Data Warehousing & Data Mining

UNIT-II

Data Preprocessing

B.Tech(CSE)-V SEM

Course Outcomes

After Successful completion of the Course, the student will be able to:

CO1: Discuss the basic concepts and techniques of data warehousing and data mining. (K2)

CO2: **Demonstrate the types of the data to be mined and apply pre-processing methods on raw data. (K3)**

CO3: Illustrate various Classification Techniques. **(K3)**

CO4: Discover interesting patterns, analyze supervised and unsupervised models and estimate the accuracy of the algorithms. **(K3)**

CO5: Use different clustering techniques to cluster data. **(K3)**

UNIT : II -Data Preprocessing

- Overview of Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation
- Data Discretization

Data Preprocessing

- It is a data mining technique that involves transforming raw data in to an understandable format.
- To make data more suitable for data mining.
- To improve the data mining analysis with respect to time, cost and quality.

Why preprocess the data?

- Data in the real world is:

  - 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: lack of compatibility or similarity between two or more facts. e.g.,

  - *Age*=“42”, *Birthday*=“03/07/2010”
  - Was rating “1, 2, 3”, now rating “A, B, C”
  - discrepancy between duplicate records

- No quality data , no quality mining results:

  - Quality decisions must be based on quality data
  - Data warehouse needs consistent integration of quality data

Why preprocess the data?

- No quality data, no quality mining results!
  - Quality decisions must be based on quality data
    - e.g., duplicate or missing data may cause incorrect or even misleading statistics.

Data preparation, cleaning, and transformation comprises the majority of the work in a data mining application (70%).

Measures of Data Quality

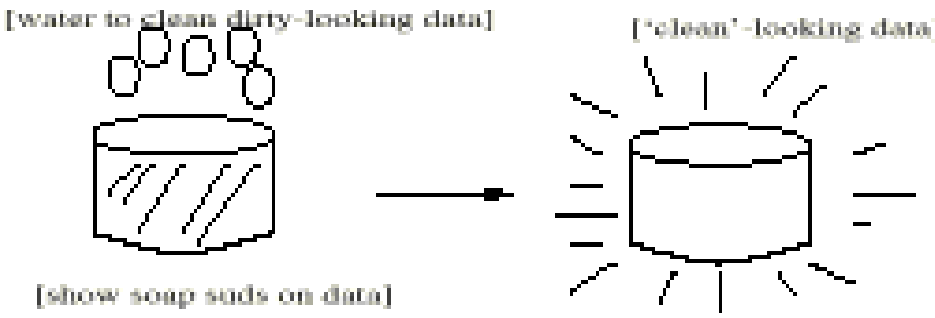
- Accuracy
- Completeness
- Consistency
- Timeliness
- Believability
- Value added
- Interpretability
- Accessibility

Major Tasks in Data Preprocessing

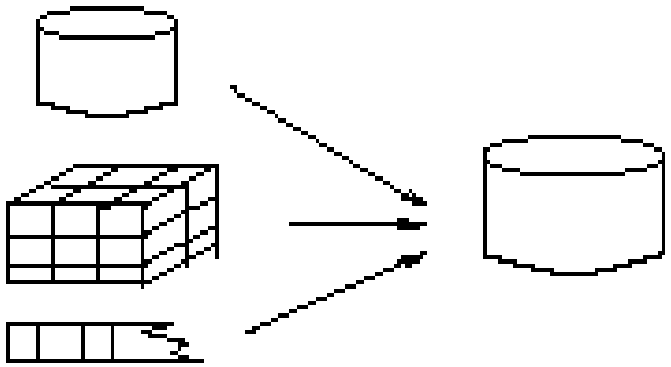
- **Data Cleaning:** Fill in Missing Values, Smooth noisy data, identify or remove outliers, and resolve inconsistencies.
- **Data Integration:** Integration of Multiple databases, data cubes, or files.
- **Data Reduction:** Obtains reduced representation in volume but produces the same or similar analytical results
  - Dimensionality reduction
  - Numerosity reduction
  - Data compression
- **Data Transformation and Discretization (for numerical data):** Normalization and Concept hierarchy generation

Forms of Data preprocessing

Data Cleaning



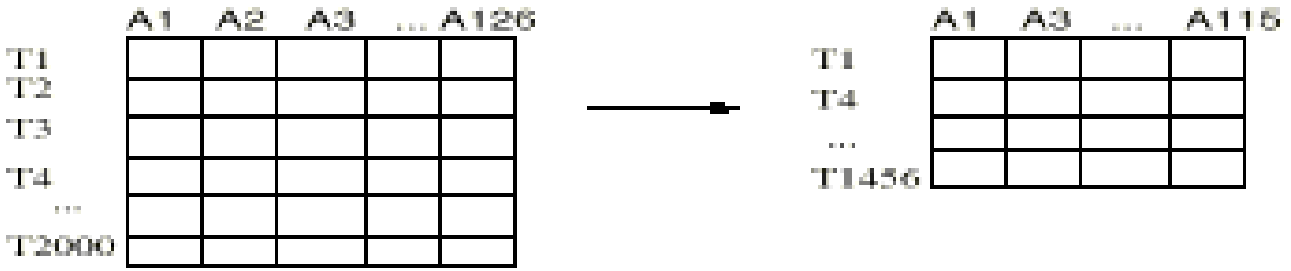
Data Integration



Data Transformation

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

Data Reduction



Data cleaning

- Data cleaning attempts to fill in missing values, smooth out noise while identifying outliers and correct inconsistencies in the data.

- **Data cleaning Tasks:**

1. Missing Values
2. Noisy Data
3. Inconsistent Data

Data Cleaning – Missing Values

1. **Ignore the tuple:** This is usually done when the class label is missing. This method is not very effective, unless the tuple contains several attributes with missing values.
2. **Fill in the missing values manually:** In general, this approach is time- consuming and may not be feasible given a large data set with many missing values.
3. **Use a global constant to fill in the missing value:** Replace all missing attribute values by the same constant, such as a label like "unknown" or "-∞".
4. **Use the attribute mean to fill in the missing values**
5. **Use the attribute mean for all samples belonging to the same class as the given tuple.**
6. **Use the most probable value to fill in the missing value:** This may be determined with inference-based such as Bayesian formula or decision tree

Data Cleaning – 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 requires data cleaning
    - duplicate records
    - inconsistent data

How to Handle Noisy Data?

1. Binnig
2. Clustering
3. Combined Computer and human inspection
4. Regression

Data Cleaning – Noisy Data

1. **Binning:**

- First sort data and partition into (equi-depth) bins
- Then one can **smooth by bin means, smooth by bin median, smooth by bin boundaries,** etc.

