2.4 Data Preprocessing

- Aggregation
- Sampling
- Dimensionality Reduction
- Feature subset selection
- Feature creation
- Discretization and Binarization
- Attribute Transformation

© UAB. All Rights Reserved.

Aggregation

- **Less is more**: Combining two or more attributes (or objects) into a single attribute (or object)

Table 2.4. Data set containing information about customer purchases.

Transaction ID	Item	Store Location	Date	Price	...
⋮	⋮	⋮	⋮	⋮	
101123	Watch	Chicago	09/06/04	$25.99	...
101123	Battery	Chicago	09/06/04	$5.99	...
101124	Shoes	Minneapolis	09/06/04	$75.00	...
⋮	⋮	⋮	⋮	⋮	

© UAB. All Rights Reserved.

Aggregation

- **Less is more**: Combining two or more attributes (or objects) into a single attribute (or object)
- Purpose
  - Data reduction – less memory and processing time, more expensive data analysis techniques

© UAB. All Rights Reserved.

Aggregation

- **Less is more**: Combining two or more attributes (or objects) into a single attribute (or object)
- Purpose
  - Data reduction – less memory and processing time, more expensive data analysis techniques
  - Change of scale – low level view to high level view
    - Cities aggregated into regions, states, countries, etc.
    - Days aggregated into weeks, months, or years

© UAB. All Rights Reserved.

Aggregation

- **Less is more**: Combining two or more attributes (or objects) into a single attribute (or object)
- Purpose
  - Data reduction – less memory and processing time, more expensive data analysis techniques
  - Change of scale – low level view to high level view
    - Cities aggregated into regions, states, countries, etc.
    - Days aggregated into weeks, months, or years
  - Stability – groups of objects or attributes is often more stable

© UAB. All Rights Reserved.

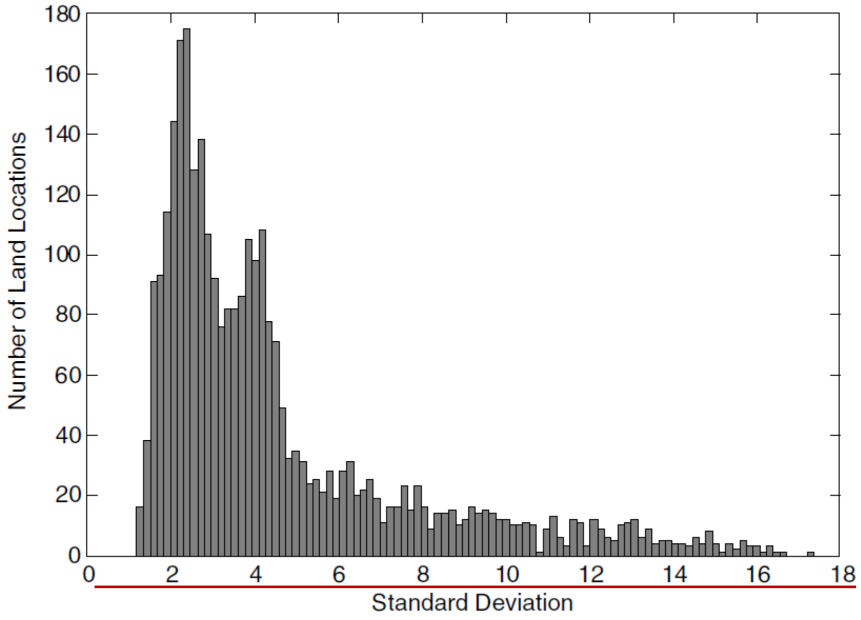
Aggregation

- **Less is more**: Combining two or more attributes (or objects) into a single attribute (or object)
- **Disadvantage**: Potential loss of interesting details.

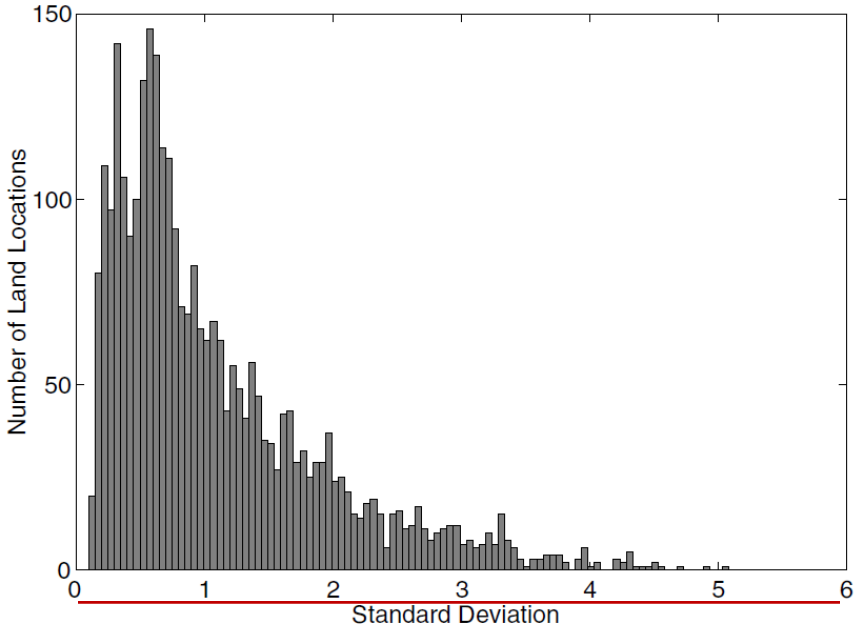
© UAB. All Rights Reserved.

Example: Precipitation in Australia

- This example is based on precipitation in Australia from the period 1982 to 1993.



(a) Histogram of standard deviation of average monthly precipitation



(b) Histogram of standard deviation of average yearly precipitation

© UAB. All Rights Reserved.

Sampling

- Select a **subset** of data objects.
- Sampling is the main technique employed for **data reduction**.
- Allow usage of more expensive algorithms.

© UAB. All Rights Reserved.

Sampling

- The key principle for effective sampling:
  - Choose **representative** samples.
  - A sample is **representative** if it has the same properties as the set of data.
  - The representativeness will vary.

© UAB. All Rights Reserved.

Sampling

- The key principle for effective sampling:
  - Choose **representative** samples.
  - A sample is **representative** if it has the same properties as the set of data.
  - The representativeness will vary.

Q: How to guarantee a high probability of getting representative samples?

© UAB. All Rights Reserved.

Sampling Approaches

- **Simple Random Sampling**: There is an **equal probability** of selecting any particular item.
  1. Sampling without replacement
     - As each item is selected, it is removed from the population
  2. Sampling with replacement
     - Items are not removed from the population.
     - The same item can be picked up more than once.

© UAB. All Rights Reserved.

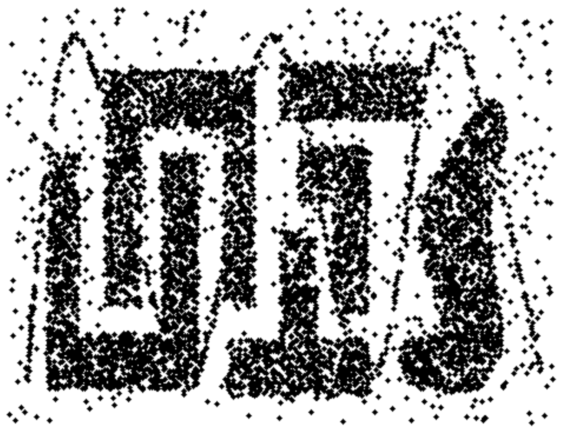
Sampling Approaches

- **Simple Random Sampling**: There is an **equal probability** of selecting any particular item.
    1. Sampling without replacement
    2. Sampling with replacement
- **Stratified sampling**:
    - Split the data into several partitions
    - Draw random samples from each partition. The number of selected objects can be equal or proportional to the group size.

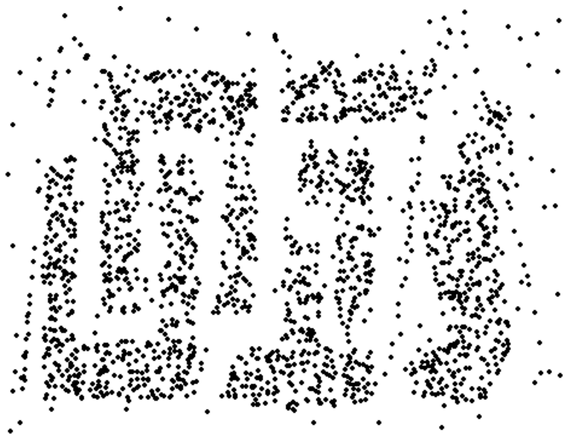
© UAB. All Rights Reserved.

