DATA PRE-PROCESSING

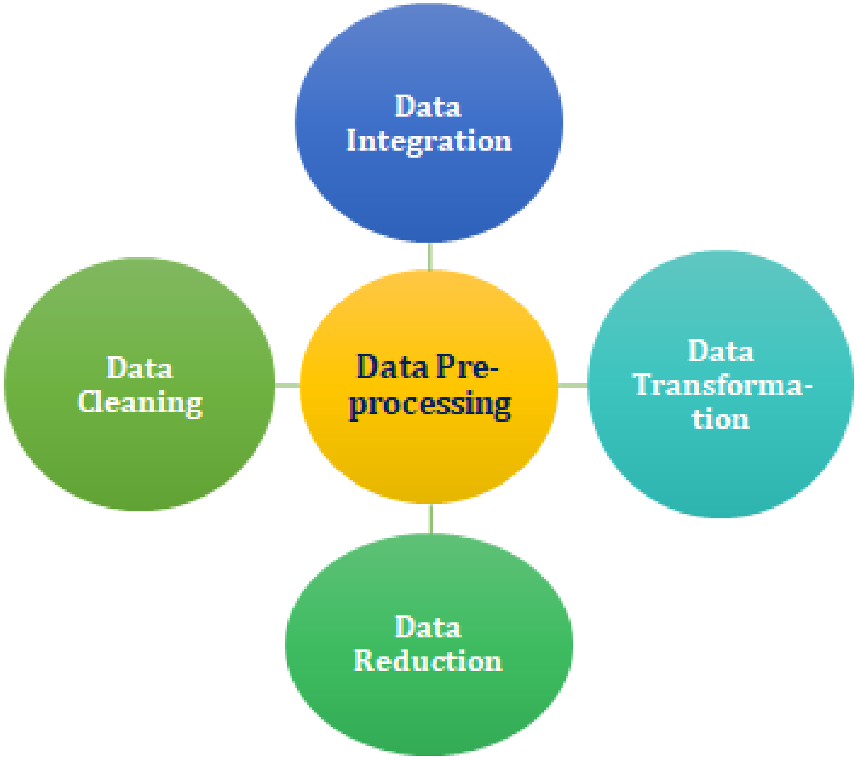
Courtesy:
Jiawei Han, Micheline Kamber, and Jian Pei
Anjali Jivani

WHY PREPROCESS THE DATA?

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?

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 and Data Discretization**
  - Normalization
  - Concept hierarchy generation
- **Data Reduction**
  - Dimensionality reduction
  - Numerosity reduction
  - Data compression

Every task of pre-processing is interrelated and Many sub-tasks under them are common too.

DATA CLEANING

Data in the real world is dirty: lots of potentially incorrect data, e.g., Instrument faulty, human or computer error, 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)

MISSING
Little strip of paper with phone number on it.
Piraro 11-9-12 w/Andy Cowan
BIZARROCOMICS.COM
Facebook.com/BizarroComics

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 registered history or changes of the data

MISSING DATA MAY NEED TO BE TAKEN CARE OF

HANDLING 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 which require data cleaning**
  - duplicate records
  - incomplete data
  - inconsistent data

HANDLING 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
- Combined computer and human inspection
  - detect suspicious values and check by human (e.g., deal with possible outliers)

BINNING METHODS

➢ 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

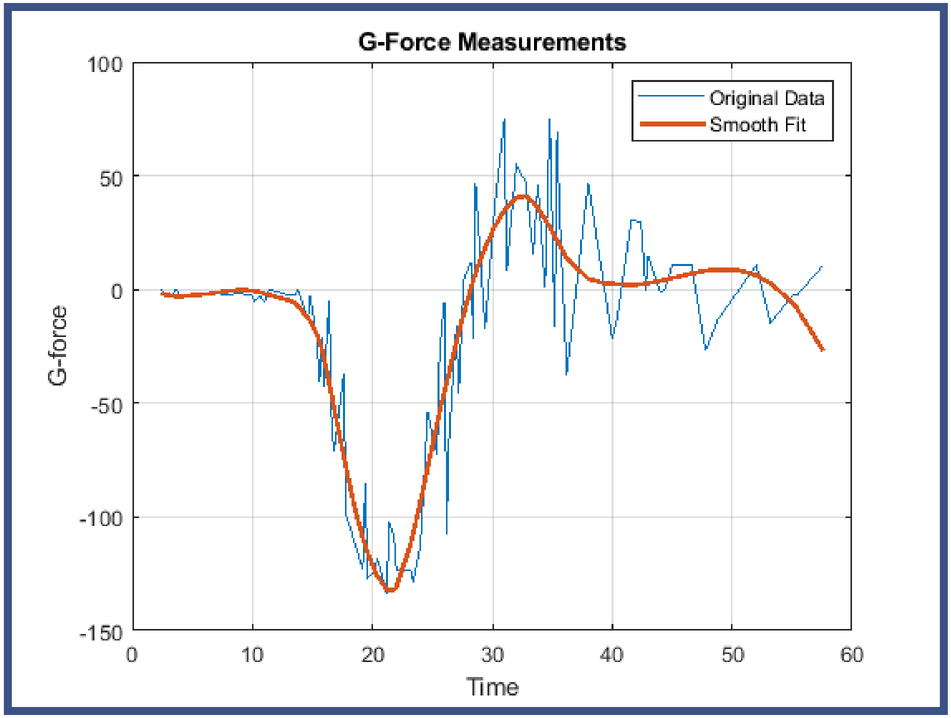
* Smoothing by bin means:

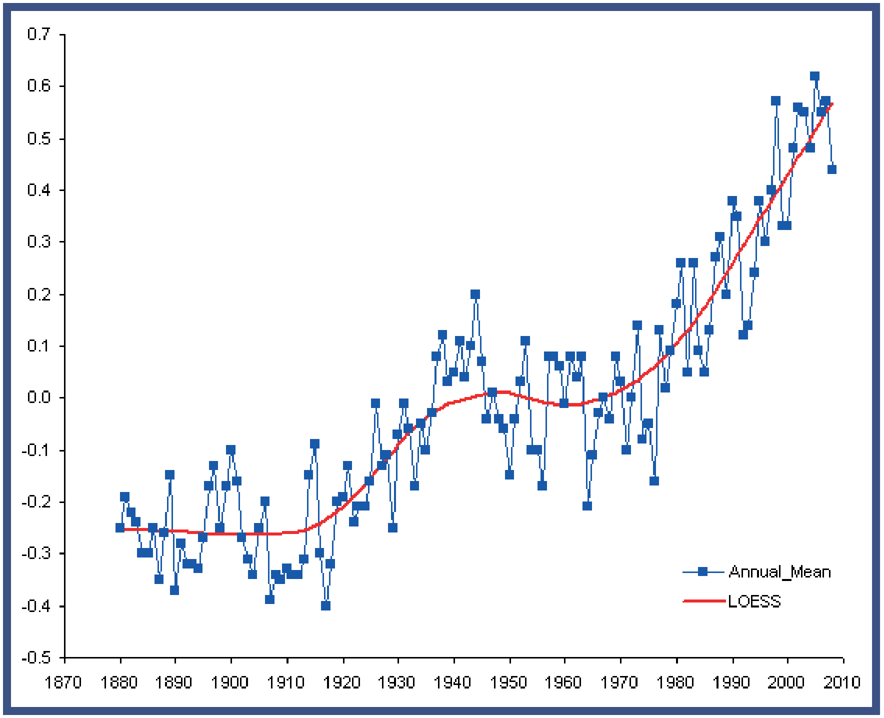
- Bin 1: 9, 9, 9, 9
- Bin 2: 23, 23, 23, 23
- Bin 3: 29, 29, 29, 29

* Smoothing by bin boundaries:

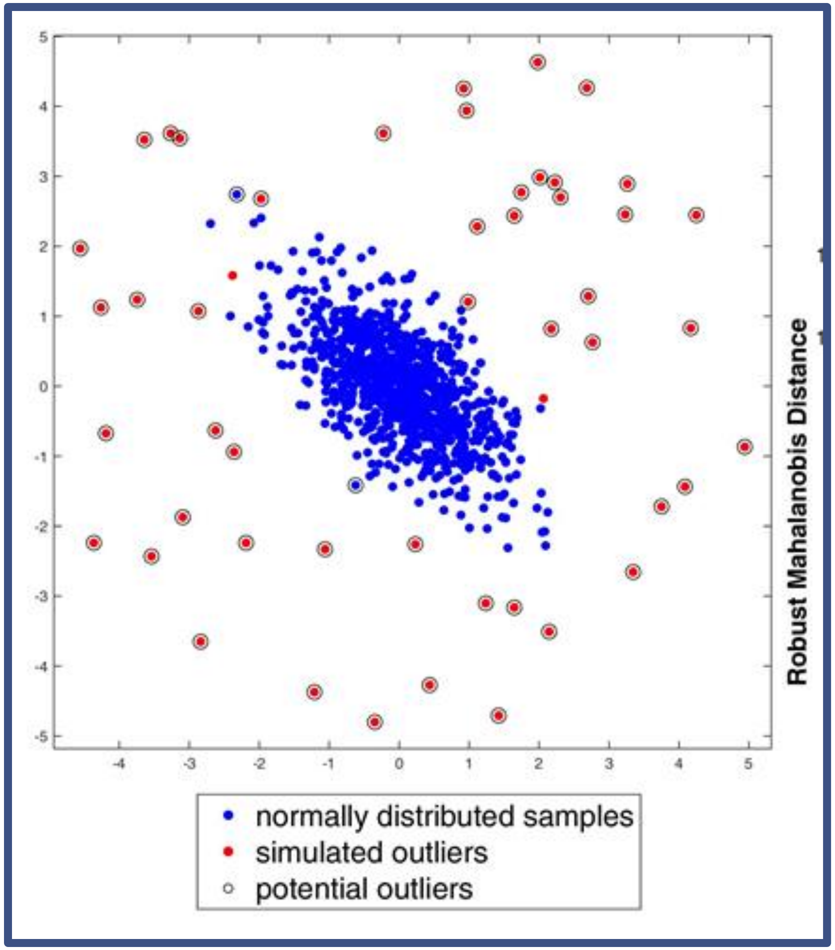
- Bin 1: 4, 4, 4, 15
- Bin 2: 21, 21, 25, 25
- Bin 3: 26, 26, 26, 34

