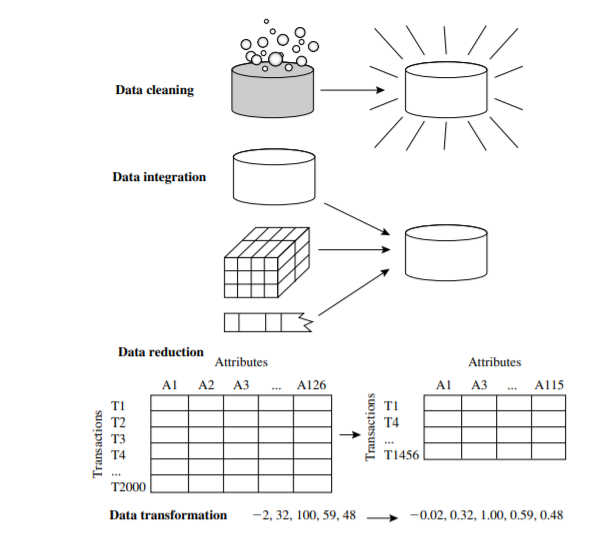
**DATA PREPROCESSING**

**Data preprocessing** is a **data mining** technique that involves transforming raw **data** into an understandable format.

**Steps Involved in Data Preprocessing:**



**Data cleaning** - clean the data by filling in missing values, smoothing noisy data, identifying or removing outliers, and resolving inconsistencies.

**Data integration**- integrating multiple databases, data cubes, or files .

**Data reduction** - a reduced representation of the data set that is much smaller in volume. Data reduction strategies include **dimensionality reduction and numerosity reduction**.

**Dimensionality reduction**-data encoding schemes are applied so as to obtain a reduced or “compressed” representation of the original data. Examples include data compression techniques (e.g., wavelet transforms and principal components analysis), attribute subset selection (e.g., removing irrelevant attributes), and attribute construction (e.g., where a small set of more useful attributes is derived from the original set).

numerosity reduction-the data are replaced by alternative, smaller representations using parametric models (e.g., regression or log-linear models) or nonparametric models (e.g., histograms, clusters, sampling, or data aggregation).

**Data transformation-** Discretization and concept hierarchy generation are powerful tools for data mining in that they allow data mining at multiple abstraction levels. Normalization, data discretization, and concept hierarchy generation.

**DATA CLEANING**

**Missing Values**

**1.Ignore the tuple-** when the class label is missing. This method is not very effective, unless the tuple contains several attributes with missing values. It is especially poor when the percentage of missing values per attribute varies considerably. By ignoring the tuple, we do not make use of the remaining attributes’ values in the tuple. Such data could have been useful to the task at hand.

**2. Fill in the missing value manually**-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 a measure of central tendency for the attribute (e.g., the mean or median) to fill in the missing value:** indicate the “middle” value of a data distribution. For normal (symmetric) data distributions, the mean can be used, while skewed data distribution should employ the median.

**5. Use the attribute mean or median for all samples belonging to the same class as the given tuple:** For example, if classifying customers according to credit risk, we may replace the missing value with the mean income value for customers in the same credit risk category as that of the given tuple. If the data distribution for a given class is skewed, the median value is a better choice.

**6. Use the most probable value to fill in the missing value:** This may be determined with regression, inference-based tools using a Bayesian formalism, or decision tree induction.

**Noisy Data**

Noise is a random error or variance in a measured variable.

**Binning:** Binning methods smooth a sorted data value by consulting its “neighborhood,” that is, the values around it. The sorted values are distributed into a number of “buckets,” or bins. Bins may be equal width, where the interval range of values in each bin is constant.

