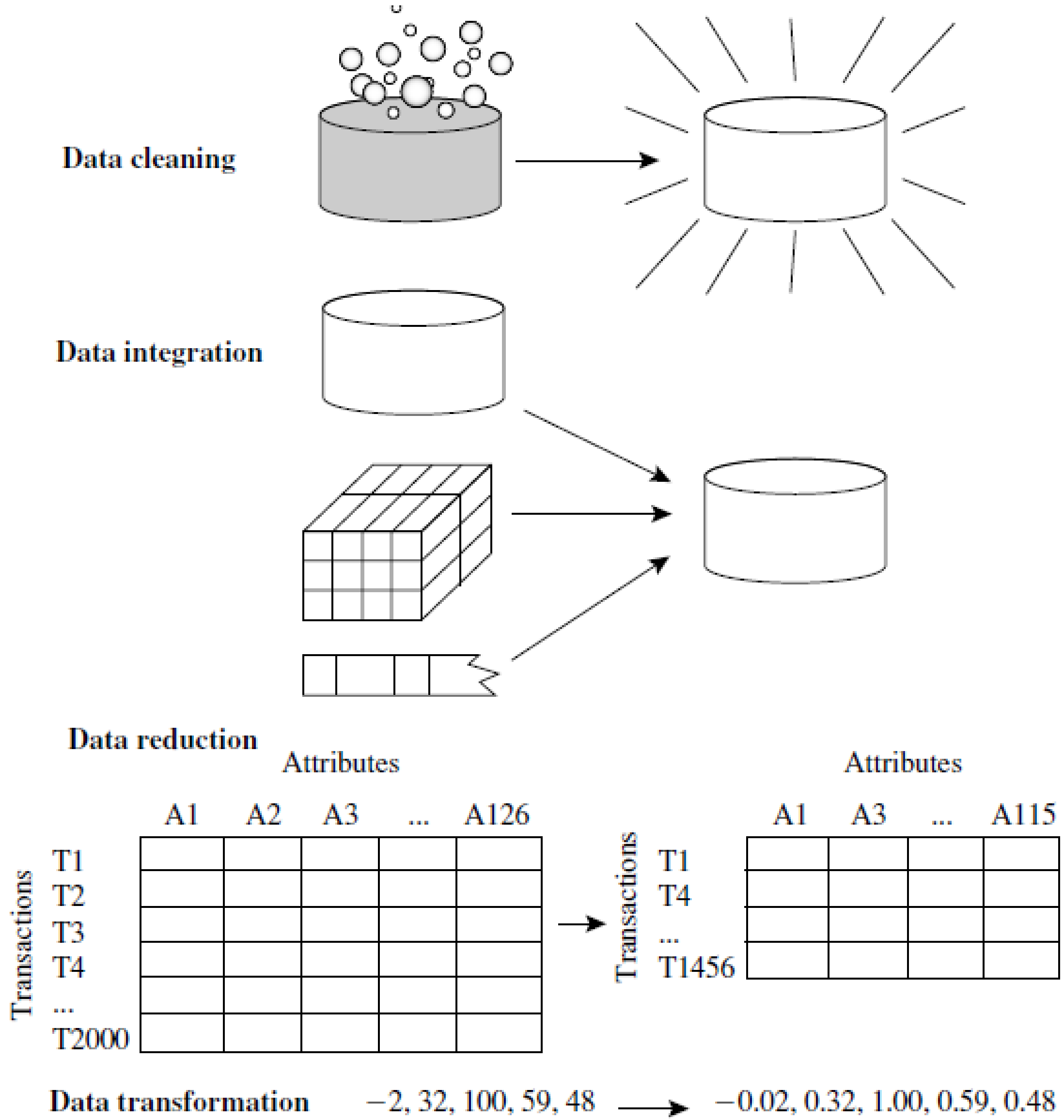
KDD process steps

- **Data cleaning** - to remove noise and inconsistent data
- **Data integration** - where multiple data sources may be combined
- **Data selection** - where data relevant to the analysis task are retrieved from the database
- **Data transformation** - where data are transformed and consolidated into forms appropriate for mining by performing summary or aggregation operations
- **Data mining** - an essential process where intelligent methods are applied to extract data patterns
- **Pattern evaluation** - to identify the truly interesting patterns representing knowledge based on *interestingness measures*
- **Knowledge presentation** - where visualization and knowledge representation techniques are used to present mined knowledge to users

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 reduction**
  - **Dimensionality reduction** – data compression techniques like PCA, attribute subset selection, attribute construction
  - **Numerosity reduction** – smaller representations using parametric models (regression) or nonparametric models (histograms, clusters, sampling or data aggregation)
- **Data transformation and data discretization**
  - Normalization
  - Concept hierarchy generation

Forms of data preprocessing



Data Cleaning

Data Cleaning

- **Data in the Real World Is Dirty: Why?** Lots of potentially incorrect data, e.g., instrument faulty, human or computer error, transmission error
  - <u>**incomplete**</u>: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
    - e.g., *Occupation*=" " (missing data)
  - <u>**noisy**</u>: containing noise, errors, or outliers
    - e.g., *Salary*="−10" (an error)
  - <u>**inconsistent**</u>: 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
  - <u>**Intentional**</u> (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 not entered due to misunderstanding (left Blank)
  - certain data may not be considered important at the time of entry (left Blank)
  - not register history or changes of the data

How to Handle Missing Data?

- **Ignore the tuple:** usually done when class label is missing (when doing classification) - not effective, unless the tuple contains several attributes with missing values
- **Fill in the missing value manually:** tedious + infeasible?
- **Fill in it automatically with**
  - a global constant : e.g., "unknown"
  - the attribute mean (Central Tendency: Mean, Median, Mode)
  - the attribute mean for all samples belonging to the same class
  - 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

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

How Binning is done?