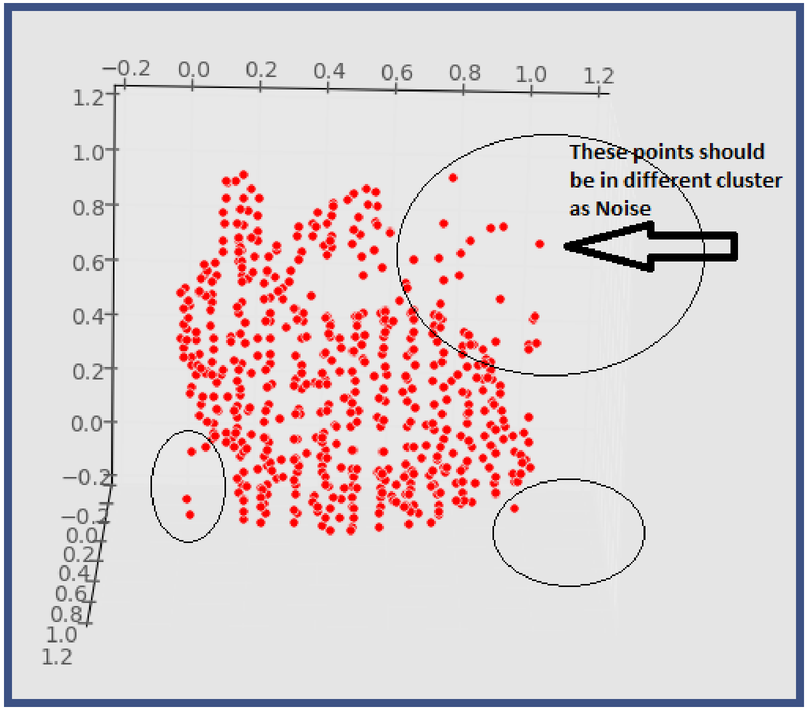
REGRESSION FOR SMOOTHING





CLUSTERING FOR NOISY DATA



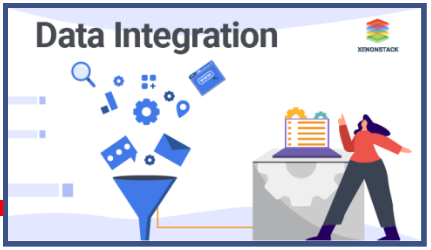


These points should be in different cluster as Noise

DATA CLEANING AS A PROCESS

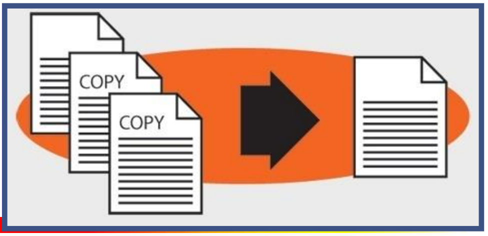
- Data discrepancy detection
  - Use metadata (e.g., domain, range, dependency, distribution)
  - Check field overloading
  - Check uniqueness rule, consecutive rule and null rule
  - Use commercial tools
    - Data scrubbing: use simple domain knowledge (e.g., postal code, spell-check) to detect errors 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)

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* and *covariance analysis*
- Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality

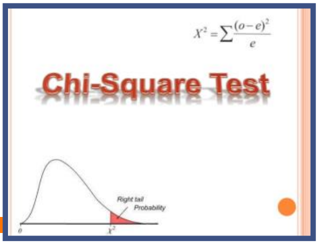
Correlation Analysis (Nominal Data)

- **X^2 (chi-square) test**

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

- The larger the X^2 value, the more likely the variables are related and not independent
- The cells that contribute the most to the X^2 value are those whose actual count is very different from the expected count
- 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



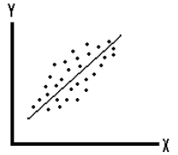
	Play chess	Not play chess	Sum (row)
Like science fiction	250(90)	200(360)	450
Not like science fiction	50(210)	1000(840)	1050
Sum(col.)	300	1200	1500

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

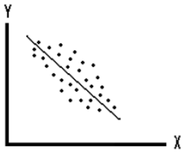
$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840} = 507.93$$

- It shows that like_science_fiction and play_chess are correlated in the group (reject the hypothesis that they are independent)

Correlation Analysis (Numeric Data)



Positive Correlation



Negative Correlation

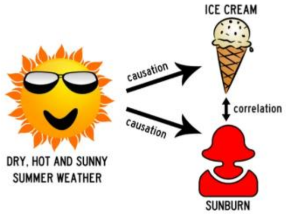
- Correlation coefficient (also called Pearson's product moment coefficient)

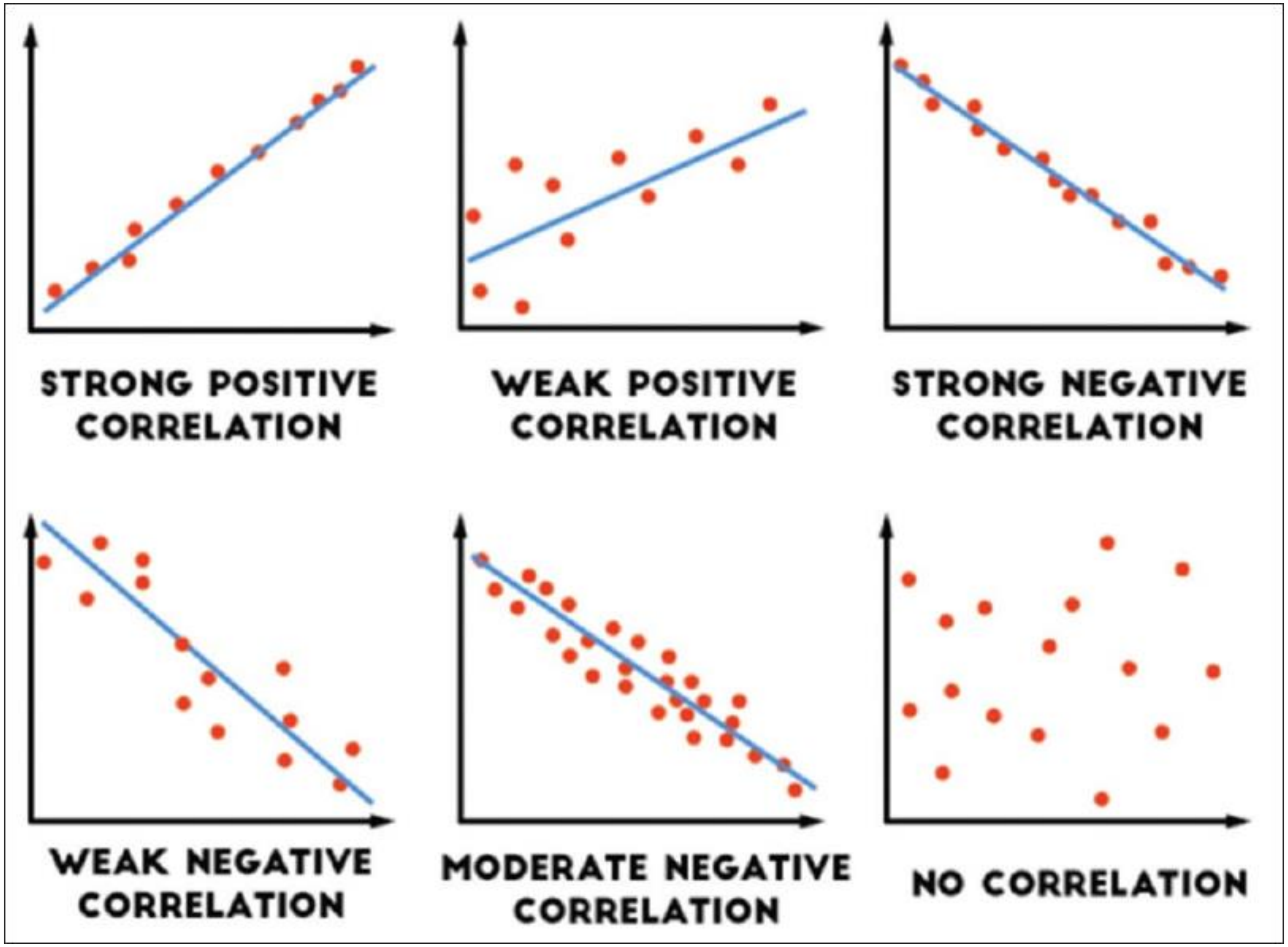
$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i - \bar{A})(b_i - \bar{B})}{(n-1)\sigma_A\sigma_B} = \frac{\sum_{i=1}^{n}(a_i b_i) - n\bar{A}\bar{B}}{(n-1)\sigma_A\sigma_B}$$

Where, n is the number of tuples, $\bar{A}$ and $\bar{B}$ are the respective means of A and B, σ_A and σ_B are the respective standard deviation of A and B, and $\Sigma(a_i b_i)$ is the sum of the AB cross-product.

