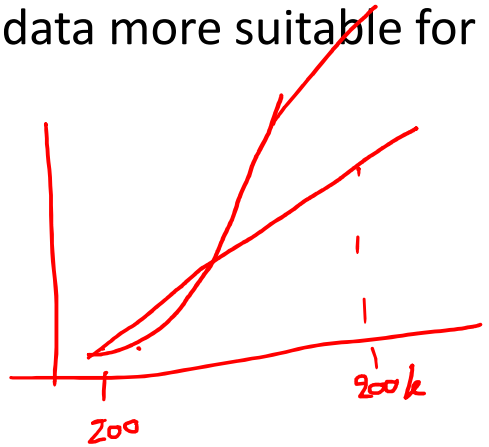
Data Mining

Data Preprocessing
Similarity and Dissimilarity

Slides by Tan, Steinbach, Kumar adapted by Pimprapai Thainiam

Data Preprocessing

- **Data preprocessing** is the process used to make the data more suitable for data mining.



- **Two categories** of data preprocessing:
  1. Selecting data objects and attributes
  2. Creating/changing the attributes

- **Goal** of data preprocessing is to improve the data mining analysis with respect to time, cost, and quality.

Data Preprocessing

- **7 most important ideas and approaches:**
    1) Aggregation
    2) Sampling
    3) Dimensionality Reduction
    4) Attribute Subset Selection
    5) Attribute Creation
    6) Discretization and Binarization
    7) Attribute Transformation

m = objects, n = attributes
n dimension data set

Topics

- **Data preprocessing:**
- Similarity and Dissimilarity

Aggregation

- **Aggregation** is the process of reducing number of objects by combining two or more objects into a single object.

- **How the values of each attribute are combined across all the record.**

  Quantitative attributes: The attribute can be aggregated by taking a sum or an average.

  Qualitative attributes: The attribute values can either be omitted or summarized as the set of all the values.

Aggregation

- **Advantages of Aggregation:**
  1) Data reduction: Reducing the number of data objects (rows) results in less memory and processing time because some data mining algorithms are expensive (take too much time to process).
  2) Change of scope or scale: This will provide a high-level (less detailed data) view of the data instead of a low-level (more detailed data) view.
  3) More stable data: Aggregated data by using average (mean, expected value) or total tends to have less variability than the individual objects being aggregated.

- **Disadvantages of Aggregation:**
  1) Interesting detail of the data might be lost.

Aggregation

Examples:

- Cities aggregated into regions, states, or countries, etc. (Reducing the possible values for locations)
- Sales at a particular store location aggregated into daily, weekly, or monthly sales (Reducing the possible values for transaction ID at a particular store location from multiple transactions to 365 days, 52 weeks or 12 months)

Transaction ID	Item	Store Location	Date	Price	…
101123	Watch	Chicago	09/06/16	$25.99	…
101123	Battery	Chicago	09/06/16	$5.99	…
101124	Shoes	Dallas	09/06/16	$75.00	…
101124	Socks	Dallas	09/06/16	$9.00	…
101125	Milk	Dallas	09/06/16	$4.00	…
⋮	⋮	⋮	⋮	⋮	

Aggregation

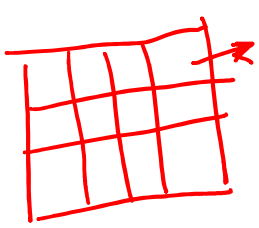
Examples:

- Replace all the transactions of a single store with a single storewide transaction.
- This reduces the hundreds or thousands transactions that occur daily a specific store to a single daily transaction, and the number of data objects is reduced to the number of stores.

Items	Store Location	Date	Price	…
Watch, Battery	Chicago	09/06/16	$31.98	…
Shoes, Socks, Milk	Dallas	09/06/16	$88.00	…
⋮	⋮	⋮	⋮	

- Aggregation steps
  1) Transaction ID is omitted.
  2) Items are summarized.
  3) Prices are summed up to one value.

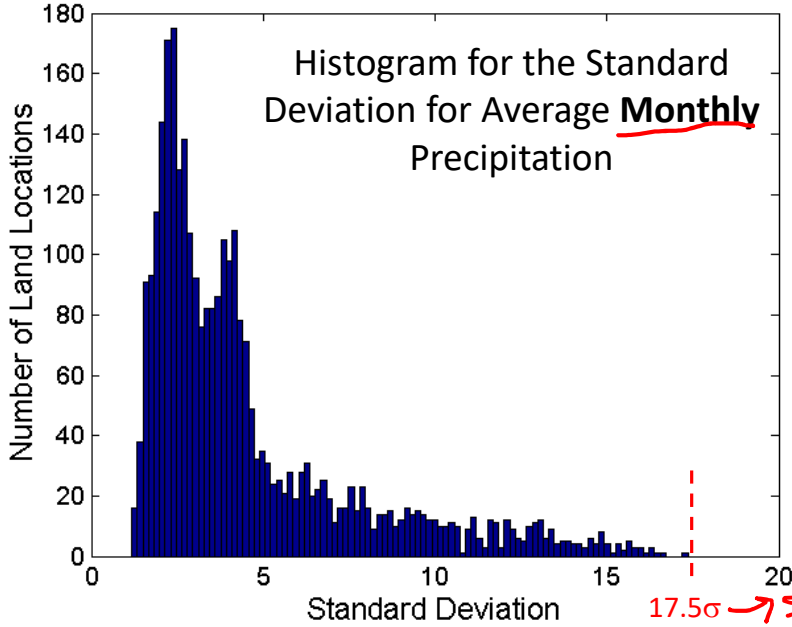
Aggregation

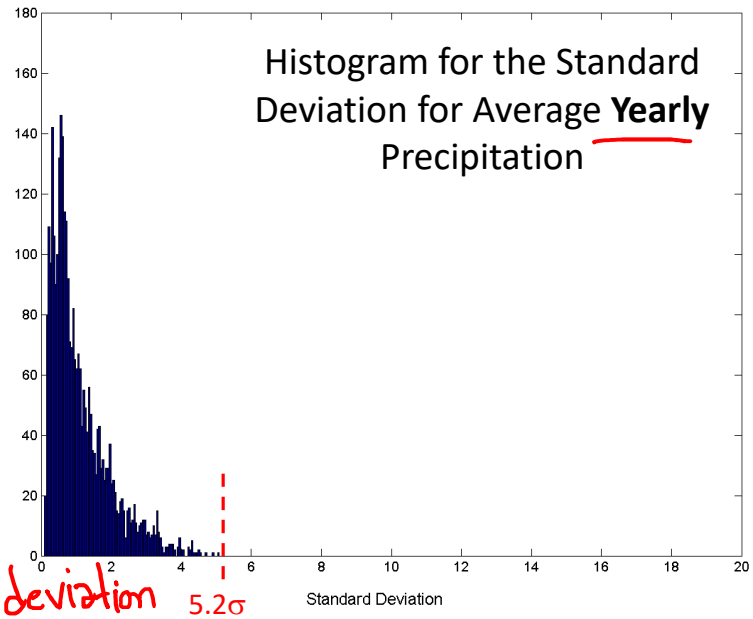


daily data → average monthly → 144 values
→ 1 yearly → 12 values
SD SD

Example:

- The variation of Precipitation in Australia from the period 1982 to 1993 (12 years) of 3,030 grid cells. The average yearly precipitation has less variability than the average monthly precipitation.





17.5σ → standard deviation 5.2σ

Sampling

- **Sampling** is a commonly used approach for selecting a subset of the data objects to be analyzed and it is also the main technique employed for data selection.

- The **motivations for sampling** in statistics and data mining are different.

  Statisticians sample because obtaining the entire set of data of interest is too expensive or time consuming. For examples, census data of people living in Bangkok.

  Data miners sample because processing the entire set of data of interest is too expensive or time consuming. In some cases, using a sampling algorithm can reduce the data size to the point where a better, but more expensive (in terms of processing time, and memory) algorithm can be used.