Steps:

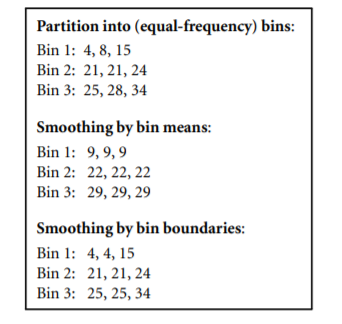
The data are first sorted and then partitioned into equal-frequency bins

**Smoothing By Bin Means-** each value in a bin is replaced by the mean value of the bin.

**Smoothing By Bin Medians**- in which each bin value is replaced by the bin median.

**Smoothing By Bin Boundaries**- the minimum and maximum values in a given bin are identified as the bin boundaries. Each bin value is then replaced by the closest boundary value.

Example: consider the Sorted data : 4, 8, 15, 21, 21, 24, 25, 28, 34



Example:

for the attribute age: 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.

1. Use smoothing by bin means, median, boundary to smooth these data, using a bin depth of 3

**BIN MEANS**

* 1. **Step 1:** Sort the data.
  2. **Step 2:** Partition the data into equidepth bins of depth 3.

|  |  |  |
| --- | --- | --- |
| Bin 1: 13, 15, 16  Bin 2: 16, 19, 20  Bin 3: 20, 21, 22 |  |  |
| Bin 4: 22, 25, 25  Bin 5: 25, 25, 30  Bin 6: 33, 33, 35 |  |  |
| Bin 7: 35, 35, 35  Bin 8: 36, 40, 45  Bin 9: 46, 52, 70 |  |  |

* 1. **Step 3:** Calculate the arithmetic mean of each bin.
  2. **Step 4:** Replace each of the values in each bin by the arithmetic mean calculated for the bin.

Bin 1: 14.7, 14.7, 14.7

Bin 2: 18, 118, 18 ……..

**BIN Median**

1. **Step 1:** Sort the data.
2. **Step 2:** Partition the data into equidepth bins of depth 3.

|  |  |  |
| --- | --- | --- |
| Bin 1: 13, 15, 16  Bin 2: 16, 19, 20  Bin 3: 20, 21, 22 |  |  |
| Bin 4: 22, 25, 25  Bin 5: 25, 25, 30  Bin 6: 33, 33, 35 |  |  |
| Bin 7: 35, 35, 35  Bin 8: 36, 40, 45  Bin 9: 46, 52, 70 |  |  |

1. **Step 3:** Calculate the median of each bin.
2. **Step 4:** Replace each of the values in each bin by the median calculated for the bin.

**BIN Boundary**

**Step 1:** Sort the data.

**Step 2:** Partition the data into equidepth bins of depth 3.

|  |  |  |
| --- | --- | --- |
| Bin 1: 13, 15, 16  Bin 2: 16, 19, 20  Bin 3: 20, 21, 22 |  |  |
| Bin 4: 22, 25, 25  Bin 5: 25, 25, 30  Bin 6: 33, 33, 35 |  |  |
| Bin 7: 35, 35, 35 |  |  |

Bin 8: 36, 40, 45

|  |  |  |
| --- | --- | --- |
| Bin 9: 46, 52, 70  **Step 4:** Replace each of the values in each bin by the closest boundary value.  Bin 1: 13, 13, 16  Bin 2: 16, 16, 20  Bin 3: 20, 20, 22…. |  |  |

**Regression:** Data smoothing can also be done by regression, a technique that conforms data values to a function. Linear regression involves finding the “best” line to fit two attributes (or variables) so that one attribute can be used to predict the other. Multiple linear regression is an extension of linear regression, where more than two attributes are involved and the data are fit to a multidimensional surface.

**Outlier analysis:** Outliers may be detected by clustering, for example, where similar values are organized into groups, or “clusters.”

**DATA INTEGRATION**

**Entity Identification Problem:** This is the **problem** of identifying object instances from different databases that correspond to the same real-world **entity**

**Redundancy and Correlation Analysis:**

**Redundancy** is another important issue in data integration. An attribute may be redundant if it can be “derived” from another attribute or set of attributes. Inconsistencies in attribute or dimension naming can also cause redundancies in the resulting data set.

Some redundancies can be detected by **correlation analysis.** Given two attributes, such analysis can measure how strongly one attribute implies the other, based on the available data. For nominal data, we use the χ 2 (chi-square) test. For numeric attributes, we can use the correlation coefficient and covariance, both of which access how one attribute’s values vary from those of another.

