** (also known as **explanatory variables** or **predictors**)
- ❑ The parameters are estimated so as to give a "**best fit**" of the data
- ❑ Most commonly the best fit is evaluated by using the **least squares method**, but other criteria have also been used



- ❑ Used for prediction (including forecasting of time-series data), inference, hypothesis testing, and modeling of causal relationships

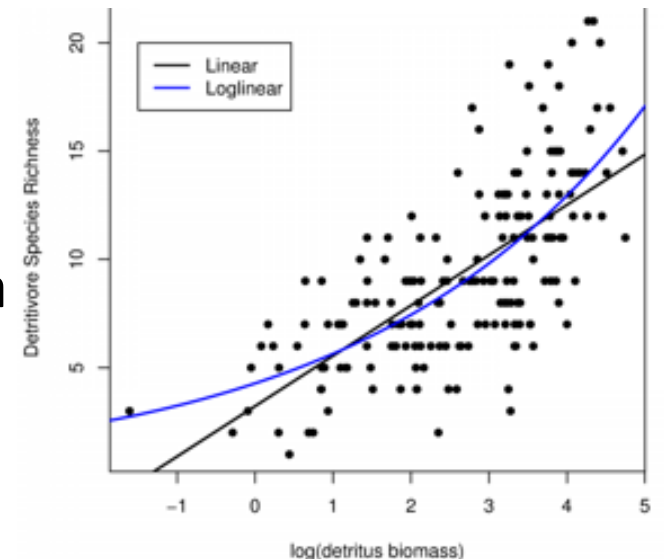
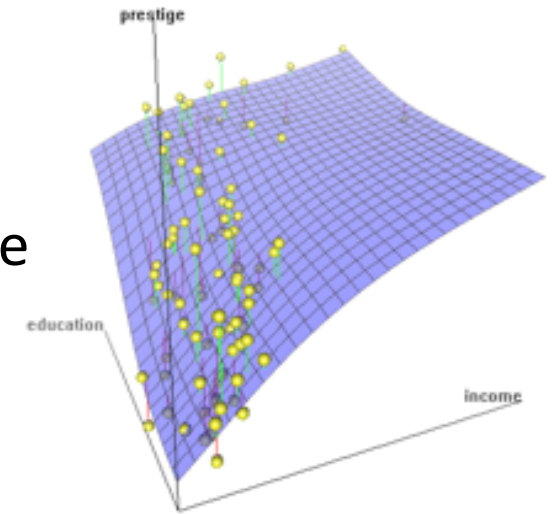
Linear and Multiple Regression

- Linear regression: $Y = wX + b$
 - Data modeled to fit a straight line
 - Often uses the least-square method to fit the line
 - 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 $Y_1, Y_2, \dots, X_1, X_2, \dots$
- Nonlinear regression:
 - Data are modeled by a function which is a nonlinear combination of the model parameters and depends on one or more independent variables
 - The data are fitted by a method of successive approximations



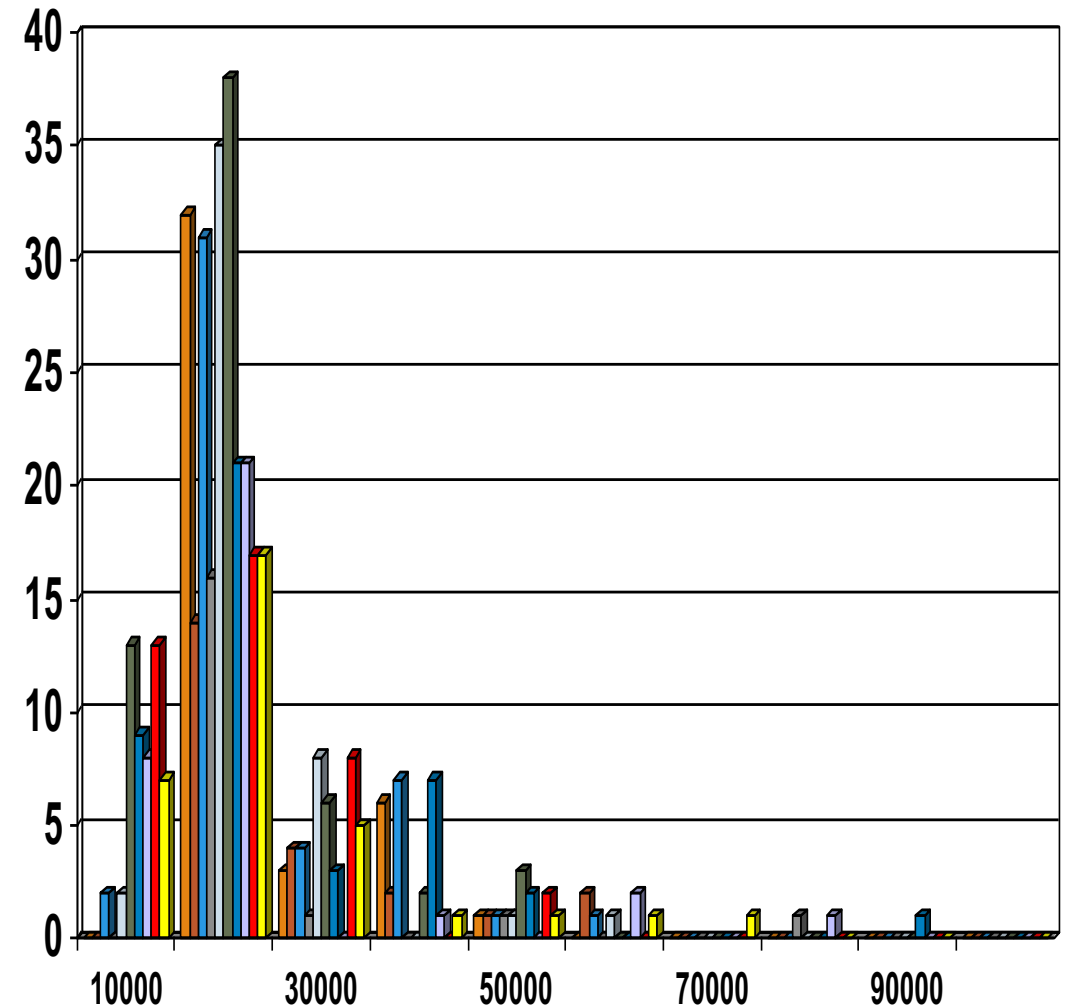
Multiple Regression and Log-Linear Models