Sampling

avg. attribute x → 10. (whole data set)
→ 9.8 (100 data objects)

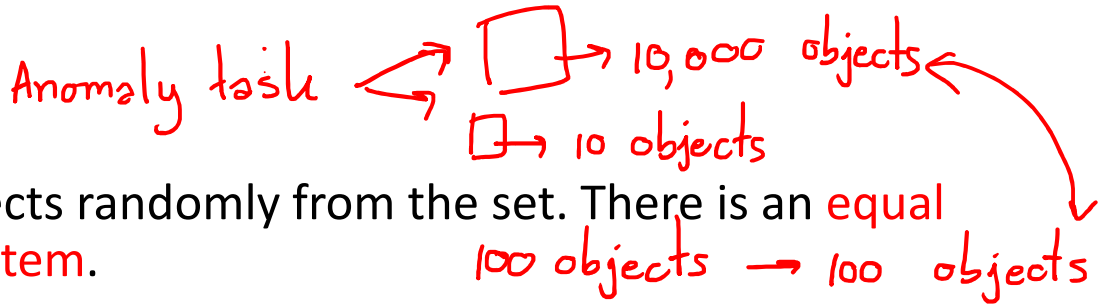
- The **key principle for effective sampling** is the following:
  - Using a sample will work almost as well as using the entire data sets, if the sample is **representative**.
  - A sample is representative if it has approximately the same property (of interest) as the original set of data. For example,

    Average (mean) of the data objects is the property of interest, then a sample is representative if it has an average that is close to that of the original data.

- Because sampling is a statistical process, the representativeness of any particular sample will vary, and the best that we can do is choose a sampling scheme that guarantees a high probability of getting a representative sample.

- **Two sampling issues:** - Sampling technique
  - Sample size

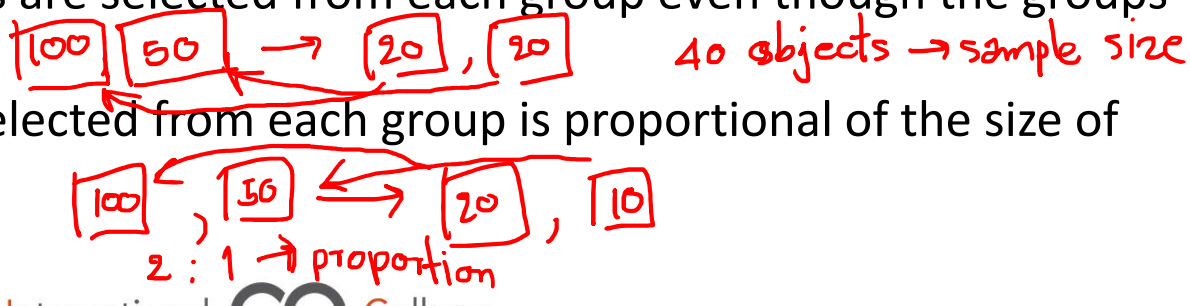
Sampling Approaches



1. **Simple Random Sampling:** Select objects randomly from the set. There is an equal probability of selecting any particular item.

> However, the population may consists of different types of objects, with widely different number of objects, thus simple random sampling can fail to adequately represent those types of objects that are less frequent.

2. **Stratified Sampling:** Split the objects into several groups based on a selected attribute; then select objects randomly from each group. Two variations:
   (1) Equal numbers of objects are selected from each group even though the groups are of different sizes.
   (2) The number of objects selected from each group is proportional of the size of that group.



Sampling Approaches

Both type of sampling approaches can have two variations:

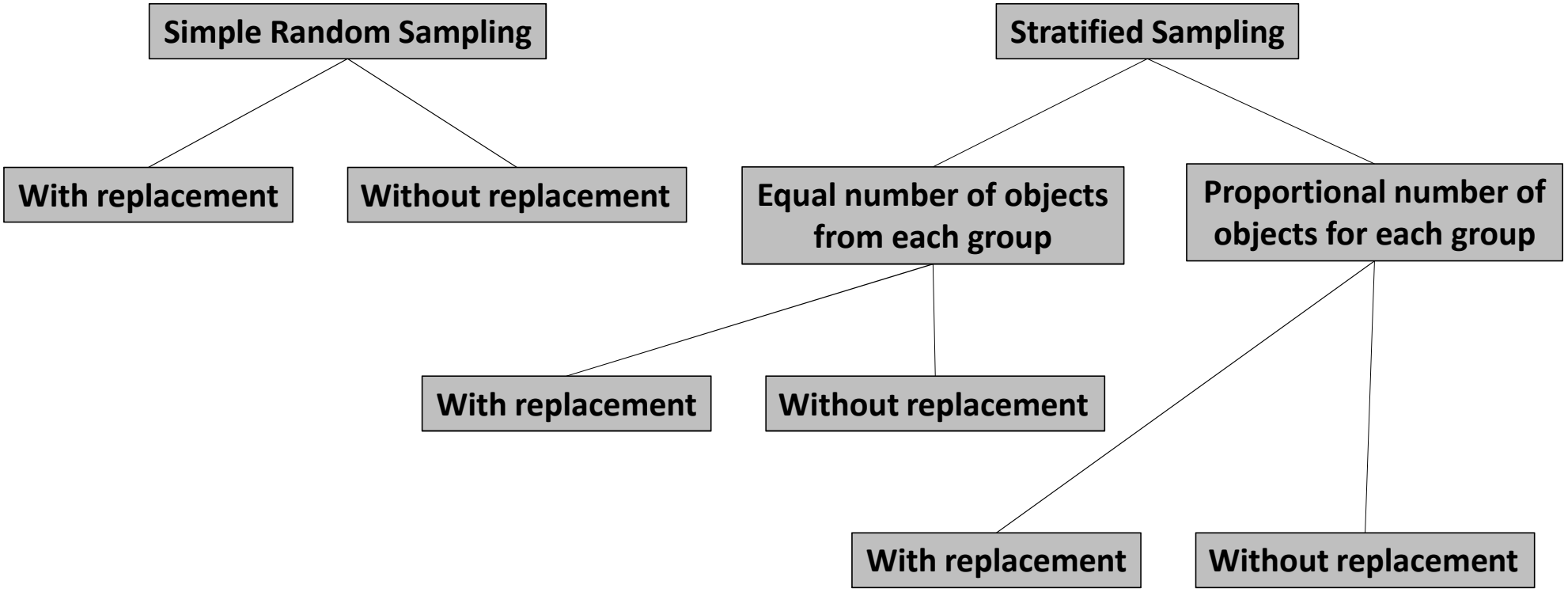
(1) Sampling without replacement:

- As each object is selected, it is removed from the population.
- Probability of selecting any object is getting higher during the sampling process.

(2) Sampling with replacement:

- Objects are not removed from the population as they are selected for the sample.
- The same object can be selected more than once.
- Probability of selecting any object remains constant during the sampling process.

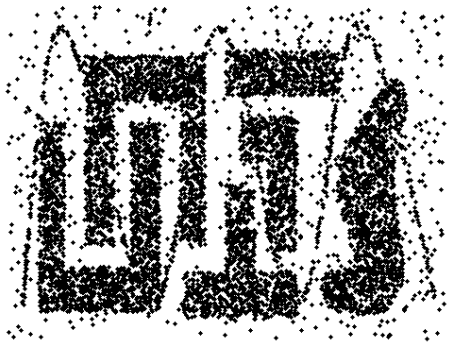
Sampling Approaches



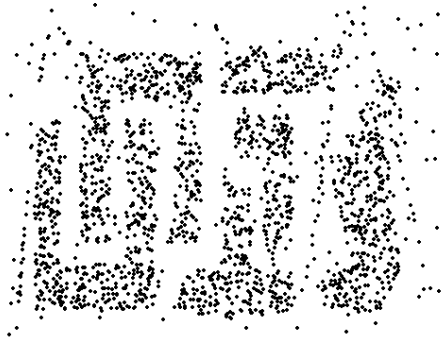
Sample Size

Once a sampling technique has been selected, it is still necessary to choose the sample size.

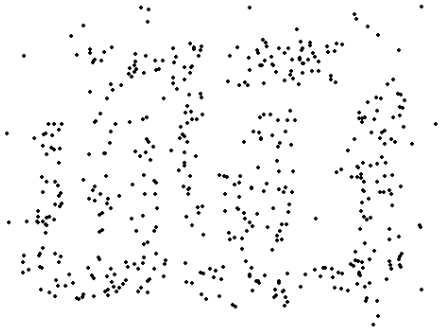
- **Larger sample size** increases the probability that sample will be representative, but they also eliminate much of the advantage of sampling (reduce processing time).
- **Smaller sample size** may create erroneous pattern or miss the original pattern.



8,000 points



2,000 Points



500 Points

Determining Proper Sample Size

The data mining performance improves as the sample sizes increases.

For a set of data that consists of a small number of almost equal-sized groups.

- The desired set of representative points is obtained by taking one point from each of these groups.
- To follow this approach, we need to determine a sample size that would guarantee, with a high probability, the desired outcome; that is, that at least one point will be obtained from each cluster.

Determining Proper Sample Size

Progressive (Adaptive) Sampling:

- It starts with a small sample , and then increase the sample size gradually until a sample of sufficient size have been obtained.
- We use a stopping criterion as a criterion to tell if a particular sample size is large enough.
- Stopping criteria could be as follows
  - Improvement rate of results falls below 10%
  - Reach a prespecified-calculation time

Dimensionality Reduction

- **Dimensionality reduction** is the process of reducing the number of attributes under consideration.