Binning Methods for Data Smoothing

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

*Partition into (equi-depth) bins:

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

*Smoothing by bin means:

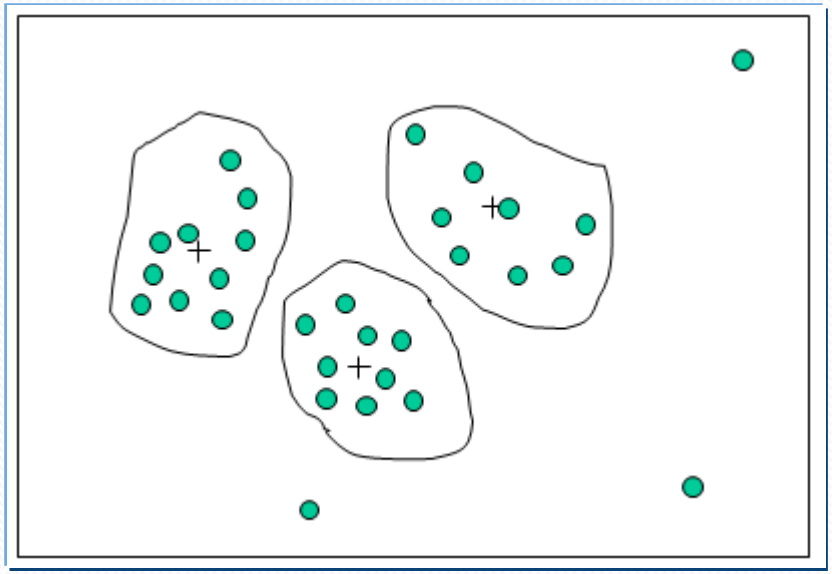
- **Bin 1:** 9, 9, 9, 9-
- **Bin 2:** 23, 23, 23, 23
- **Bin 3:** 29, 29, 29, 29

*Smoothing by bin boundaries:

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

Data Cleaning – Noisy Data

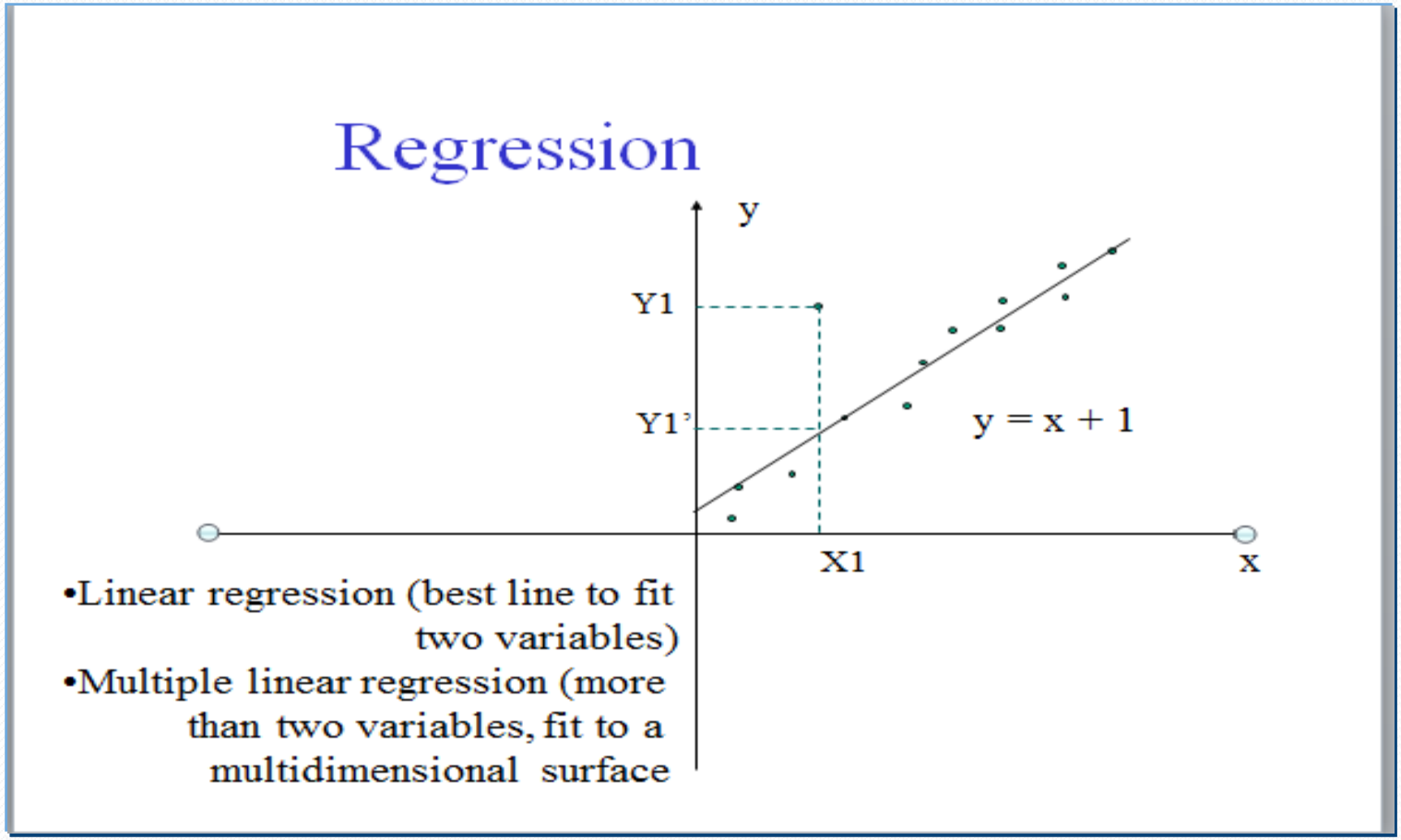
2. **Clustering:**
   - Similar values are organized into groups (clusters).
   - Values that fall outside of clusters considered outliers



3. **Combined computer and human inspection:** Outliers may be identified through a combination of computer and human inspection. Outlier patterns may be informative or garbage.

Data Cleaning – Noisy Data

4. **Regression:** Data can be smoothed by fitting the data to a function such as with regression. (linear regression/multiple linear regression)



Data Integration

- The merging of data from multiple data sources. The data may also need to be transformed into forms appropriate for mining. The data sources may include multiple databases, datacubes (or) flat files.

- **Data Integration – Issues**

  1. Entity Identification Problem
  2. Redundancy and Correlation Analysis
  3. Tuple Duplication
  4. Data Conflict Detection and Resolution

Data Integration

1. Entity Identification Problem :

- Integrate metadata (about the data) from different sources.
- The real world entities from multiple source be matched referred to as the **entity identification problem.**

- **Schema integration** and **Object matching** are very important issues in Data integration.

