Unit – II

Data Preprocessing

- An Overview
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation and Data Discretization

AN OVERVIEW

Why Preprocessing?

Data have quality if they satisfy the requirements of the intended use. There are many factors comprising **data quality**, including *accuracy, completeness, consistency, timeliness, believability*, and *interpretability*.

Major Tasks in Data Preprocessing

- ✓ Data Cleaning
- ✓ Data Integration
- ✓ Data Reduction
- ✓ Data Transformation

DATA CLEANING

- ✓ Real-world data tend to be incomplete, noisy, and inconsistent
- ✓ *Data cleaning* attempt to fill in Missing Values, Smooth out Noise while identifying outliers, and correct inconsistencies in the data

Handling Missing Values

1. Ignore the tuple
   - This is usually done when the class label is missing.
   - It is not very effective, unless the tuple contains several attributes with missing values.
2. Fill in the missing value manually
   - It is time consuming and not feasible when there are many missing values
3. Use a global constant to fill in the missing value
4. Use a measure of central tendency for the attribute
5. Use the attribute mean or median for all samples belonging to the same class as the given tuple
6. Use the most probable value to fill in the missing value

Handling Noisy Data

Noise is a random error or variance in a measured variable. Techniques used to remove the noise:

1) **Binning:**

Binning methods smooth a sorted data value by consulting its "neighbour hood" values around it.

Stored values are distributed into a number of buckets (or) bins.

Sorted data for price (in dollars): 48,15,21,21,24,25,28,34

Partition into equal-frequency bins of size 3:

Bin1: 4,8,15

Bin2: 21,21,24

Bin3: 25,28,34

Smoothing by bin means:

Bin1: 9,9,9

Bin2: 22,22,22

Bin3: 29,29,29

Smoothing by bin medians:

Bin1: 8,8,8

Bin2: 21,21,21

Bin3: 28,28,28

Smoothing by bin boundaries:

Bin1: 4,4,4

Bin2: 21,21,24

Bin3: 25,25,34

2) **Regression:**

Data can be smoothed by fitting the data to a function, such as with linear regression involves finding the best line to fit two attributes.

Multiple linear regression is an extension, where more than two attributes are involved and the data are fit to a multidimensional surface.

3) **Clustering:**

Outliers may be detected by clustering, where similar values are organized into groups or clusters. The values fall outside of the set of clusters may be considered outliers.

Ex: Data cleaning needs for poorly designed data entry forms, human error in data entry, data decay (out dated addresses), system errors.

DATA CLEANING AS A PROCESS

- ✓ The first step in data cleaning as a process is *discrepancy detection*
- ✓ Discrepancies can be caused by several factors
  - Poorly designed data entry forms
  - Human error in data entry

- Data decay (e.g., outdated addresses)
- Inconsistent data representations
- Errors can also be inconsistencies due to data integration

HOW TO EXAMINE DATA?

- ✓ A **unique rule** says that each value of the given attribute must be different from all other values for that attribute.
- ✓ A **consecutive rule** says that there can be no missing values between the lowest and highest values for the attribute, and that all values must also be unique (e.g., as in check numbers).
- ✓ A **null rule** specifies the use of blanks, question marks, special characters, or other strings that may indicate the null condition

COMMERCIAL TOOLS THAT CAN AID IN THE DISCREPANCY DETECTION

- ✓ **Data scrubbing tools** use simple domain knowledge (e.g., knowledge of postal addresses and spell-checking) to detect errors and make corrections in the data. These tools rely on parsing and fuzzy matching techniques when cleaning data from multiple sources.
- ✓ **Data auditing tools** find discrepancies by analyzing the data to discover rules and relationships, and detecting data that violate such conditions. They are variants of data mining tools. For example, they may employ statistical analysis to find correlations, or clustering to identify outliers.