- **Benefits of Dimensionality Reduction:**
  1) It may help to eliminate irrelevant attributes or reduce noise. This will make many data mining algorithms work better.
  2) It can lead to a more understandable model because the model may involve fewer attributes.
  3) It may allow data to be more easily visualized.
  4) It can reduce amount of time and memory required by data mining algorithms

Dimensionality Reduction

- **Techniques for Dimensionality Reduction:**
  1. **Attribute creation:** It creates new attributes that are a combination of the old attributes.
  2. **Attribute subset selection:** It selects attributes that are subset of the old attributes.

Attribute Subset Selection

- **Attribute subset selection** is used to find a smallest set of attributes by selecting only a subset of the attributes which can be done by removing redundant attributes or irrelevant to the data mining task attributes.

- **Redundant attributes** duplicate much or all of the information contained in one or more other attributes. For example,
  - Purchase price of a product and the amount of sales tax paid contain much of the same information.

- **Irrelevant attributes** contain no information that is useful for the data mining task at hand. For example,
  - Students' ID numbers are irrelevant to the task of predicting students' GPA.
  - Customers' telephone numbers are irrelevant to the task of classifying willingness to purchase.

Attribute Subset Selection

- **Approaches for Attribute Subset Selection:**

  1. Basic Approach:

  - Use your common sense or domain knowledge to select attributes

  2. Systematic Approaches:

  1) Ideal Approach:

  - Try all possible attribute subsets (complete enumeration) as input to data mining algorithm, then take the subset that produces the best results.
  - The number of subsets involving n attributes is $2^n - 1$

    If number of attributes = 3 $\Rightarrow$ number of subsets = $2^3 - 1 = 8 - 1$

    If number of attributes = 20 $\Rightarrow$ number of subsets = $2^{20} - 1$ = ~~1,048,576~~ 1,048,575
  - This approach is impractical.

{1, 2, .., 10} → {1}, {2}, ..., {1, 2}, ..., {1, 2, .., 10}

10 attributes = $2^n = 2^{10}$ → include {∅}

Attribute Subset Selection

2) Standard Approaches:

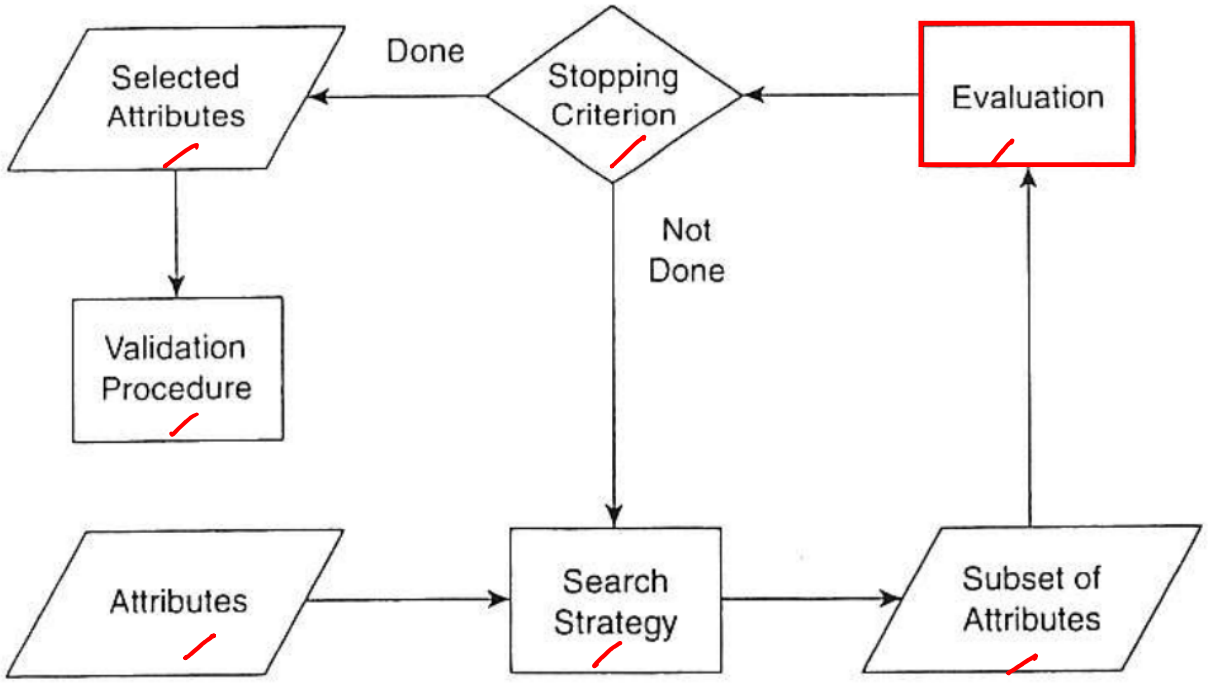
(1) Embedded approaches: Attribute selection occurs naturally as part of the data mining algorithm. The algorithm itself decides which attributes to use and which to ignore.

(2) Filter approaches: Attributes are selected before data mining algorithm is run, using some measure approaches that are independent of the data mining task, such measures attempt to predict how well the actual data mining will perform on a given set of attributes For example, we might select sets of attributes whose pairwise correlation is as low as possible.

(3) Wrapper approaches: This approach use the criterion normally used to measure the result of the target data mining algorithm, similar to the ideal approach but typically without enumerating all possible subsets.

Attribute Subset Selection

Flowchart of filter and wrapper approaches

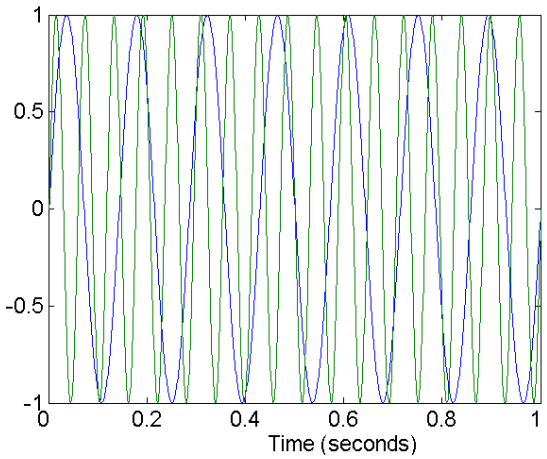


Attribute Creation

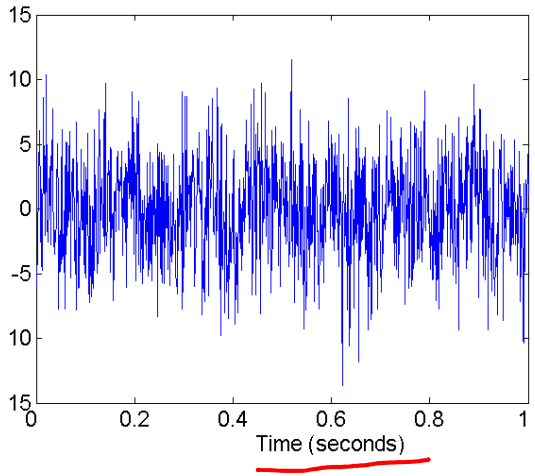
- **Attribute creation** is used to create new attributes that can capture the important information in a data set much more efficiently than the original attributes.

- **Three general methodologies:**
  1. **Attribute Extraction:** The creation of a new set of attributes from the original raw data. For example, data mining task is to predict whether or not a photograph contains a human face; its original raw data is a set of pixels. A new set of attributes could be the presence or absence of certain types of edges and areas that are highly correlated with the presence of human faces.
  2. **Mapping Data to New Space:** A totally different view of the data can reveal important and interesting attributes. For example, applying Fourier transform to the time series in order to change to a representation in which frequency information is explicit.