For example, How can the data analyst and computer be sure that customer id in one data base and customer number in another reference to the same attribute. A.cust-id=B.cust-# (same entity?)

Data Integration

Schema Integration - Mismatch in Attribute names

Ex: - Cust _id, customer_id, cust _ no, etc

handling blank,zero, null values

Object Matching _ Mismatch In structure of the data

Ex:- Discount issues

Currency type

Data Integration

2. Redundancy and Correlation Analysis:

Redundancy - An attribute may be redundant if it can be "**derived**" or obtaining from another attribute or set of attributes.

Ex:- DOB, Age

Quarter sales, year sales

- In consistencies in attribute can also cause redundancies in the resulting data set.
- Some redundancies can be detected by **correlation analysis**.
- **Correlation analysis** – Given two attributes, such analysis can measure how strongly one attribute implies the other, based on the available data.

Name	DOB	Age

Branch id	Quarter total	Year total

Data Integration

2. Redundancy and Correlation Analysis:

- For nominal data , we use the X^2 (chi-square) test.
- For numeric attributes, we can use the correlation coefficient and covariance.

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
- 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

Correlation Analysis (Numeric 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.
- $r_{A,B} = 0$: independent; $r_{AB} < 0$: negatively correlated

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

Co-Variance: An Example

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

- 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?
  - 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 Integration

3. Tuple Duplication:

The use of denormalized tables (often done to improve performance by avoiding joins) is another source of data redundancy. Inconsistencies often arise between various duplicates, due to inaccurate data entry or updating some but not all data occurrences.

Name	Age	Branch	Occupation	Address
A	35	Tpg	Govt	Tpg
B	40	Tnk	Govt	Rjy
A	35	Tpg	Private	Tpg
C	40	Tnk	Private	Rjy

Data Integration

4. Data value conflict Detection and Resolution:

- Attribute values from another different sources may differ for the same real world entity.
- This may be due to differences in representation, scaling, or encoding.
- An attribute in one system may be recorded at a lower level abstraction then the "same" attribute in another.

For Example:

1. Weight attribute representation in
   metric units in one system (gms,milligms, kilos..) and
   british empherial units in another system(lb,gr,st....).

2. **Grading System in unversity**
   on (1-10) scale grade
   A+ to F grade

Data Reduction

- **Data Reduction** techniques are applied to obtain a reduced representation of the data set that is much smaller in volume, yet closely maintain the integrity of the original data.

- **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

- **Data reduction strategies:**

1. Dimensionality reduction: where irrelevant, weakly relevant (or) redundant attribute (or) dimensions may be detected (or) removed.

a) Wavelet transforms

b) Principal Components Analysis (PCA)

c) Attribute subset selection (Feature subset selection)

2. Numerosity reduction (some simply call it: Data Reduction)

a) Regression and Log-Linear Models

b) Histograms, clustering, sampling

c) Data cube aggregation

3. Data compression

Data Reduction: Dimensionality reduction

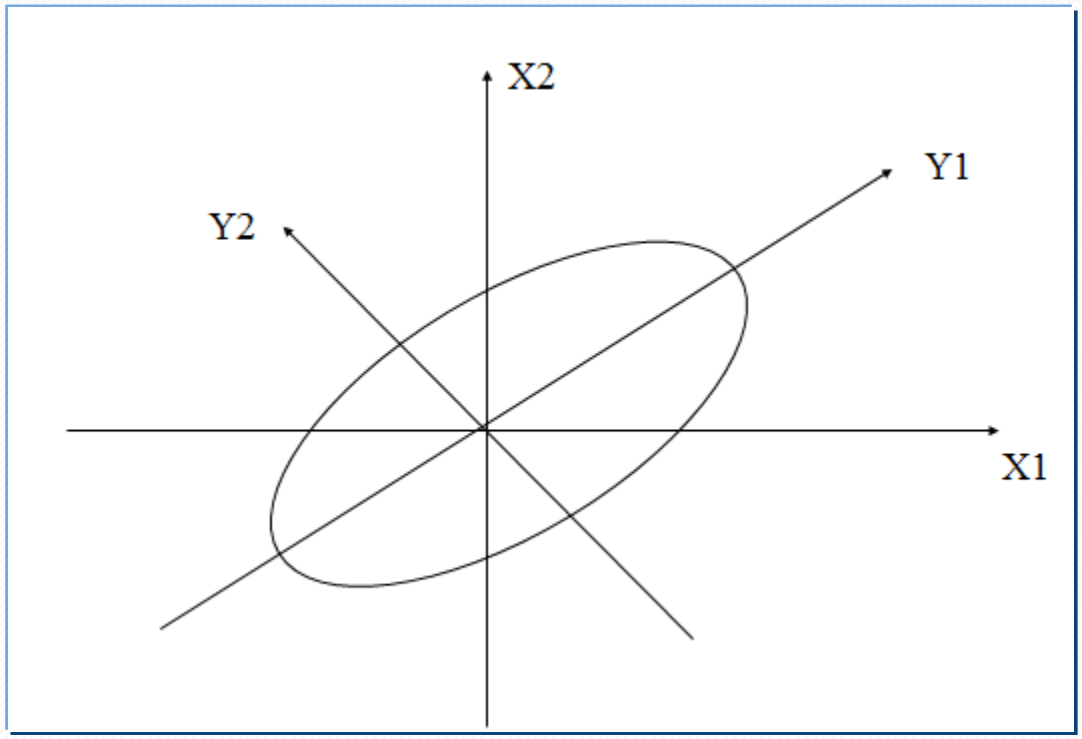
a) Discrete wavelet transforms: Transforms pixels of the images into wavelets, those will be used for wavelet based compression and coding.

Ex: An image of size 10MB compressed to 100KB

b) Principal Components Analysis : Principal components analysis (PCA; also called the Karhunen-Loeve, or K-L, method) searches for k n-dimensional orthogonal vectors that can best be used to represent the data, where k n.

Data Reduction: Dimensionality reduction

- Find a projection that captures the largest amount of variation in data
- Used to reduce data size using 'k' orthogonal vectors. Unit vectors that each point in a direction perpendicular to the others. These vectors are referred to as principal components.



Principal components analysis. Y1 and Y2 are the first two Principal components for the given data.

Data Reduction: Dimensionality reduction

c) Attribute subset selection (Feature subset selection):

- Reduces the data set size by removing irrelevant or redundant attributes (or dimensions).

- The goal of attribute subset selection is to find a minimum set of attributes such that the resulting probability distribution of the data classes is as close as possible to the original distribution obtained using all attributes.