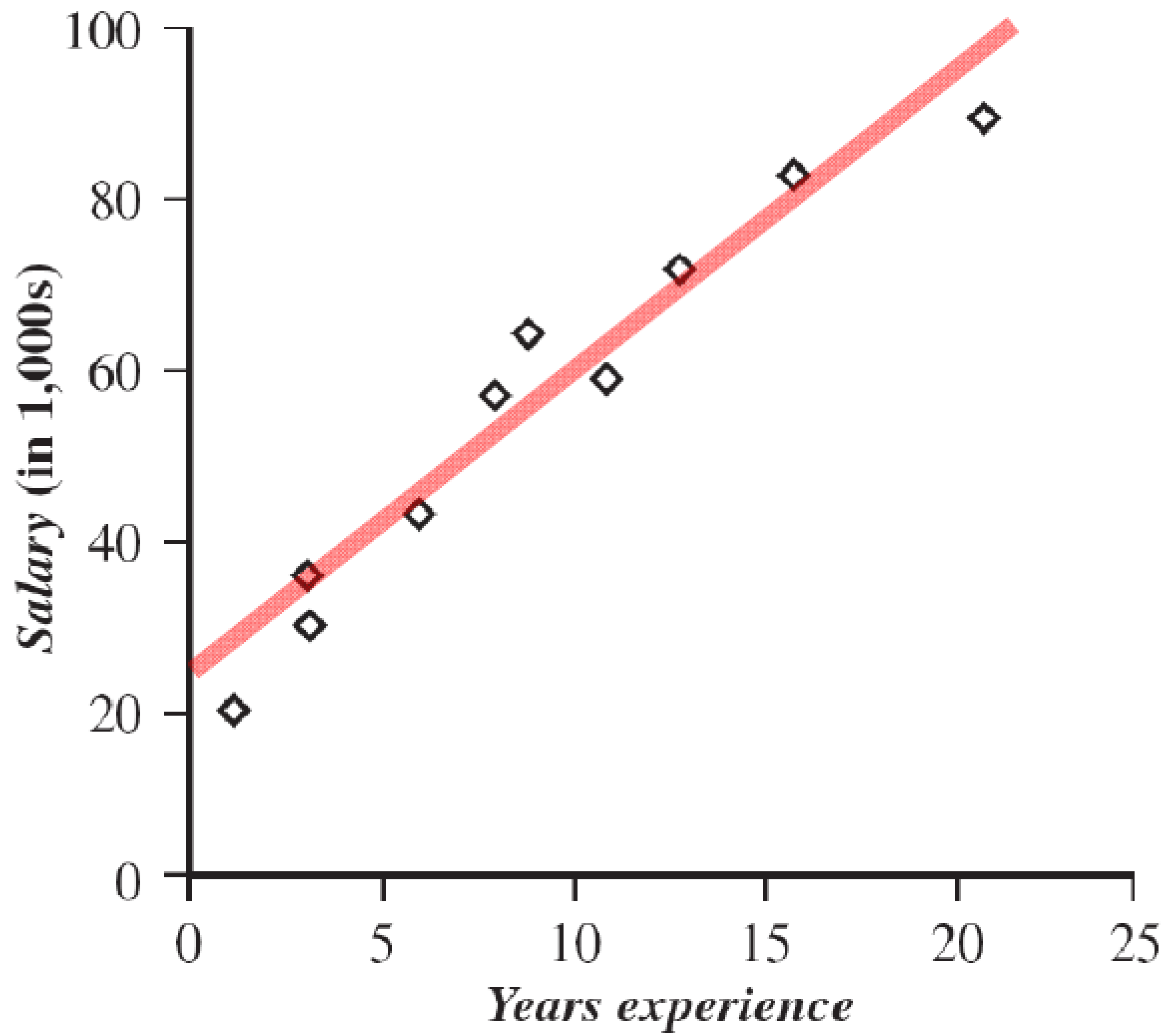
- Equal-width(distance) partitioning
  - Divides the range into N intervals of equal size: uniform grid
  - if A and B are the lowest and highest values of the attribute, the width of intervals will be: $W = (B - A)/N$.
  - The most straightforward, but outliers may dominate presentation
  - Skewed data is not handled well

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

Equal-width (distance) partitioning

- Sorted data for price (in dollars):
  - 4, 8, 15, 21, 21, 24, 25, 28, 34
- W = (B –A)/N = (34 – 4) / 3 = 10
  - Bin 1: 4-14, Bin2: 15-24, Bin 3: 25-34
- Equal-width (distance) partitioning:
  - Bin 1: 4, 8
  - Bin 2: 15, 21, 21, 24
  - Bin 3: 25, 28, 34

Regression



100
80
60
40
20
0
Salary (in 1,000s)
0
5
10
15
20
25
Years experience

Clustering

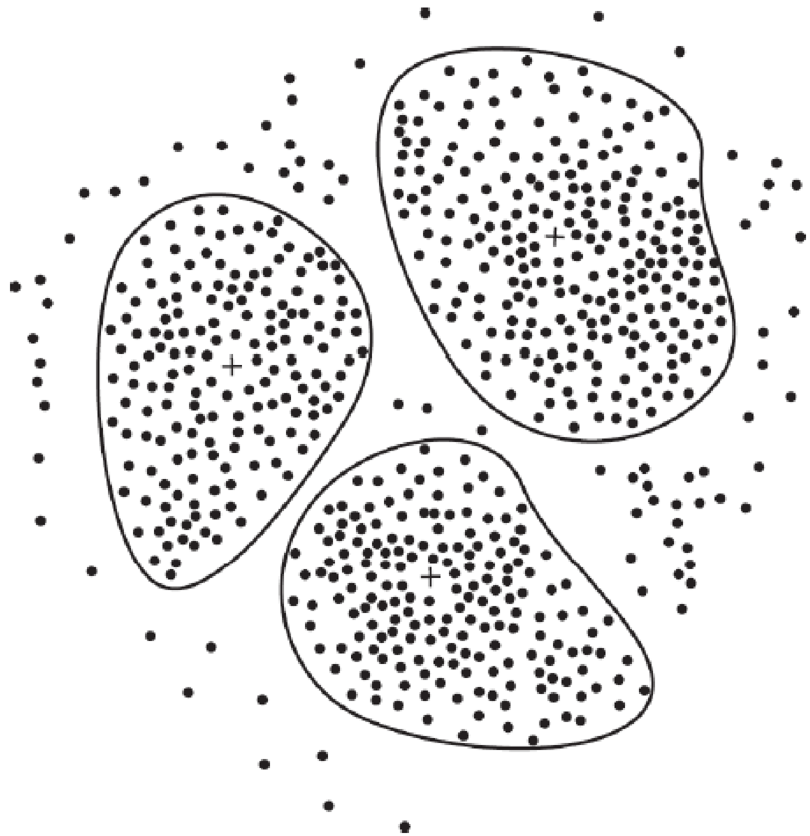


Figure: A 2-D plot of customer data with respect to customer locations in a city, showing three data clusters. Each cluster centroid is marked with a "+", representing the average point on space that cluster. Outliers may be detected as values that fall outside of the sets of clusters.