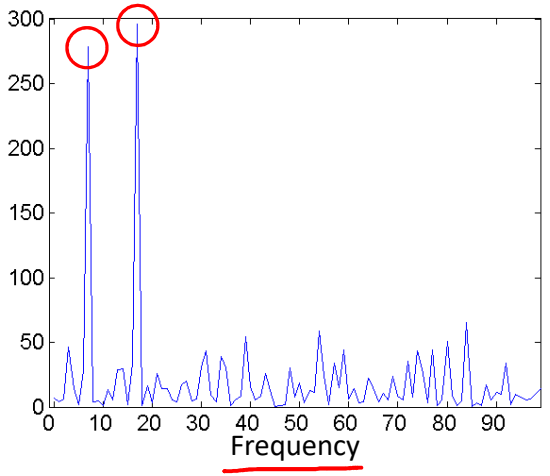
Attribute Creation



Two Sine Waves (frequencies of 7 and 17 cycles per second)



Two Sine Waves + Random Noise

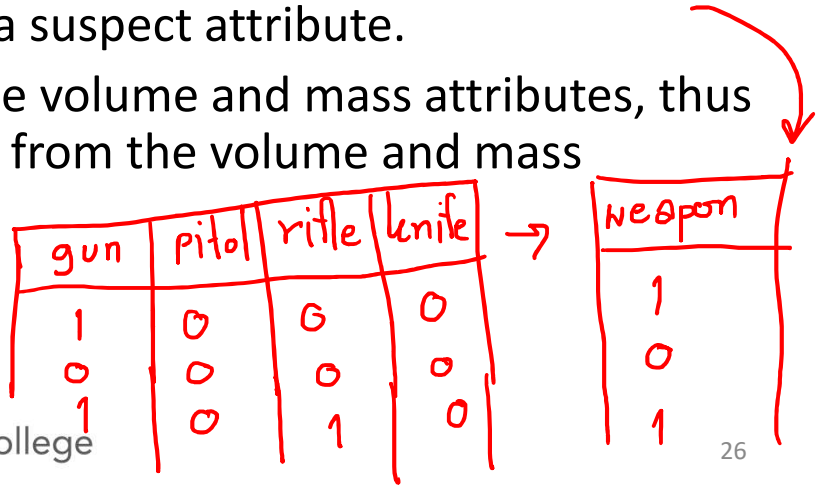


Power Spectrum

Attribute Creation

3. Attribute Construction:

- Sometime the attributes in the original data sets have the necessary information, but it is not in a form suitable for the data mining algorithm.
- One or more new attributes constructed out of the original attributes (combining attributes) can be more useful than the original attributes.
- For example
  - Stop, question, and frisk data set contains many attributes indicating if weapons like gun, pistol, or rifle were found on a suspect; this could be combined to if weapon was found on a suspect attribute.
  - Historical artifacts data set contains the volume and mass attributes, thus a density attribute can be constructed from the volume and mass attributes where $density = \frac{mass}{volume}$



Discretization and Binarization

- **Discretization** is the process used to
  1) transform a continuous attribute into a categorical attribute
  2) transform a large-levels attribute into a small-levels attribute

- **Binarization** is the process used to transform a continuous attribute or a discrete (categorical) attribute into one or more binary attributes.

- **Reasons for Discretization and Binarization:**
  - For some classification algorithms, they require that the data must be in the form of categorical attributes. → discretization
  - For association rules mining, it requires that the data must be in the form of binary attributes. → binarization

Binarization

Binarization of Categorical (Discrete) Attributes

The simple technique is the following:

Step 1: If there are m categorical values, then uniquely assign each original value to an integer in the interval [0, m-1].

Note that

- If the attribute is ordinal, then order must be maintained by the assignment, and
- If the attribute is originally represented using integers, this process is necessary if the integers are not in the interval [0, m -1].

Step 2: Convert each of these m categories to a binary number using n binary attributes where $n = \lceil \log_2 m \rceil$. → round up

For example, $m = 5$ original values

Binarization

Example: A categorical attribute with 5 values {awful, poor, OK, good, great} → $m = 5$

$$\lceil \log_2 5 \rceil = \lceil 2.322 \rceil = 3$$

2 new attributes → 3 new attribute

This would require three new binary variables x_1, x_2, x_3.

0,0
1,0
0,1
1,1
4 unique values

Categorical Value	Integer Value	x_1	x_2	x_3
Awful	0	0	0	0
Poor	1	1	0	0
OK	2	0	1	0
Good	3	0	0	1
Great	4	1	1	0

0,0,0
1,0,0
0,1,0
0,0,1
1,1,0
1,0,1
0,1,1
1,1,1
5 unique values

Note: - 2^n can be used to determine maximum number of attribute values that a set of n binary attributes can represent.

- This approach can create unintended relationships among the transformed attributes.

Binarization

- Association rule mining requires asymmetric binary attributes where only the presence of the attribute (value = 1) is important
- For association problems, it is necessary to introduce one binary attribute for each categorical value.

Categorical Value	Integer Value	x_1	x_2	x_3	x_4	x_5
Awful	0	1	0	0	0	0
Poor	1	0	1	0	0	0
OK	2	0	0	1	0	0
Good	3	0	0	0	1	0
Great	4	0	0	0	0	1

- If number of attribute values is too large, then the discretization of categorical attribute can be implemented in order to reduce number of levels before implementing binarization.

Discretization

Discretization of Continuous Attribute

Two subtasks:

1) Deciding how many categories (n) to have (How many split points ($n-1$) to choose?)
2) Determining how to map the values of the continuous attribute to these categories (Where to place those split points?)

General discretization steps:

Step 1: Sort all values

Step 2: Divide sorted values into n intervals by specifying $n-1$ split points which are $x_1, x_2, \ldots, x_{n-1}$

Step 3: Map all values in the same interval to the same categorical value.

Discretization

Discretization of Continuous Attribute

- The result can be represented either as

  1) A set of intervals

  $$\{(x_0, x_1], (x_1, x_2], \ldots, (x_{n-1}, x_n)\}$$

  where x_0 and x_n ~~may be +∞ or -∞~~ may be $-\infty$ and $+\infty$, respectively

  2) A series on inequalities

  $$x_0 < x \leq x_1, x_1 < x \leq x_2, \ldots, x_{n-1} < x < x_n$$

  where x_0 and x_n may be ~~+∞ or -∞~~ x_0 and x_n may be $-\infty$ or $+\infty$

Discretization

Discretization Methods of Numerical Attribute:

There are 2 types of discretization of numerical attribute:

1. **Unsupervised Discretization** is the discretization method where class information is not used. There are 3 methods.

   1) Equal width approach:

   - This approach divides the range of the attribute into a prespecified number of intervals each having the same width.
   - It can be badly affected by outliers.

$$\frac{\text{max} - \text{min}}{\text{no. groups.}}$$

   2) Equal frequency (equal depth) approach:

   - This approach tries to put the same number of objects into each interval.
   - Mostly used approach

$$\frac{\text{No. values}}{\text{No. groups}} = \text{No. of obj. in each group.}$$

Discretization

3) K-means approach: (see Chapter 8)

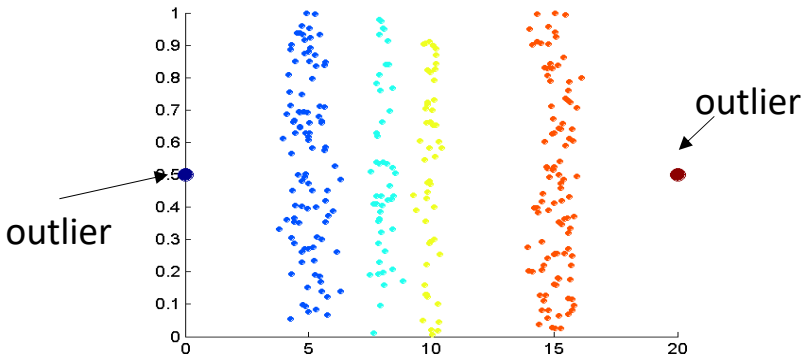
1) Randomly select n cluster centers $\{v_1, v_2, \dots, v_n\}$
2) Calculate the distance between each data point and cluster centers.
3) Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers.
4) Recalculate the new cluster center using:

$$v_i = \frac{1}{c_i} \sum_{j=1}^{c_i} x_i$$

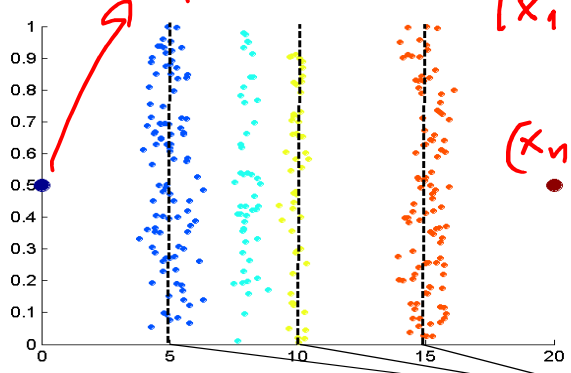
where c_i is the number of data points in i^{th} cluster

5) Recalculate the distance between each data point and new obtained cluster centers.
6) If no data point was reassigned then stop, otherwise repeat steps 3) – 6)