- ❑ Multiple regression: $Y = b_0 + b_1 X_1 + b_2 X_2$
 - ❑ Allows a response variable Y to be modeled as a linear function of multidimensional feature vector
 - ❑ Many nonlinear functions can be transformed into the above
- ❑ Log-linear model:
 - ❑ A math model that takes the form of a function whose logarithm is a linear combination of the parameters of the model, which makes it possible to apply (possibly multivariate) linear regression
 - ❑ Estimate the probability of each point (tuple) in a multi-dimen. space for a set of discretized attributes, based on a smaller subset of dimensional combinations
 - ❑ Useful for dimensionality reduction and data smoothing



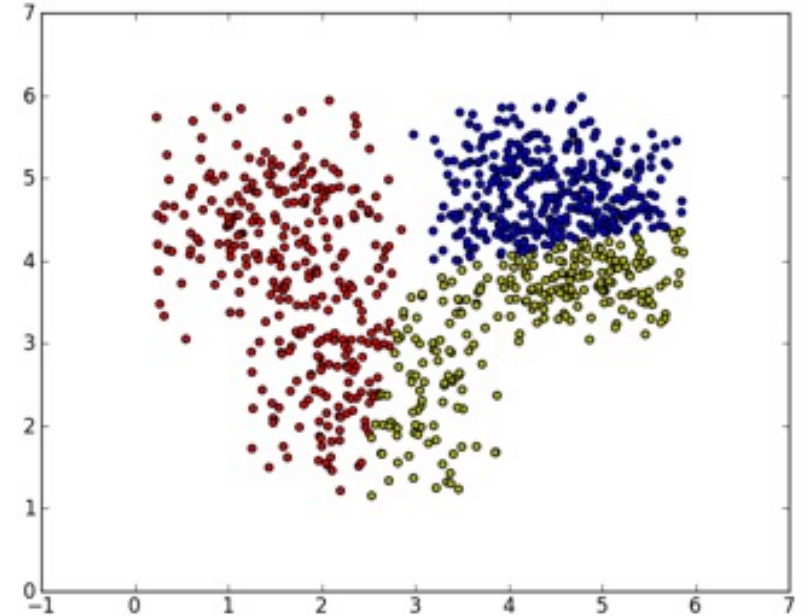
Histogram Analysis

- Divide data into buckets and store average (sum) for each bucket
- Partitioning rules:
 - Equal-width: equal bucket range
 - Equal-frequency (or equal-depth)



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 10



Announcement 1: Registration of the Overflow Session

- ❑ For those who would like to try to register our online session of the CS412 Fall classes
- ❑ Register before the end of Thursday (i.e., 11:59pm CT Thursday, Sept. 14)!
<http://go.cs.illinois.edu/CS412OverflowSection>
- ❑ Fill up all the needed information, waiting until the end of the week
- ❑ We will evaluate your application and decide if you can get into the overflown session