Data Integration

Data Integration

- **Data integration**: Combines data from multiple sources into a coherent store
- **[1] Schema integration:** e.g., A.cust-id ≡ B.cust-#
  - Solution: To resolve errors, use **metadata** for integration of data from different sources.
- **Entity identification problem:**
  - Identify real world entities from multiple data sources
- **[2] Detecting and resolving data value conflicts (Solution: Sec. 3.2.3 Book)**
  - For same real world entity, attribute values from different sources are different.
  - Possible reasons: different representations, different scales, encoding.
  - **Eg1:** A weight attribute may be stored in metric units in one system and British imperial units in another.
  - **Eg2:** For a hotel chain, the price of rooms in different cities may involve not only different currencies but also different services (e.g., free breakfast) and taxes.

Handling Redundancy in Data Integration

- **[3] Redundancy and correlation analysis:** Redundant data occurs 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
- Redundant attributes may 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 (Categorical Data)

- For categorical (discrete) data, a correlation relationship between two attributes, A and B, can be discovered by a $\chi 2$ test

- Given the degree of freedom, the value of $\chi 2$ is used to decide correlation based on a significance level

- For nominal data, use the $\chi 2$ (chi-square) test.
- For numeric attributes, use the correlation coefficient and covariance.

Correlation Analysis (Categorical Data)

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

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

$$\chi^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

$$e_{ij} = \frac{count(A = a_i) \times count(B = b_j)}{n}$$

- The χ**2** tests the hypothesis that *A and B are independent,* that is, there is no correlation between them.
- The test is based on a significance level, with **(r – 1) x (c – 1)** degrees of freedom.
- If the hypothesis can be rejected, then we say that *A and B are statistically correlated.*
- **Note:**
  - The larger the χ**2** value, the more likely the variables are related.
  - The cells that contribute the most to the χ**2** value are those whose actual count is very different from the expected count.

Chi-Square Calculation: An Example

	male	female	Sum (row)
fiction	250(90)	200(360)	450
Non-fiction	50(210)	1000(840)	1050
Sum(col.)	300	1200	1500

- **Note:** Are *gender* and *preferred reading* correlated?
- χ**2** calculation (numbers in parenthesis are expected counts)

$$e_{11} = \frac{count(male) \times count(fiction)}{n} = \frac{300 \times 450}{1500} = 90$$

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

- DoF is 1; χ2 value reject the hypothesis at the 0.001 significance level is 10.828.
- It shows that *gender* and *preferred reading* are correlated in the group

Correlation Analysis (Numerical Data)

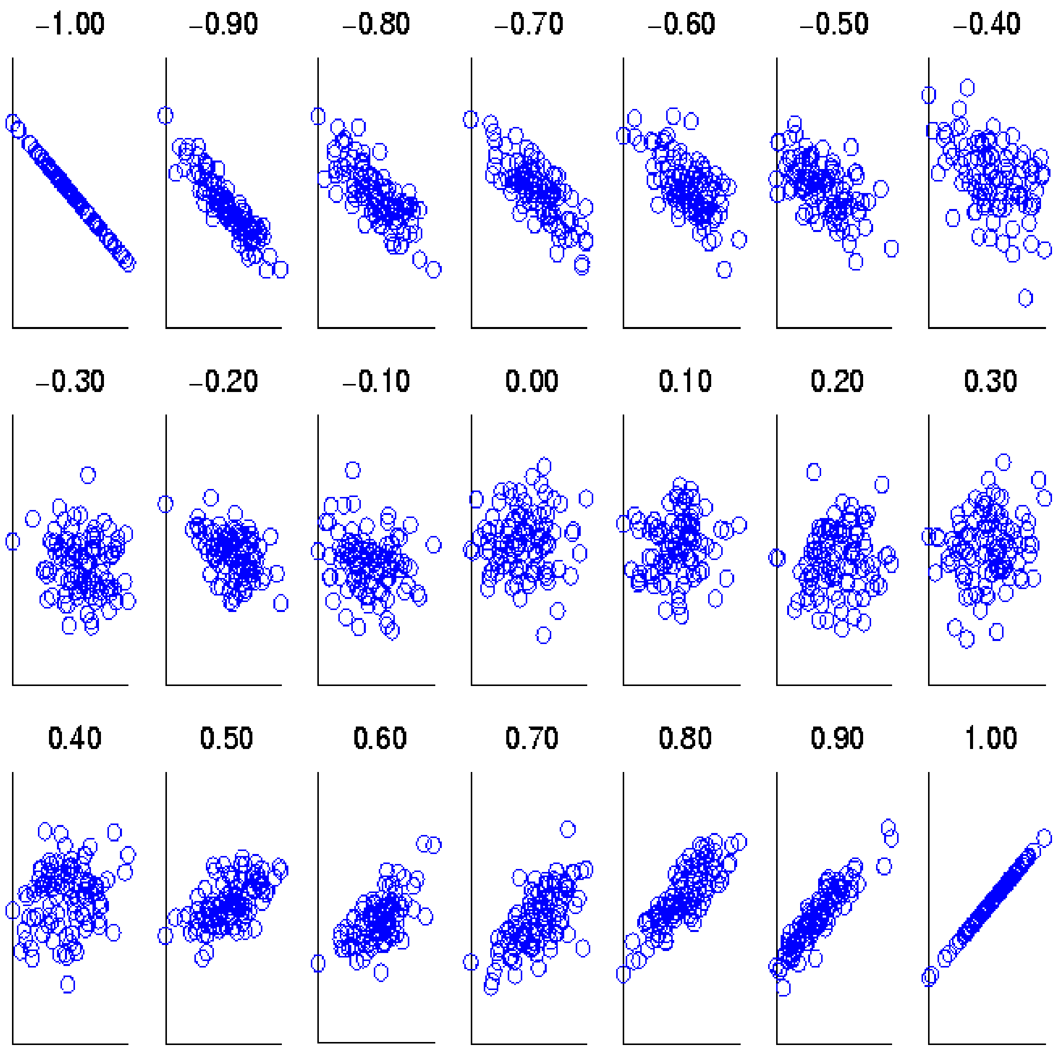
- Correlation coefficient (also called Pearson's product moment coefficient)

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

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. Thus A (or B) redundant.
- If $r_{A,B} = 0$, A and B are independent, *no correlation*;
- If $r_{AB} < 0$, A and B are *negatively correlated*.