**χ 2 Correlation Test for Nominal Data:**

For nominal data, a correlation relationship between two attributes, A and B, can be done by a χ 2 (chi-square) test.

* Suppose A has c distinct values, namely a1,a2,...ac . B has r distinct values, namely b1,b2,...br .
* The data tuples described by A and B can be shown as a contingency table, with the c values of A making up the columns and the r values of B making up the rows.
* Let (Ai ,Bj) denote the joint event that attribute A takes on value ai and attribute B takes on value bj , that is, where (A = ai ,B = bj). Each and every possible (Ai ,Bj) joint event has its own cell .

The χ 2 value (also known as the Pearson χ 2 statistic) is computed as



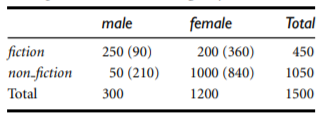
where oij is the observed frequency (i.e., actual count) of the joint event (Ai ,Bj) and eij is the expected frequency of (Ai ,Bj), which can be computed as



where n is the number of data tuples, count(A = ai) is the number of tuples having value ai for A, and count(B = bj) is the number of tuples having value bj for B

Example

Suppose that a group of 1500 people was surveyed. The gender of each person was noted. Each person was polled as to whether his or her preferred type of reading material was fiction or nonfiction. Thus, we have two attributes, gender and preferred reading. The observed frequency (or count) of each possible joint event is summarized in the contingency table shown in Table below



e11 = count(male) × count(fiction)/n =

(300 × 450) /1500 = 90

e12 = count(female) × count(fiction)/n=

(1200 × 450) /1500 = 360.

e21 = count(male) × count(non-fiction)/n =

(300 × 1050) /1500 = 210

e22 = count(female) × count(non-fiction)/n=

(1200 × 1050) /1500 = 840.

**χ 2 is computed as**

χ 2 = (250 − 90) 2 /90 + (50 − 210) 2 /210 + (200 − 360) 2 /360 + (1000 − 840) 2 /840

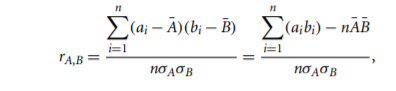
= 284.44 + 121.90 + 71.11 + 30.48

= 507.93.

**Correlation Coefficient for Numeric Data:**

For numeric attributes, we can evaluate the correlation between two attributes, A and B, by computing the correlation coefficient (also known as Pearson’s product moment coefficient).

This is given as

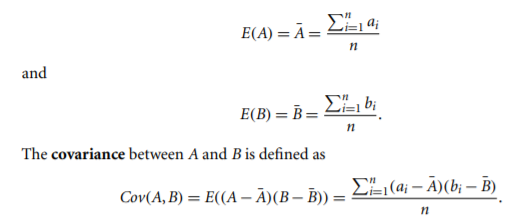


where n is the number of tuples, ai and bi are the respective values of A and B in tuple i, A¯ and B¯ are the respective mean values of A and B, σA and σB are the respective standard deviations of A and B.

is the sum of the AB cross-product

Covariance of Numeric Data:

correlation and covariance are two similar measures for assessing how much two attributes change together. Consider two numeric attributes A and B, and a set of n observations {(a1,b1),...,(an,bn)}. The mean values of A and B, respectively, are also known as the expected values on A and B, that is,



If we compare Eq for rA,B (correlation coefficient) with Eq. for covariance,



where σA and σB are the standard deviations of A and B, respectively

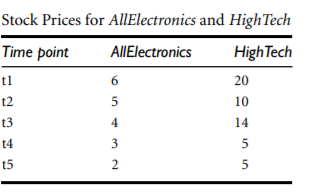


For two attributes A and B that tend to change together, if A is larger than A¯ (the expected value of A), then B is likely to be larger than B¯ (the expected value of B). Therefore, the covariance between A and B is positive. On the other hand, if one of the attributes tends to be above its expected value when the other attribute is below its expected value, then the covariance of A and B is negative.