- 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_{AB} < 0: negatively correlated

Visually Evaluating Correlation





Correlation

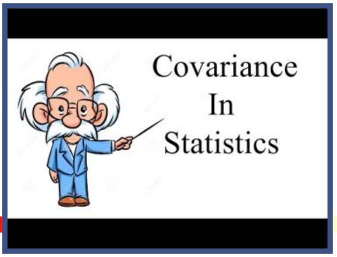
- Correlation measures the linear relationship between objects
- To compute correlation, we standardize data objects, A and B, and then take their dot product

$$a'_k = (a_k - mean(A)) / std(A)$$

$$b'_k = (b_k - mean(B)) / std(B)$$

$$correlation(A, B) = A' \bullet B'$$

Covariance (Numeric Data)



- Covariance is similar to correlation

$$Cov(A,B) = E((A-\bar{A})(B-\bar{B})) = \frac{\sum_{i=1}^{n}(a_i-\bar{A})(b_i-\bar{B})}{n}$$

  - where n is the number of tuples, $\bar{A}$ and $\bar{B}$ are the respective mean or **expected values** of A and B, σ_A and σ_B are the respective standard deviation of A and B.

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

- **Positive covariance**: If $Cov_{A,B} > 0$, then A and B both tend to be larger than their expected values.
- **Negative covariance**: If $Cov_{A,B} < 0$ then if A is larger than its expected value, B is likely to be smaller than its expected value.
- **Independence**: $Cov_{A,B} = 0$ but the converse is not true

Co-variance: An example

$$Cov(A,B) = E((A-\bar{A})(B-\bar{B})) = \frac{\sum_{i=1}^{n}(a_i-\bar{A})(b_i-\bar{B})}{n}$$

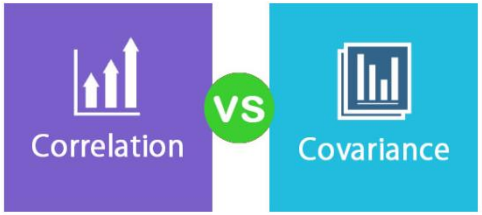
- It can be simplified in computation as

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

- 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.**

Covariance and Correlation

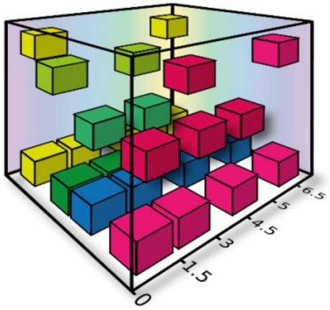


Covariance	Correlation
Covariance is a measure to indicate the extent to which two random variables change in tandem.	Correlation is a measure used to represent how strongly two random variables are related to each other.
Covariance is nothing but a measure of correlation.	Correlation refers to the scaled form of covariance.
Covariance indicates the direction of the linear relationship between variables.	Correlation on the other hand measures both the strength and direction of the linear relationship between two variables.
Covariance can vary between -∞ and +∞	Correlation ranges between -1 and +1
Covariance is affected by the change in scale. If all the values of one variable are multiplied by a constant and all the values of another variable are multiplied, by a similar or different constant, then the covariance is changed.	Correlation is not influenced by the change in scale.
Covariance assumes the units from the product of the units of the two variables.	Correlation is dimensionless, i.e. It's a unit-free measure of the relationship between variables.
Covariance of two dependent variables measures how much in real quantity (i.e. cm, kg, liters) on average they co-vary.	Correlation of two dependent variables measures the proportion of how much on average these variables vary w.r.t one another.
Covariance is zero in case of independent variables (if one variable moves and the other doesn't) because then the variables do not necessarily move together.	Independent movements do not contribute to the total correlation. Therefore, completely independent variables have a zero correlation.

DATA REDUCTION STRATEGIES

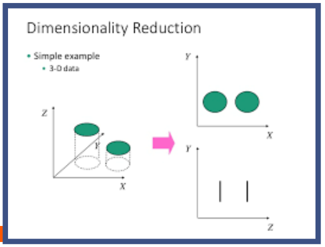
- Data reduction: Obtain a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results
- Why data reduction? — A database/data warehouse may store terabytes of data. Complex data analysis may take a very long time to run on the complete data set.
- Data reduction strategies:
  - Dimensionality reduction, e.g., remove unimportant attributes
    - Wavelet transforms
    - Principal Components Analysis (PCA)
    - Feature subset selection, feature creation
  - Numerosity reduction (some simply call it: Data Reduction)
    - Regression and Log-Linear Models
    - Histograms, clustering, sampling
    - Data cube aggregation
  - Data compression

DIMENSIONALITY REDUCTION



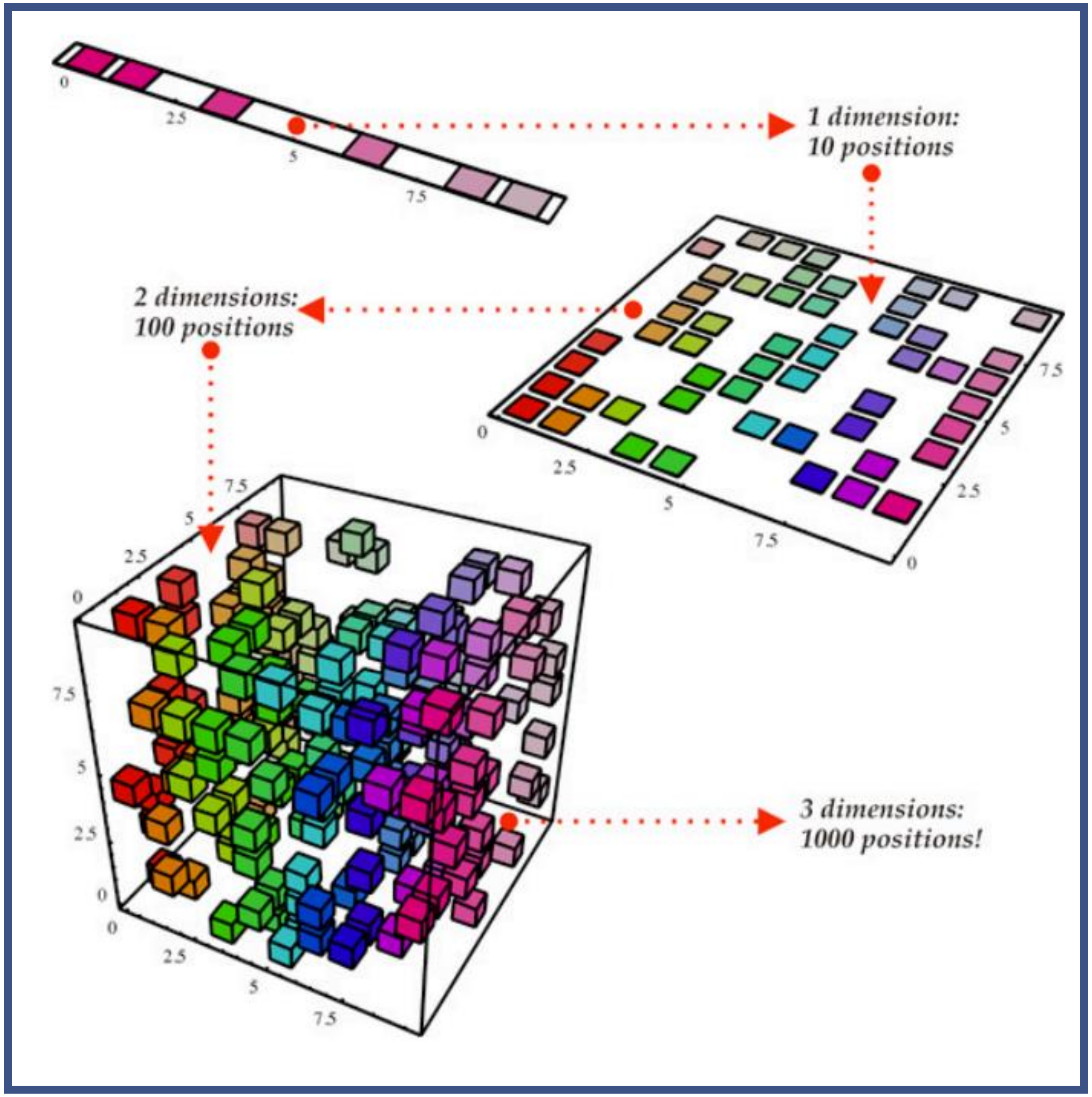
- **Curse of dimensionality**
  - When dimensionality increases, data becomes increasingly sparse
  - Density and distance between points, which is critical to clustering, outlier analysis, becomes less meaningful
  - The possible combinations of subspaces will grow exponentially
- **Dimensionality reduction**
  - 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**
  - Wavelet transforms
  - Principal Component Analysis
  - Supervised and nonlinear techniques (e.g., feature selection)