Greedy (heuristic) methods for attribute subset selection are:

i) Stepwise forward Selection

ii) Stepwise backward elimination

iii) Decision Tree Induction

Data Reduction: Dimensionality reduction

Stepwise forward selection:

- The procedure starts with an empty set of attributes as the reduced set.
- **First:** The best single-feature is picked.
- **Next:** At each subsequent iteration or step, the best of the remaining original attributes is added to the set

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:
$\{A_1, A_4, A_6\}$

Data Reduction: Dimensionality reduction

ii) Stepwise backward selection:

- The procedure starts with the full set of attributes.

– At each step, it removes the worst attribute remaining in the set.

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:
$\{A_1, A_4, A_6\}$

Data Reduction: Dimensionality reduction

iii) Decision tree induction:

- Decision tree algorithms, such as ID3, C4.5, and CART, were originally intended for classification.

- Decision tree induction constructs a flowchart-like structure where each internal (nonleaf) node denotes a test on an attribute, each branch corresponds to an outcome of the test, and each external (leaf) node denotes a class prediction.

- At each node, the algorithm chooses the "best" attribute to partition the data into individual classes.

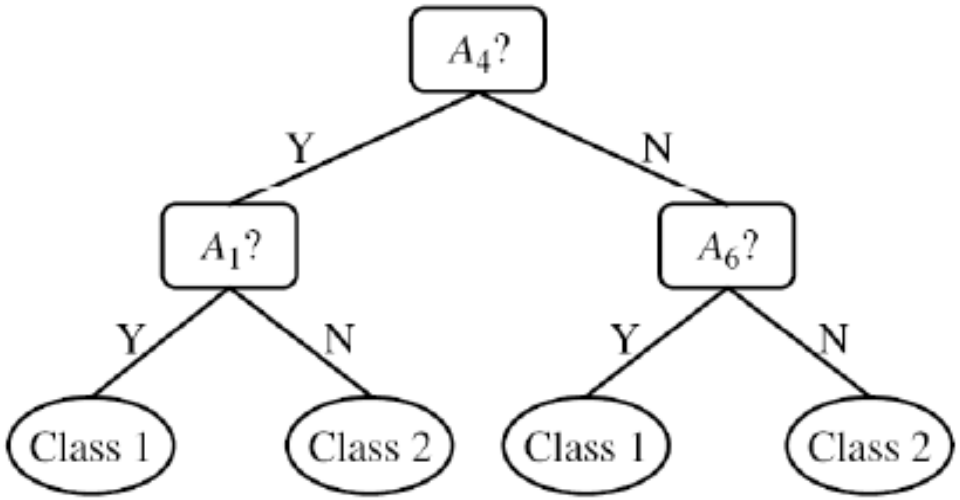
- When decision tree induction is used for attribute subset selection, a tree is constructed from the given data.

- All attributes that do not appear in the tree are assumed to be irrelevant.

Data Reduction : Dimensionality reduction

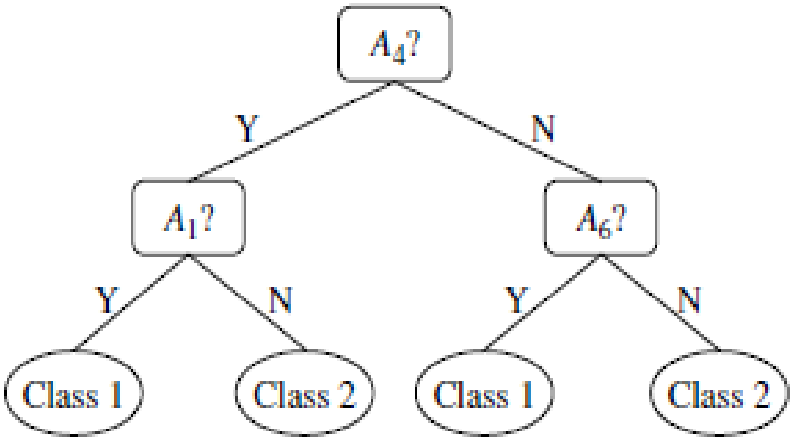
Decision tree induction

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: Dimensionality reduction

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: $\{\}$ $\Rightarrow \{A_1\}$ $\Rightarrow \{A_1, A_4\}$ $\Rightarrow$ Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ $\Rightarrow \{A_1, A_3, A_4, A_5, A_6\}$ $\Rightarrow \{A_1, A_4, A_5, A_6\}$ $\Rightarrow$ Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$  $\Rightarrow$ Reduced attribute set: $\{A_1, A_4, A_6\}$

Data Reduction : Numerosity reduction

2.**Numerosity reduction:** The data volume is decreased by selecting an alternative, smaller form of data representation. These techniques can be parametric or non-parametric.

i)Parametric method: Used to estimate the data, so that only parameters of data are required to be stored, instead of the actual data,

for example, Regression and Log-linear models.

(a)**Regression:** the data are modeled to fit a straight line. Y (called a),

$y=ax+b$ *(y is response variable and x is a predictor variable)*

- Multiple linear regression(with 2 or more predictor variables)

Data Reduction : Numerosity reduction

(b)Log-linear models: used to estimate the probability of each data point in a multidimensional space for a set of discretized attributes , based on a smaller subset of dimensional combinations.

$$\log(y)=ax+b$$

This allows a higher-dimensional data space to be constructed from lower-dimensional attributes..

Data Reduction : Numerosity reduction

ii) Non-parametric method: used to store a reduced representation of the data. It includes

a) Histogram

b) Clustering

c) Sampling

d) Data Cube Aggregation

Data Reduction : Numerosity reduction

a) Histogram: Histograms use binning to approximate data distributions

The following data are a list of *AllElectronics prices for commonly sold*

- items (rounded to the nearest dollar). The numbers have been sorted: 1, 1, 5, 5, 5,5, 5, 8, 8, 10, 10, 10, 10, 12, 14, 14, 14, 15, 15, 15, 15, 15, 15, 18, 18, 18, 18, 18, 18, 18, 18, 20, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 25, 25, 25, 25, 25, 28, 28, 30, 30, 30.

Data Reduction : Numerosity reduction

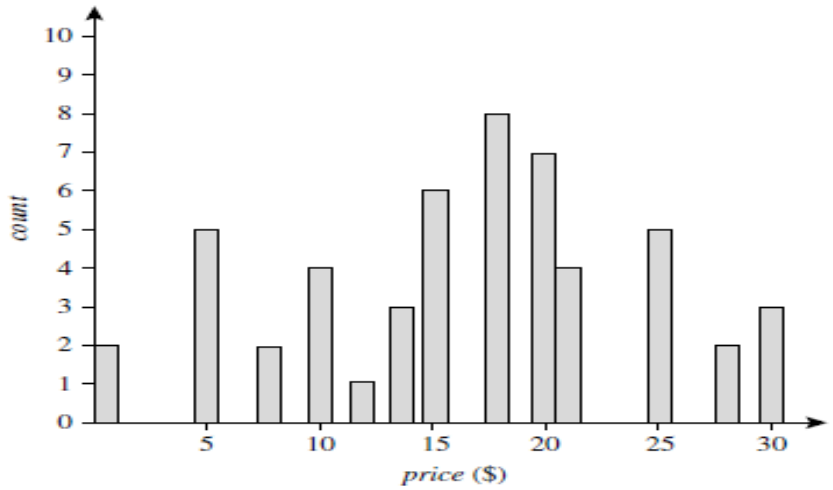


Figure 3.7 A histogram for *price* using singleton buckets—each bucket represents one price–value/frequency pair.