Visually Evaluating Correlation



Scatter plots showing the similarity from

–1 to 1

($-1 <= r_{A,B} <= 1$)

Correlation Vs Causality

- **Correlation does not imply causality**
- If *A and B are correlated, this* does not necessarily imply that *A causes B or that B causes A.*
- **Eg:**
  - Attributes # of hospitals and # of car-theft in a city are correlated.
  - Does this mean that one causes the other?
  - Both are causally linked to the third variable, namely, *population*.

Covariance (Numeric Data)

- Covariance is similar to correlation

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

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

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.

- **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.

Covariance: 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,

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

- **Eg:** 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?
  - E(A) = (2 + 3 + 5 + 4 + 6)/ 5 = 20/5 = 4
  - E(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.**

Data Reduction

Data Reduction

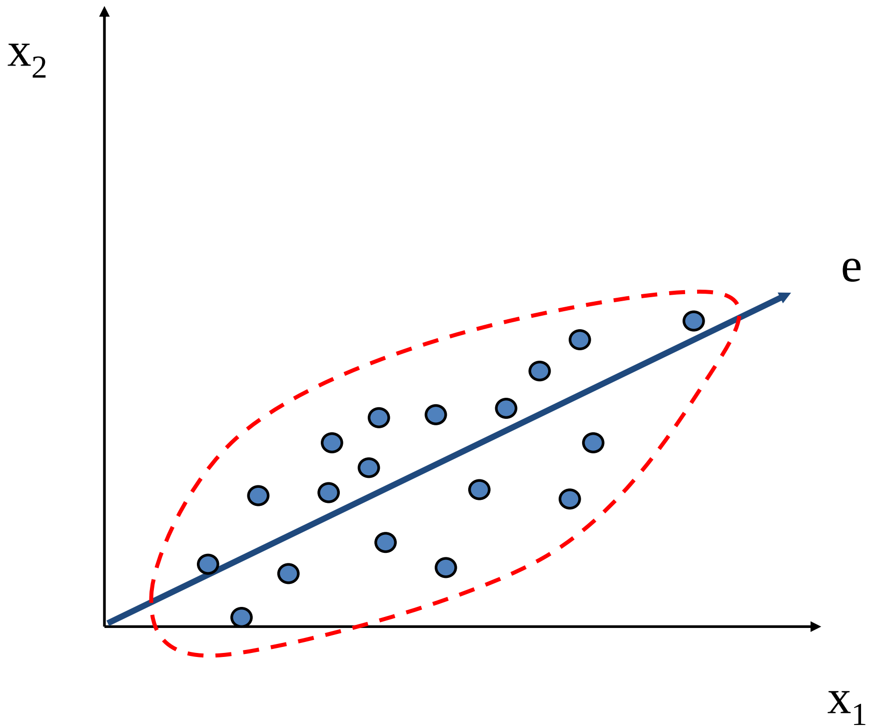
- **Data reduction**: Obtain a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results.
- **Why data reduction?** — A database/data warehouse may store terabytes of data. Complex data analysis may take a very long time to run on the complete data set.
- Data reduction strategies
  - **Dimensionality reduction**
    - remove attributes that are the same or similar to other attributes
  - **Numerosity reduction**
    - represent or aggregate the data, sometimes with precision loss
  - **Data compression**
    - generalized techniques to decrease the number of bytes needed to store data
  - **Data cube aggregation**

Data Reduction 1: Dimensionality Reduction

- **Dimensionality reduction techniques**
  - Wavelet transforms
  - Principal Component Analysis
  - **Attribute subset selection (e.g., feature selection)**

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 $k \leq n$ orthogonal vectors (*principal components*) that can be best used to represent data
  - Normalize input data: Each attribute falls within the same range
  - Compute k ortho-normal (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

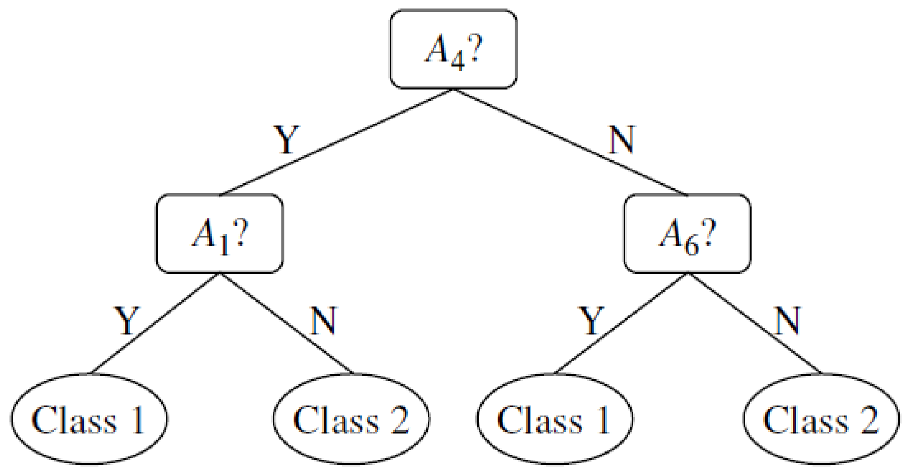
Attribute Subset Selection

- **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.
  - Best single attribute under the attribute independence assumption: choose by significance tests.
- **Typical heuristic attribute selection methods:**
  1. **Stepwise forward selection** (best step-wise feature selection):
     - The best single-attribute is picked first
     - Then next best attribute is added, ...
  2. **Stepwise backward elimination** (step-wise attribute elimination):
     - Repeatedly eliminate the worst attribute
  3. **Best combined attribute selection and elimination**
  4. **Decision tree induction:**
     - Tree is constructed from given data. At each node, the algorithm chooses the "best" attribute to partition the data into individual classes