Sample Size

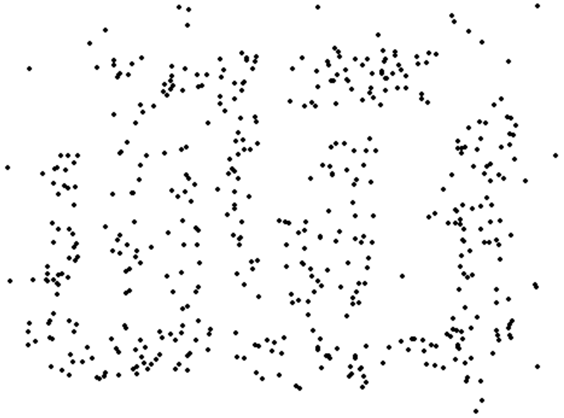
- Larger sample sizes: keep representativeness but lose advantage of sampling.
- Smaller sample sizes: Patterns may be missed or wrong patterns can be detected



8000 points



2000 Points



500 Points

© UAB. All Rights Reserved.

Dimensionality Reduction

- Datasets can have a large number of features:
  - Documents represented by vectors whose components are the word frequencies.
  - Time series of daily stock price over 30 years.

© UAB. All Rights Reserved.

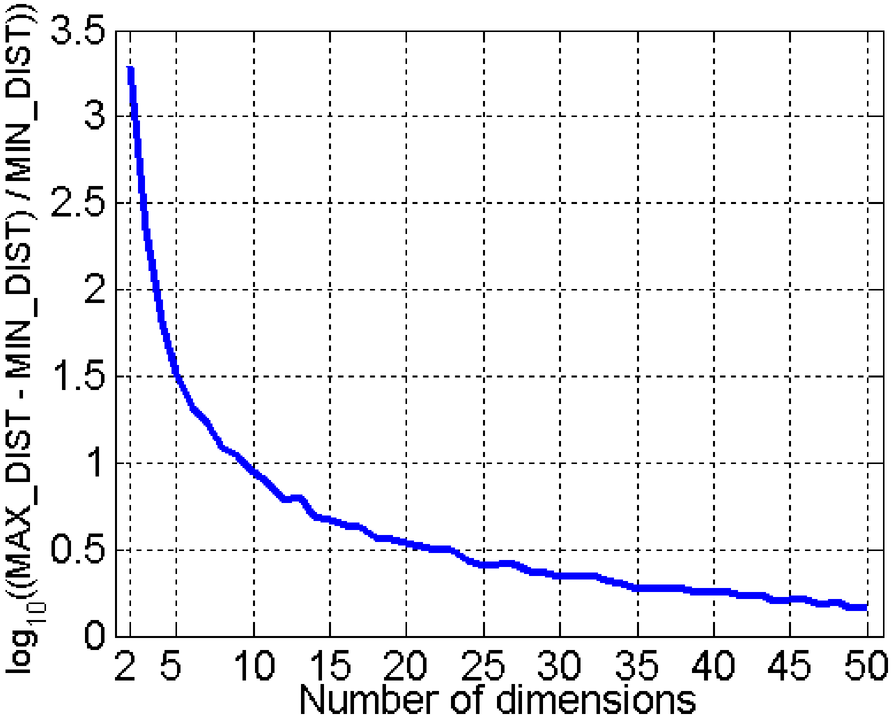
Dimensionality Reduction

- Datasets can have a large number of features:
  - Documents represented by vectors whose components are the word frequencies.
  - Time series of daily stock price over 30 years.
- Purpose of dimensionality reduction:
  - Eliminate irrelevant features and reduce noises
  - Mitigate curse of dimensionality
  - Lead to more understandable model
  - Easy visualization
  - Save time and memory

© UAB. All Rights Reserved.

Curse of Dimensionality

- Data analysis becomes **harder** as dimensionality increases.
  - Data becomes **increasingly sparse** in the high-dimensional space.
  - Definitions of **density** and **distance** between points become **less meaningful**



Difference between max and min distance vs. the number of dimensions

© UAB. All Rights Reserved.

Dimensionality Reduction

- Techniques
  - Singular Value Decomposition (SVD)
  - Principal Components Analysis (PCA)
  - Others: supervised and non-linear techniques

© UAB. All Rights Reserved.

Feature Subset Selection

- Another way to **reduce dimensionality** of data
  - **Redundant** features
  - **Irrelevant** features

© UAB. All Rights Reserved.

Feature Subset Selection

- Redundant features
  - **Duplicate** information contained in other attributes
  - Example: purchase price of a product and the amount of sales tax paid
- Irrelevant features
  - Contain **no useful** information
  - Example: students' ID is often irrelevant to the task of predicting students' GPA

© UAB. All Rights Reserved.

Feature Subset Selection

- Naïve: try all possible 2^n subsets of features

© UAB. All Rights Reserved.

Feature Subset Selection

- ~~Naïve: try all possible 2^n subsets of features~~
- Embedded approaches
- Filter approaches
- Wrapper approaches

© UAB. All Rights Reserved.

Feature Subset Selection

- ~~Naïve: try all possible 2^n subsets of features~~
- Embedded approaches
  - Feature selection occurs **as part of** the data analysis algorithm.
- Filter approaches
  - Features are selected **independently** from the data analysis algorithm.
- Wrapper approaches
  - Use the target algorithm as a black box to find the best subset of attributes.

© UAB. All Rights Reserved.

Feature Subset Selection

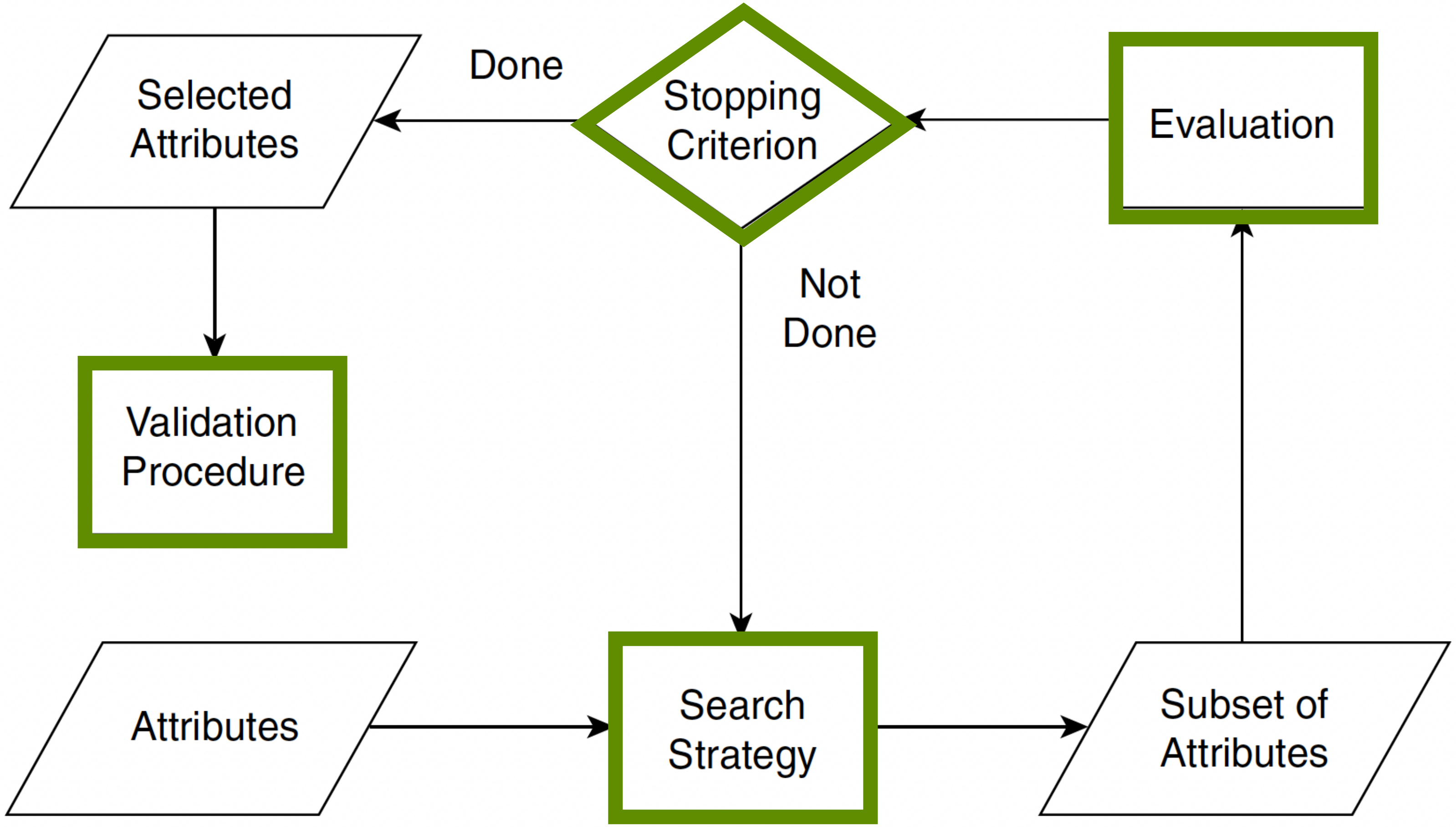


Figure 2.11. Flowchart of a feature subset selection process.