DIMENSIONALITY REDUCTION

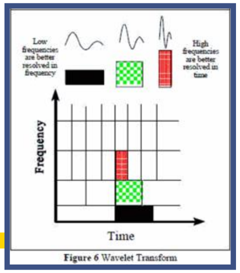


Dimensionality Reduction

- Simple example
  - 3-D data



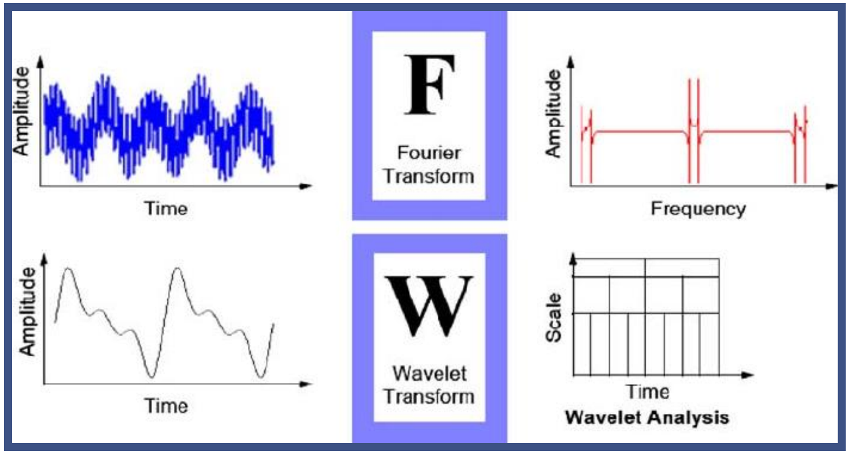
MAPPING DATA TO A NEW SPACE

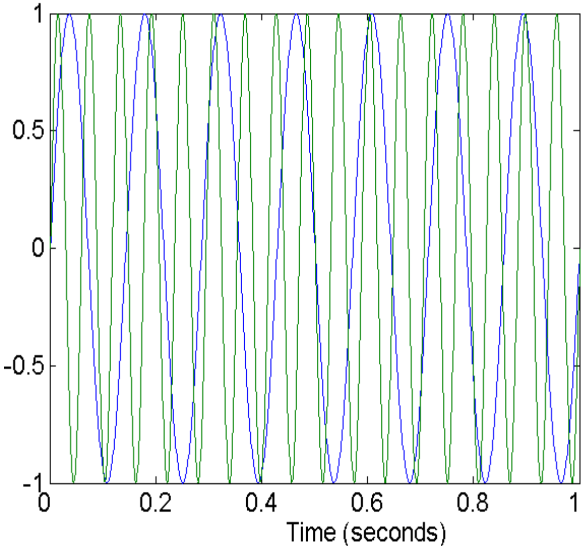


Low frequencies are better resolved in frequency
High frequencies are better resolved in time
Frequency
Time

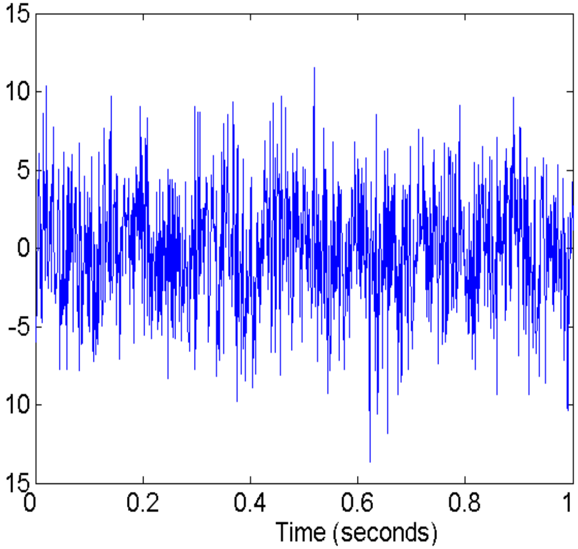
Figure 6 Wavelet Transform

- ➢ **Fourier transform**
- ➢ **Wavelet transform**

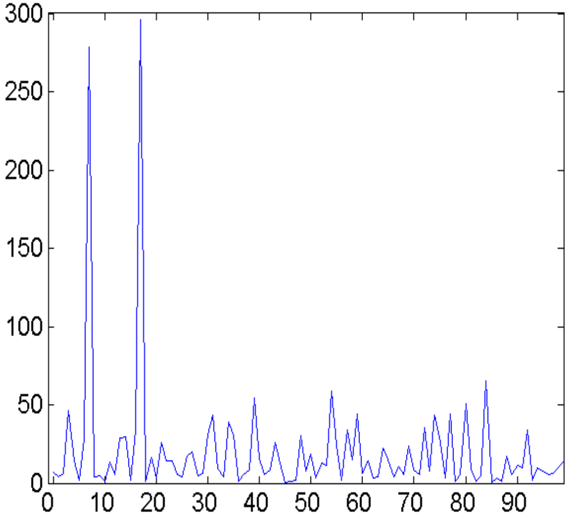




Two Sine Waves

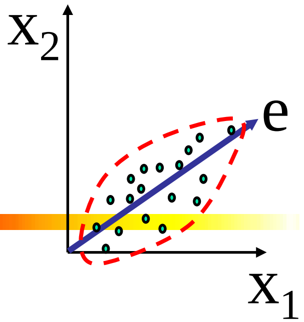


Two Sine Waves + Noise

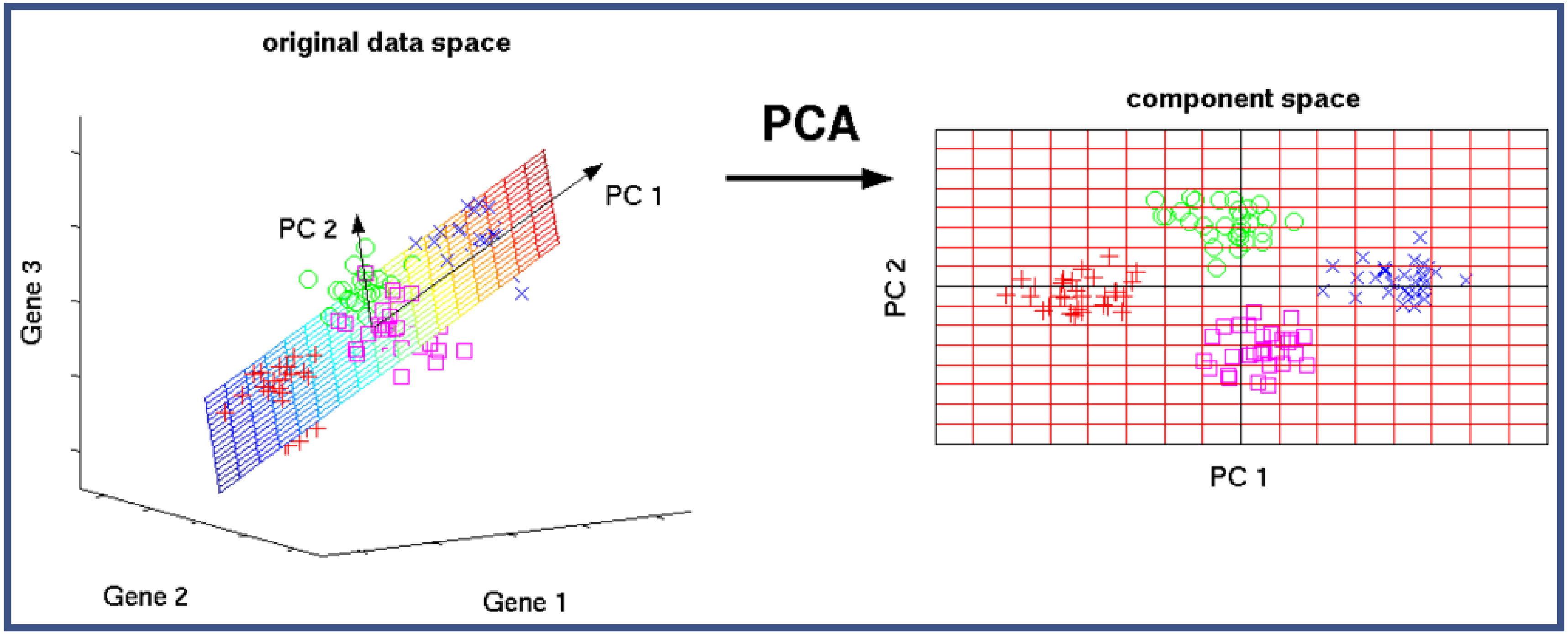


Frequency

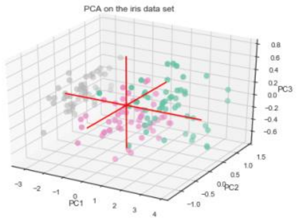
Principal Component Analysis (PCA)



- Find a projection that captures the largest amount of variation in data
- The original data are projected onto a much smaller space, resulting in dimensionality reduction. We find the eigenvectors of the covariance matrix, and these eigenvectors define the new space



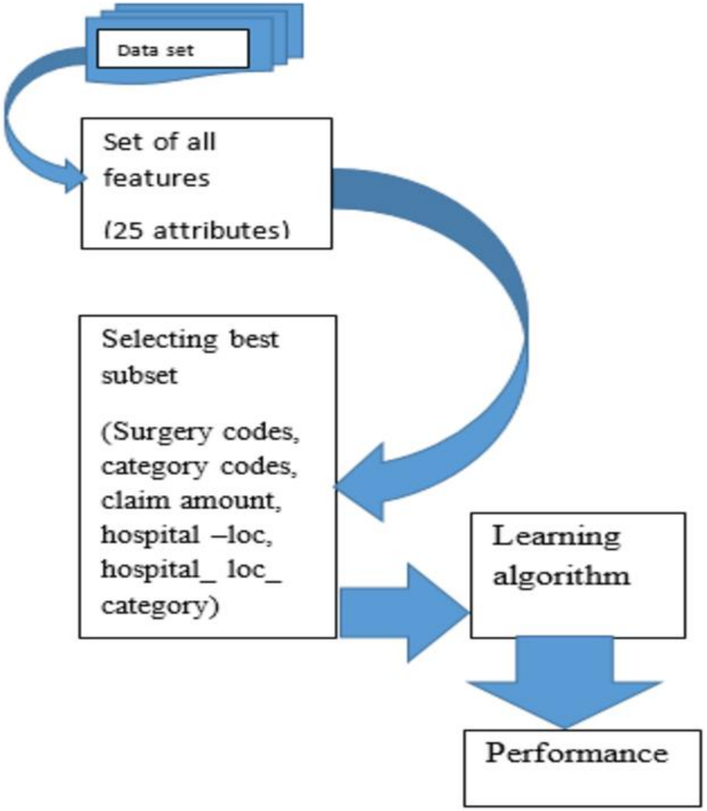
Principal Component Analysis (Steps)