DATA INTEGRATION

- ✓ Data mining often requires data integration—the merging of data from multiple data stores
- ✓ Reduces and avoids redundancies and inconsistencies
- ✓ Improves the accuracy

Problems in Data Integration

- ✓ Entity Identification Problem
- ✓ Redundancy and Correlation Analysis
- ✓ Tuple Duplication
- ✓ Data Value Conflict Detection and Resolution

Entity Identification Problem

Shcema integration and object matching

How can equivalent real-world entities from multiple data sources be matched up? -> this is called as " entity – identification problem".

Ex: Attributes with different names in different sources of a warehouse.

Redundancy and Correlation Analysis

An attribute may be redundant if it can be derived from another attribute (or) set of attributes.

- ✓ For **nominal data**, we use the **χ2 (*Chi-Square*) Test**
- ✓ For **numeric attributes**, we can use the ***Correlation Coefficient*** and ***Covariance***

For Nominal Attribute

χ2 (Chi-square) Test (Pearson χ2 Statistic)

A has c distinct values, namely $a_1, a_2, \ldots, a_c$
B has r distinct values, namely $b_1, b_2, \ldots, b_r$
Let (A_i, B_j) denote the joint event that attribute A takes on value a_i and attribute B takes on value b_j

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

where o_{ij} is the observed frequency (i.e., actual count) of the joint event (A_i, B_j) and e_{ij} is the expected frequency of (A_i, B_j), which can be computed as

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

where *n* is the number of data tuples

- χ2 (*Chi-square*) Test is based on significant level with respect to degree of freedom (r-1) x (c-1)

	male	female	Total
fiction	250 (90)	200 (360)	450
non_fiction	50 (210)	1000 (840)	1050
Total	300	1200	1500

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

- The degrees of freedom are (2-1) x (2-1) = 1
- For 1 degree of freedom, the χ2 value needed to reject the hypothesis at the 0.001 significance level is 10.828
- Since our computed value is above this, we can reject the hypothesis that *gender* and *preferred reading* are independent

For Numeric Data
Correlation Analysis
Redundancies can be detected by correlation analysis_-> how strongly one attribute implies another.
For **Numerical Attributes A&B**, Correlation Coefficient

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

N-> No. of tuples

a_i and b_i-> respective values of Attributes A and B in tuple i,

$\overline{A}$ and $\overline{B}$-> respective mean values of A & B

σ_A ans σ_B -> respective standard deviations of A & B

S(*aibi*)-> sum of the AB cross-product i.e; for each tuple, the value for *A* is multiplied by the value for *B* in that tuple

If $\gamma_{A,B}$ value:
= 0, then *A* & *B* are *independent*
> 0, then A & B are positively correlated; either A or B can be removed
<0, then then A & B are negatively correlated; each attribute discourages the other

Similarly **Covariance**,

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

Tuple Duplication

- In addition to detecting redundancies between attributes, duplication should also be detected at the tuple level
- The use of denormalized tables 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.
- To overcome this data tables must be normalized using normalization

Data Value Conflict Detection & Resolution

- For example, for the same real-world entity, attribute values from different sources may differ. This may be due to differences in representation, scaling, or encoding. All such attributes must be brought into common domain space.

DATA REDUCTION

- ✓ **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.
- ✓ Data reduction strategies include *Dimensionality Reduction*, *Numerosity Reduction* and *Data compression*.

Dimensionality Reduction

- ✓ **Dimensionality Reduction** is the process of reducing the number of random variables or attributes

- ✓ Methods:
  - *Wavelet Transforms*
  - *Principal Components Analysis*
  - *Attribute Subset Selection*

Wavelet Transforms

- ✓ Discrete wavelet transform (DWT) is a linear signal processing technique that, when applied to a data vector *X*, transforms it to a numerically different vector, *X'*, of Wavelet Coefficients.
- ✓ Wavelet transformed data are truncated
- ✓ Wavelet coefficients larger than some user-specified threshold can be retained. All other coefficients are set to 0
- ✓ Resulting data representation is therefore very sparse, so that operations that can take advantage of data sparsity are computationally very fast
- ✓ Popular wavelet transforms include the Haar-2, Daubechies-4, and Daubechies-6.
- ✓ The general procedure for applying a discrete wavelet transform uses a hierarchical *pyramid algorithm* that halves the data at each iteration. The method is as follows:
  - Input Data Set Length, L, (in power of 2; pad zeros if necessary)
  - Apply two functions to pairs of Input Data Vector. (a) Data Smoothing (b) Weighted Difference
  - This results in two data sets of length *L*/2
  - The above procedure is applied recursively until the input data vectors become of size 2