Announcement 2: Additional Work for the 4th Credit

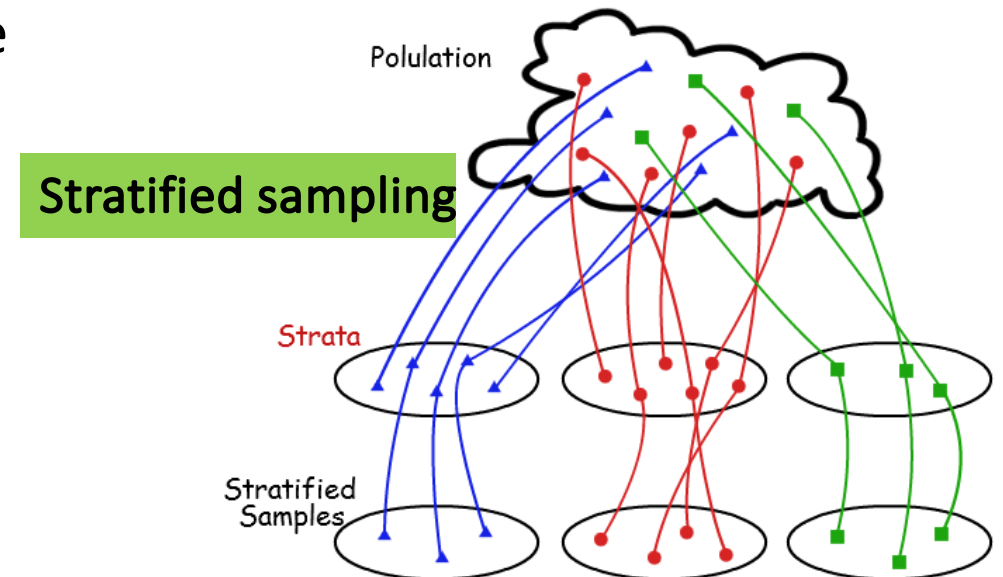
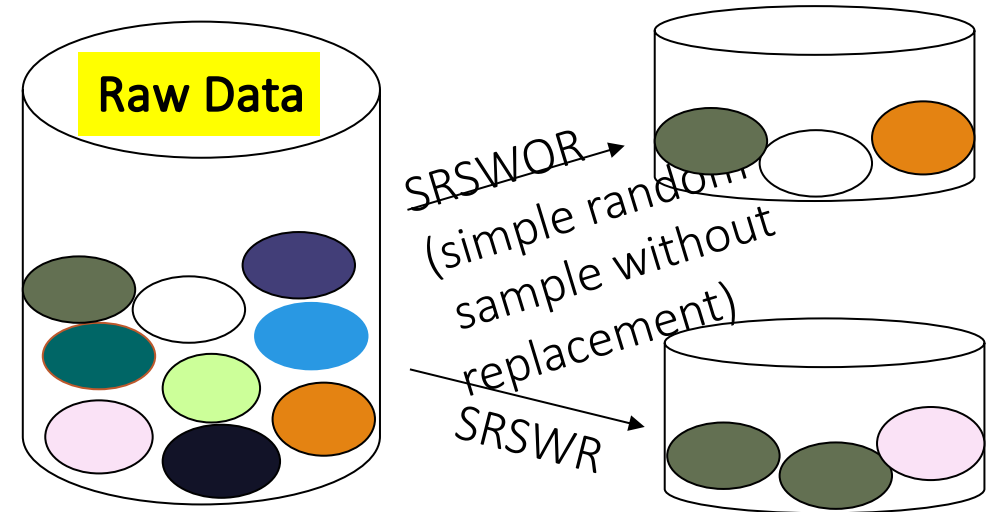
- ❑ Two choices, select one:
 - ❑ **Survey:** A focused, comprehensive survey on a focused topic of data mining
 - ❑ Ex. Clustering methods for heterogeneous information networks
 - ❑ **Research Project:** Working on a research problem related to data mining and write a paper/report, including a literature review, proposed method, datasets, and experiment results
- ❑ **Grouping:** Single person or a group of maximum 3 students
- ❑ **Submission on Compass2g**
 - ❑ One page proposal @the end of 4th week: Title, abstract, datasets, references
 - ❑ Final report: similar to a conference paper, usually 10-12 pages in double columns
- ❑ **Piazza:** Containing more detailed description, please raise questions on Piazza

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
- ❑ Key principle: Choose a **representative** subset of the data
 - ❑ Simple random sampling may have very poor performance in the presence of skew
 - ❑ Develop adaptive sampling methods, e.g., stratified sampling:
- ❑ Note: Sampling may not reduce database I/Os (page at a time)

Types of Sampling

- ❑ **Simple random sampling:** equal probability of selecting any particular item
- ❑ **Sampling without replacement**
 - ❑ Once an object is selected, it is removed from the population
- ❑ **Sampling with replacement**
 - ❑ A selected object is not removed from the population
- ❑ **Stratified sampling**
 - ❑ Partition (or cluster) the data set, and draw samples from each partition (proportionally, i.e., approximately the same percentage of the data)