Greedy (heuristic) methods for attribute subset selection

Forward selection	Backward elimination	Decision tree induction
Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ Initial reduced set: $\{\}$ => $\{A_1\}$ => $\{A_1, A_4\}$ => Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ => $\{A_1, A_3, A_4, A_5, A_6\}$ => $\{A_1, A_4, A_5, A_6\}$ => Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$  => Reduced attribute set: $\{A_1, A_4, A_6\}$

Data Reduction 2: Numerosity Reduction

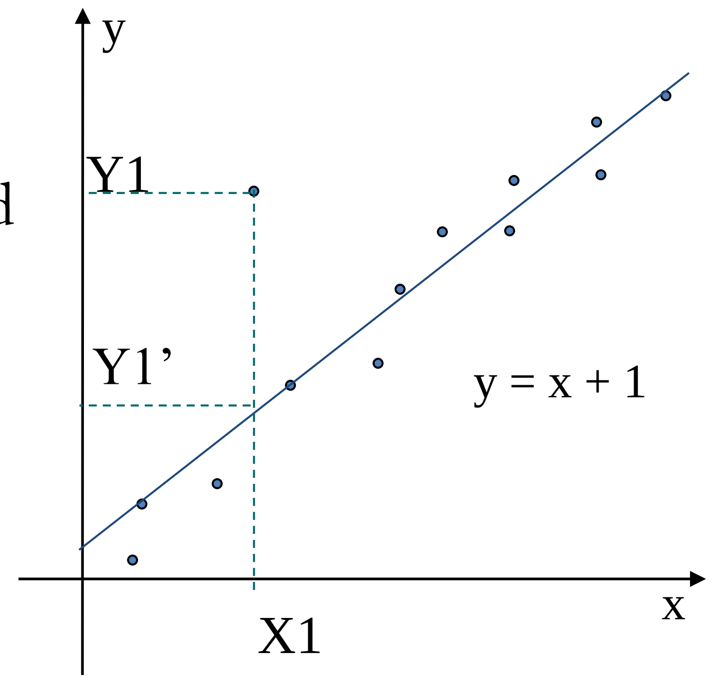
- Reduce data volume by choosing alternative, *smaller forms* of data representation

- **Parametric methods**
  - Assume the data fits some model, estimate model parameters, store only the parameters, and discard the data (except possible outliers)
  - Ex: Regression, Log-linear models

- **Non-parametric methods**
  - histograms, clustering, sampling, and data cube aggregation

Parametric Data Reduction: Regression and Log-Linear Models

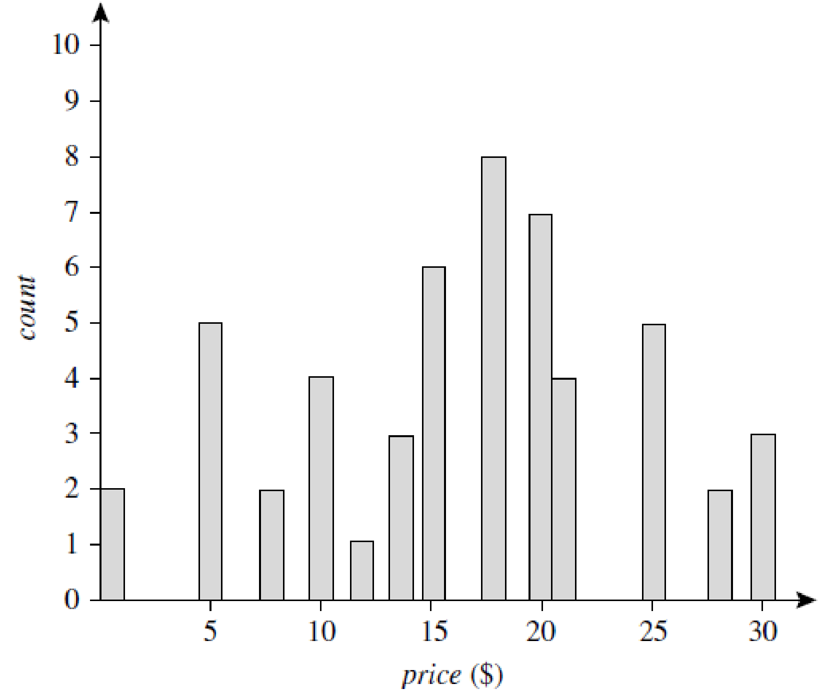
- **Linear regression**
  - Data modeled to fit a straight line
  - Often uses the least-square method to fit the line
  - $Y = w\,X + b$ *where* w and b *are* regression coefficients
- **Multiple regression**
  - Allows a response variable Y to be modeled as a linear function of multidimensional feature vector
  - $Y = b_0 + b_1\,X_1 + b_2\,X_2$

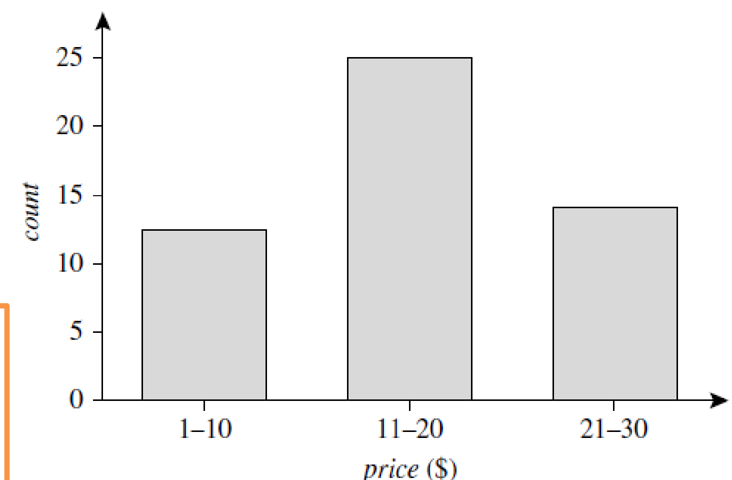


Note: see Section 3.4.5 for more details.