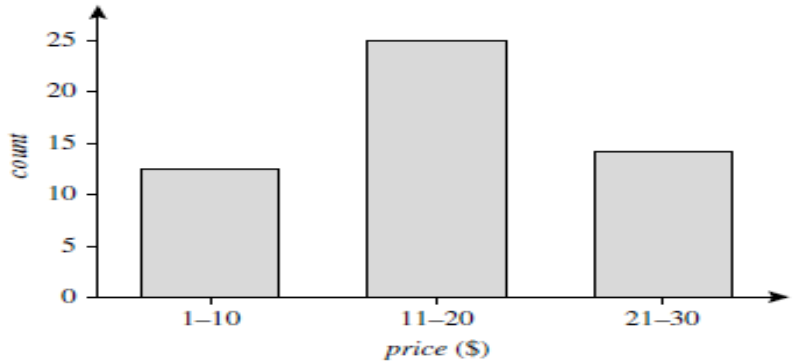
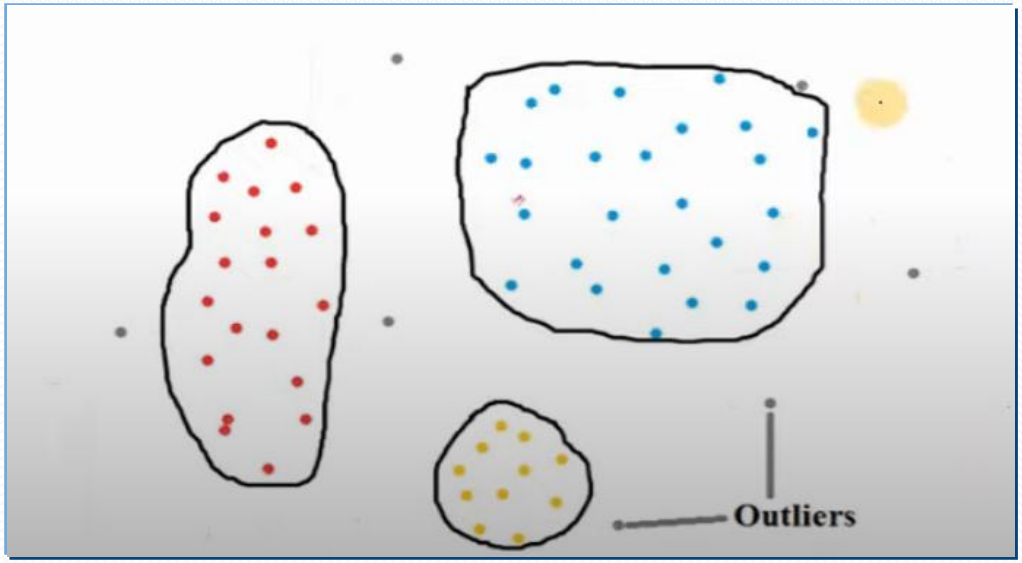


Figure 3.8 An equal-width histogram for *price*, where values are aggregated so that each bucket has a uniform width of $10.

Data Reduction : Numerosity reduction

b) **Clustering:** partition the objects into **groups, or *clust*ers**, *so that objects within a cluster are "similar" to one another and "dissimilar"* to objects in other clusters.

Centroid distance is an alternative measure of **cluster quality** and is defined as the average distance of each cluster object from the cluster centroid.



Data Reduction : Numerosity reduction

c) Sampling: can be used as a data reduction technique since it allows a large data set to be represented by a much smaller random sample (or subset) of the data.

- Suppose that a large data set, D, contains N instances.

- The most common ways that we could sample D for data reduction:

  -Simple random sample without replacement (SRSWOR)

  -Simple random sample with replacement (SRSWR)

  -Cluster sample

  -Stratified sample

Data Reduction : Numerosity reduction

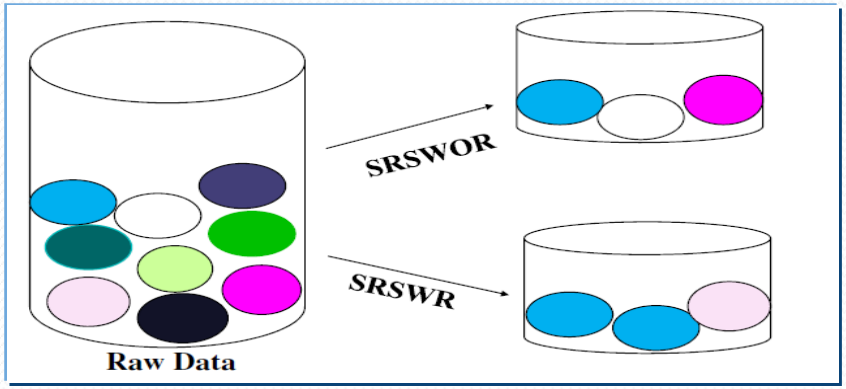
- **Simple random sample without replacement (SRSWOR) of size n:**

  This is created by drawing n of the N tuples from D (n < N), where the probably of drawing any tuple in D is 1=N, i.e., all tuples are equally likely

- **Simple random sample with replacement (SRSWR) of size n:**

  - This is similar to SRSWOR, except that each time a tuple is drawn from D, it is recorded and then replaced.

  - That is, after a instance is drawn, it is placed back in D so that it may be drawn again.