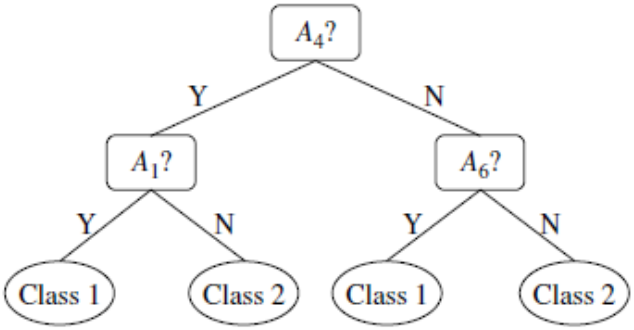
Principal Components Analysis

- ✓ The input data are normalized, so that each attribute falls within the same range. This step helps ensure that attributes with large domains will not dominate attributes with smaller domains.
- ✓ PCA computes *k* orthonormal (k<n) vectors that provide a basis for the normalized input data. These are unit vectors that each point in a direction perpendicular to the others, called as *principal components*. The input data are a linear combination of the principal components.
- ✓ The principal components are sorted in order of decreasing "significance" or strength. The principal components essentially serve as a new set of axes for the data
- ✓ Because the components are sorted in decreasing order of "significance," the data size can be reduced by eliminating the weaker components, that is, those with low variance. Using the strongest principal components, it should be possible to reconstruct a good approximation of the original data.
- ✓ PCA can be applied to ordered and unordered attributes, and can handle sparse data and skewed data.
- ✓ PCA tends to be better at handling sparse data, whereas wavelet transforms are more suitable for data of high dimensionality

Attribute Subset Selection

- ✓ Attribute subset selection reduces the data set size by removing irrelevant or redundant attributes (or dimensions)
- ✓ Heuristic Methods
  1. Stepwise forward selection
  2. Stepwise backward elimination
  3. Combination of forward selection and backward elimination
  4. Decision tree induction

1. **Stepwise forward selection**: The procedure starts with an empty set of attributes as the reduced set. The best of the original attributes is determined and added to the reduced set. At each subsequent iteration or step, the best of the remaining original attributes is added to the set.
2. **Stepwise backward elimination**: The procedure starts with the full set of attributes. At each step, it removes the worst attribute remaining in the set.
3. **Combination of forward selection and backward elimination**: The stepwise forward selection and backward elimination methods can be combined so that, at each step, the procedure selects the best attribute and removes the worst from among the remaining attributes.
4. **Decision tree induction**: Decision tree algorithms (e.g., ID3, C4.5, and CART) were originally intended for classification. Decision tree induction constructs a flow chart like structure where each internal (non-leaf) 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. The set of attributes appearing in the tree form the reduced subset of attributes.

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

Numerosity Reduction

- ✓ Replaces the original data volume by alternative, smaller forms of data representation.
- ✓ *Parametric methods & Non-parametric methods*
- ✓ *Parametric methods*
  - Regression and log-linear models
- ✓ *Non-parametric methods*
  - *Histograms*
  - *Clustering*
  - *Sampling*
  - *Data cube aggregation*

Parametric methods: Regression and log-linear models

Regression and log-linear models can be used to approximate the given data.