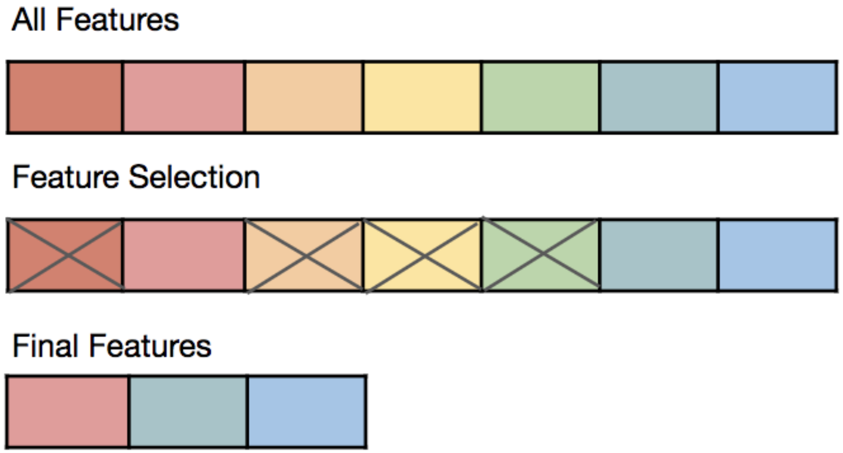
- Given *N* data vectors from *n*-dimensions, find *k* ≤ *n* orthogonal vectors (*principal components*) that can be best used to represent data
  - Normalize input data: Each attribute falls within the same range
  - Compute *k* orthonormal (unit) vectors, i.e., *principal components*
  - Each input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

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
  - E.g., students' ID is often irrelevant to the task of predicting students' GPA



Heuristic Search in Attribute Selection

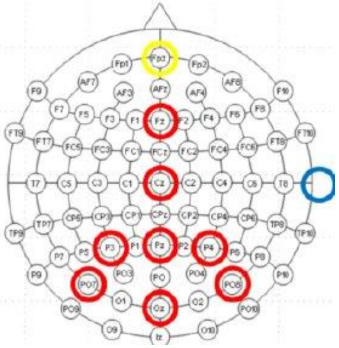


- There are 2^d possible attribute combinations of *d* attributes
- Typical heuristic attribute selection methods:
  - Best single attribute under the attribute independence assumption: choose by significance tests
  - Best step-wise feature selection:
    - The best single-attribute is picked first, then next best, …
  - 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)

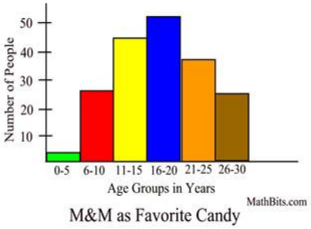
- Create new attributes (features) that can capture the important information in a data set more effectively than the original ones
- Three general methodologies
  - Attribute extraction
    - Domain-specific
  - Mapping data to new space
    - E.g., Fourier transformation, wavelet transformation, manifold approaches
  - Attribute construction
    - Combining features
    - Data discretization

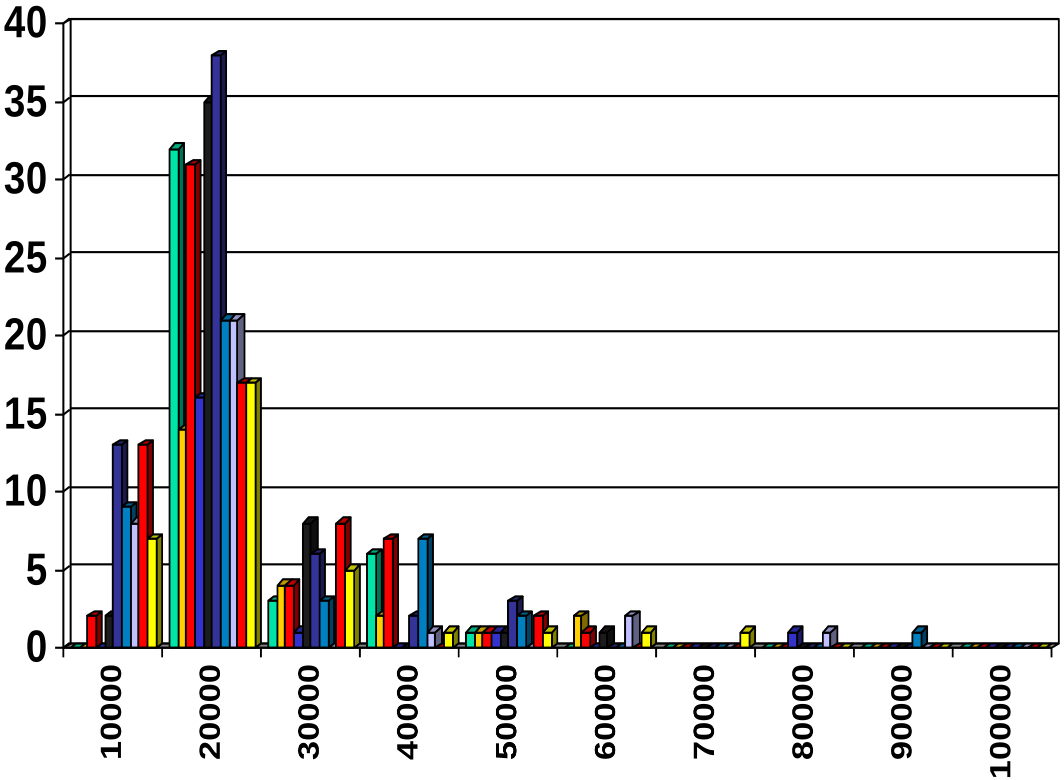
Numerosity Reduction



- 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)
- **Non-parametric** methods
  - Do not assume models
  - Major families: histograms, clustering, sampling, …

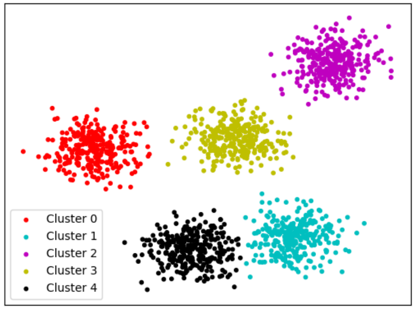
Histogram Analysis





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

Clustering

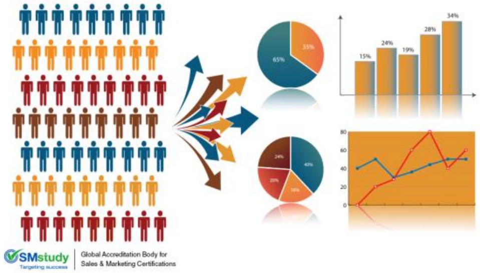


- Partition data set into clusters based on similarity, and store cluster representation (e.g., centroid and diameter) only
- Can be very effective if data is clustered but not if data is “smeared”
- Can have hierarchical clustering and be stored in multi-dimensional index tree structures
- There are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

Sampling

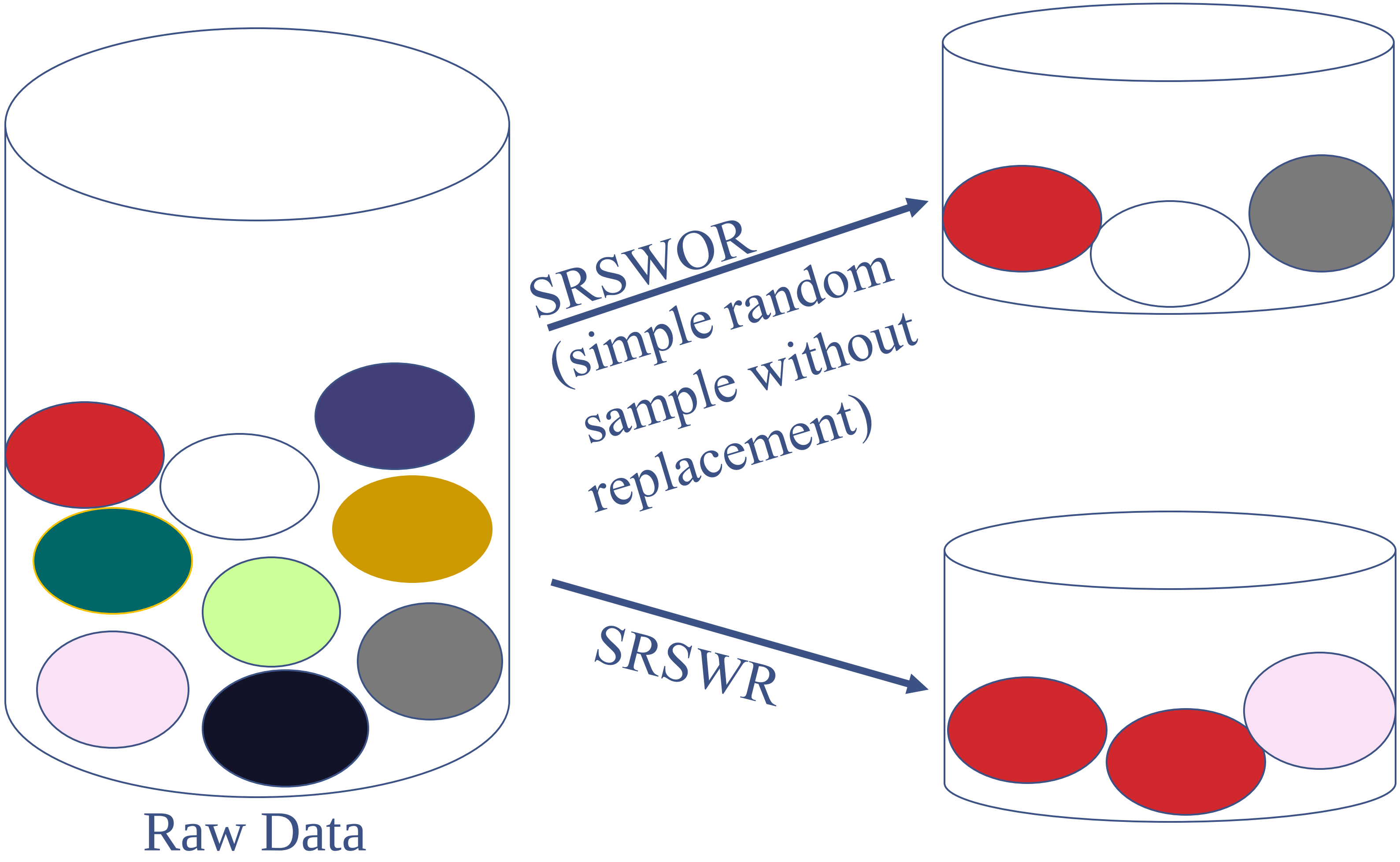
- Sampling: obtaining a small sample *s* to represent the whole data set *N*
- Allow a mining algorithm to run in complexity that is potentially sub-linear to the size of the data
- Key principle: Choose a representative subset of the data
  - Simple random sampling may have very poor performance in 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

