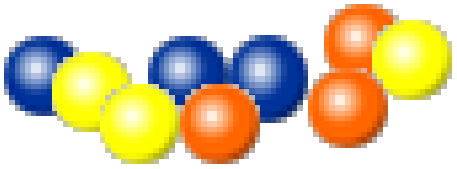
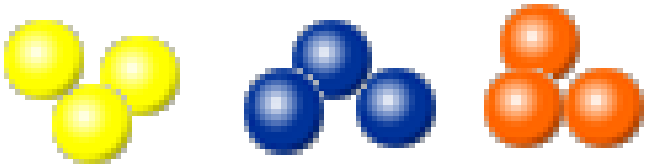
Clustering (slide from Han and Kamber)

Clustering of data is a method by which large sets of data is grouped into clusters of smaller sets of similar data.

The example below demonstrates the clustering of balls of same colour. There are a total of 10 balls which are of three different colours. We are interested in clustering of balls of the three different colours into three different groups.



The balls of same colour are clustered into a group as shown below :



Thus, we see clustering means grouping of data or dividing a large data set into smaller data sets of some similarity.

Clustering Algorithms

A clustering algorithm attempts to find natural groups of components (or data) based on some similarity. Also, the clustering algorithm finds the centroid of a group of data sets.To determine cluster membership, most algorithms evaluate the distance between a point and the cluster centroids. The output from a clustering algorithm is basically a statistical description of the cluster centroids with the number of components in each cluster.

Usual Working Data Structures

- Data matrix
  - (two modes)

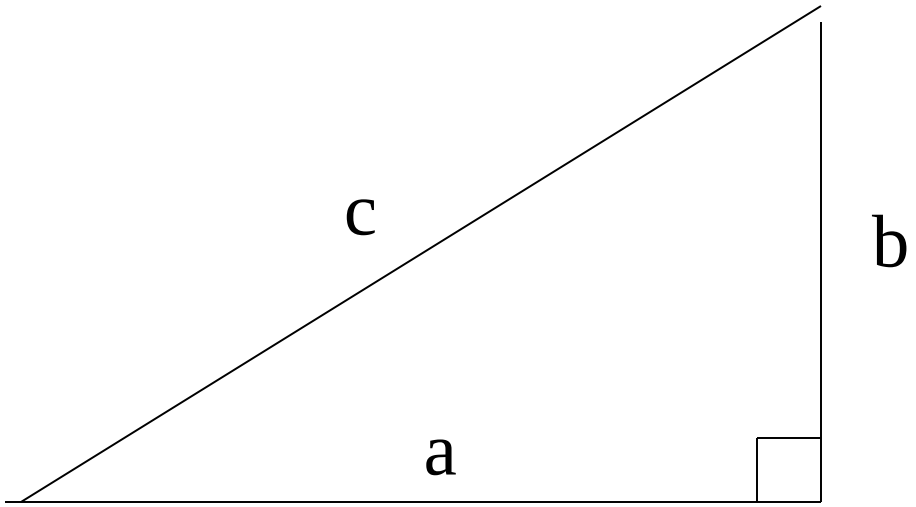
(Flat File of Attributes/coordinates)