In (simple) **linear regression**, the data are modeled to fit a straight line. For example, a random variable, *y* (called a *response variable*), can be modeled as a linear function of another random variable, *x* (called a *predictor variable*), with the equation

$$y = wx + b$$

where the variance of *y* is assumed to be constant. In the context of data mining, *x* and *y* are numeric database attributes. The coefficients, *w* and *b* (called *regression coefficients*), specify the slope of the line and the *y*-intercept. These coefficients can be solved for by the *method of least squares*, which minimizes the error between the actual line separating the data and the estimate of the line.

Log-linear models approximate discrete multidimensional probability distributions. Given a set of tuples in *n* dimensions (e.g., described by *n* attributes), we can consider each tuple as a point in an *n*-dimensional space. Log-linear models can be used to estimate the probability of each point in a multidimensional space for a set of discretized attributes, based on a smaller subset of dimensional combinations. This allows a higher-dimensional data space to be constructed from lower-dimensional spaces. Log-linear models are therefore also useful for dimensionality reduction

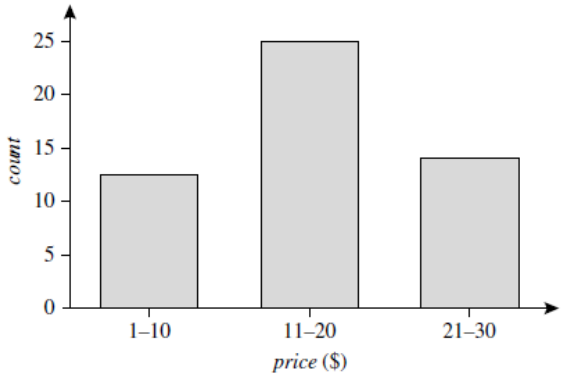
Non-parametric methods

Histograms

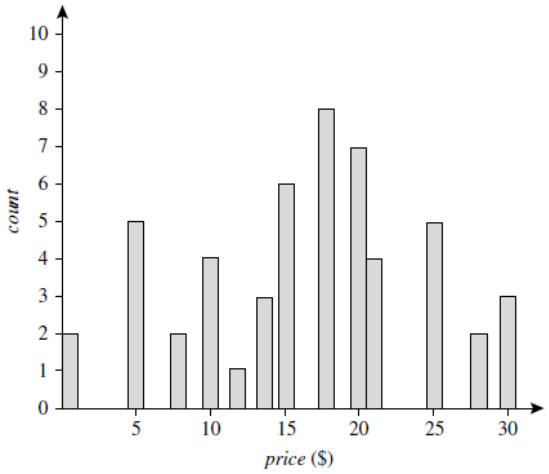
Histograms use binning to approximate data distributions and are a popular form of data reduction. A **histogram** for an attribute, *A*, partitions the data distribution of *A* into disjoint subsets, referred to as *buckets* or *bins*. If each bucket represents only a single attribute–value/frequency pair, the buckets are called *singleton buckets*. Often, buckets instead represent continuous ranges for the given attribute.

Bucket partitioning rules, including the following:

- **Equal-width**: In an equal-width histogram, the width of each bucket range is uniform
- **Equal-frequency** (or equal-depth): In an equal-frequency histogram, the buckets are created so that, roughly, the frequency of each bucket is constant (i.e., each bucket contains roughly the same number of contiguous data samples).



An equal-width histogram for *price*



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

Note: Singleton buckets are useful for storing high-frequency outliers.