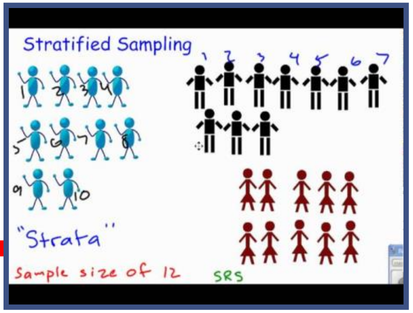


- **Simple random sampling**
  - There is an equal probability of selecting any particular item
- **Sampling without replacement**
  - Once an object is selected, it is removed from the population
- **Sampling with replacement**
  - A selected object is not removed from the population
- **Stratified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

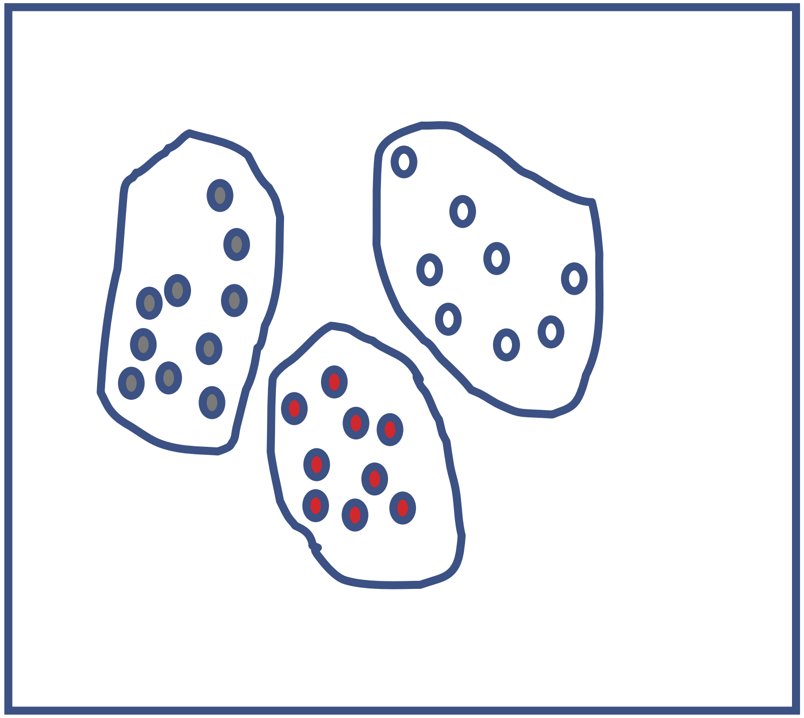
Sampling: With or without Replacement



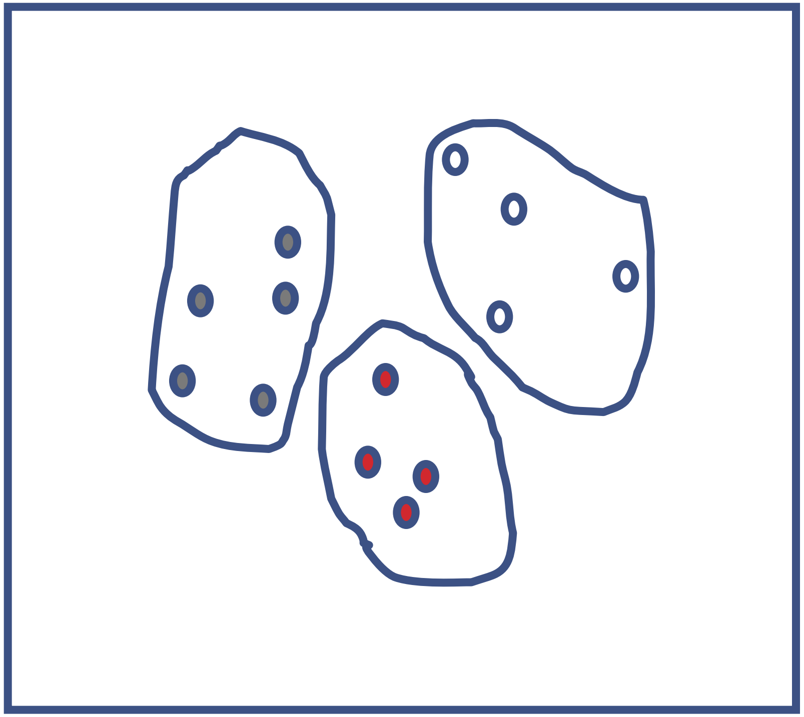
CLUSTER OR STRATIFIED SAMPLING



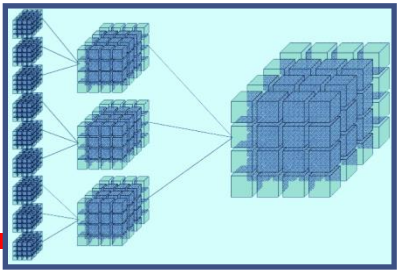
Raw Data



Cluster/Stratified Sample

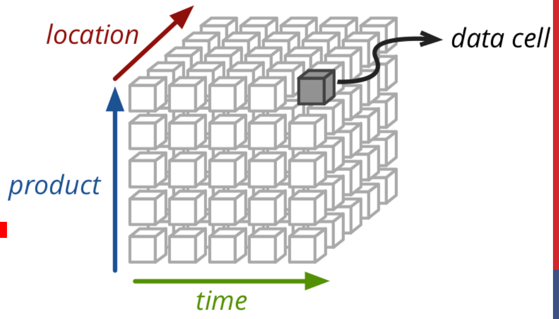


DATA CUBE AGGREGATION

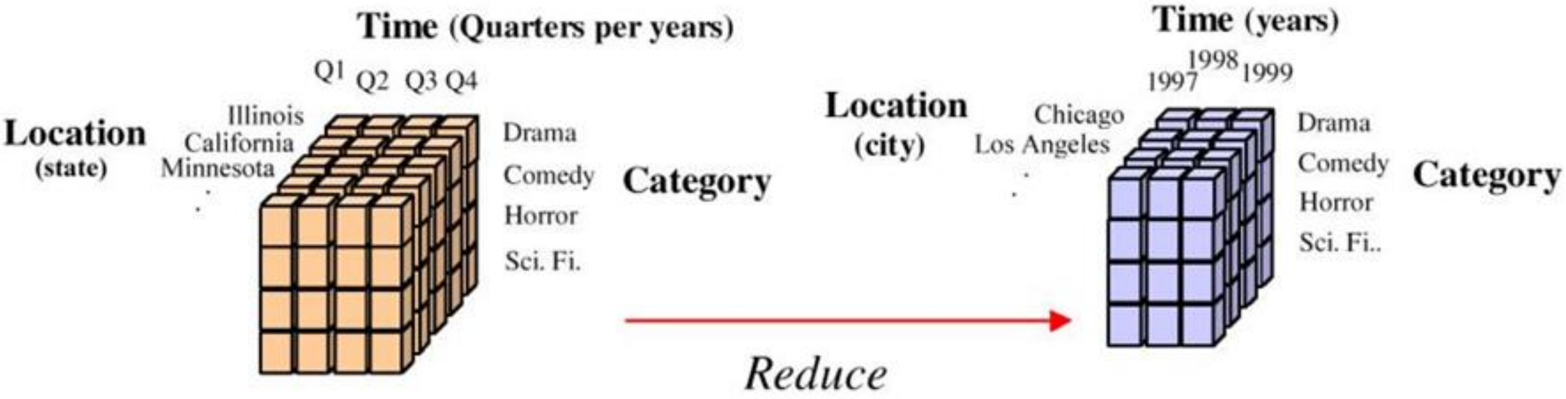


- **The lowest level of a data cube (base cuboid)**
  - The aggregated data for an individual entity of interest
  - E.g., a customer in a phone calling data warehouse
- **Multiple levels of aggregation in data cubes**
  - Further reduce the size of data to deal with
- **Reference appropriate levels**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