Histogram A

- Divide data into bins (buckets) and store average (sum) for each bin.
- ***What's singleton buckets?***
  - each bucket represents only a single attribute –value/frequency pair
- **Partitioning rules:**
  - Equal-width histogram
  - Equal-frequency histogram





Note: Histograms are highly effective at approximating both **sparse and dense data**, as well as **highly skewed and uniform data**.

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".

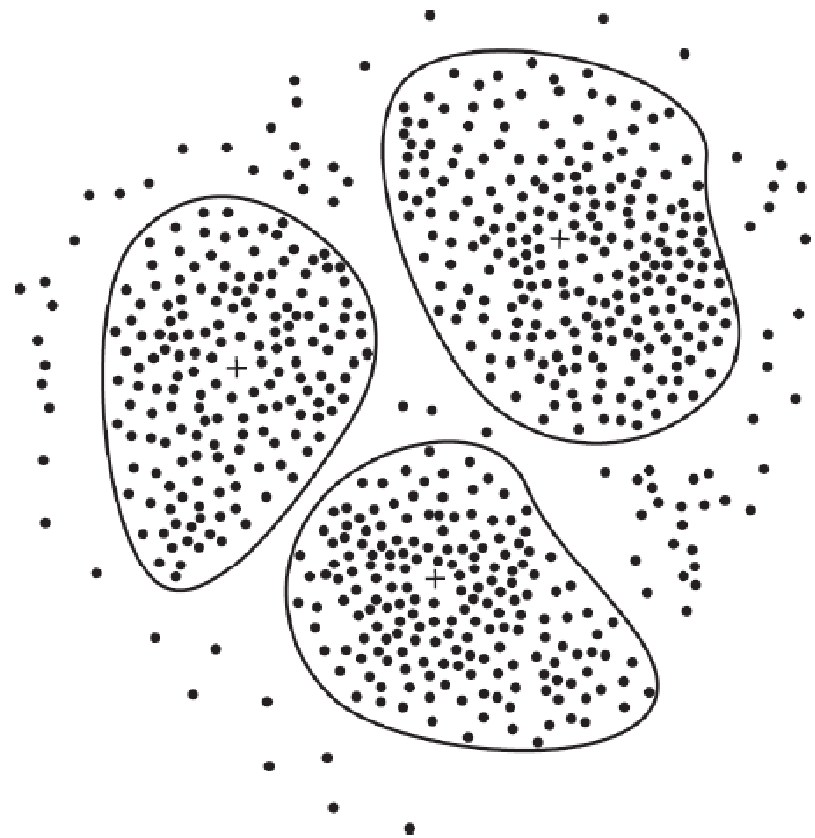


Figure: A 2-D plot of customer data with respect to customer locations in a city, showing three data clusters. Each cluster centroid is marked with a "+", representing the average point on space that cluster. Outliers may be detected as values that fall outside of the sets of clusters.

Sampling

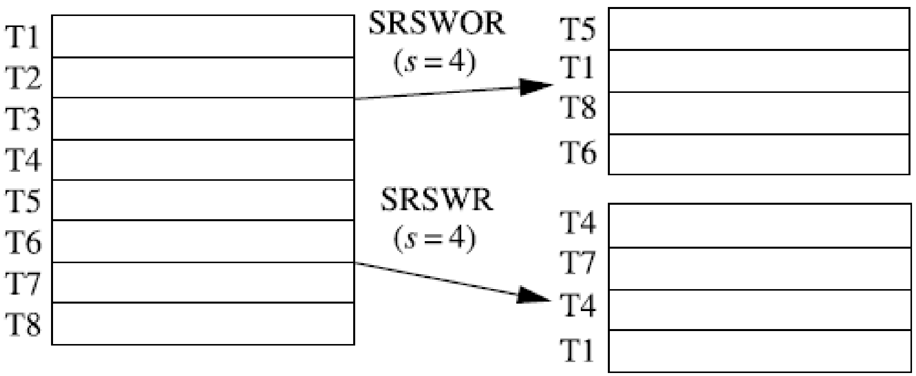
- **Sampling:** obtaining a small sample s to represent the whole data set N.

- **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**

Types of Sampling

- **Simple random sampling (SRS)**
  - There is an equal probability of selecting any particular item
- **SRS without replacement (SRSWOR)**
  - Once an object is selected, it is removed from the population
- **SRS with replacement (SRSWR)**
  - 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

Types of Sampling



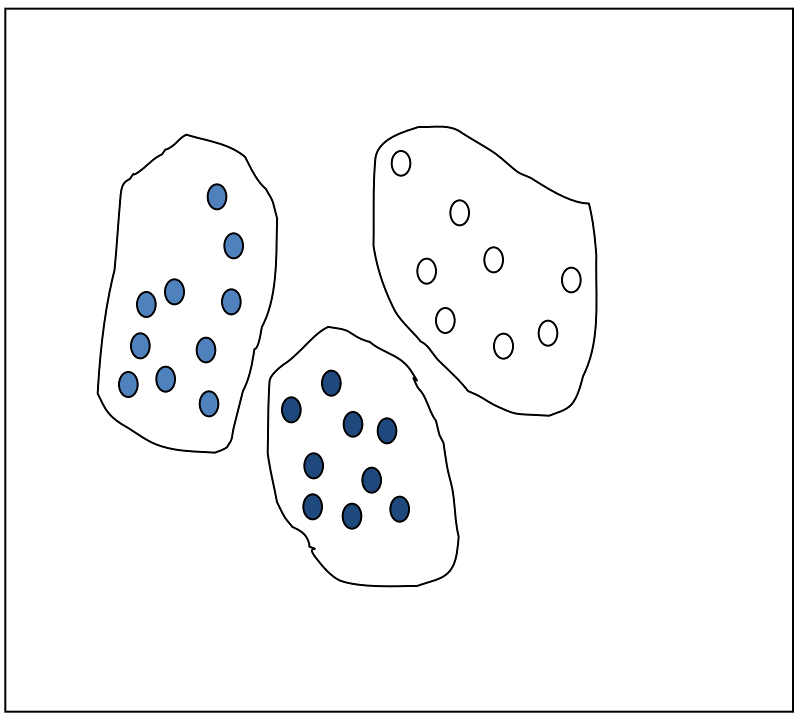
T38	youth
T256	youth
T307	youth
T391	youth
T96	middle_aged
T117	middle_aged
T138	middle_aged
T263	middle_aged
T290	middle_aged
T308	middle_aged
T326	middle_aged
T387	middle_aged
T69	senior
T284	senior