© UAB. All Rights Reserved.

Feature Subset Selection

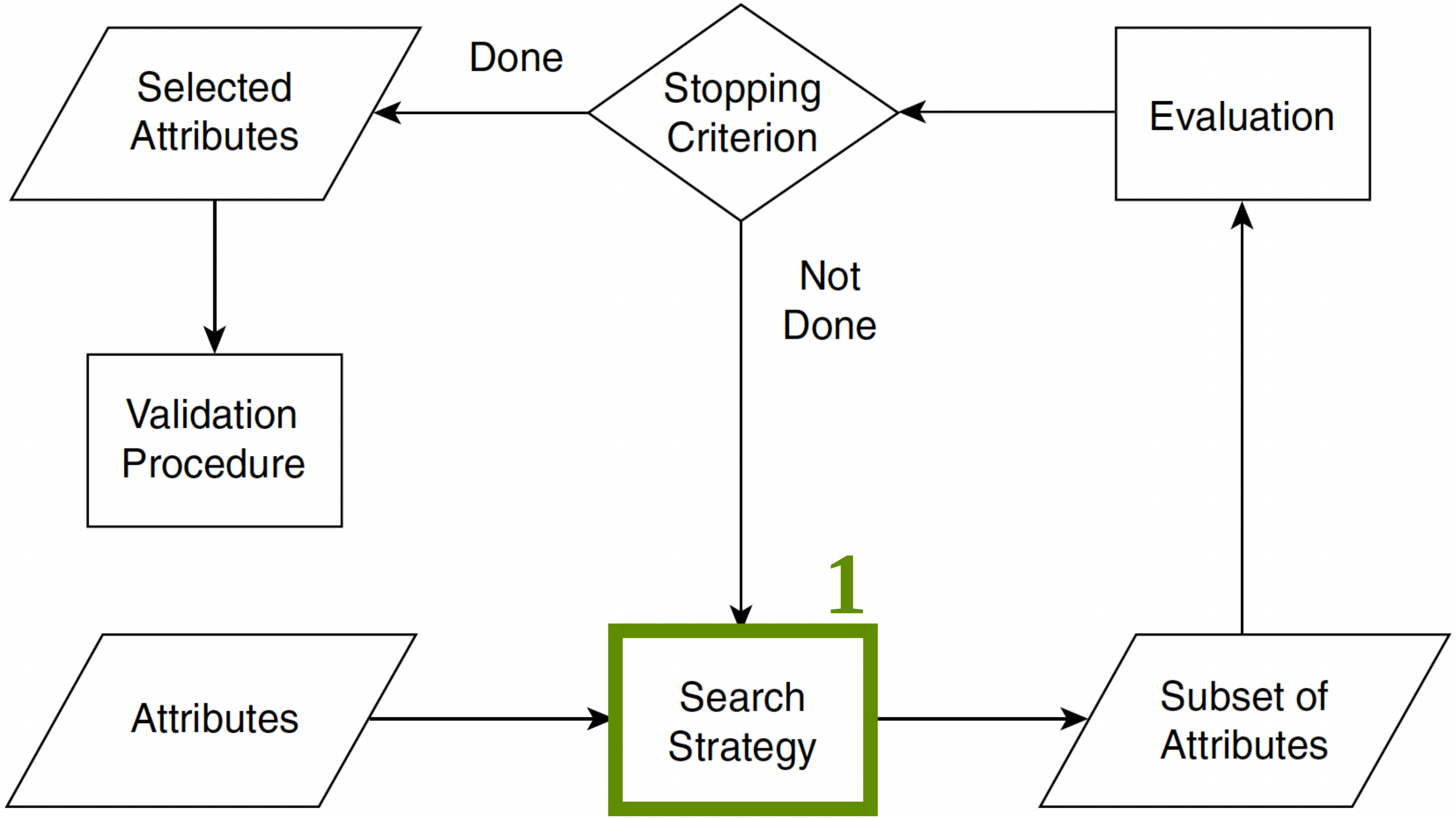


Figure 2.11. Flowchart of a feature subset selection process.

© UAB. All Rights Reserved.

Feature Subset Selection

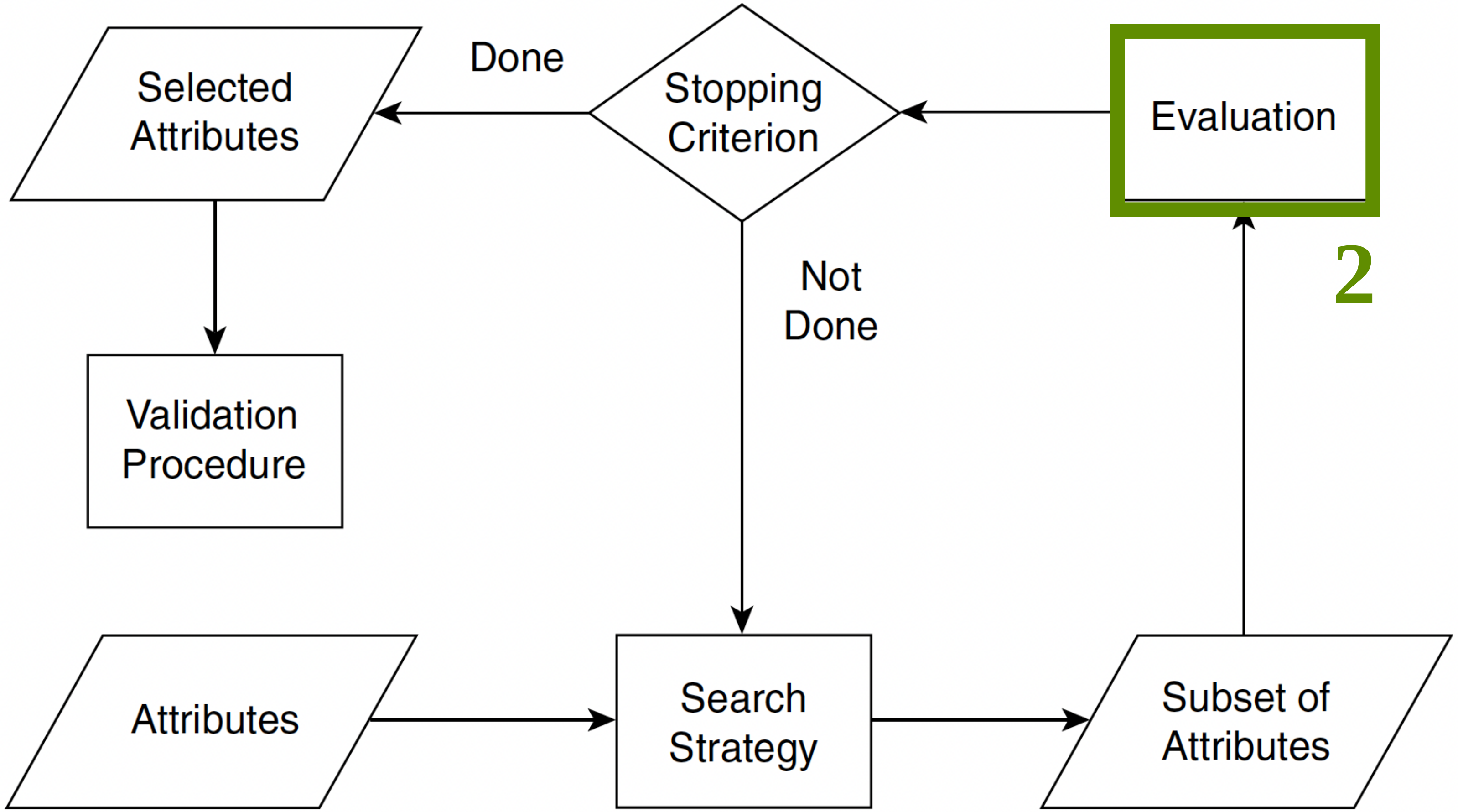


Figure 2.11. Flowchart of a feature subset selection process.

© UAB. All Rights Reserved.