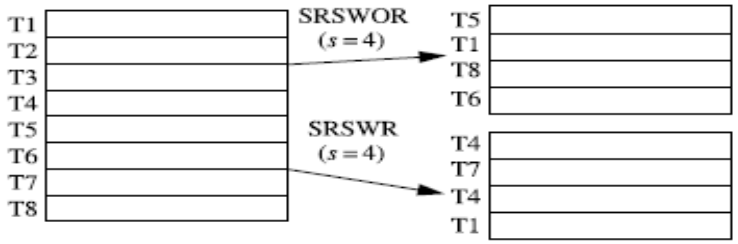


Data Reduction : Numerosity reduction

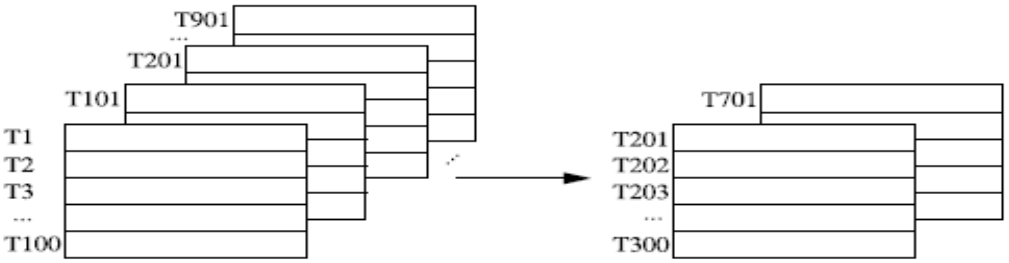
- **Cluster sample:** If the tuples in D are grouped into M mutually disjoint —clusters", then a SRS of m clusters can be obtained, where m < M. A reduced data representation can be obtained by applying, say, SRSWOR to the pages, resulting in a cluster sample of the tuples.

- **Stratified sample:** If D is divided into mutually disjoint parts called —strata", a stratified sample of D is generated by obtaining a SRS at each stratum. This helps to ensure a representative sample, especially when the data are skewed. For example, a stratified sample may be obtained from customer data, where stratum is created for each customer age group.

Data Reduction : Numerosity reduction



Cluster sample
$(s = 2)$



Startified sample
(according to *age*)

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

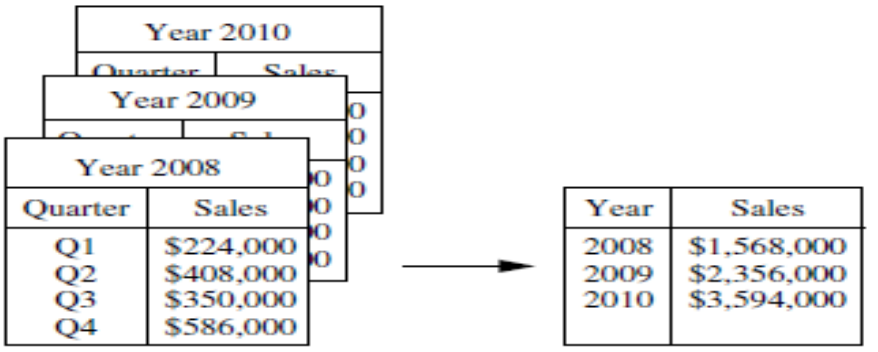
T38	youth
T391	youth
T117	middle_aged
T138	middle_aged
T290	middle_aged
T326	middle_aged
T69	senior

Sampling can be used for data reduction.

Data Reduction: Data Cube Aggregation

- **Data reduction techniques can be applied to** obtain a reduced representation of the data set that is much smaller in volume, yet closely maintains the integrity of the original data.

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



Sales data for a given branch of *AllElectronics* 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 Reduction : Data Compression

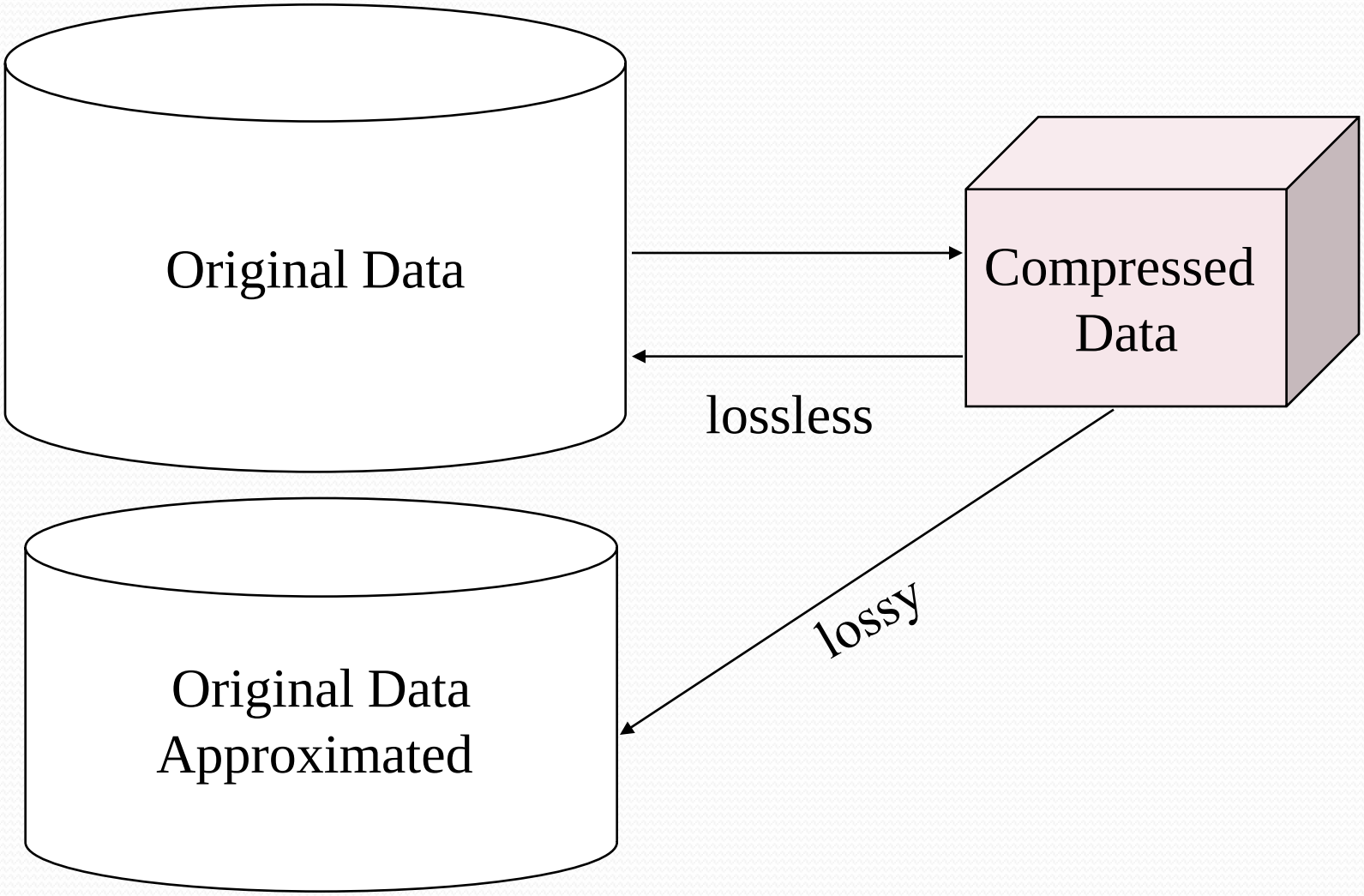
Data Compression:

In data compression, transformations are applied so as to obtain a reduced or "compressed" representation of the original data.

i) **Lossless :** compressed data without any information loss.

ii) Lossy: The decompressed data may differ to the original data but are useful enough to retrieve information from them.