Startified sample
(according to *age*)

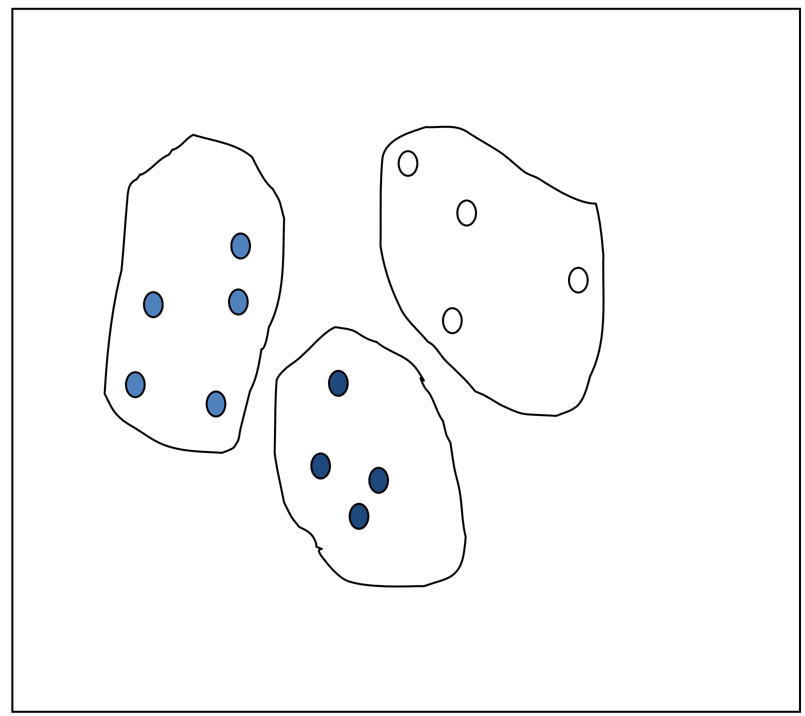
T38	youth
T391	youth
T117	middle_aged
T138	middle_aged
T290	middle_aged
T326	middle_aged
T69	senior

Sampling: Cluster or Stratified Sampling

Raw Data

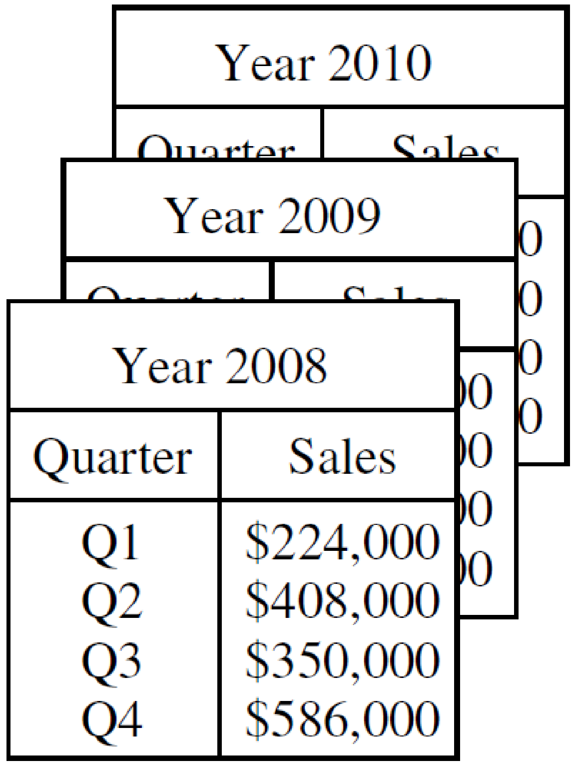


Cluster/Stratified Sample



Data Reduction 3: Data Cube Aggregation

Note: The resulting data set is smaller in volume, without loss of information necessary for the analysis task.

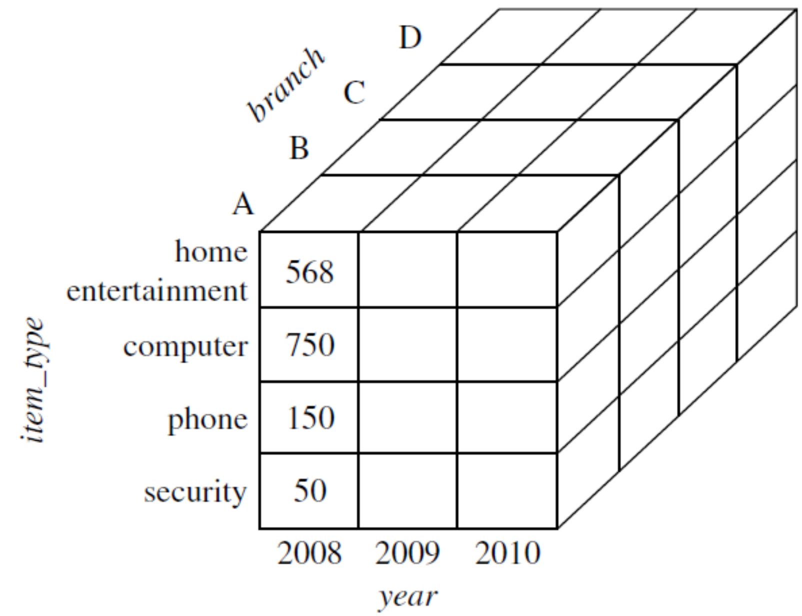


Year	Sales
2008	$1,568,000
2009	$2,356,000
2010	$3,594,000

Fig: Sales data for a given branch for the years 2008 through 2010. On the left, the sales are shown per quarter. On the right, the **data are aggregated** to provide the annual sales.

Data Cube Aggregation

- **Data cubes** store multidimensional aggregated information.
- **Figure** shows a data cube for multidimensional analysis of sales data with respect to ***annual sales*** per ***item type*** for each ***branch***.
- Each cell holds an aggregate data value.
- **Adv:** provides fast access to pre-computed and summarized data.