DATA CUBE AGGREGATION



- **Reduce the data to the concept level needed in the analysis**
  - Use the smallest (most detailed) level necessary to solve the problem



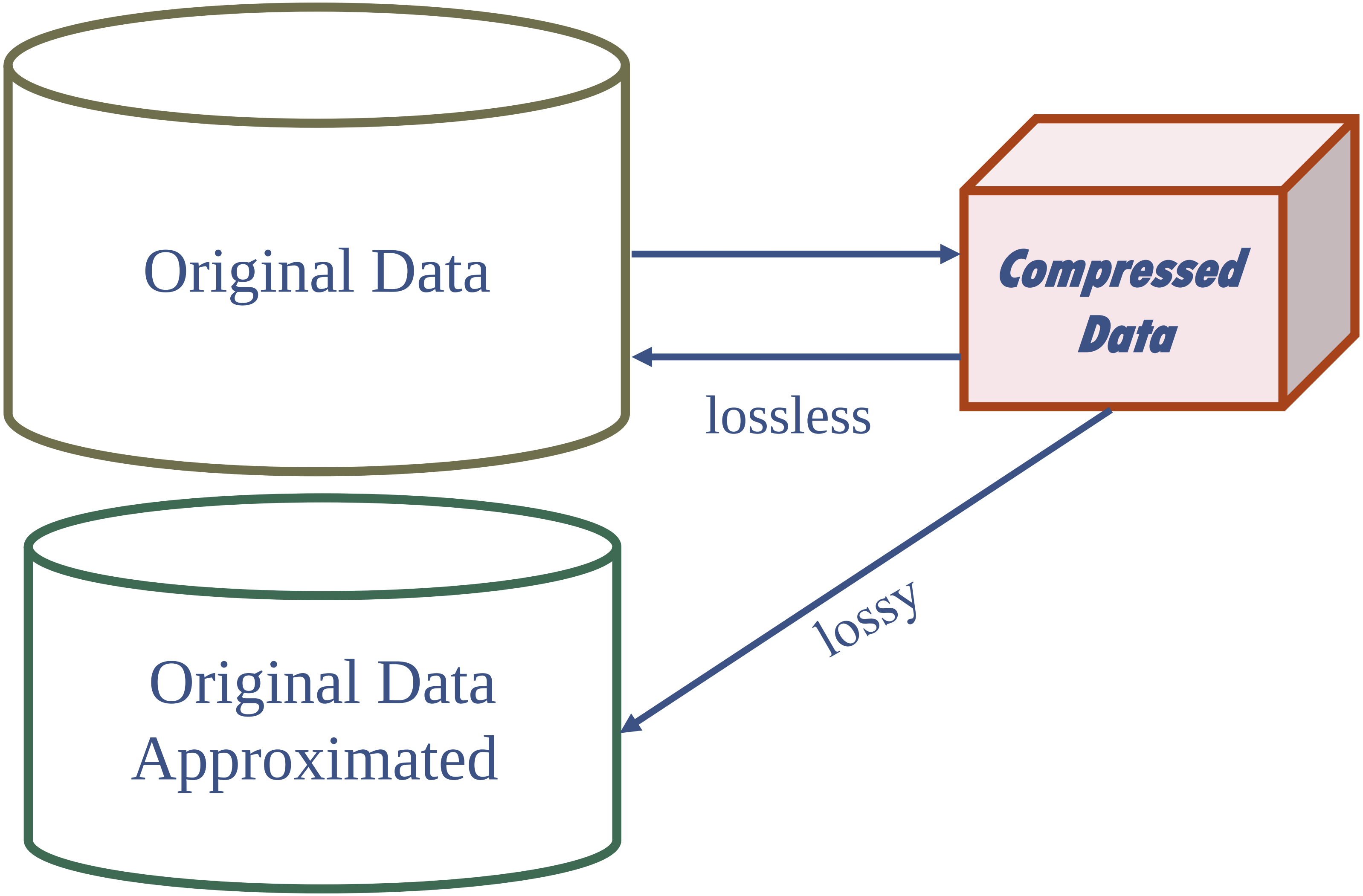
- **Queries regarding aggregated information should be answered using data cube when possible**

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

- **Dimensionality and numerosity reduction may also be considered as forms of data compression**

DATA COMPRESSION



Original Data

Compressed Data

lossless

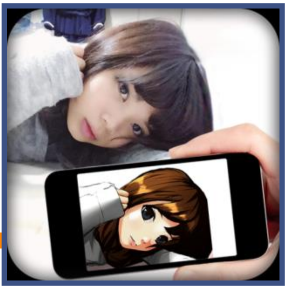
lossy

Original Data Approximated

DATA TRANSFORMATION

- **A function that maps the entire set of values of a given attribute to a new set of replacement values such that each old value can be identified with one of the new values**
- **Methods**
  - Smoothing: Remove noise from data (already covered)
  - 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 \$73,600 is mapped to:

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

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

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

Ex. Let μ = 54,000, σ = 16,000. Then

$$\boxed{\frac{73{,}600 - 54{,}000}{16{,}000} = 1.225}$$

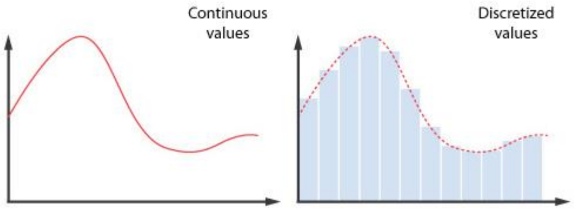
- **Normalization by decimal scaling:**

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

Ex. 989 becomes 989/1000 = 0.989 and 76 becomes 76/1000 = .076

(Assuming maximum value is 989)

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

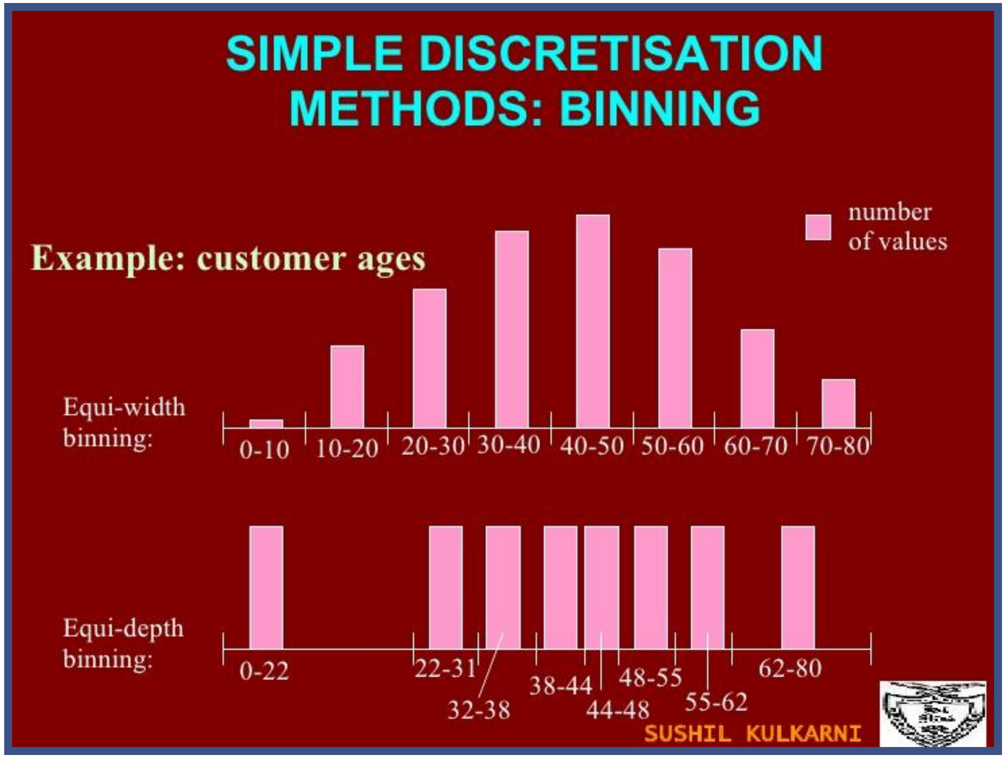
Data Discretization

Hadi Mabaki

Faculty of computer science , Iran university of science & technology

- **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)