Feature Subset Selection

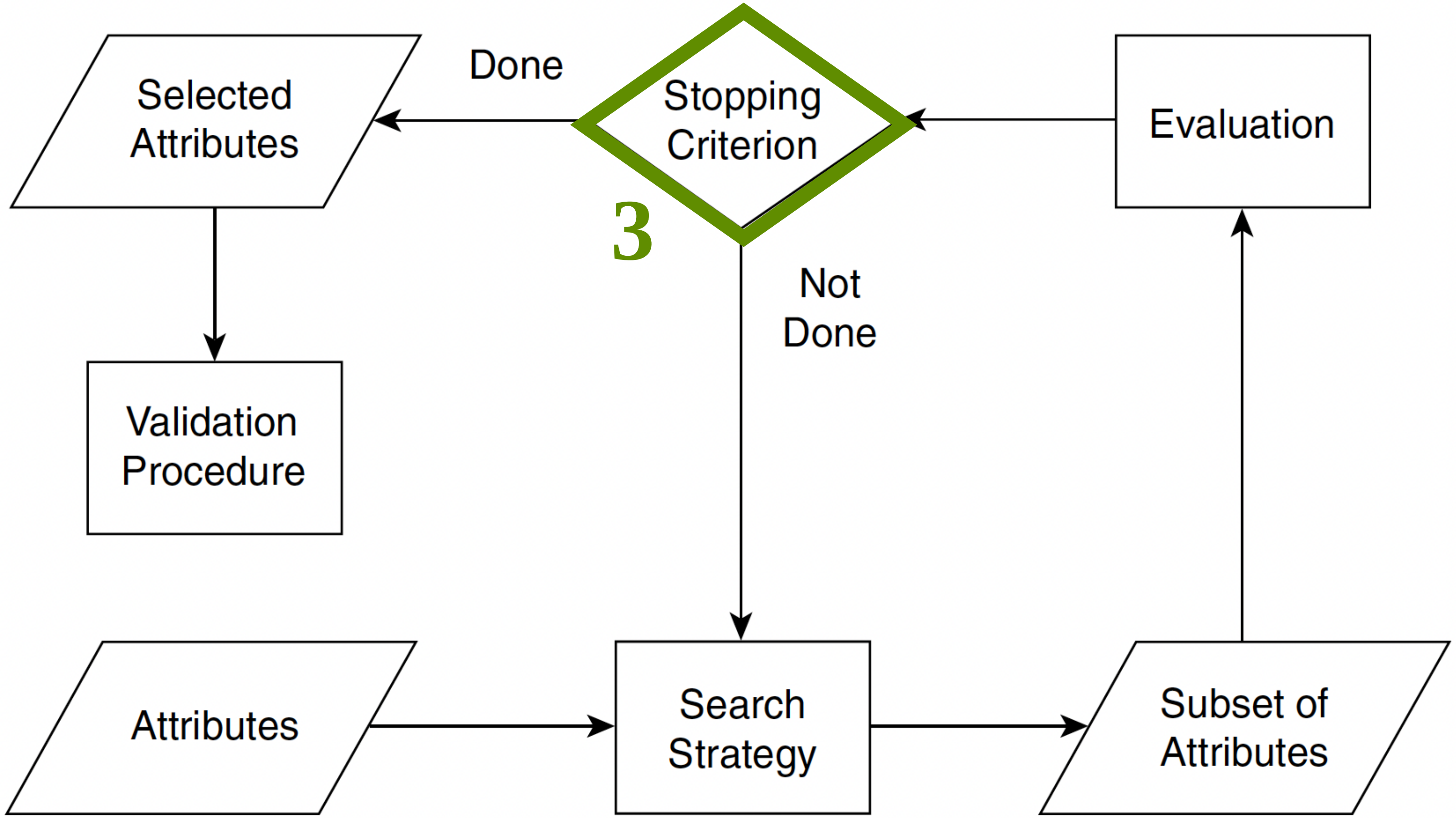


Figure 2.11. Flowchart of a feature subset selection process.

© UAB. All Rights Reserved.

Feature Subset Selection

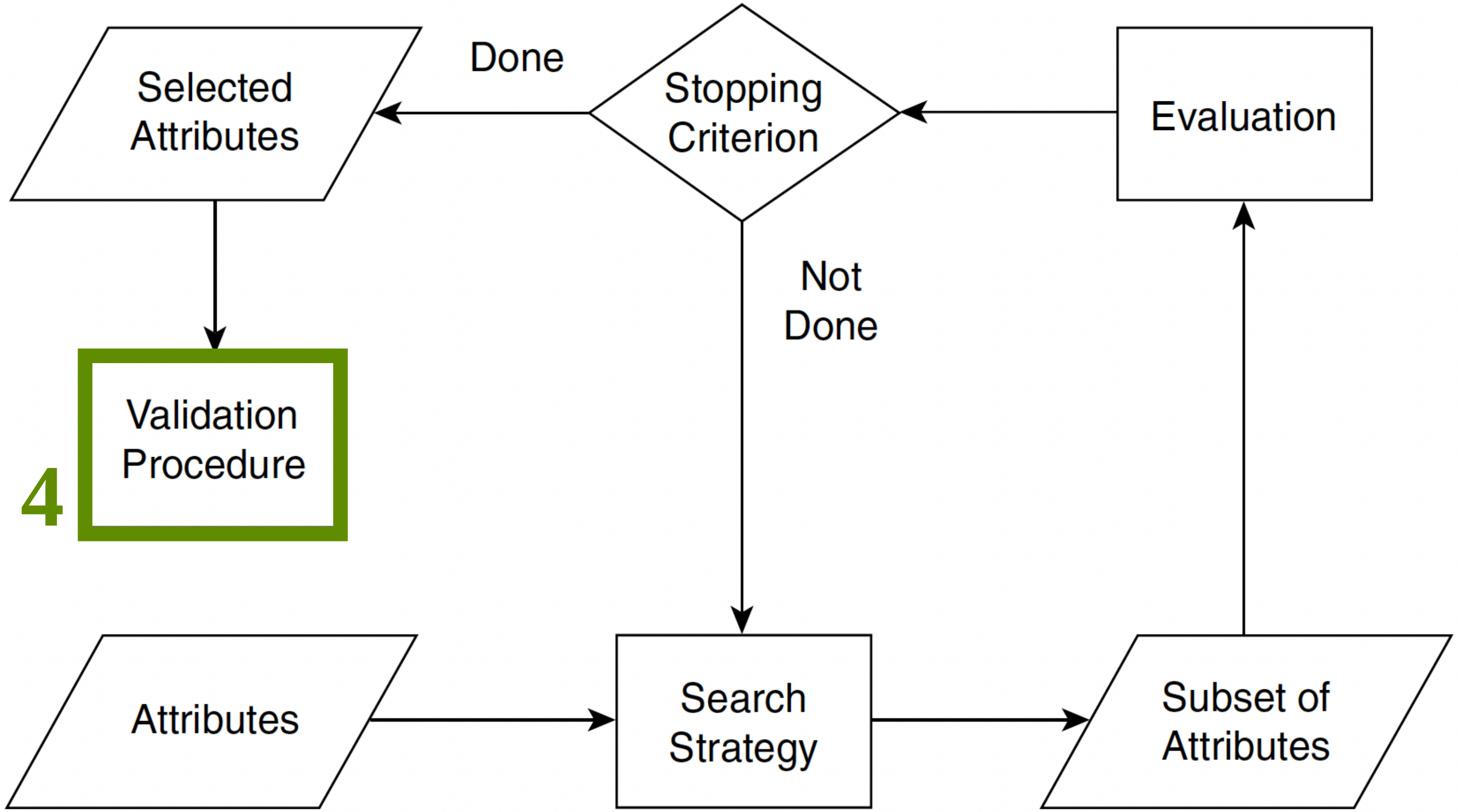


Figure 2.11. Flowchart of a feature subset selection process.

© UAB. All Rights Reserved.

Feature Creation

- Create new attributes that can capture the important information more efficiently
- Three general methodologies:
  - Feature extraction
  - Feature construction
  - Mapping data to new space

© UAB. All Rights Reserved.

Feature Creation

- Feature extraction: extract new features from the original raw data
  - Example: extracting edges from images
- Feature construction
- Mapping data to new space

© UAB. All Rights Reserved.