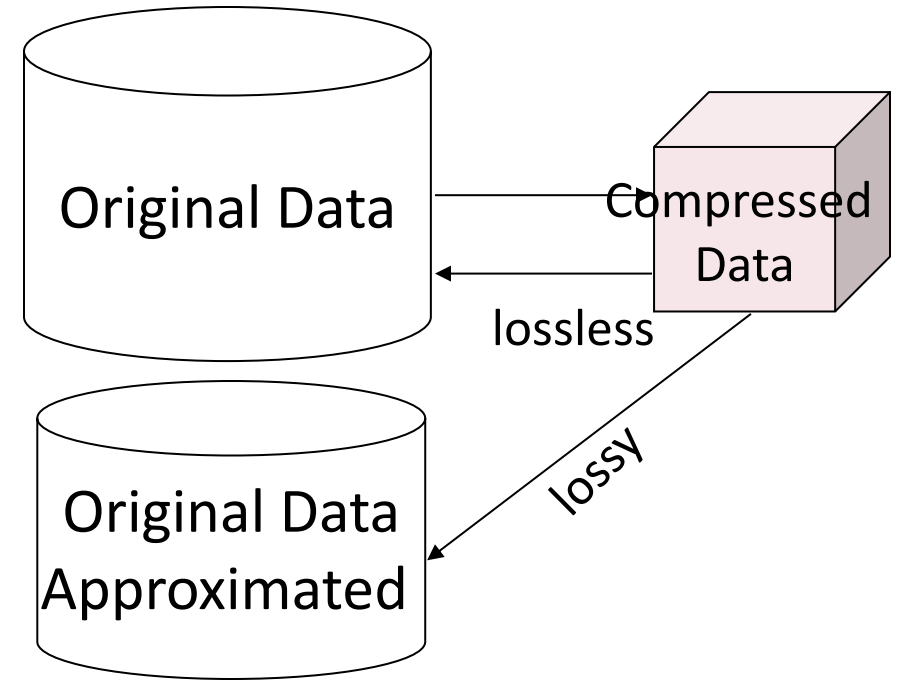
Data Cube Aggregation

- 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



Data Compression

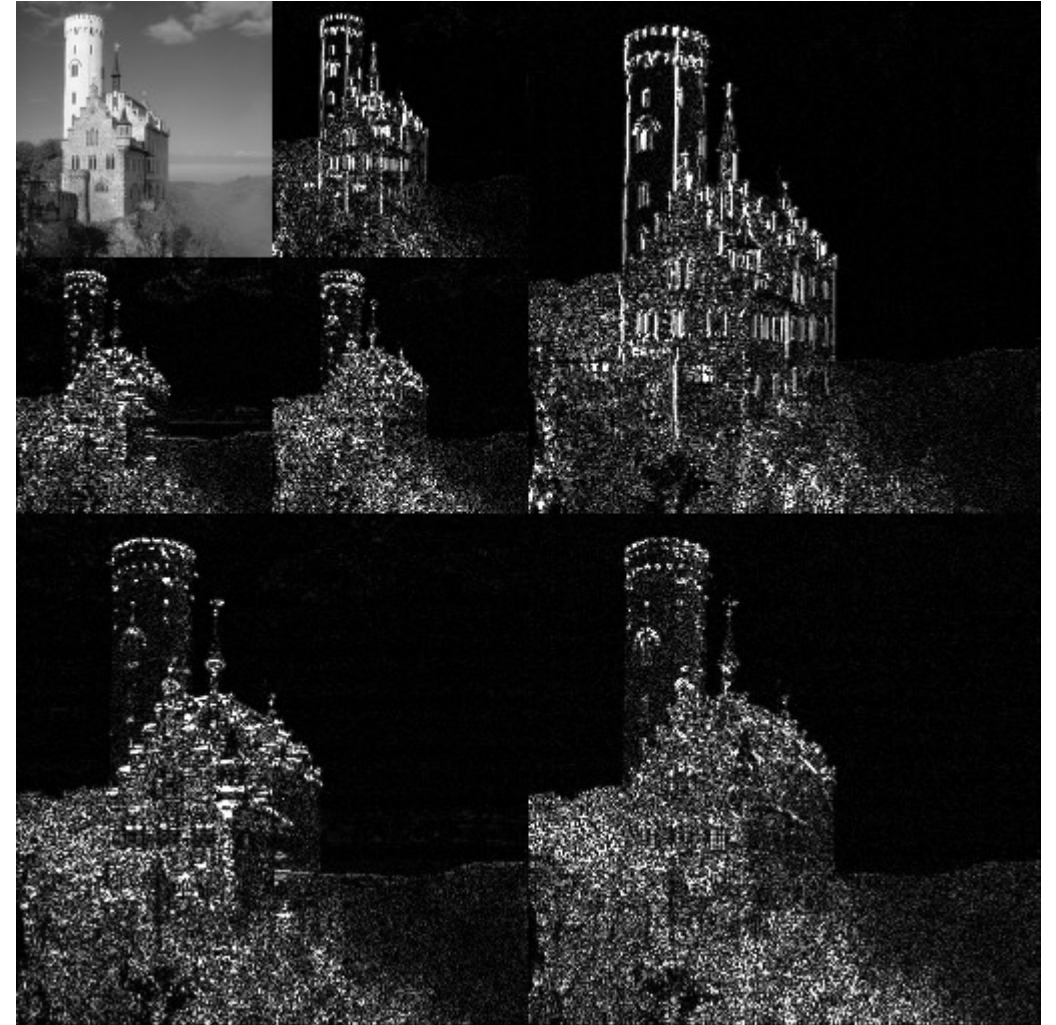
- ❑ String compression
 - ❑ There are extensive theories and well-tuned algorithms
 - ❑ Typically lossless, but only limited manipulation is possible without expansion
- ❑ Audio/video compression
 - ❑ Typically lossy compression, with progressive refinement
 - ❑ Sometimes small fragments of signal can be reconstructed without reconstructing the whole
- ❑ Time sequence is not audio
 - ❑ Typically short and vary slowly with time
- ❑ Data reduction and dimensionality reduction may also be considered as forms of data compression



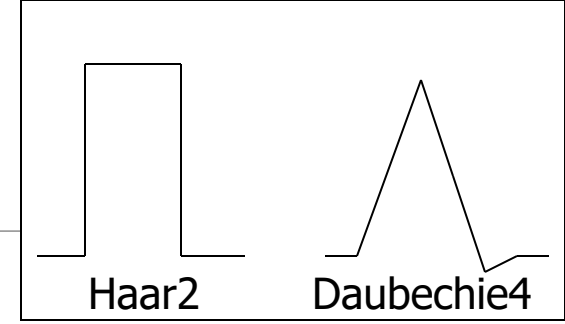
Lossy vs. lossless compression

Wavelet Transform: A Data Compression Technique

- ❑ Wavelet Transform
 - ❑ Decomposes a signal into different frequency subbands
 - ❑ Applicable to n-dimensional signals
- ❑ Data are transformed to preserve relative distance between objects at different levels of resolution
- ❑ Allow natural clusters to become more distinguishable
- ❑ Used for image compression



Wavelet Transformation



- ❑ Discrete wavelet transform (DWT) for linear signal processing, multi-resolution 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

Wavelet Decomposition

- ❑ Wavelets: A math tool for space-efficient hierarchical decomposition of functions
- ❑ $S = [2, 2, 0, 2, 3, 5, 4, 4]$ can be transformed to $S_\wedge = [2^{3/4}, -1^{1/4}, 1/2, 0, 0, -1, -1, 0]$
- ❑ Compression: many small detail coefficients can be replaced by 0's, and only the significant coefficients are retained

| Resolution | Averages | Detail Coefficients |
|------------|----------------------------|---------------------|
| 8 | $[2, 2, 0, 2, 3, 5, 4, 4]$ | |
| 4 | $[2, 1, 4, 4]$ | $[0, -1, -1, 0]$ |
| 2 | $[1\frac{1}{2}, 4]$ | $[\frac{1}{2}, 0]$ |
| 1 | $[2\frac{3}{4}]$ | $[-1\frac{1}{4}]$ |

Why Wavelet Transform?