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

BINNING 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

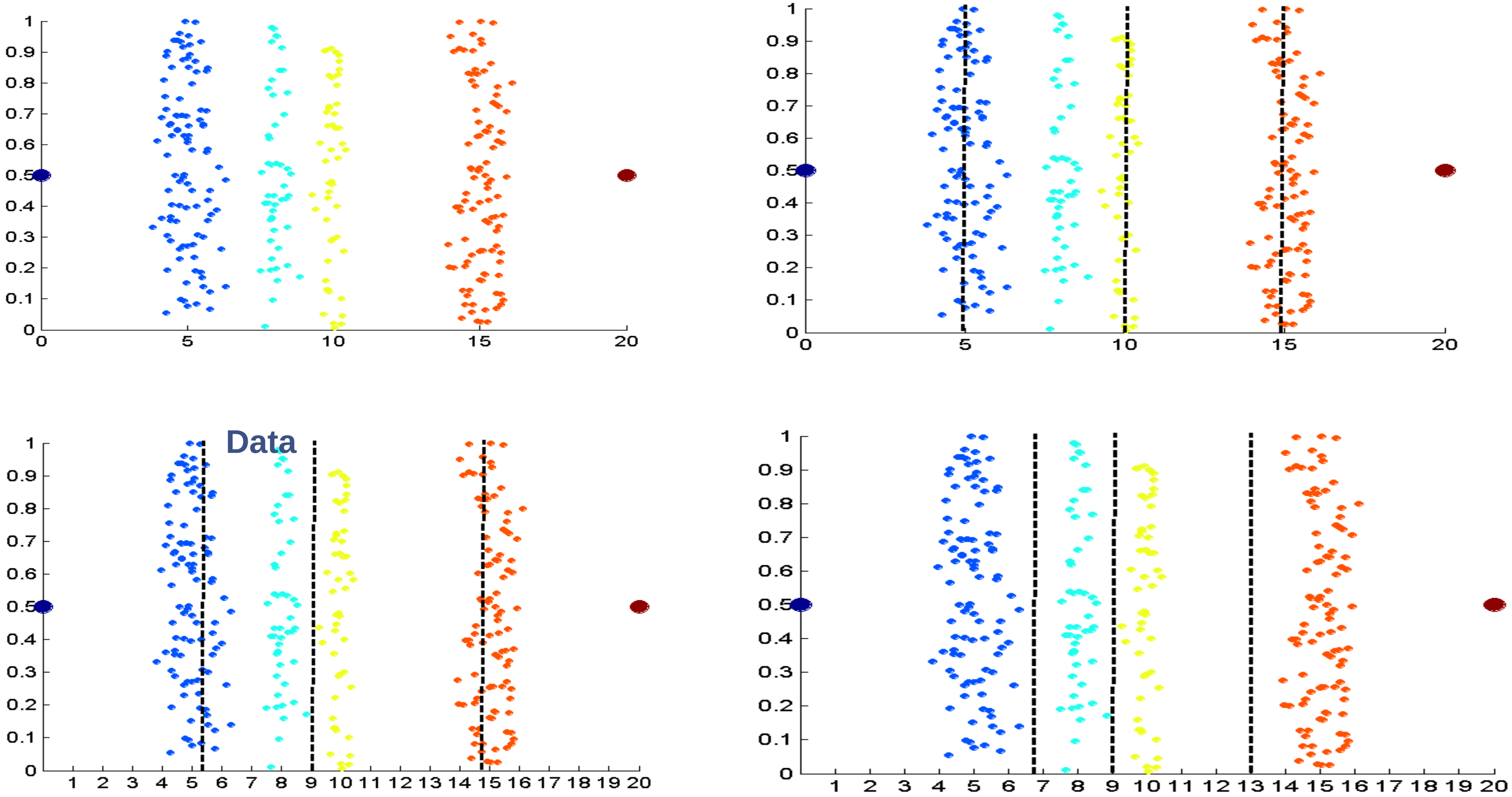
* Smoothing by bin means:

- 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 USING CLASS LABELS (BINNING VS. CLUSTERING)



Equal frequency (binning)

K-means clustering leads to better results

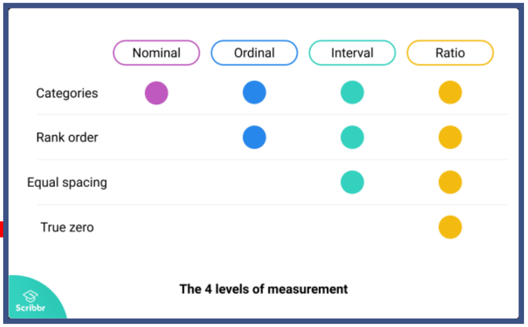
DISCRETIZATION BY CLASSIFICATION & CORRELATION ANALYSIS

- **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
- **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

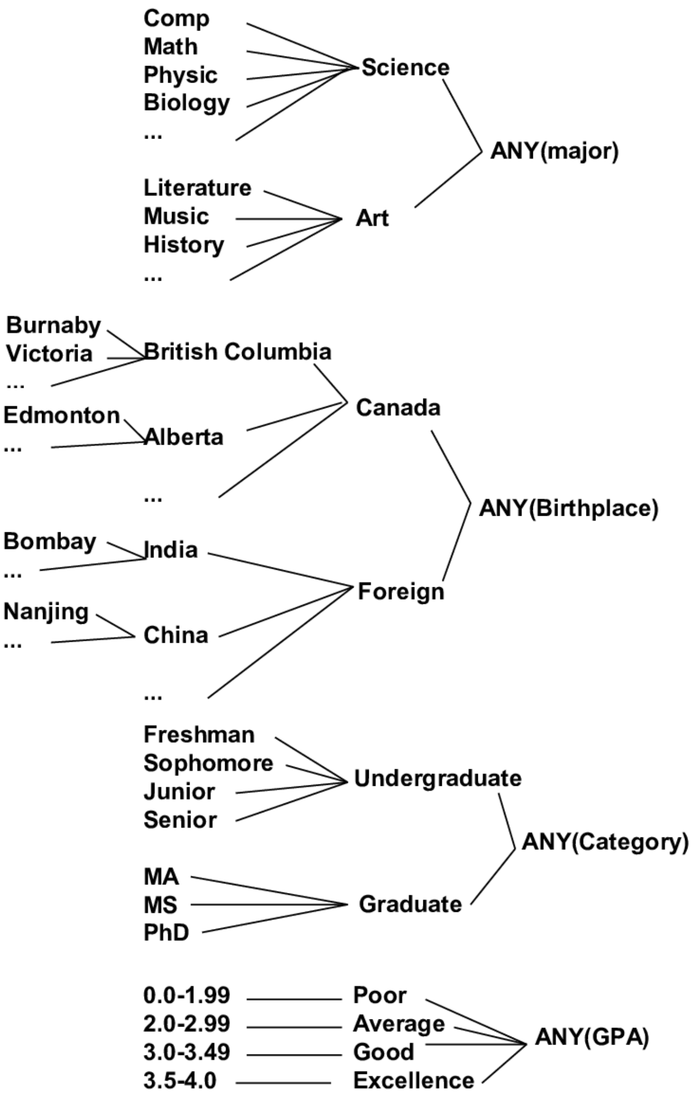
- Concept hierarchy organizes concepts (i.e., attribute values) hierarchically and is usually associated with each dimension in a data warehouse
- Concept hierarchies facilitate drilling and rolling in data warehouses to view data in multiple granularity
- Concept hierarchy formation: Recursively reduce 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



	Nominal	Ordinal	Interval	Ratio
Categories	●	●	●	●
Rank order		●	●	●
Equal spacing			●	●
True zero				●

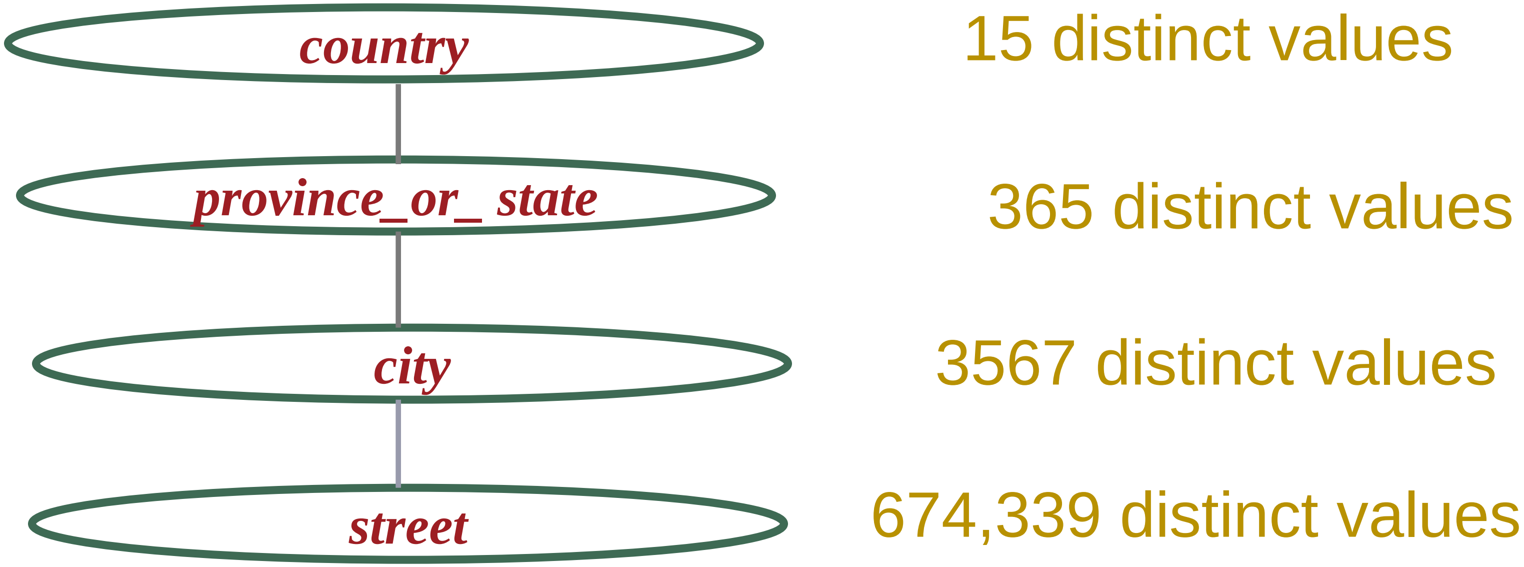
The 4 levels of measurement



- **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



country 15 distinct values

province_or_ state 365 distinct values

city 3567 distinct values

street 674,339 distinct values

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**
  - Dimensionality reduction
  - Numerosity reduction
  - Data compression
- **Data transformation and data discretization**
  - Normalization
  - Concept hierarchy generation

END