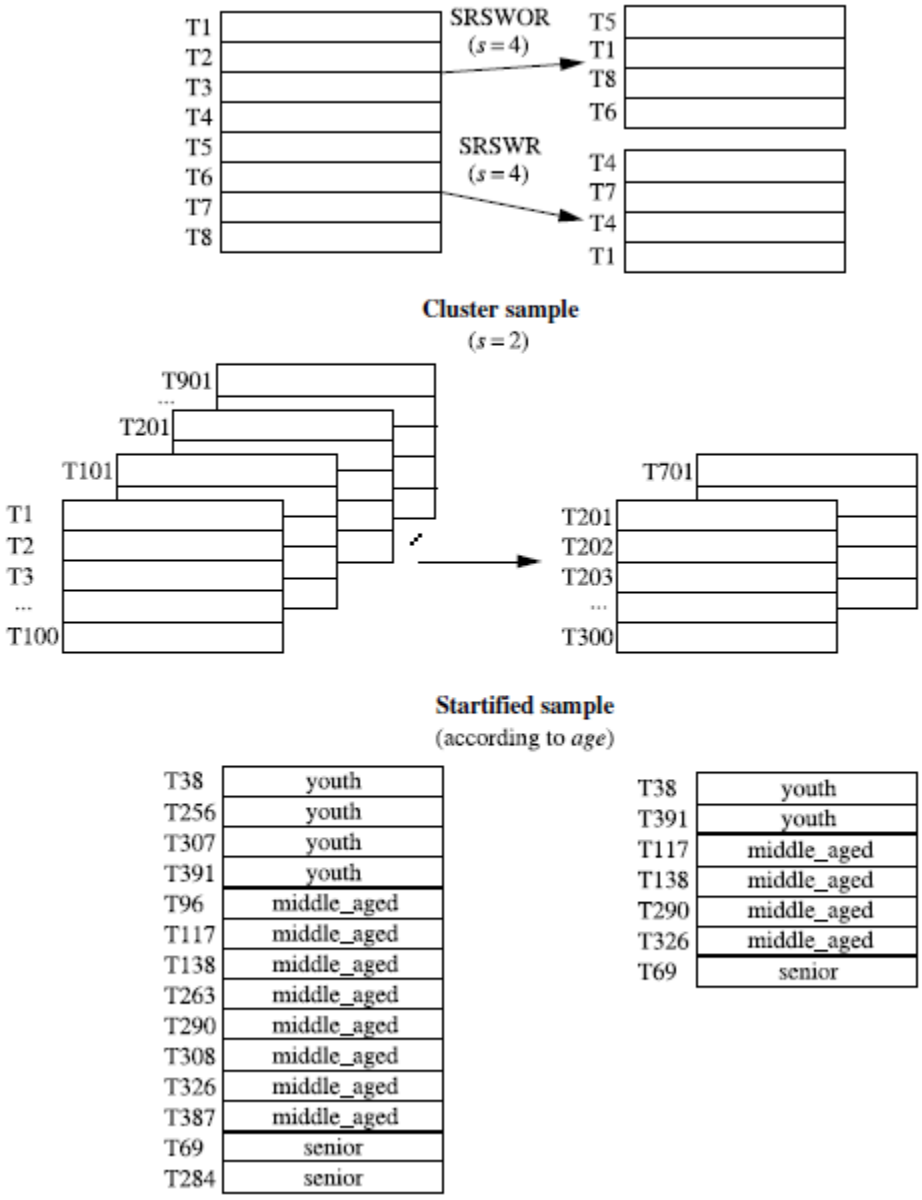
Clustering

Clustering techniques consider data tuples as objects. They partition the objects into groups, or *clusters*, so that objects within a cluster are "similar" to one another and "dissimilar" to objects in other clusters. Similarity is commonly defined in terms of how "close" the objects are in space, based on a distance function.

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

Sampling

Sampling can be used as a data reduction technique because it allows a large data set to be represented by a much smaller random data sample (or subset). Suppose that a large data set, *D*, contains *N* tuples. Let's look at the most common ways that we could sample *D* for data reduction.



Simple random sample without replacement (SRSWOR) of size *s*: This is created by drawing *s* of the *N* tuples from *D* (*s* < *N*), where the probability of drawing any tuple in *D* is 1/*N*, that is, all tuples are equally likely to be sampled.

Simple random sample with replacement (SRSWR) of size *s*: 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 tuple is drawn, it is placed back in *D* so that it may be drawn again.