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

- Dissimilarity matrix
  - (one mode)
  - Or distance matrix

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

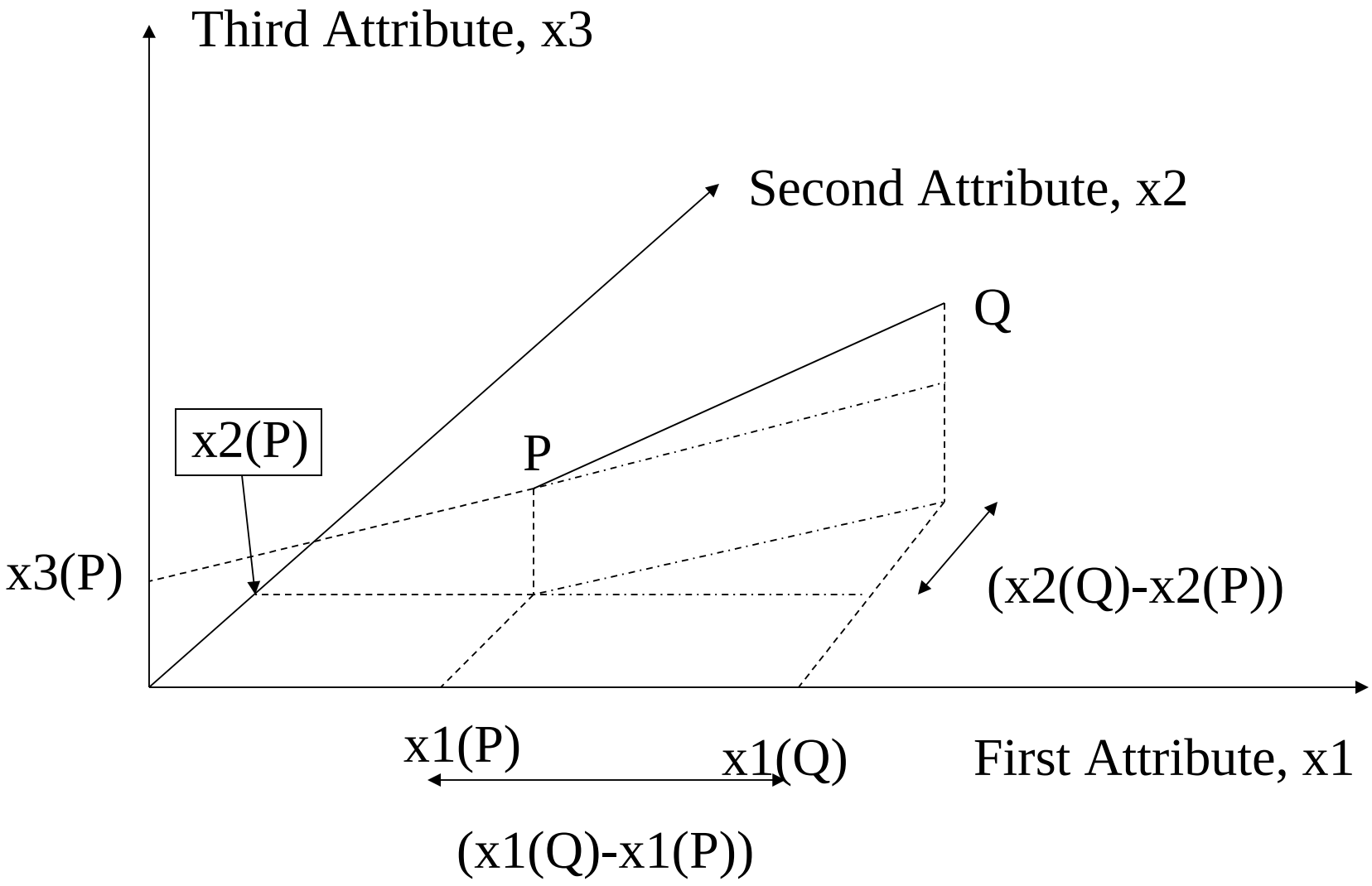
PYTHAGORUS



$$c^2 = a^2 + b^2 \quad \Rightarrow \quad c = \sqrt{\left(a^2 + b^2\right)}$$

Graphical Data [3D]

(ASSUMING ATTRIBUTES/VARIABLES ARE REAL(interval/ratio))



Distances and Cluster Centroid

Distance

Generally, the distance between two points is taken as a common metric to assess the similarity among the instances of a population. The commonly used distance measure is the ***Euclidean metric*** which defines the distance between two points P= (x1(P), x2(P),…) and Q = (x1(Q), x2(Q),…) is given by :

$$d(P,Q) = \sqrt{(x1(P) - x1(Q))^2 + (x2(P) - x2(Q))^2 + \ldots}$$

$$= \sqrt{\sum_{j=1}^{p} (xj(P) - xj(Q))^2}$$

Cluster centroid :

The centroid of a cluster is a point whose coordinates are the mean of the coordinates of all the points in the clusters.