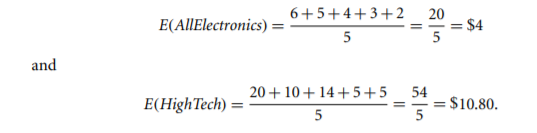
If A and B are independent (i.e., they do not have correlation), then E(A · B) = E(A)· E(B). Therefore, the covariance is Cov(A,B) = E(A · B) − A¯ B¯ = E(A)· E(B) − A¯ B¯ = 0.

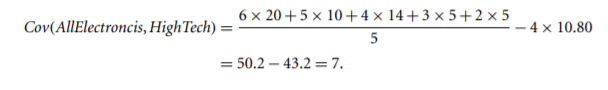
**Example:**

Covariance analysis of numeric attributes



which presents a simplified example of stock prices observed at five time points for AllElectronics and HighTech, a high-tech company.





**Data Reduction**

Data reduction strategies include dimensionality reduction, numerosity reduction, and data compression.

Dimensionality reduction is the process of reducing the number of random variables or attributes under consideration.

**Dimensionality reduction** methods include **wavelet transforms and principal components analysis** which transform or project the original data onto a smaller space. **Attribute subset selection** is a method of dimensionality reduction in which irrelevant, weakly relevant, or redundant attributes or dimensions are detected and removed

**Numerosity reduction** techniques replace the original data volume by alternative, smaller forms of data representation. These techniques may be **parametric or nonparametric.**

Parametric methods- a model is used to estimate the data, so that typically only the data parameters need to be stored, instead of the actual data. **Regression and log-linear models** are the examples.

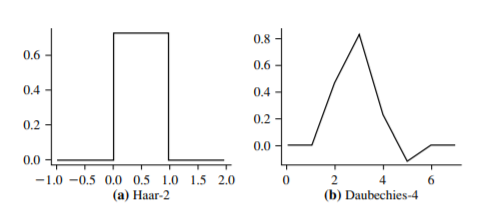
Nonparametric methods for storing reduced representations of the data include histograms, clustering, sampling and data cube aggregation.

**Data compression**-transformations are applied so as to obtain a reduced or “compressed” representation of the original data. If the original data can be reconstructed from the compressed data without any information loss, the data reduction is called **lossless**. If, instead, we can reconstruct only an approximation of the original data, then the data reduction is called **lossy**.

Wavelet Transforms

The 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, X1 , of wavelet coefficients. The two vectors are of the same length. When applying this technique to data reduction, we consider each tuple as an n-dimensional data vector, that is, X = (x1,x2,...,xn), depicting n measurements made on the tuple from n database attributes.

The DWT is closely related to the discrete Fourier transform (DFT), a signal processing technique involving sines and cosines. Figure shows some wavelet families.



The method is as follows:

1. The length, L, of the input data vector must be an integer power of 2. This condition can be met by padding the data vector with zeros as necessary (L ≥ n).

2. Each transform involves applying two functions. The first applies some data smoothing, such as a sum or weighted average. The second performs a weighted difference, which acts to bring out the detailed features of the data.

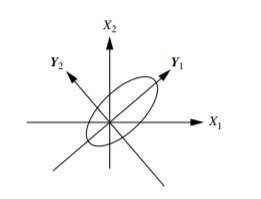
3. The two functions are applied to pairs of data points in X, that is, to all pairs of measurements (x2i ,x2i+1). This results in two data sets of length L/2. In general, these represent a smoothed or low-frequency version of the input data and the highfrequency content of it, respectively.

4. The two functions are recursively applied to the data sets obtained in the previous loop, until the resulting data sets obtained are of length 2.

5. Selected values from the data sets obtained in the previous iterations are designated the wavelet coefficients of the transformed data.

**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. The original data are thus projected onto a much smaller space, resulting in dimensionality reduction



The basic procedure is as follows:

1. 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.

2. PCA computes k orthonormal vectors that provide a basis for the normalized input data. These are unit vectors that each point in a direction perpendicular to the others. These vectors are referred to as the principal components. The input data are a linear combination of the principal components.