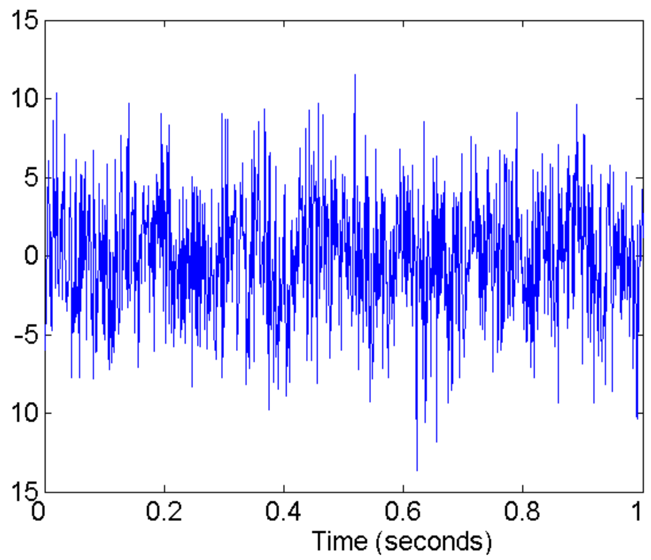
Feature Creation

- Feature extraction
- Feature construction: construct new features based on the original features
  - Example: density = mass/volume
- Mapping data to new space

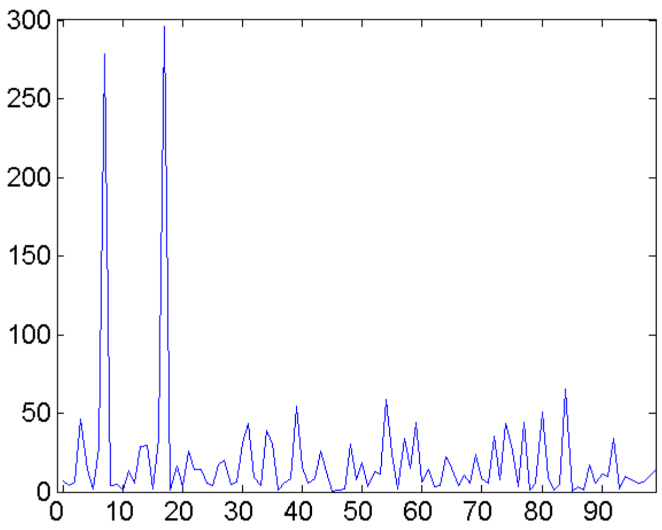
© UAB. All Rights Reserved.

Feature Creation

- Feature extraction
- Feature construction
- Mapping data to new space: apply transforms to get a different view of data
  - Example: Fourier transform



Two Sine Waves + Noise



Frequency

© UAB. All Rights Reserved.

Discretization and Binarization

- Categorical attribute could benefit classification.
- Binary attribute could benefit association pattern discovery.
- **Discretization**: transform a continuous attribute into a categorical attribute
- **Binarization**: transform both continuous and discrete attributes into binary attributes

© UAB. All Rights Reserved.

Binarization

- Transform both continuous and discrete attributes into binary attributes
- Assign each categorical value (out of m values) to an integer in [0, m-1]
- n = log2(m) binary digits are required

Table 2.5. Conversion of a categorical attribute to three binary attributes.

Categorical Value	Integer Value	x_1	x_2	x_3
awful	0	0	0	0
poor	1	0	0	1
OK	2	0	1	0
good	3	0	1	1
great	4	1	0	0

© UAB. All Rights Reserved.

Binarization

- Transform both continuous and discrete attributes into binary attributes
- Assign each categorical value (out of m values) to an integer in [0, m-1]
- n = log2(m) binary digits are required

Table 2.5. Conversion of a categorical attribute to three binary attributes.

Categorical Value	Integer Value	x_1	x_2	x_3
awful	0	0	0	0
poor	1	0	0	1
OK	2	0	1	0
good	3	0	1	1
great	4	1	0	0

© UAB. All Rights Reserved.

Binarization

- Transform both continuous and discrete attributes into binary attributes
- Assign each categorical value (out of m values) to an integer in [0, m-1]
- n = m binary digits are required

Table 2.6. Conversion of a categorical attribute to five asymmetric binary attributes.

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

© UAB. All Rights Reserved.

Discretization

- Transform a continuous attribute into a categorical attribute
  1. Decide how many categories
  2. Decide how to map continuous values to these categories

© UAB. All Rights Reserved.

Discretization

- Transform a continuous attribute into a categorical attribute
  1. Suppose we sort the continuous values and have n intervals (categories)
  2. We map all the values in one interval to the same categorical value

$$\{(x_0, x_1], (x_1, x_2], \ldots, (x_{n-1}, x_n)\}, where\ x_0\ and\ x_n\ can\ be + \infty\ and\ - \infty$$

© UAB. All Rights Reserved.

Discretization

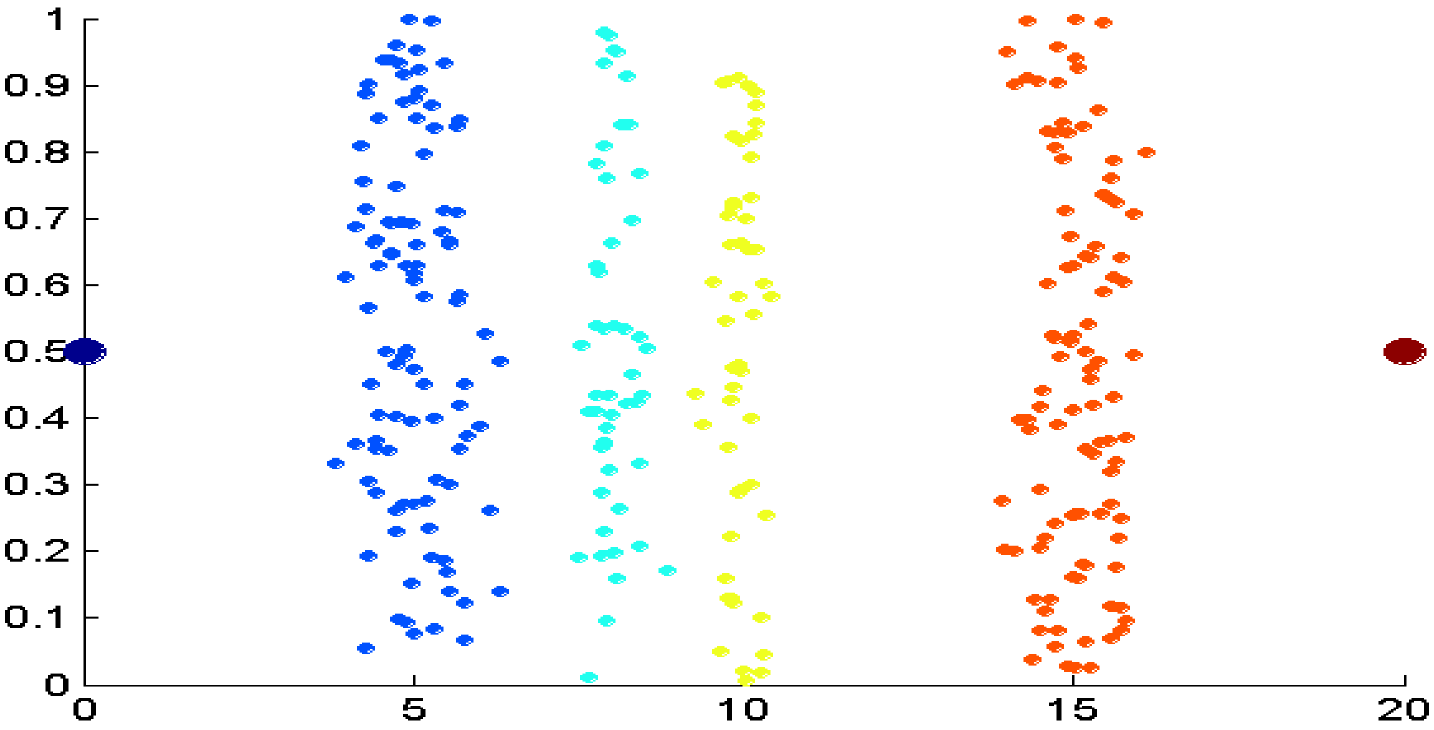
- Transform a continuous attribute into a categorical attribute
  1. Suppose we sort the continuous values and have n intervals (categories)
  2. We map all the values in one interval to the same categorical value

     $\{(x_0, x_1], (x_1, x_2], \ldots, (x_{n-1}, x_n)\}, where\ x_0\ and\ x_n\ can\ be + \infty\ and\ - \infty$

Challenge is to choose appropriate n.

© UAB. All Rights Reserved.