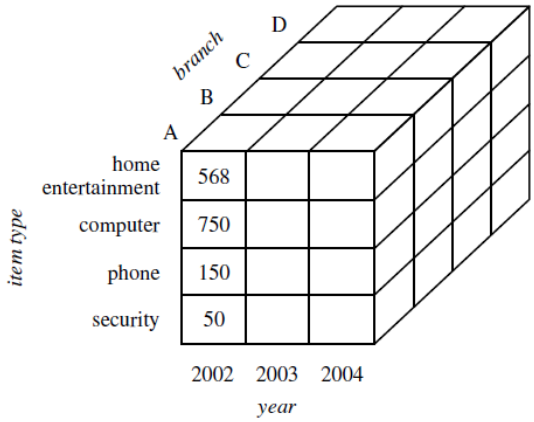
Cluster sample: If the tuples in *D* are grouped into *M* mutually disjoint "clusters," then an SRS of *s* clusters can be obtained, where *s* < *M*. For example, tuples in a database are usually retrieved a page at a time, so that each page can be considered a cluster. A reduced data representation 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 an SRS at each stratum. This helps ensure a representative sample, especially when the data are

skewed. For example, a stratified sample may be obtained from customer data, where a stratum is created for each customer age group.

Data cube Aggregation

Data can be aggregated from Quarters to years in data cubes.



Concept hierarchies used for performing the analysis at multiple levels of abstraction

DATA TRANSFORMATION AND DATA DISCRETIZATION

In *data transformation*, the data are transformed or consolidated into forms appropriate for mining. Strategies for data transformation include the following:

1. Smoothing, which works to remove noise from the data. Techniques include binning, regression, and clustering.

2. Attribute construction (or *feature construction*), where new attributes are constructed and added from the given set of attributes to help the mining process.

3. Aggregation, where summary or aggregation operations are applied to the data. For example, the daily sales data may be aggregated so as to compute monthly and annual total amounts. This step is typically used in constructing a data cube for data analysis at multiple abstraction levels.

4. Normalization, where the attribute data are scaled so as to fall within a smaller range, such as -1.0 to 1.0, or 0.0 to 1.0.

5. Discretization, where 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*). The labels, in turn, can be recursively organized into higher-level concepts, resulting in a *concept hierarchy* for the numeric attribute.

6. Concept hierarchy generation for nominal data, where attributes such as *street* can be generalized to higher-level concepts, like *city* or *country*. Many hierarchies for nominal attributes are implicit within the database schema and can be automatically defined at the schema definition level.

Discretization techniques can be categorized based on how the discretization is performed, such as whether it uses class information or which direction it proceeds (i.e., top-down vs. bottom-up). If the discretization process uses class information, then we say it is *supervised discretization*. Otherwise, it is *unsupervised*. 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, it is called *top-down discretization* or *splitting*. This

contrasts with *bottom-up discretization* or *merging*, which 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 discretization and concept hierarchy generation are also forms of data reduction. The raw data are replaced by a smaller number of interval or concept labels. This simplifies the original data and makes the mining more efficient. The resulting patterns mined are typically easier to understand. Concept hierarchies are also useful for mining at multiple abstraction levels.

Data Transformation by Normalization

- ✓ Involves transforming the data to fall within a smaller or common range such as [-1, 1] or [0.0, 1.0]
- ✓ Methods:
  1. Min-Max Normalization
  2. Z-Score Normalization
  3. Normalization by Decimal Scaling

Min-max normalization performs a linear transformation on the original data. Suppose that min_A and max_A are the minimum and maximum values of an attribute, A. Min-max normalization maps a value, v_i , of A to v_i' in the range [new_min_A, new_max_A] by computing,

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

Z-score normalization (or *zero-mean normalization*), the values for an attribute, *A*, are normalized based on the mean (i.e., average) and standard deviation of *A*. A value, v_i , of *A* is normalized to v_i' by computing,

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

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_i , of *A* is normalized to v_i' by computing,

$$v_i' = \frac{v_i}{10^j}$$

where j is the smallest integer such that $max(|v_i'|) < 1$

Discretization by Binning

Binning is a top-down splitting technique based on a specified number of bins. These methods are also used as discretization methods for data reduction and concept hierarchy generation.