Data Reduction : Data Compression



Data Transformation :

- Transforming or Consolidating data into alternate forms appropriate for mining.

- Data Transformation strategies:
  1. Smoothing
  2. Attribute/feature construction
  3. Aggregation
  4. Normalization
  5. Discretization
  6. Concept hierarchy generation for nominal data

Data Transformation

1. **Smoothing:**

   which works to remove noise from data. Techniques include binning, clustering, regression.*(Data Cleaning)*

2. **Attribute construction (or feature construction):**

   New attributes are constructed and added from the given of attributes to help the mining process. Where new attributes can be created and added to given set of attributes to simplify the mining process more efficient.

3. **Aggregation:**

   where summary or aggregation operations are applied to the data. **For example**, daily Sales data may be aggregated to compute monthly & annual total amounts.*(Data reduction)*

Data Transformation

4. **Normalization:** Attribute values are normalized by scaling their values. So that they fall within a specified range such as,

-1.0 to 1.0 (or)
0.0 to 1.0.

Normalizing the data attempts to give all attributes an equal weight. Normalization is particularly useful for classification algorithms involving neural networks or distance measurements such as nearest-neighbor classification and clustering.

Data Transformation :

- Data normalization involves converting all data variable into a given range.
- Techniques that are used for normalization are:
  - min-max normalization
  - z-score normalization
  - normalization by decimal scaling
- **Min-Max Normalization:** It performs a linear transformation on the original data.

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

Where,

- v is original attribute value
- v' is the new value you get after normalizing the old value.
- min_A is the minimum value of attribute
- max_A is the maximum value of attribute

Ex. Let income range $12,000 to $98,000 normalized to [0.0, 1.0]. Then $73,000 is mapped to

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

Data Transformation

- **z-score Normalization:**
  - In **z-score normalization** (or *zero-mean normalization*) the values of an attribute (A), are normalized based on the mean of A and its standard deviation.
  - A value, v, of attribute A is normalized to v' by computing.

$$v_i' = \frac{v_i - \bar{A}}{\sigma_A}$$

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

Data Transformation

- **Normalization by Decimal Scaling:**
  - It normalizes the values of an attribute by changing the position of their decimal points.

  - The number of points by which the decimal point is moved can be determined by the absolute maximum value of attribute A.

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

Example: 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, we therefore* divide each value by 1000 (i.e., *j=3) so that*

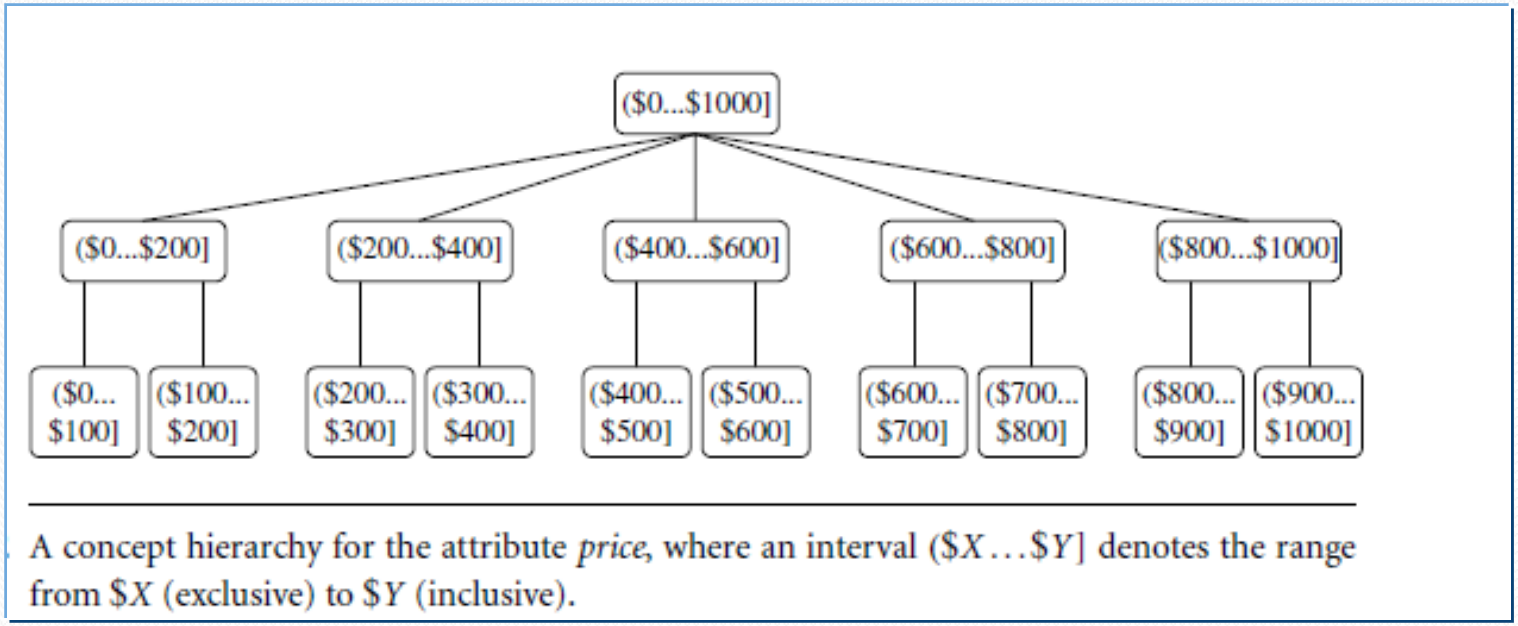
-986 normalizes to -0.986 and
917 normalizes to 0.917.

Data Transformation: Discretization

5. Discretization:

Numerical values are replaced by interval labels or conceptual labels. The labels, The data where low-level or "primitive" data are replaced by higher-level concepts through the use of concept hierarchies.

- interval labels (e.g.,0-10,11-20, etc.) or
- conceptual labels (e.g.,youth,adult,senior)



A concept hierarchy for the attribute *price*, where an interval ($X...$Y] denotes the range from $X (exclusive) to $Y (inclusive).

Data Transformation: Discretization

- Discretization techniques can be categorized based on whether it uses class information, as:

– **Supervised discretization**

the discretization process uses class information

– **Unsupervised discretization**

the discretization process does not use class information

Data Transformation: Discretization

- **Discretization techniques can be categorized based on which direction it proceeds, as:**
- **Top-down**

  If the process starts by first finding one or a few points (called split points or cut points) to split the entire attribute range, and then repeats this recursively on the resulting intervals Data Discretization and Concept Hierarchy Generation.