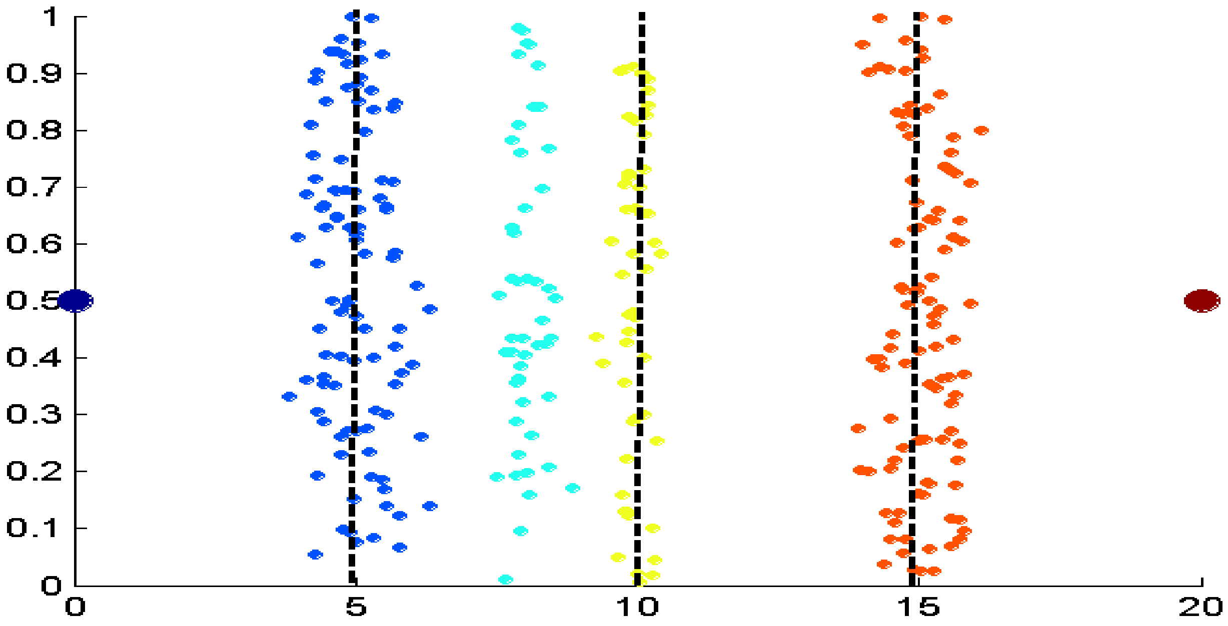
Unsupervised Discretization



Data consists of four groups of points and two outliers.

© UAB. All Rights Reserved.

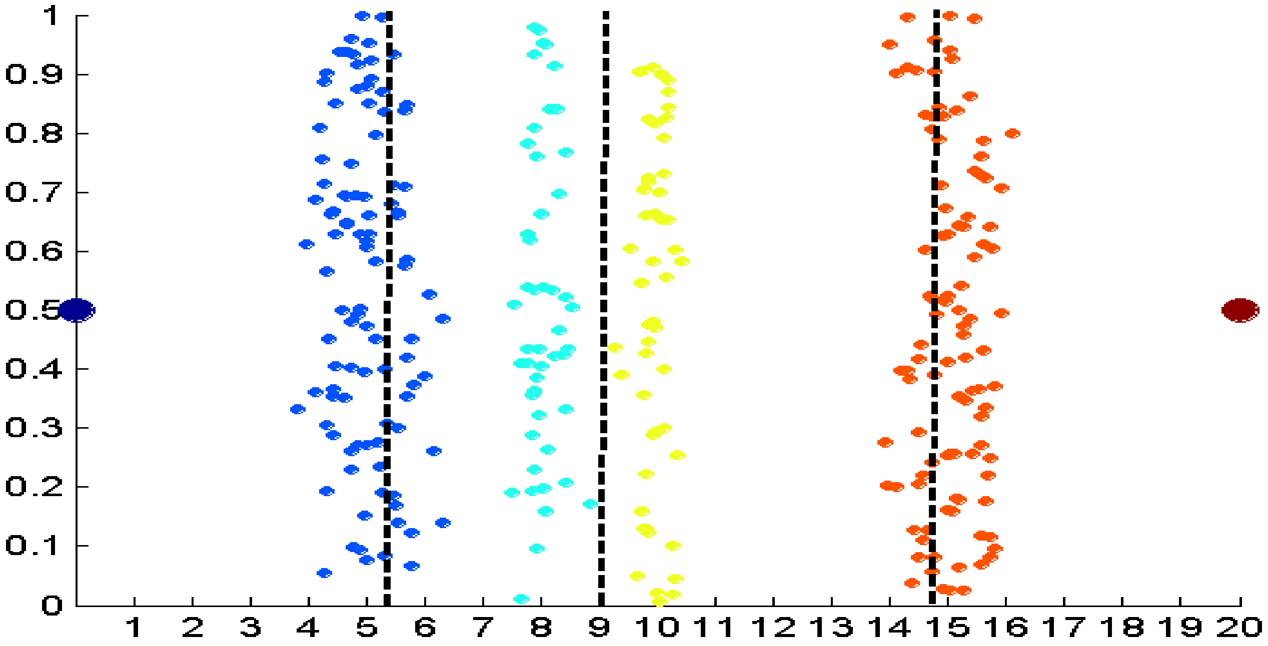
Unsupervised Discretization



Equal interval width approach used to obtain 4 values.

© UAB. All Rights Reserved.

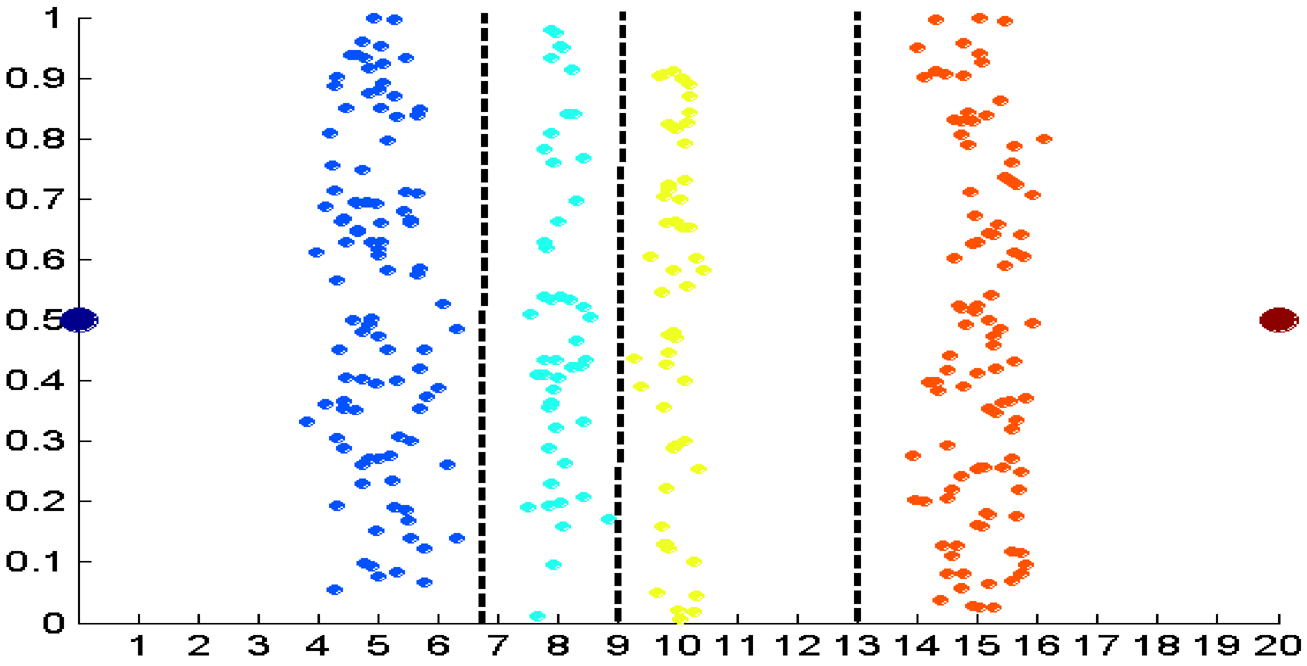
Unsupervised Discretization



Equal frequency approach used to obtain 4 values.

© UAB. All Rights Reserved.