- ❑ Use hat-shape filters
 - ❑ Emphasize region where points cluster
 - ❑ Suppress weaker information in their boundaries
- ❑ Effective removal of outliers
 - ❑ Insensitive to noise, insensitive to input order
- ❑ Multi-resolution
 - ❑ Detect arbitrary shaped clusters at different scales
- ❑ Efficient
 - ❑ Complexity $O(N)$
- ❑ Only applicable to low dimensional data

Data Transformation

- ❑ A function that maps the entire set of values of a given attribute to a new set of replacement values s.t. each old value can be identified with one of the new values
- ❑ Methods
 - ❑ Smoothing: Remove noise from data
 - ❑ Attribute/feature construction
 - ❑ New attributes constructed from the given ones
 - ❑ Aggregation: Summarization, data cube construction
 - ❑ Normalization: Scaled to fall within a smaller, specified range
 - ❑ min-max normalization
 - ❑ z-score normalization
 - ❑ normalization by decimal scaling
 - ❑ Discretization: Concept hierarchy climbing

Normalization

- **Min-max normalization:** to $[new_min_A, new_max_A]$

$$v' = \frac{v - min_A}{max_A - min_A} (new_max_A - new_min_A) + new_min_A$$

- Ex. Let income range \$12,000 to \$98,000 normalized to $[0.0, 1.0]$

□ Then \$73,000 is mapped to $\frac{73,600 - 12,000}{98,000 - 12,000} (1.0 - 0) + 0 = 0.716$

- **Z-score normalization** (μ : mean, σ : standard deviation):

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

Z-score: The distance between the raw score and the population mean in the unit of the standard deviation

- Ex. Let $\mu = 54,000$, $\sigma = 16,000$. Then $\frac{73,600 - 54,000}{16,000} = 1.225$

- **Normalization by decimal scaling**

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

Discretization

- ❑ Three types of attributes
 - ❑ Nominal—values from an unordered set, e.g., color, profession
 - ❑ Ordinal—values from an ordered set, e.g., military or academic rank
 - ❑ Numeric—real numbers, e.g., integer or real numbers
- ❑ Discretization: Divide the range of a continuous attribute into intervals
 - ❑ Interval labels can then be used to replace actual data values
 - ❑ Reduce data size by discretization
 - ❑ Supervised vs. unsupervised
 - ❑ Split (top-down) vs. merge (bottom-up)
 - ❑ Discretization can be performed recursively on an attribute
 - ❑ Prepare for further analysis, e.g., classification

Data Discretization Methods

- ❑ 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
- ❑ Note: All the methods can be applied recursively

Simple Discretization: Binning

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

Example: 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 equal-frequency (**equi-depth**) bins:

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

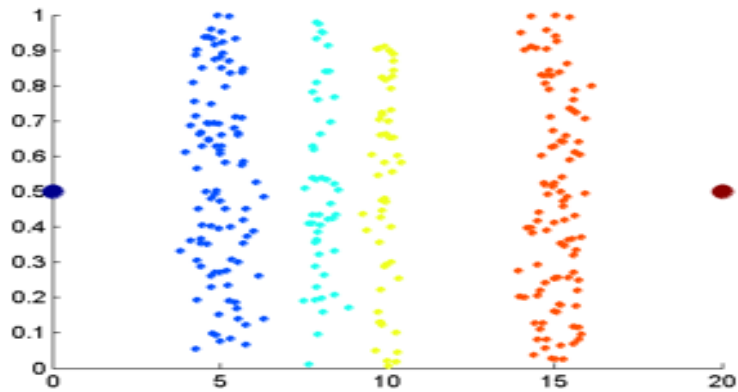
* Smoothing by **bin means**:

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

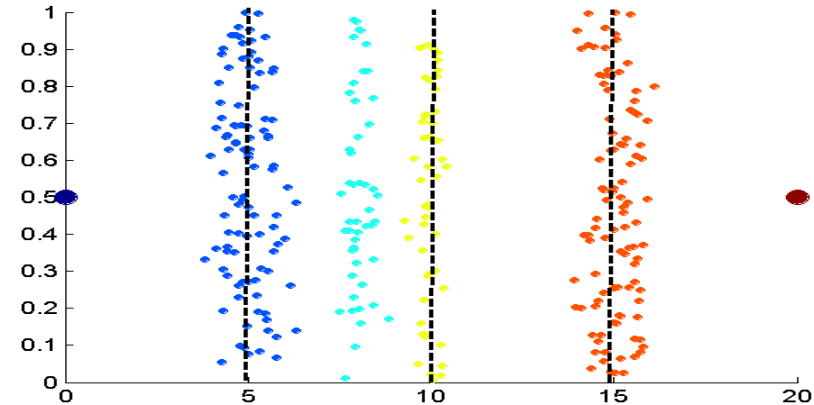
* Smoothing by **bin boundaries**:

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

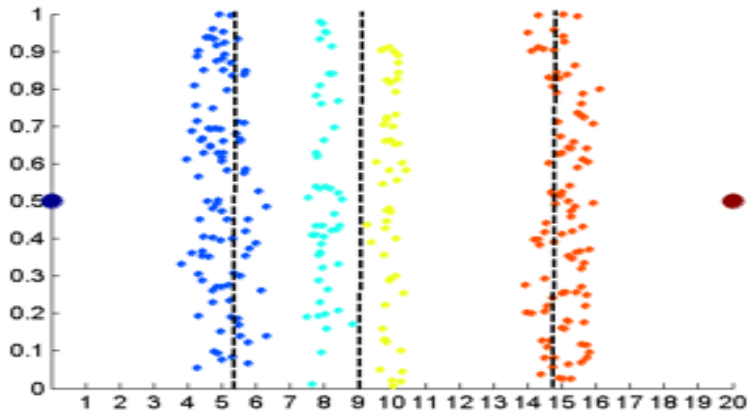
Discretization Without Supervision: Binning vs. Clustering



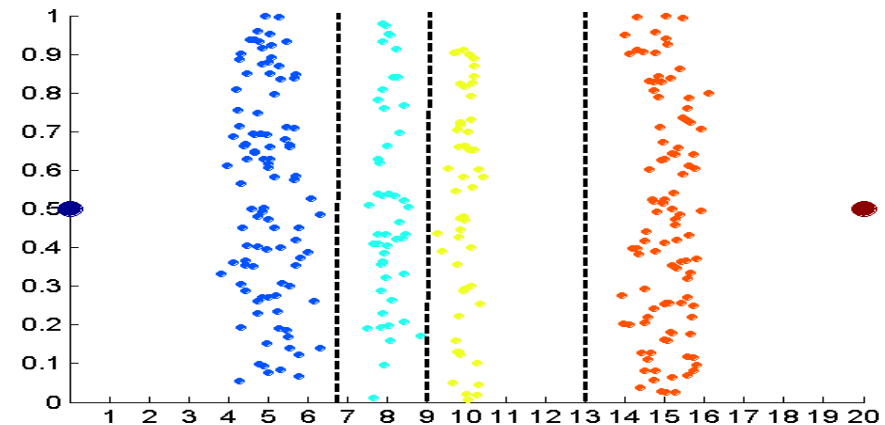
Data



Equal width (distance) binning



Equal depth (frequency) (binning)



K-means clustering leads to better results

Discretization by Classification & Correlation Analysis

- ❑ Classification (e.g., decision tree analysis)
 - ❑ Supervised: Given class labels, e.g., cancerous vs. benign
 - ❑ Using *entropy* to determine split point (discretization point)
 - ❑ Top-down, recursive split
 - ❑ Details to be covered in Chapter “Classification”
- ❑ Correlation analysis (e.g., Chi-merge: χ^2 -based discretization)
 - ❑ Supervised: use class information
 - ❑ Bottom-up merge: Find the best neighboring intervals (those having similar distributions of classes, i.e., low χ^2 values) to merge
 - ❑ Merge performed recursively, until a predefined stopping condition

Concept Hierarchy Generation

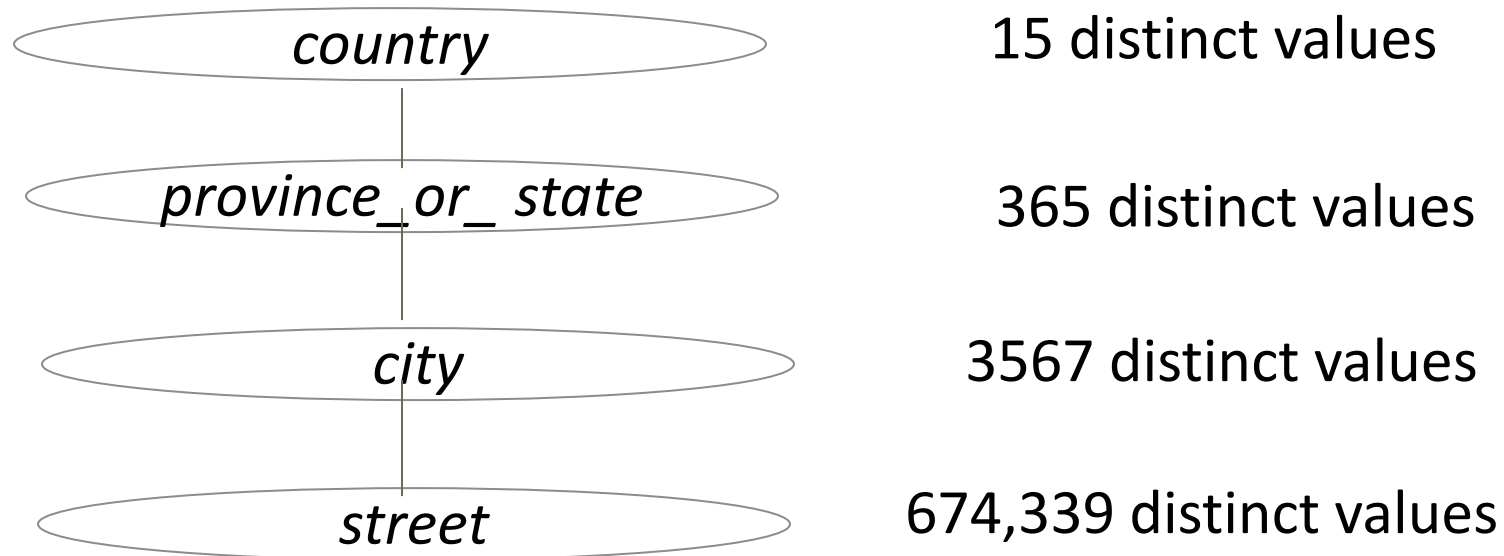
- ❑ **Concept hierarchy** organizes concepts (i.e., attribute values) hierarchically and is usually associated with each dimension in a data warehouse
- ❑ Concept hierarchies facilitate drilling and rolling in data warehouses to view data in multiple granularity
- ❑ Concept hierarchy formation: Recursively 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 can be automatically formed for both numeric and nominal data—For numeric data, use discretization methods shown