Discretization Examples



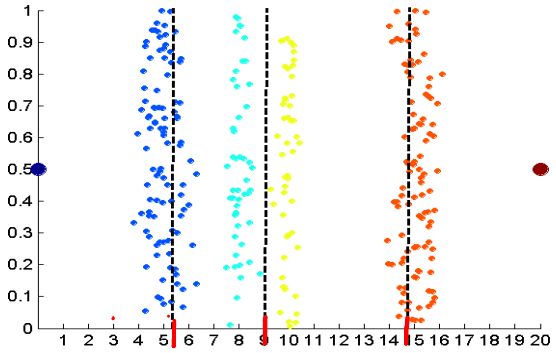
Original data



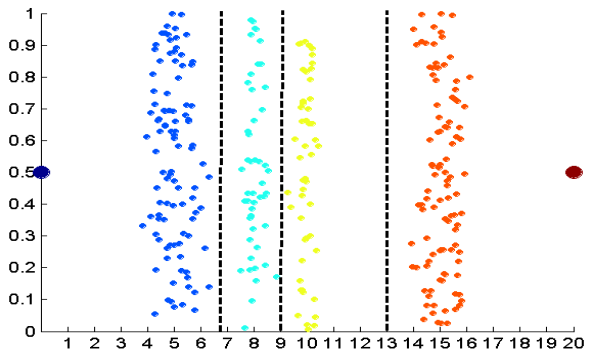
Equal width discretization

Split points

Discretize the x values into 4 categorical values.



Equal frequency discretization



K-means discretization

Discretization

2. Supervised Discretization is the discretization method where class information is used by trying to place the split points in a way that maximize the purity of the intervals.

Entropy-based Approaches are statistically based approaches where the entropy of an interval is a measure of the purity of an interval.

The entropy of the i^{th} interval :

$$e_i = -\sum_{i=1}^{k} p_{ij} \log_2 p_{ij}$$

The total entropy :

$$e = \sum_{i=1}^{n} w_i e_i$$

$\boldsymbol{k}$ is the number of different class labels

$\boldsymbol{m}$ is the number of values (objects)

$\boldsymbol{n}$ is the number of intervals (prespecified)

$\boldsymbol{m_i}$ is the number of values in the i^{th} interval of a partition

$\boldsymbol{m_{ij}}$ is the number of values of class j in interval i where $j = \{1, \dots, k\}$

$\boldsymbol{p_{ij}} = \frac{\boldsymbol{m_{ij}}}{\boldsymbol{m_i}}$ is the probability (fraction of values) of class j in the i^{th} interval

$\boldsymbol{w_i} = \frac{\boldsymbol{m_i}}{\boldsymbol{m}}$ is the fraction of values in the i^{th} interval

Discretization

- If an interval contains only values of one class (is perfectly pure), then the entropy is 0 and it contributes nothing to the overall entropy.
- If the classes of values in an interval occur equally often (the interval is as impure as possible), then the entropy is a maximum.

- **Entropy-based Approach Steps:**

  Step 1: Create partitions by bisecting the interval values using each value as a possible split point (number of partitions = number of values – 1)

  Step 2: Calculate the total entropy (e) of each partition

  Step 3: Select a partition that give minimum total entropy (e)

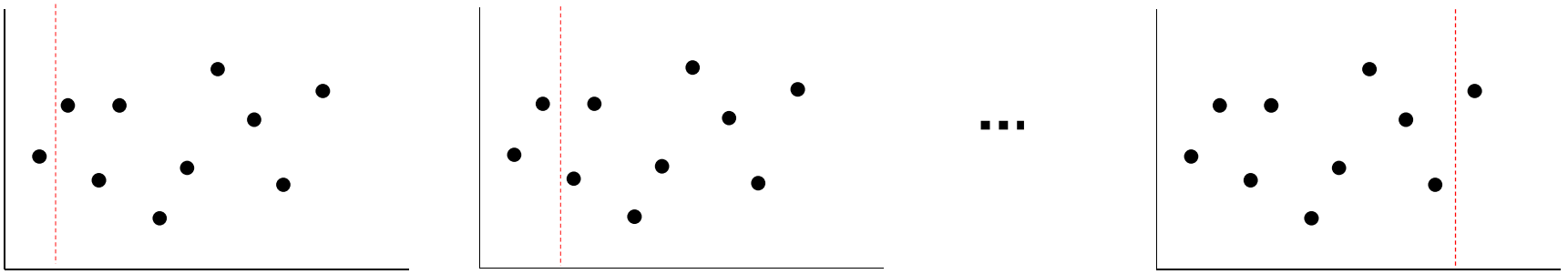
  Step 4: Repeat steps 1 – 3 with the interval that has highest entropy (e_i)

  Step 5: Stop the process when a pre-specified number of intervals is reached or another stopping criterion (e.g., if the total entropy is reduced by less than 0.1) is satisfied

Discretization

Example: Discretize the x values where $m = 10, n = 4$ (m: no. objects, n: no. groups)

Step 1: Create partitions by bisecting the interval values using each value as a possible split point (number of partitions = number of values – 1)



$e_1 = p_1, e_2 = p_2$ $\quad$ $e_1 = p_3, e_2 = p_4$ $\quad$ ... $\quad$ $e_1 = p_{17}, e_2 = p_{18}$

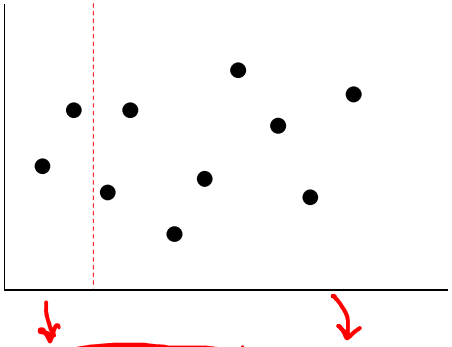
$e = q_1$ $\quad$ $e = q_2$ $\quad$ ... $\quad$ $e = q_9$

Step 2: Calculate the total entropy (e) of each partition → $\{q_1, q_2, \dots, q_9\}$

Discretization

Step 3: Select a partition that give minimum total entropy (e)
(assume that the second partition has the lowest $e \rightarrow q_2 < q_1, q_3, \ldots, q_9$)

the lower e = the purer the groups

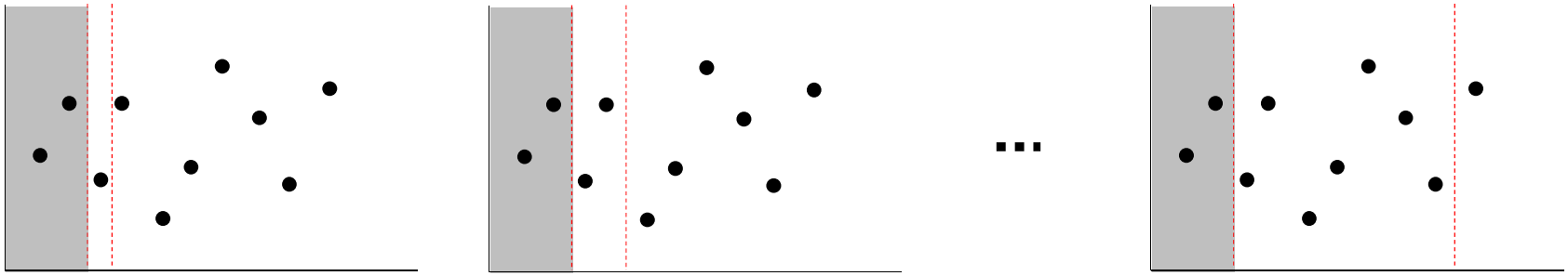


$e_1 = p_3, e_2 = p_4$

$e = q_2$

Step 4: Repeat steps 1 – 3 with the interval that has highest entropy (e_i)
(assume that the second group has the highest $e_i \rightarrow p_2 > p_1$)