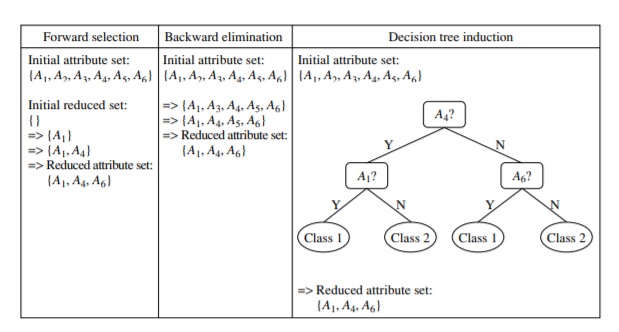
3. The principal components are sorted in order of decreasing “significance” or strength. The principal components essentially serve as a new set of axes for the data, providing important information about variance. That is, the sorted axes are such that the first axis shows the most variance among the data, the second axis shows the next highest variance, and so on. For example, the above Figure shows the first two principal components, Y1 and Y2, for the given set of data originally mapped to the axes X1 and X2. This information helps identify groups or patterns within the data.

4. 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.

**Attribute Subset Selection**

**Attribute 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.



**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 flowchartlike 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.

**Regression and Log-Linear Models: Parametric Data Reduction**

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



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.

**Multiple linear regression** is an extension of (simple) linear regression, which allows a response variable, y, to be modeled as a linear function of two or more predictor variables.

**Log-linear models** approximate discrete multidimensional probability distributions. Given a set of tuples in n dimensions , 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.

**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.

Example:

Histograms. 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.

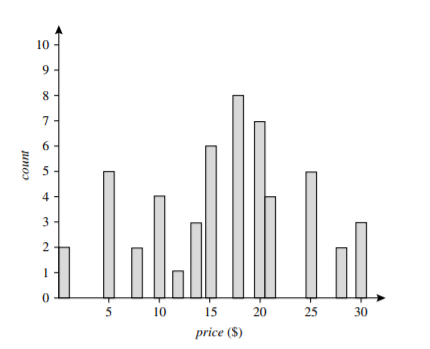


Fig:1 A histogram for price using singleton buckets—each bucket represents one price–value/ frequency pair

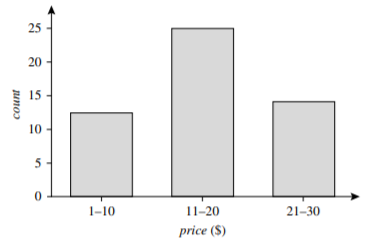


Fig:2 An equal-width histogram for price, where values are aggregated so that each bucket has a uniform width of $10

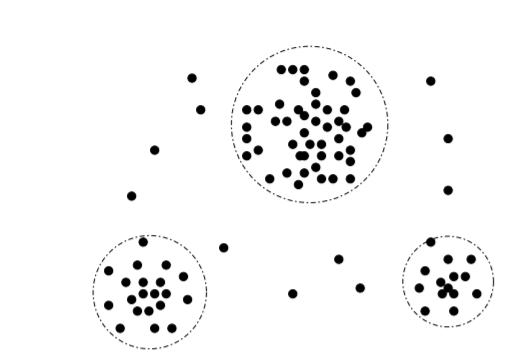
There are several partitioning rules, including the following:

**Equal-width:** In an equal-width histogram, the width of each bucket range is uniform (e.g., the width of $10 for the buckets in Figure 2).

**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).

**Clustering**

Clustering techniques consider data tuples as objects.Centroid distance is an alternative measure of cluster quality and is defined as the average distance of each cluster object from the cluster centroid (denoting the “average object,” or average point in space for the cluster).



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.
* 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.

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.

**DATA TRANSFORMATION BY NORMALIZATION**

Data normalization methods **are min-max normalization, z-score normalization, and normalization by decimal scaling**.

let A be a numeric attribute with n observed values, v1, v2,..., vn.