Unsupervised Discretization

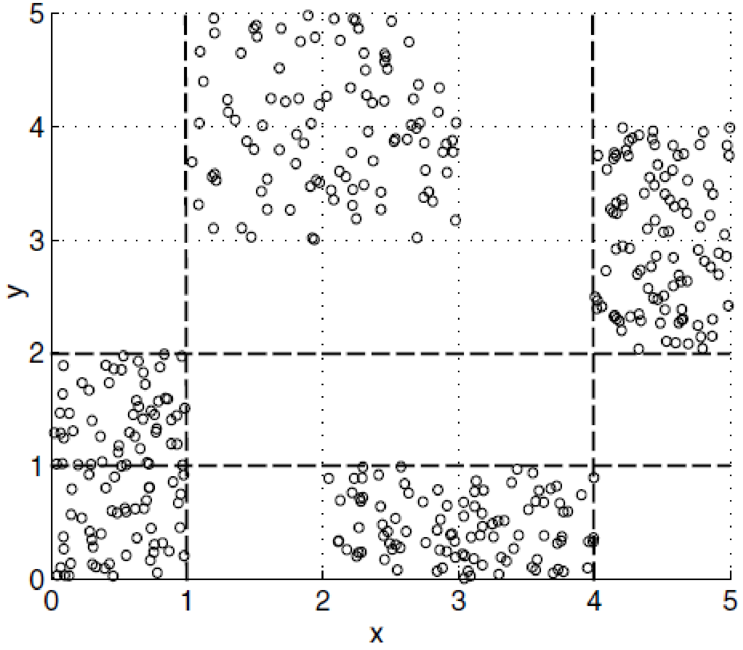


K-means approach to obtain 4 values.

© UAB. All Rights Reserved.

Supervised Discretization

- Maximize the purity of the intervals – the intervals mostly contain data from the same class.



(a) Three intervals

© UAB. All Rights Reserved.

Attribute Transformation

- An attribute transform is a transformation applied to all the values of an attribute.
  - Simple functions
  - Normalization

© UAB. All Rights Reserved.

Attribute Transformation

- Simple functions:
  - Apply a simple math function to each value individually.
  - x^k, $\log(x)$, e^x, $|x|$, $\sqrt{x}$, $\frac{1}{x}$, $\sin x$, ...
  - Usually used on non-Gaussian distributed attributes
  - Change the nature of data (e.g., $\log_{10}(x)$, $\frac{1}{x}$)

© UAB. All Rights Reserved.

Attribute Transformation

- Normalization (Standardization)
  - μ is mean and δ is standard deviation of values of an attribute.
  - $x' = \frac{x-\mu}{\delta}$, x' is the new attribute having a mean of 0 and a standard deviation of 1.
  - Avoid large attribute dominating the results.

© UAB. All Rights Reserved.

Attribute Transformation

- Normalization (Standardization)
  - μ is mean and δ is standard deviation of values of an attribute.
  - $x' = \frac{x-\mu}{\delta}$, x' is the new attribute having a mean of 0 and a standard deviation of 1.
  - Avoid large attribute dominating the results.
  - Affected by outliers
  - Variation:
    - mean -> median
    - standard deviation -> absolute standard deviation: $\delta_A = \sum|x_i - \mu|$, μ is mean or median

© UAB. All Rights Reserved.

2.5 Similarity and Dissimilarity

- Proximity refers to a similarity or dissimilarity
- Proximity between objects => proximity between corresponding attributes

© UAB. All Rights Reserved.

2.5 Similarity and Dissimilarity

- Similarity measure
  - Numerical measure of how alike two data objects are.
  - Is higher when objects are more alike.
  - Often falls in the range [0,1]
- Dissimilarity (Distance) measure
  - Numerical measure of how different two data objects are
  - Lower when objects are more alike
  - Minimum dissimilarity is often 0
  - Upper limit varies

© UAB. All Rights Reserved.

Similarity/Dissimilarity for Simple Attributes

- The following table shows the similarity and dissimilarity between two objects, x and y, with respect to a single, simple attribute.

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

© UAB. All Rights Reserved.

Similarity/Dissimilarity for Simple Attributes

- The following table shows the similarity and dissimilarity between two objects, x and y, with respect to a single, simple attribute.

Attribute Type	**Dissimilarity**	**Similarity**
Nominal	$d = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x \neq y \end{cases}$	$s = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$
Ordinal	$d = \|x - y\|/(n - 1)$ (values mapped to integers 0 to $n-1$, where n is the number of values)	$s = 1 - d$

Example: {poor=0, fair=1, OK=2, good=3, wonderful=4}
P1 is rated wonderful and P2 is rated good
D(P1, P2) = (4-3)/4=0.25

© UAB. All Rights Reserved.

Similarity/Dissimilarity for Simple Attributes

- The following table shows the similarity and dissimilarity between two objects, x and y, with respect to a single, simple attribute.

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

© UAB. All Rights Reserved.

Dissimilarity between objects

- Distance metrics
  - Euclidean distance
  - Minkowski distance
  - Hamming distance
- Distance properties
- Distance definition

© UAB. All Rights Reserved.

Euclidean Distance

- Euclidean Distance

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

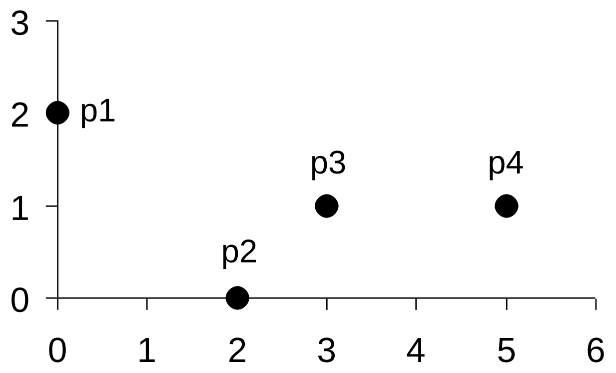
where n is the number of dimensions (attributes) and x_k and y_k are, respectively, the kth attributes (components) or data objects x and y.

- Standardization is necessary, if attribute scales differ.

© UAB. All Rights Reserved.

Euclidean Distance

- Example



Data points

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

	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Distance Matrix

$$d(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$
$$= \sqrt{(0-2)^2 + (2-0)^2}$$
$$= \sqrt{8} = 2.828$$

© UAB. All Rights Reserved.

Minkowski Distance

- A generalization of Euclidean Distance

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

Where r is a parameter, n is the number of dimensions (attributes) and x_k and y_k are, respectively, the kth attributes (components) or data objects x and y.

© UAB. All Rights Reserved.

Minkowski Distance

- Common examples of Minkowski distances
  - $r = 1$. City block (Manhattan, taxicab, L_1 norm) distance.
    - A common example is the Hamming distance -- the number of bits that are different between two binary vectors
  - $r = 2$. Euclidean distance (L_2 norm)
  - $r \rightarrow \infty$. "supremum" (l_{max} norm, l_∞ norm) distance.
    - This is the maximum difference between any attribute of the objects
  - Note: do not confuse r with n
    - n – the numbers of dimensions (attributes).
    - r – parameter of distance metric.

© UAB. All Rights Reserved.

Minkowski Distance

Data points

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

$r = 1$, City block distance

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

$r = 2$, Euclidean distance

L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

$r = \infty$, Supremum distance

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

Distance Matrix

© UAB. All Rights Reserved.

Common Properties of a Distance

$d(x, y)$ is the distance (dissimilarity) between points (data objects), x and y.

- **Positivity**: $d(x, y) \geq 0$ for all x and y, $d(x, y) = 0$ if and only if $x = y$.
- **Symmetry**: $d(x, y) = d(y, x)$ for all x and y.
- **Triangle Inequality**: $d(x, z) \leq d(x, y) + d(y, z)$ for all points x, y, and z.

A distance that satisfies these properties is a **metric.**

© UAB. All Rights Reserved.

Common Properties of a Similarity

$s(x, y)$ is the similarity between points (data objects), x and y.

- $s(x, y) = 1$ only if $x = y$ $(0 \leq s(x, y) \leq 1)$.
- $s(x, y) = s(y, x)$ for all x and y.
- Triangle inequality does not always hold for similarity.

© UAB. All Rights Reserved.

Similarity Between Binary Vectors

- Objects, x and y, have only binary attributes
- Compute similarities using the following:
  - f_{01} = the number of attributes where x was 0 and y was 1
  - f_{10} = the number of attributes where x was 1 and y was 0
  - f_{00} = the number of attributes where x was 0 and y was 0
  - f_{11} = the number of attributes where x was 1 and y was 1

© UAB. All Rights Reserved.

Similarity Between Binary Vectors

- Simple Matching Coefficient (SMC)

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

- Jaccard Coefficients

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

© UAB. All Rights Reserved.

SMC versus Jaccard: Example

x = 1 0 0 0 0 0 0 0 0 0

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

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

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

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

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

SMC $= (f_{11} + f_{00}) / (f_{01} + f_{10} + f_{11} + f_{00}) = (0+7) / (2+1+0+7) = 0.7$

J $= (f_{11}) / (f_{01} + f_{10} + f_{11}) = 0 / (2 + 1 + 0) = 0$

© UAB. All Rights Reserved.

Cosine Similarity

- If x and y are two document vectors, then

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

where $< x, y >= \sum x_k \, y_k$ is inner product, and $||x|| = \sqrt{\sum x_k^2}$ is the length of vector x.