Discretization



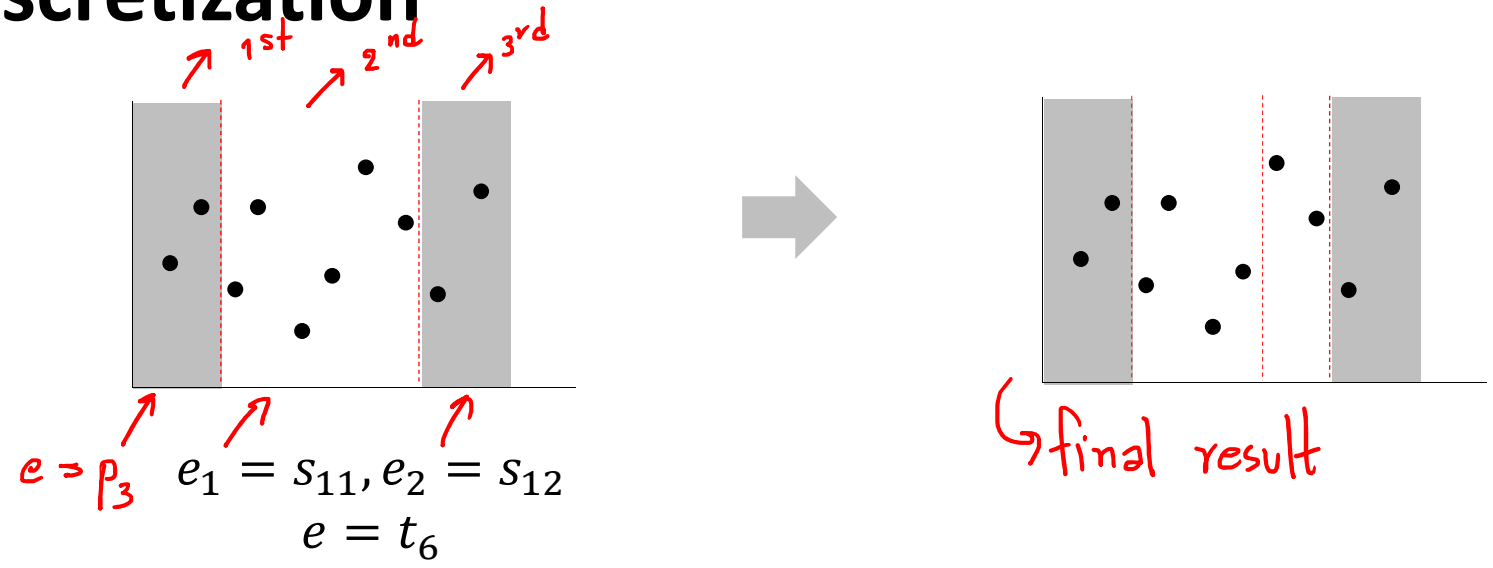
$e_1 = s_1, e_2 = s_2$ $e_1 = s_3, e_2 = s_4$ ... $e_1 = s_{13}, e_2 = s_{14}$

$e = t_1$ $e = t_2$ ... $e = t_7$

- assume that the sixth partition has the lowest $e \rightarrow t_6 < t_1, t_2, \dots, t_7$
- Now, we already have 3 groups, assume that the second group has the highest $e_i \rightarrow s_{11} > p_3, s_{12},$ thus continue partitioning with the second group.

highest entropy value

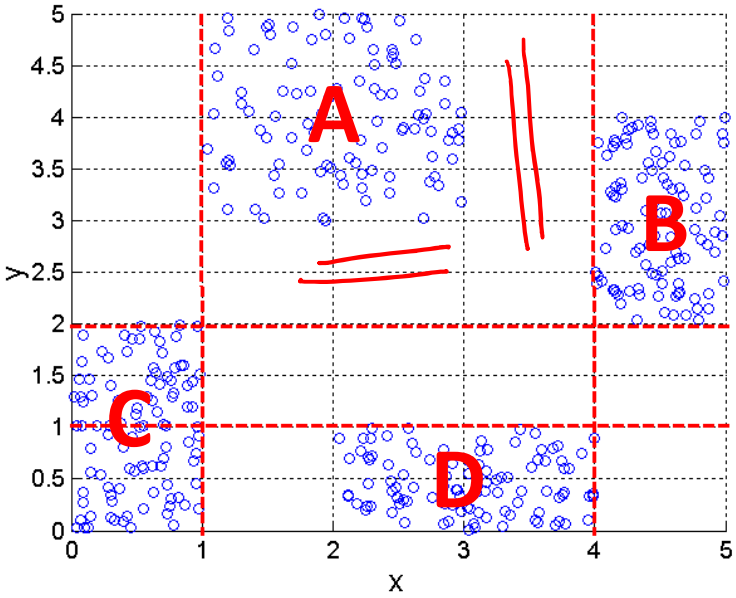
Discretization



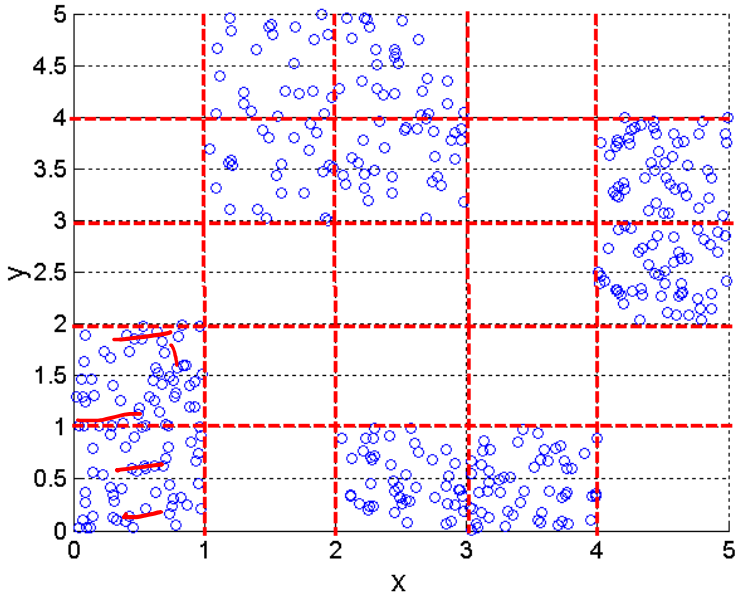
Step 5: Stop the process when a pre-specified number of intervals is reached (n = 4)

Discretization Using Entropy-based Approach Example

Independently discretize both x and y attributes of the two-dimensional data



3 categories for both x and y



5 categories for both x and y

Five intervals work better than three, but six intervals do not improve the discretization much in terms of entropy.

Discretization

Discretization of Categorical Attribute:

- Categorical attributes can sometimes have too many values; so, we need to reduce the number of attribute values.
- For the **ordinal attribute**, we can implement the algorithms similar to those for the continuous attribute.
- For the **nominal attribute**, we could use the knowledge of the relationships among different values to combine values into larger groups (domain knowledge approach). For example, combining a department name attribute into larger groups, such as engineering, social sciences.
- If domain knowledge approach results in poor classification performance, then **grouping values together** can be implemented only if such a grouping results in improved classification accuracy or achieves some other data mining objective.

Attribute Transformation

- **Variable transformation** refers to a transformation that apply a function which is used to map the entire set of values of a given attribute to a new set of values.
- The two types of variable transformations are:
  1. **Simple Functions:**
     - A simple mathematical function is applied to each value individually.
     - For examples, x^k, $\log x$, e^x, $\sqrt{x}$, $\frac{1}{x}$, $\sin x$, or $|x|$
     - Attribute transformations should be applied with caution since they change the nature of the data, for example, the transformation $\frac{1}{x}$ reduces the magnitude of values that are 1 or larger, but increases the magnitude of values between 0 and 1, thus for all set of values, the transformation $\frac{1}{x}$ reverses the order.

Attribute Transformation

2. Normalization or Standardization:

- The goal is to make an entire set of values have a particular property such a mean of 0 and a standard deviation of 1.

calculate distance

	age	height (cm)
1	15	160
2	20	180

$\sqrt{(160-180)^2 + (15-20)^2}$

original − average

$$x' = \frac{x - \bar{x}}{s_x}$$

standard deviation

normalize it → calculate

where $\bar{x}$ is the mean (average) of the attribute values

s_x is the standard deviation of the attribute values

- This transformation function creates a new set of values that has a mean of 0 and a standard deviation of 1.
- This transformation function is used to avoid having a variable with large values dominate the results of the calculation, for example, when we have two attributes – age and income - to represent each person. The income values will dominate the dissimilarity calculation.

Attribute Transformation

- Since mean and standard deviation are strongly affected by outliers.
- If the dataset contains some outliers, then the normalization formula could be modified as follows:

$$x' = \frac{x - M}{\sigma_A}$$

there are some outliers

where M is the median of the variable values

σ_A is the absolute standard deviation of the variable

$$\sigma_A = \sum_{i=1}^{m} |x_i - \mu|$$

where x_i is the i^{th} value of the variable

μ is either mean of the variable

Topics

- Data preprocessing:
- **Similarity and Dissimilarity**

Similarity and Dissimilarity

- **Similarity and dissimilarity** is important because they are used by a number of data mining techniques, such as clustering, nearest neighbor classification, and anomaly detection.

- **Proximity** refers to either similarity or dissimilarity.