- **Bottom-up**

  - starts by considering all of the continuous values as potential split-points,
  - removes some by merging neighborhood values to form intervals, and
  - then recursively applies this process to the resulting intervals.

Data Transformation: Discretization

- 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)

- All the methods can be applied recursively.
- Each method assumes that the values to be discretized are sorted in ascending order.

Data Transformation: Discretization

Binning

- Binning is a top-down splitting technique based on a specified number of bins.

- Binning is an unsupervised discretization technique because it does not use class information.

- In this, The sorted values are distributed into several buckets or bins and then replaced with each bin value by the bin mean or median.

- It is further classified into

- ***Equal-width (distance) partitioning***
- ***Equal-depth (frequency) partitioning***

Data Transformation: Discretization

Histogram Analysis

- It is an unsupervised discretization technique because histogram analysis does not use class information.

- Histograms partition the values for an attribute into disjoint ranges called buckets.

- It is also further classified into
  - ***Equal-width histogram***
  - ***Equal frequency histogram***

- The histogram analysis algorithm can be applied recursively to each partition to automatically generate a multilevel concept hierarchy, with the procedure terminating once a pre-specified number of concept levels has been reached.

Data Transformation: Discretization

Cluster Analysis

- Cluster analysis is a popular data discretization method.

- A clustering algorithm can be applied to discretize a numerical attribute of A by partitioning the values of A into clusters or groups.

- Clustering considers the distribution of A, as well as the closeness of data points, and therefore can produce high-quality discretization results.

- Each initial cluster or partition may be further decomposed into several subcultures, forming a lower level of the hierarchy.

Data Transformation: Discretization

- **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

Data Transformation : Concept hierarchy generation for nominal data

6. Concept hierarchy generation for nominal data:

Attributes of low-level concepts can be generalized to higher-level concepts.

Example: Street can be generalized to higher-level concept city or country.

- **There are several methods for the generation of concept hierarchies for categorical data:**

– Specification of a partial ordering of attributes explicitly at the schema level by users or experts.

– Specification of a portion of a hierarchy by explicit data grouping Data Discretization and Concept Hierarchy Generation.

– Specification of a set of attributes, but not of their partial ordering.

Data Transformation : Concept hierarchy generation for nominal data

- **Specification of a partial ordering of attributes explicitly at the schema level by users or experts**

– **Example:** a relational database or a dimension location of a data warehouse may contain the following group of attributes: street, city, province or state, and country.

– A user or expert can easily define a concept hierarchy by Data Discretization and Concept Hierarchy Generation specifying ordering of the attributes at the schema level.

– A hierarchy can be defined by specifying the total ordering among these attributes at the schema level, such as:

street < city < province or state < country

Data Transformation : Concept hierarchy generation for nominal data

- **Specification of a portion of a hierarchy by explicit data grouping**

– we can easily specify explicit groupings for a small portion of intermediate-level data.

– For example, after specifying that province and country form a hierarchy at the schema level, a user could define Data Discretization and Concept Hierarchy Generation some intermediate levels manually, such as:

{Urbana, Champaign, Chicago} < Illinois

Data Transformation : Concept hierarchy generation for nominal data

- **Specification of a set of attributes, but not of their partial ordering**

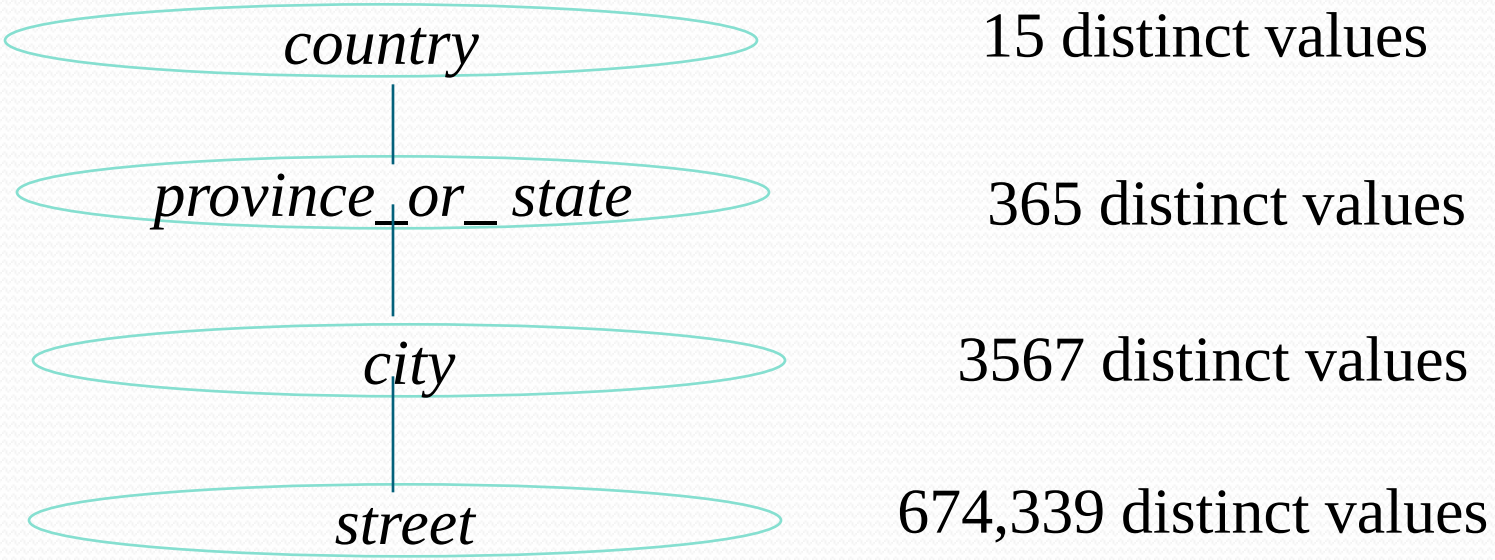
– A user may specify a set of attributes forming a concept hierarchy, but omit to explicitly state their partial ordering.

– The system can then try to automatically generate the attribute ordering so as to construct a meaningful concept Data Discretization and Concept Hierarchy Generation hierarchy.

– **Example:** Suppose a user selects a set of location-oriented attributes, *street, country, province_or_state, and city*, from the *AllElectronics* database, but does not specify the hierarchical ordering among the attributes.

Data Transformation : Concept hierarchy generation for nominal data

- 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



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

- **Reference:**

Data Mining Concepts and Techniques, Jiawei Han, Micheline Kamber, Jian Pei,3rd Edition, Morgan Kaufmann Publishers