Concept Hierarchy Generation for Nominal Data


- ❑ Specification of a partial/total ordering of attributes explicitly at the schema level by users or experts
 - ❑ *street < city < state < country*
- ❑ Specification of a hierarchy for a set of values by explicit data grouping
 - ❑ {Urbana, Champaign, Chicago} < Illinois
- ❑ Specification of only a partial set of attributes
 - ❑ E.g., only *street < city*, not others
- ❑ Automatic generation of hierarchies (or attribute levels) by the analysis of the number of distinct values
 - ❑ E.g., for a set of attributes: {*street, city, state, country*}

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
 - ❑ Exceptions, e.g., weekday, month, quarter, year



Chapter 3: Data Preprocessing

- ❑ Data Preprocessing: An Overview
- ❑ Data Cleaning
- ❑ Data Integration
- ❑ Data Reduction and Transformation
- ❑ Dimensionality Reduction 
- ❑ Summary

Dimensionality Reduction

❑ Curse of dimensionality

- ❑ When dimensionality increases, data becomes increasingly sparse
- ❑ Density and distance between points, which is critical to clustering, outlier analysis, becomes less meaningful
- ❑ The possible combinations of subspaces will grow exponentially

❑ Dimensionality reduction

- ❑ Reducing the number of random variables under consideration, via obtaining a set of principal variables

❑ Advantages of dimensionality reduction

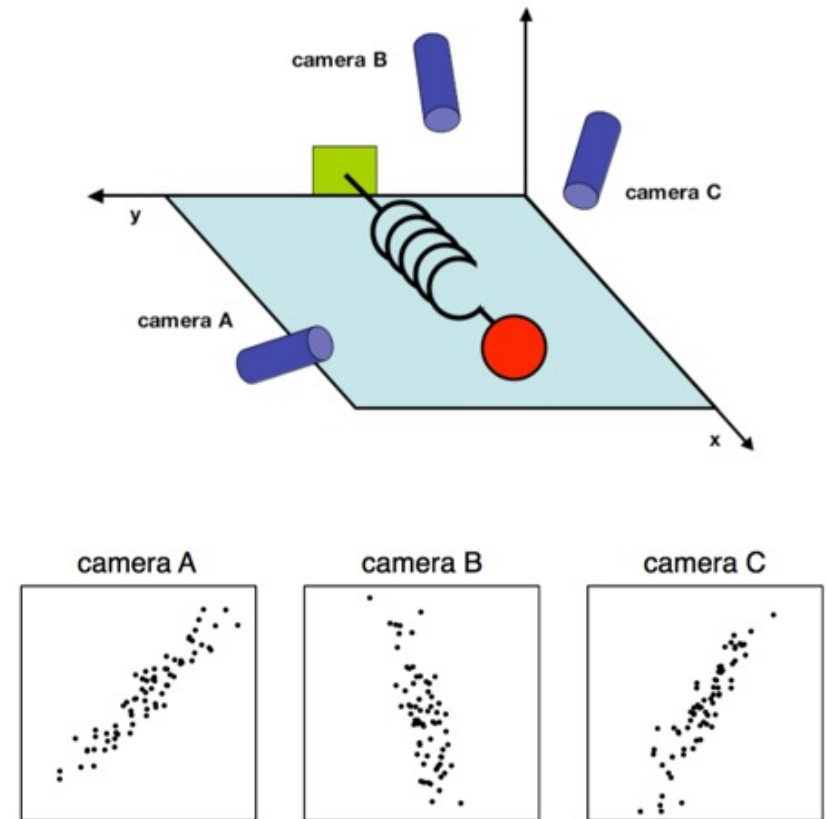
- ❑ Avoid the curse of dimensionality
- ❑ Help eliminate irrelevant features and reduce noise
- ❑ Reduce time and space required in data mining
- ❑ Allow easier visualization

Dimensionality Reduction Techniques

- ❑ Dimensionality reduction methodologies
 - ❑ **Feature selection:** Find a subset of the original variables (or features, attributes)
 - ❑ **Feature extraction:** Transform the data in the high-dimensional space to a space of fewer dimensions
- ❑ Some typical dimensionality methods
 - ❑ Principal Component Analysis
 - ❑ Supervised and nonlinear techniques
 - ❑ Feature subset selection
 - ❑ Feature creation

Principal Component Analysis (PCA)

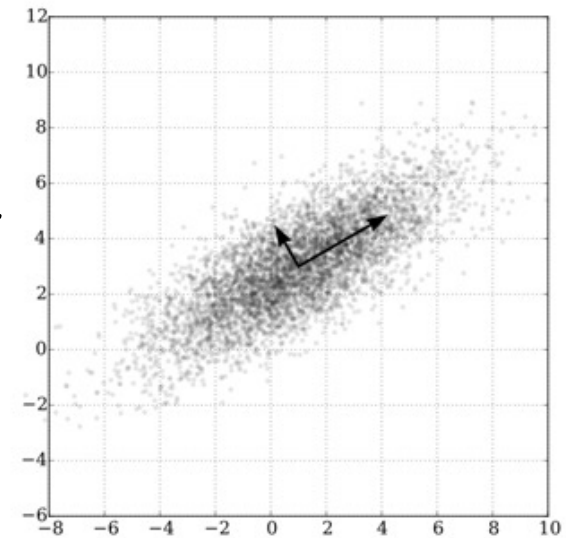
- ❑ PCA: A statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called *principal components*
- ❑ The original data are projected onto a much smaller space, resulting in dimensionality reduction
- ❑ Method: Find the eigenvectors of the covariance matrix, and these eigenvectors define the new space