- **Similarity**
  - The similarity between two objects is a numerical measure of the degree to which the two data objects are alike.
  - Similarities are higher for pairs of objects that are more alike.
  - Similarities are usually non-negative and are often falls in the range [0,1] where 0 means no similarity and 1 means complete similarity

Similarity and Dissimilarity

- **Dissimilarity**
  - The dissimilarity between two objects is a numerical measure of the degree to which the two objects are different.
  - Dissimilarities are lower for pairs of objects that are more alike.
  - The term **distance** is used as a synonym for dissimilarity
  - Minimum dissimilarity is often 0
  - Dissimilarities sometimes fall in the interval [0,1], but it is also common for them to range from 0 to ∞.

Transformation

Transformations are often applied

1) To transform a proximity (similarity or dissimilarity) measure to fall within a particular range, such as [0,1]
2) To convert a similarity to a dissimilarity
3) To convert a dissimilarity to a similarity

1. Converting into range [0,1]:

Similarity:

$$s' = \frac{(s - \min\_s)}{(\max\_s - min\_s)}$$

where s and s' are the original and new similarity values

$\max\_s$ and $\min\_s$ are the maximum and minimum similarity values

Transformation

Dissimilarity:

$$d' = \frac{(d - \min\_d)}{(\max\_d - min\_d)}$$

where d and d' are the original and new dissimilarity values

$\max\_d$ and $\min\_d$ are the maximum and minimum dissimilarity values

Dissimilarity that range from 0 to ∞ :

$$d' = \frac{d}{(1 + d)}$$

Transformation

2. Similarity ⇔ Dissimilarity:

- **If the values are already in the interval [0,1]:**

  1) We can use the followings formulas to convert values

  Similarity ⇒ Dissimilarity: $d = 1 - s$

  Dissimilarity ⇒ Similarity: $s = 1 - d$

  2) We can use **negation transformation**:

  It defines similarity as the negative of the dissimilarity (or vice versa); the results from this approach do not fall in interval [0,1].

- **If the values are not in the interval [0,1]:**

$$s = \frac{1}{d+1} \qquad s = e^{-d} \qquad s = 1 - \frac{d - \min\_d}{\max\_d - min\_d}$$

Similarity and Dissimilarity of Objects

x and y are two objects that have a single attribute

Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0, & if\ x = y \\ 1, & if\ x \neq y \end{cases}$	$s = \begin{cases} 1, & if\ x = y \\ 0, & if\ x \neq y \end{cases}$
Ordinal	$d = \frac{\lvert x - y \rvert}{(n-1)}$ (values mapped to integers 0 to $n-1$, when n is the number of values)	$s = 1 - d$
Interval or Ratio	$d = \lvert x - y \rvert$	$s = -d$ $s = \frac{1}{1+d}$ $s = e^{-d}$, $s = 1 - \frac{d - \min\_d}{\max\_d - \min\_d}$

Dissimilarities between Data Objects

Distance between two objects is considered as a dissimilarity. There are several distance measures:

1. Euclidean Distance:

$$d(x,y) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2}$$

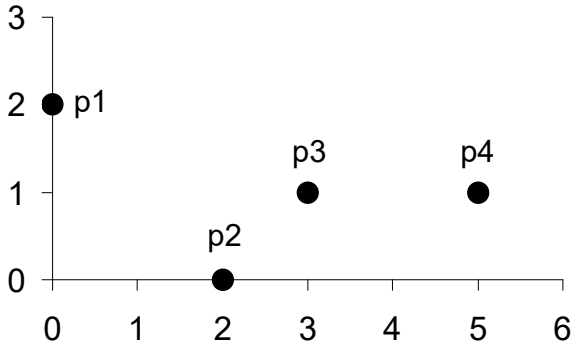
where n is the number of dimensions (attributes)

x_k and y_k are the k^{th} attribute values of data objects x and y, respectively

Standardization is necessary, if scales of attributes differ!

Dissimilarities between Data Objects

Example:



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

Distance Matrix

	p1	p2	p3	p4
p1	0.0	2.8	3.2	5.1
p2	2.8	0.0	1.4	3.2
p3	3.2	1.4	0.0	2.0
p4	5.1	3.2	2.0	0.0

This distance matrix is symmetric; the ij^{th} entry is the same as the ji^{th} entry.

Dissimilarities between Data Objects

2. Minkowski Distance is a generalization of Euclidean Distance

$$d(x, y) = \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{\frac{1}{r}}$$

where r is a parameter

n is the number of dimensions (attributes)

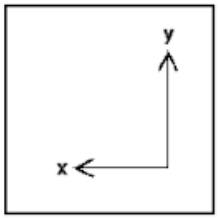
x_k and y_k are the k^{th} attributes (components) or data objects x and y

- Different r values means different ways of combining the differences in each dimension (attribute) into an overall distance.

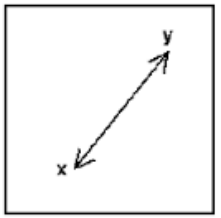
Dissimilarities between Data Objects

Three most commonly used *r* values of Minkowski Distance :

1) ***r* = 1** : City block (Manhattan, taxicab, L_1 norm) distance
2) ***r* = 2** : Euclidean (L_2 norm) distance
3) ***r* = ∞** : Supremum (L_{max} norm, L_∞ norm) distance
   - This is the maximum difference between any attribute of the objects



Manhattan



Euclidean

$$d(x,y) = \lim_{r\to\infty}\left(\sum_{k=1}^{n}|x_k - y_k|^r\right)^{\frac{1}{r}}$$

Dissimilarities between Data Objects

Example of Minkowski Distance using various r values:

4 objects, 2 attributes

point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

Distance Matrices

L1	p1	p2	p3	p4
p1	0	4	4	6
p2	4	0	2	4
p3	4	2	0	2
p4	6	4	2	0

r = 1

L2	p1	p2	p3	p4
p1	0.0	2.8	3.2	5.1
p2	2.8	0.0	1.4	3.2
p3	3.2	1.4	0.0	2.0
p4	5.1	3.2	2.0	0.0

r = 2

L_∞	p1	p2	p3	p4
p1	0	2	3	5
p2	2	0	1	3
p3	3	1	0	2
p4	5	3	2	0

r = ∞

Dissimilarities between Data Objects

Common Properties of a Distance

1. Positivity:

(a) $d(x, y) \geq 0$ for all x and y

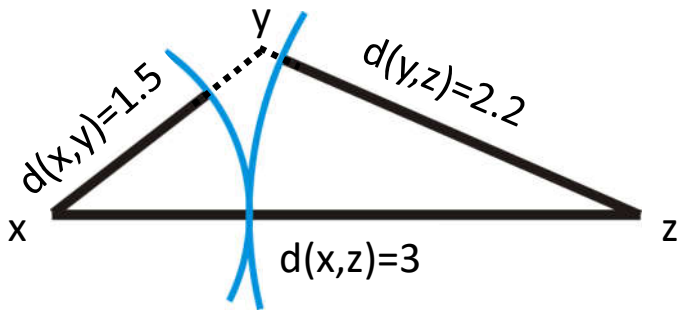
(b) $d(x, y) = 0$ only if $x = y$

2. Symmetry:

$d(x, y) = d(y, x)$ for all x and y

3. Triangle Inequality:

$d(x, z) \leq d(x, y) + d(y, z)$ for all points x, y, and z



Note that

- Measures that satisfy all three properties are known as metrics.
- Dissimilarity, such as the Euclidean distance, have all these properties.
- Many dissimilarities do not satisfy one or more of the metric properties
- For similarities, the triangle inequality typically does not hold, but positivity and symmetry do.

Similarity Measures for Binary Data

- **Similarity measures** between objects that contain only binary attributes are called **similarity coefficients**, and typically have values between 0 and 1 where a value of 1 indicates that the two objects are very similar, while a value of 0 indicates that the objects are not at all similar.