Binning does not use class information and is therefore an unsupervised discretization technique.

Discretization by Histogram Analysis

Histogram analysis is an unsupervised discretization technique because it does not use class information. A histogram partitions the values of an attribute, *A*, into disjoint ranges called *buckets* or *bins*.

Various partitioning rules can be used to define histograms: *equal-width* histogram, for example, the values are partitioned into equal-size partitions or ranges and *equal-frequency* histogram; the values are partitioned so that, ideally, each partition contains the same number of data tuples. The histogram analysis algorithm can be applied recursively to each partition in order to automatically generate a multilevel concept hierarchy.

Discretization by Cluster, Decision Tree, and Correlation Analyses

Cluster analysis is a popular data discretization method. A clustering algorithm can be applied to discretize a numeric attribute, *A*, by partitioning the values of *A* into clusters or groups. Clustering takes the distribution of *A* into consideration, as well as the closeness of data points, and therefore is able to produce high-quality discretization results. Clustering can be used to generate a concept hierarchy for *A* by following either a top-down splitting strategy or a bottom-up merging strategy, where each cluster forms a node of the concept hierarchy.

Decision trees for classification can be applied to discretization. Such techniques employ a top-down splitting approach. Unlike the other methods mentioned so far, decision tree approaches to discretization are supervised, that is, they make use of class label information. *Entropy* is the most commonly used measure for this purpose.

Measures of correlation can be used for discretization. *ChiMerge* is a **χ2** based discretization method. ChiMerge, which employs a bottom-up approach by finding the best neighboring intervals and then merging them to form larger intervals, recursively. As with decision tree analysis, ChiMerge is supervised in that it uses class information. **χ2** tests are performed for every pair of adjacent intervals. Adjacent intervals with the least **χ2** values are merged together, because low **χ2** values for a pair indicate similar class distributions.

Concept Hierarchy Generation for Nominal Data

- ✓ Data transformation for nominal data.
- ✓ Nominal attributes have a finite number of distinct values, with no ordering among the values. Examples include *geographic_location*, *job_category*, and *item_type*.
- ✓ Manual definition of concept hierarchies can be a tedious and time-consuming task for a user or a domain expert.
- ✓ Four methods for the generation of concept hierarchies for nominal data, as follows:
    1. *Specification of a partial ordering of attributes explicitly at the schema level by users or experts:*

       Concept hierarchies for nominal attributes or dimensions typically involve a group of attributes. A user or expert can easily define a concept hierarchy by specifying a partial or total ordering of the attributes at the schema level. For example, suppose that a relational database contains the following group of attributes: *street, city, province or state*, and *country*. Similarly, a data warehouse *location* dimension may contain the same attributes. 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*.

2. *Specification of a portion of a hierarchy by explicit data grouping:*
   This is essentially the manual definition of a portion of a concept hierarchy. In a large database, it is unrealistic to define an entire concept hierarchy by explicit value enumeration. On the contrary, 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 some intermediate levels manually, such as "{*Alberta, Saskatchewan, Manitoba*} ∈ *prairies Canada*" and "{*British Columbia, prairies Canada*} ∈ *Western Canada*."
3. *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 hierarchy.
   Consider the observation that since higher-level concepts generally cover several subordinate lower-level concepts, an attribute defining a high concept level (e.g., *country*) will usually contain a smaller number of distinct values than an attribute defining a lower concept level (e.g., *street*). Based on this observation, a concept hierarchy can be automatically generated based on the number of distinct values per attribute in the given attribute set. The attribute with the most distinct values is placed at the lowest hierarchy level. The lower the number of distinct values an attribute has, the higher it is in the generated concept hierarchy.
4. *Specification of only a partial set of attributes:*
   Sometimes a user have only a vague idea about what should be included in a hierarchy. Consequently, the user may have included only a small subset of the relevant attributes in the hierarchy specification. For example, instead of including all of the hierarchically relevant attributes for *location*, the user may have specified only *street* and *city*.