- Example:

$$x = (3,2,0,5,0,0,0,2,0,0)$$
$$y = (1,0,0,0,0,0,0,1,0,2)$$
$$< x, y > = 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2=5$$
$$||x|| = = (3*3+2*2+0*0+5*5+0*0+0*0+0*0+2*2+0*0+0*0)0.5 = (42)\ 0.5 = 6.481$$
$$||y|| = (1*1+0*0+0*0+0*0+0*0+0*0+0*0+1*1+0*0+2*2)\ 0.5 = (6)\ 0.5 = 2.449$$
$$\cos(x, y) = \frac{5}{6.481 * 2.449} = 0.3150$$

© UAB. All Rights Reserved.

Cosine Similarity

- A measure of the (cosine of the) angle between **x** and **y.**
- cos(x, y) = 1, angle is 0
- cos(x, y)= 0, angle is 90

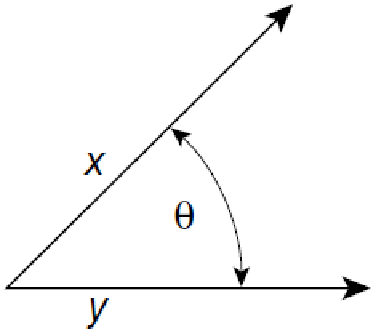


Figure 2.16. Geometric illustration of the cosine measure.

© UAB. All Rights Reserved.

Correlation

- A measure of linear relationship between objects
- In the range $[-1, 1]$
- 1(-1) indicates perfect positive (negative) linear relationship: $y = ax + b$
- 0 means no linear relationship (but non-linear relationship may exist)

© UAB. All Rights Reserved.

Correlation

- Pearson's correlation

$$\text{corr}(\mathbf{x},\mathbf{y}) = \frac{\text{covariance}(\mathbf{x},\mathbf{y})}{\text{standard\_deviation}(\mathbf{x}) * \text{standard\_deviation}(\mathbf{y})} = \frac{s_{xy}}{s_x\ s_y},$$

$$\text{covariance}(\mathbf{x},\mathbf{y}) = s_{xy} = \frac{1}{n-1}\sum_{k=1}^{n}(x_k-\overline{x})(y_k-\overline{y})$$

$$\overline{x} = \frac{1}{n}\sum_{k=1}^{n} x_k \text{ is the mean of } \mathbf{x}$$

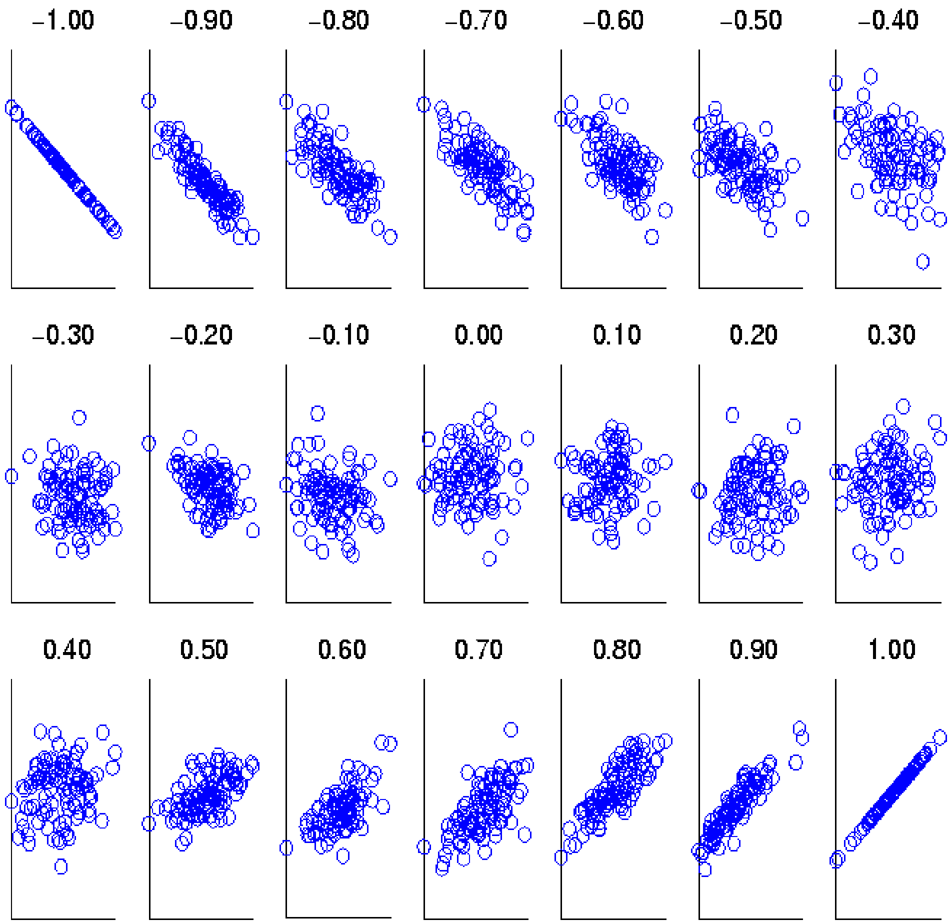
$$\overline{y} = \frac{1}{n}\sum_{k=1}^{n} y_k \text{ is the mean of } \mathbf{y}$$

$$\text{standard\_deviation}(\mathbf{x}) = s_x = \sqrt{\frac{1}{n-1}\sum_{k=1}^{n}(x_k-\overline{x})^2}$$

$$\text{standard\_deviation}(\mathbf{y}) = s_y = \sqrt{\frac{1}{n-1}\sum_{k=1}^{n}(y_k-\overline{y})^2}$$

© UAB. All Rights Reserved.

Visualize Pearson’s correlation



Scatter plots showing the correlation from –1 to 1.

© UAB. All Rights Reserved.

General Approach for Combining Similarities

Sometimes attributes are of many different types, but an overall similarity is needed.

1. For the k[th] attribute, compute a similarity, $s_k(x, y)$, in the range [0, 1].
2. Compute

$$similarity(x, y) = \frac{1}{K} s_k(x, y)$$

Does not work with asymmetric attributes.

© UAB. All Rights Reserved.

General Approach for Combining Similarities

Sometimes attributes are of many different types, but an overall similarity is needed.

1. For the k^th attribute, compute a similarity, $s_k(x, y)$, in the range [0, 1].
2. Define an indicator variable, δ_k, for the k^th attribute as follows:
   1. $\delta_k = 0$ if the k^th attribute is an asymmetric attribute and both objects have a value of 0, or if one of the objects has a missing value for the k^th attribute
   2. $\delta_k = 1$ otherwise
3. Compute

$$\text{similarity}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{k=1}^{n} \delta_k s_k(\mathbf{x}, \mathbf{y})}{\sum_{k=1}^{n} \delta_k}$$

© UAB. All Rights Reserved.

Using Weights to Combine Similarities

- May not want to treat all attributes the same.
  - Use non-negative weights ω_k
  - $similarity(x, y) = \frac{\sum_{k=1}^{n} \omega_k \delta_k s_k(x,y)}{\sum_{k=1}^{n} \omega_k \delta_k}$
- Can also define a weighted form of distance

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

© UAB. All Rights Reserved.

Select the Right Proximity Measure

- The type of proximity measure should match the type of data
  - Dense, continuous data: Euclidean distance
  - Sparse data (asymmetric attributes): similarity measures that ignore 0-0 matches, e.g., Cosine and Jaccard

© UAB. All Rights Reserved.

Select the Right Proximity Measure

- The type of proximity measure should match the type of data
  - Dense, continuous data: Euclidean distance
  - Sparse data (asymmetric attributes): similarity measures that ignore 0-0 matches, e.g., Cosine and Jaccard
- Data characteristics:
  - Time series:
    - Use Euclidean distance if magnitude is important.
    - Use correlation if the shape of the series is more important than magnitude.
  - Transformations and normalizations: Necessary for proper similarity computation, especially for time series with trends or patterns.

© UAB. All Rights Reserved.

Select the Right Proximity Measure

- The type of proximity measure should match the type of data
  - Dense, continuous data: Euclidean distance
  - Sparse data (asymmetric attributes): similarity measures that ignore 0-0 matches, e.g., Cosine and Jaccard
- Data characteristics:
  - Time series:
    - Use Euclidean distance if magnitude is important.
    - Use correlation if the shape of the series is more important than magnitude.
  - Transformations and normalizations: Necessary for proper similarity computation, especially for time series with trends or patterns.
- Practical considerations:
  - Efficiency
  - Software or algorithm limitations

© UAB. All Rights Reserved.