- Let x and y be two objects that consists of n binary attributes; this leads to the following four quantities (frequencies):

  f_{00} = the number of attributes where x is 0 and y is 0

  f_{01} = the number of attributes where x is 0 and y is 1

  f_{10} = the number of attributes where x is 1 and y is 0

  f_{11} = the number of attributes where x is 1 and y is 1

Similarity Measures for Binary Data

Two widely used similarity measures for binary data are:

1. Simple Matching Coefficient (SMC): Counts both presences and absences equally

$$SMC = \frac{number\ of\ matching\ attribute\ values}{number\ of\ attributes} = \frac{f_{11} + f_{00}}{f_{01} + f_{10} + f_{11} + f_{00}}$$

2. Jaccard Coefficient (J): This measure is used for asymmetric binary attributes where f_{00} far outnumbers f_{01} , f_{10} , f_{11}.

$$J = \frac{number\ of\ matching\ presences}{number\ of\ attributes\ not\ involved\ in\ 00\ matches} = \frac{f_{11}}{f_{01} + f_{10} + f_{11}}$$

Similarity Measures for Binary Data

Example:

x= 1 0 0 0 0 0 0 0 0 0 10 attributes

y= 0 0 0 0 0 0 1 0 0 1

f_{00} = 7 (the number of attributes where x is 0 and y is 0)

f_{01} = 2 (the number of attributes where x is 0 and y is 1)

f_{10} = 1 (the number of attributes where x is 1 and y is 0)

f_{11} = 0 (the number of attributes where x is 1 and y is 1)

$$SMC = \frac{f_{11} + f_{00}}{f_{01} + f_{10} + f_{11} + f_{00}}$$

$$= \frac{0+7}{2+1+0+7} = 0.7$$

$$J = \frac{f_{11}}{f_{01} + f_{10} + f_{11}}$$

$$= \frac{0}{2+1+0} = 0$$

Similarity Measures for Count Data

1. **Cosine Similarity** is similar to Jaccard Index which is trying to capture presences or absences of items.

$$cos(x, y) = \frac{(x \cdot y)}{\|x\|\|y\|}$$

where $x \cdot y = \sum_{k=1}^{n} x_k y_k$ where n is number of attributes

$\|x\|$ and $\|y\|$ are the length of vector x and y, $\|x\| = \sqrt{\sum_{k=1}^{n} x_k^2} = \sqrt{x \cdot x}$

- Cosine similarity is a measure of the angle between vector x and y.

  $cos(x, y) = 1$ means angle between x and y is 0°; x and y are the same except for magnitude (length)

  $cos(x, y) = 0$ means angle between x and y is 90°; x and y are completely different

Similarity Measures for Count Data

- **Cosine Similarity** can also be written as:

$$cos(x, y) = \frac{x}{\|x\|} \cdot \frac{y}{\|y\|} = x' \cdot y'$$

where $x' = \frac{x}{\|x\|}$ and $y' = \frac{y}{\|y\|}$

Dividing x and y by their lengths normalizes them to have a length of 1. This means that cosine similarity does not take the magnitude (size) of the two data objects into account when computing similarity.

Similarity Measures for Count Data

Example:

d_1 = 3 2 0 5 2 0

d_2 = 1 0 0 0 1 2

$$d_1 \cdot d_2 = (3 \times 1) + (2 \times 0) + (0 \times 0) + (5 \times 0) + (2 \times 1) + (0 \times 2) = 5$$

$$\|d_1\| = \sqrt{3^2 + 2^2 + 0^2 + 5^2 + 2^2 + 0^2} = 6.48$$

$$\|d_2\| = \sqrt{1^2 + 0^2 + 0^2 + 0^2 + 1^2 + 2^2} = 2.45$$

$$cos(d_1, d_2) = \frac{5}{6.48 \times 2.45} = 0.315$$

Similarity Measures for Count Data

2. Extended Jaccard Coefficient (aka Tanimoto Coefficient): This is a variation of Jaccard for count attributes.

$$EJ(x, y) = \frac{x \cdot y}{\|x\|^2 + \|y\|^2 - (x \cdot y)}$$

where $x \cdot y = \sum_{k=1}^{n} x_k y_k$ (n is number of attributes)

$\|x\|$ and $\|y\|$ are the length of vector x and y, $\|x\| = \sqrt{\sum_{k=1}^{n} x_k^2} = \sqrt{x \cdot x}$

Similarity Measures

- **Correlation** between two data objects that have binary or continuous attributes is a measure of the linear relationship between the attributes of the objects.

- Correlation is always in the range -1 to 1:
  - Correlation of 1 means that x and y have a perfect positive linear relationship between the attributes (objects)
  - Correlation of -1 means that x and y have a perfect negative linear relationship between the attributes
  - Correlation of 0 means that x and y have no linear relationship between the attributes (x and y are not correlated)

Similarity Measures

1. Pearson's correlation:

$$corr(x,y) = \frac{covariance(x,y)}{SD(x)SD(y)}$$

$$covariance(x,y) = \frac{1}{n-1}\sum_{k=1}^{n}(x_k - \bar{x})\,(y_k - \bar{y})$$

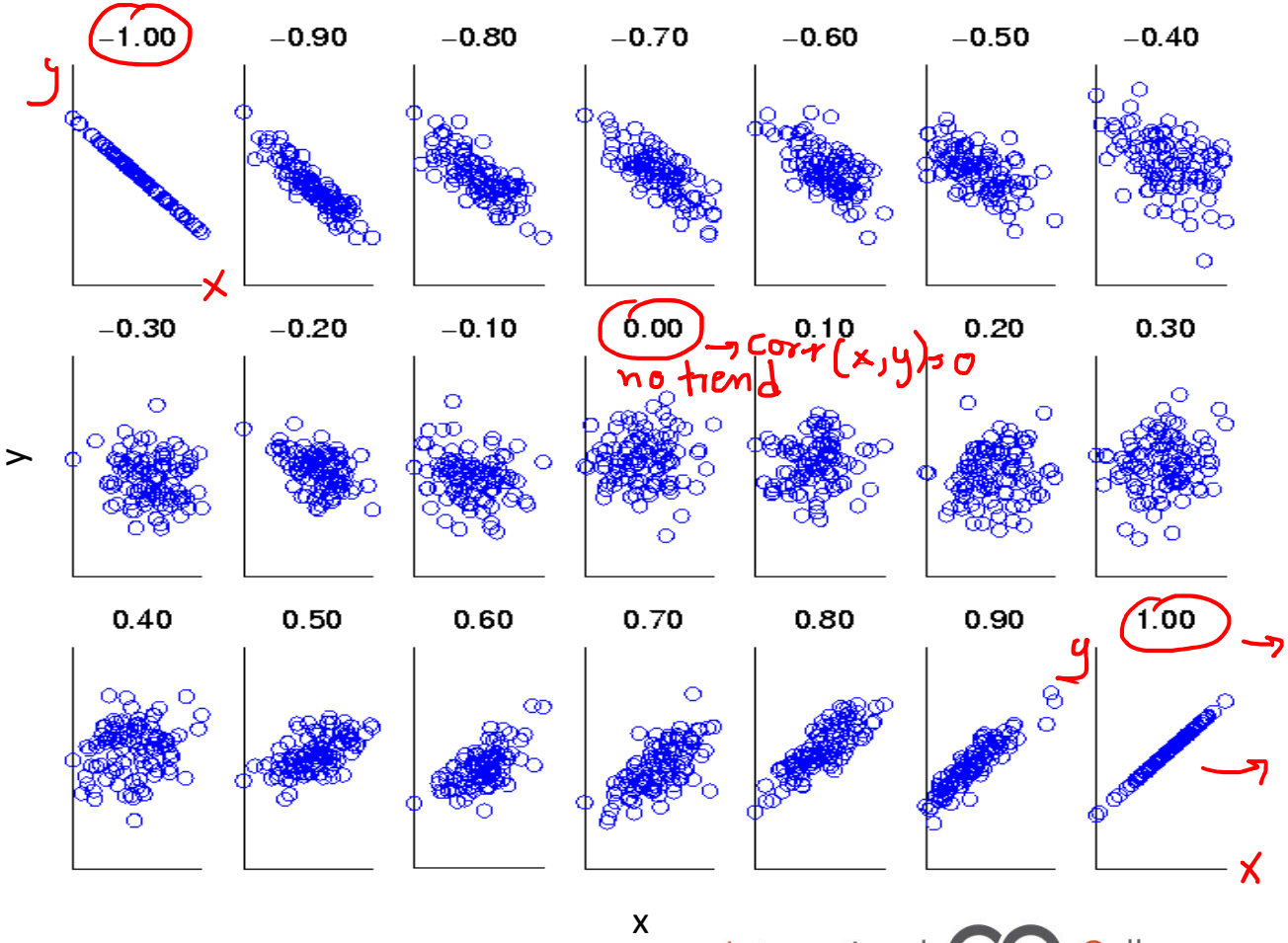
$$SD(x) = \sqrt{\frac{1}{n-1}\sum_{k=1}^{n}(x_k - \bar{x})^2}$$

$$SD(y) = \sqrt{\frac{1}{n-1}\sum_{k=1}^{n}(y_k - \bar{y})^2}$$

where n is number of attributes

$\bar{x}$ and $\bar{y}$ are the mean (average) of x and y, respectively

Visually Evaluating Correlation between Two Objects



X and y have 30 attributes

Each circle in a plot represents one of the 30 attributes

Scatter plots showing similarity between x and y ranging from -1 to 1