**Min-max normalization** performs a linear transformation on the original data. Suppose that minA and maxA are the minimum and maximum values of an attribute, A. Min-max normalization maps a value, vi , of A to vi1 in the range [new minA,new maxA] by computing



**Example:**

Min-max normalization. Suppose that the minimum and maximum values for the attribute income are $12,000 and $98,000, respectively. We would like to map income to the range [0.0,1.0]. By min-max normalization, a value of $73,600 for income is transformed to



vi1  = (73,600−12,000)/(98,000−12,000) (1.0 − 0) + 0 = 0.716.

**Example:**

for the attribute age: 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.

Use min-max normalization to transform the value 15,20,35 for age onto the range [0.0,1.0].



**For the value 15**

minA= 13

maxA=70

new minA= 0

new maxA= 1.0

v = 15

Vi1 =(15-13)/(70-13)(1-0)+0

Vi1 =0.04

**For the value 20**

minA= 13

maxA=70

new minA= 0

new maxA= 1.0

v = 20

Vi1 =(20-13)/(70-13)(1-0)+0

Vi1 =0.12

**For the value 35**

minA = 13,

maxA = 70,

new minA = 0,

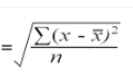
new maxA = 1.0,

v = 35

Vi1 =(35-13)/(70-13)(1-0)+0

Vi1 =0.39

**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, vi , of A is normalized to vi1 by computing

σA

where A¯ and σA are the mean and standard deviation, respectively, of attribute A.

**Example:**

z-score normalization. Suppose that the mean and standard deviation of the values for the attribute income are $54,000 and $16,000, respectively. With z-score normalization, a value of $73,600 for income is transformed to

vi1  = (73,600−54,000)/16,000 = 1.225.

**Example:**

z-score normalization for the following data

8,10,15,20

Mean=(8+10+15+20)/4

=53/4

=13.25

Standard deviation σA



= 4.6

For value 8



Vi1= (8-13.25)/4.6

= -1.14

For value 10

Vi1= (10-13.25)/4.6

=-0.7

For value 15

Vi1= (15-13.25)/4.6

=0.3

For value 20

Vi1= (20-13.25)/4.6

=1.4

|  |  |
| --- | --- |
| **data** | data after z-score normalization |
| 8 | -1.14 |
| 10 | -0.7 |
| 15 | 0.3 |
| 20 | 1.4 |

A variation of this z-score normalization replaces the standard deviation of above Eq. by the mean absolute deviation of A. The mean absolute deviation of A, denoted sA, is



Thus, z-score normalization using the mean absolute deviation is



**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, vi , of A is normalized to vi1 by computing



where j is the smallest integer such that max(|vi1 |) < 1.

**Example**

for the attribute age: 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.

Use **Normalization by decimal scaling** to transform the value 35 for age

where *j* = 2, *v* = 35 is transformed to *v*′ = 35/100=0*.*35

Use **Normalization by decimal scaling** to transform the value 45 for age

where *j* = 2, *v* = 45 is transformed to *v*′ = 45/100=0*.*45

**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. n. For example, attribute values can be discretized by applying equal-width or equal-frequency binning, and then replacing each bin value by the bin mean or median, as in smoothing by bin means or smoothing by bin medians, respectively.

**Discretization by Histogram Analysis**

A histogram partitions the values of an attribute, A, into disjoint ranges called buckets or bins. In an equal-width histogram, for example, the values are partitioned into equal-size partitions or ranges. With an equal-frequency histogram, the values are partitioned so that, ideally, each partition contains the same number of data tuples. A minimum interval size can also be used per level to control the recursive procedure.

**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.

**Decision tree–based discretization** uses class information, it is more likely that the interval boundaries (split-points) are defined to occur in places that may help improve classification accuracy.

**Measures of correlation** can be used for discretization. Chi Merge is a χ 2 -based discretization method.

**Concept Hierarchy Generation for Nominal Data**

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.
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.
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.
4. **Specification of only a partial set of attributes:** Sometimes a user can be careless when defining a hierarchy, or have only a vague idea about what should be included in a hierarchy.