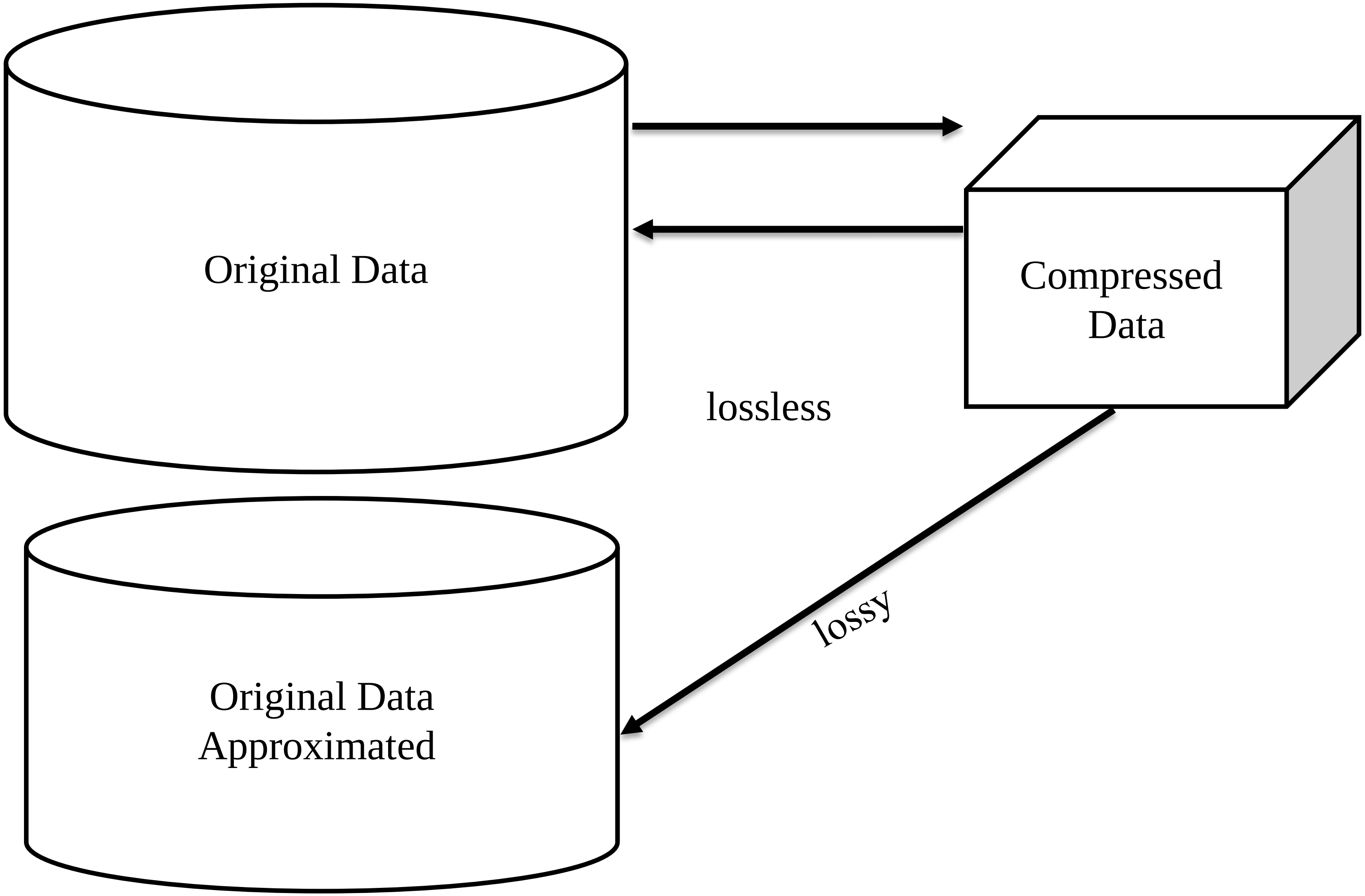


- The cube created at the lowest abstraction level is referred to as **base cuboid.**
- A cube at the highest level of abstraction is the **apex cuboid.**

Data Reduction 4: 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.
- **Dimensionality and numerosity reduction** may also be considered as forms of data compression.

Data Compression



Data Transformation

Data Transformation

- **What ?** 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

- **Why?** Transformation leads into mining process to be more efficient and as well patterns found may be easier to understand.

Data Transformation Handling Methods

- **Smoothing:** Remove noise from data
- **Attribute/feature construction:** New attributes constructed from the given ones
- **Aggregation:** Summarization and data cube construction
- **Normalization:** Scaled to fall within a smaller, specified range
  - min-max normalization, z-score normalization, normalization by decimal scaling
- **Discretization:** The raw values of a numeric attribute (e.g., *age*) are replaced by interval labels (e.g., *0–10, 11–20*, etc.) or conceptual labels (e.g., *youth, adult, senior*).
- **Concept hierarchy generation for nominal data:** Attributes such as *street* can be generalized to higher-level concepts, like *city* or *country*

Data Transformation by Normalization

- Normalizing the data attempts to give all attributes an **equal weight**.

- Normalization is useful for classification algorithms involving **neural networks** or distance measurements such as **nearest-neighbor classification** and **clustering**.

Normalization methods

- *Let A be* a numeric attribute with *n observed values, v1, v2, …, vn.*
- **[1] 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 $\frac{73{,}600 - 12{,}000}{98{,}000 - 12{,}000}(1.0 - 0) + 0 = 0.716$

- **[2] Z-score normalization** (or zero–mean normalization) :

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

  – Ex: Let μ = 54,000, σ = 16,000. Then $\frac{73{,}600 - 54{,}000}{16{,}000} = 1.225$

- **Usefulness of Z-score normalization:**
  – when the actual minimum and maximum of the attribute are unknown
  – when there are outliers that dominate the min-max normalization

Normalization methods

- **[3] Normalization by decimal scaling:** normalizes by moving the decimal point of values of attribute A.
  - The number of decimal points moved depends on the maximum absolute value of A.
  - A value v of A is normalized to v' by computing

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

– **Ex:** Suppose that the recorded values of A range from -986 to 917.
The maximum absolute value of A is 986.
To normalize by decimal scaling, divide each value by 1000 (i.e., j = 3) so that -986 normalizes to -0.986 and 917 normalizes to 0.917.