Ball travels in a straight line. Data from three cameras contain much redundancy

Principal Component Analysis (Method)

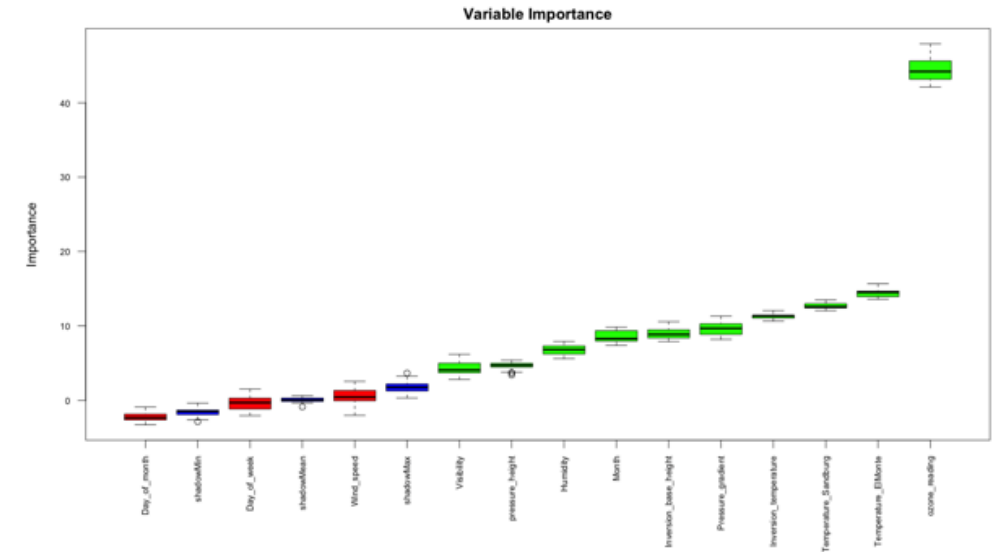
- Given N data vectors from n -dimensions, find $k \leq n$ orthogonal vectors (*principal components*) best used to represent data
 - Normalize input data: Each attribute falls within the same range
 - Compute k orthonormal (unit) vectors, i.e., *principal components*
 - Each input data (vector) is a 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, to reconstruct a good approximation of the original data)
- Works for numeric data only



Ack. Wikipedia: Principal Component Analysis

Attribute Subset Selection

- Another way to reduce dimensionality of data
- Redundant attributes
 - Duplicate much or all of the information contained in one or more other attributes
 - E.g., purchase price of a product and the amount of sales tax paid
- Irrelevant attributes
 - Contain no information that is useful for the data mining task at hand
 - Ex. A student's ID is often irrelevant to the task of predicting his/her GPA



Heuristic Search in Attribute Selection

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

Attribute Creation (Feature Generation)

- ❑ Create new attributes (features) that can capture the important information in a data set more effectively than the original ones
- ❑ Three general methodologies
 - ❑ Attribute extraction
 - ❑ Domain-specific
 - ❑ Mapping data to new space (see: data reduction)
 - ❑ E.g., Fourier transformation, wavelet transformation, manifold approaches (not covered)
 - ❑ Attribute construction
 - ❑ Combining features (see: discriminative frequent patterns in Chapter on “Advanced Classification”)
 - ❑ Data discretization

Summary

- ❑ **Data quality:** accuracy, completeness, consistency, timeliness, believability, interpretability
- ❑ **Data cleaning:** e.g. missing/noisy values, outliers
- ❑ **Data integration** from multiple sources:
 - ❑ Entity identification problem; Remove redundancies; Detect inconsistencies
- ❑ **Data reduction, data transformation and data discretization**
 - ❑ Numerosity reduction; Data compression
 - ❑ Normalization; Concept hierarchy generation
- ❑ **Dimensionality reduction**

References

- ❑ D. P. Ballou and G. K. Tayi. Enhancing data quality in data warehouse environments. *Comm. of ACM*, 42:73-78, 1999
- ❑ T. Dasu and T. Johnson. *Exploratory Data Mining and Data Cleaning*. John Wiley, 2003
- ❑ T. Dasu, T. Johnson, S. Muthukrishnan, V. Shkapenyuk. [Mining Database Structure; Or, How to Build a Data Quality Browser](#). SIGMOD'02
- ❑ H. V. Jagadish et al., Special Issue on Data Reduction Techniques. *Bulletin of the Technical Committee on Data Engineering*, 20(4), Dec. 1997
- ❑ D. Pyle. *Data Preparation for Data Mining*. Morgan Kaufmann, 1999
- ❑ E. Rahm and H. H. Do. Data Cleaning: Problems and Current Approaches. *IEEE Bulletin of the Technical Committee on Data Engineering*. Vol.23, No.4
- ❑ V. Raman and J. Hellerstein. *Potters Wheel: An Interactive Framework for Data Cleaning and Transformation*, VLDB'2001
- ❑ T. Redman. *Data Quality: Management and Technology*. Bantam Books, 1992
- ❑ 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