Distance-based Clustering

- Define/Adopt distance measure data instances
- Find a partition of the instances such that:
  - Distance between objects within partition (i.e. same cluster) is minimized
  - Distance between objects from different clusters is maximised
- Issues :
  - Requires defining a distance (similarity) measure in situation where it is unclear how to assign it
  - What relative weighting to give to one attribute vs another?
  - Number of possible partition is super-exponential in n.

Generalized Distances, and Similarity Measures

- The distance metric is a Dissimilarity measure.
- Two points are “similar” if they are “close”, or dist is near 0.
- Hence Similarity can be expressed in terms of a distance function. For example:

  $s(P,Q) = 1 / (1 + d(P,Q))$

- The definitions of distance functions are usually very different for interval-scaled, boolean, categorical, ordinal and ratio variables.
- Weights should be associated with different coordinate dimensions, based on applications and data semantics.
- It is hard to define “similar enough” or “good enough”
  - the answer is typically highly subjective.

Type of data in clustering analysis

- Interval-scaled variables:
- Binary variables:
- Nominal, ordinal, and ratio variables:
- Variables of mixed types:
- Note: The following seven slides are optional. They are given to fill in some of the background which is missed in R & G, because they do not wish to reveal their instance similarity measure, for commercial reasons. Understanding these slides really depends on some mathematical background.

Real/Interval-valued variables

- If each variable has it's own disparate scale, then we can **standardize** each of the variables to a mean of Zero, and and a "variability" of One.
- Standardizing data
  - Calculate the mean absolute deviation for variable I:

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + ... + |x_{nf} - m_f|)$$

Where

$$m_f = \frac{1}{n}(x_{1f} + x_{2f} + ... + x_{nf})$$

  - Calculate the standardized measurement (*z-score*)

$$z_{if} = \frac{x_{if} - m_f}{s_f}$$

- Then use distances/similarities based on standardized scores

Generalized Distances Between Objects

- The *Minkowski distance*:

$$d(i,j) = \sqrt[q]{(|x_{i_1} - x_{j_1}|^q + |x_{i_2} - x_{j_2}|^q + ... + |x_{i_p} - x_{j_p}|^q)}$$

where $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ are two p-dimensional data objects, and q is a positive integer

- If $q = 1$, d is Manhattan distance

$$d(i,j) = |x_{i_1} - x_{j_1}| + |x_{i_2} - x_{j_2}| + ... + |x_{i_p} - x_{j_p}|$$

Binary Variables

- A contingency table for binary data

		Object *j*		
		1	0	*sum*
Object *i*	1	*a*	*b*	*a+b*
	0	*c*	*d*	*c+d*
	sum	*a+c*	*b+d*	*p*

Simple matching coefficient (invariant, if the binary variable is *symmetric*): $d(i, j) = \frac{b+c}{a+b+c+d}$

- Jaccard coefficient of dissimilarity:

$$d(i, j) = \frac{b+c}{a+b+c}$$

Nominal Variables

- A generalization of the binary variable in that it can take more than 2 states, e.g., red, yellow, blue, green
- Method 1: Simple matching
  - *m*: # of matches, *p*: total # of variables

$$d(i, j) = \frac{p - m}{p}$$

- Method 2: use a large number of binary variables
  - creating a new binary variable for each of the *M* nominal states

Ordinal Variables

- An ordinal variable can be discrete or continuous
- order is important, e.g., rank
- Can be treated like interval-scaled
  - replacing x_{if} by their rank $r_{if} \in \{1, \ldots, M_f\}$
  - map the range of each variable onto [0, 1] by replacing *i*-th object in the *f*-th variable by

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

  - compute the dissimilarity using methods for interval-scaled variables

Ratio-Scaled Variables

- Ratio-scaled variable: a positive measurement on a multiplicative scale, correponding to exponential growth, i.e. A + B □ e^{A+B}
- Methods:
  - treat them like interval-scaled variables — *not a good choice! (why?)*
  - apply logarithmic transformation

$$y_{if} = log(x_{if})$$

  - treat them as continuous ordinal data treat their rank as interval-scaled.

Variables of Mixed Types

- A database may contain all the six types of variables
  - symmetric binary, asymmetric binary, nominal, ordinal, interval and ratio.
- One may use a weighted formula to combine their effects.
  - f is binary or nominal:
    $d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$, or $d_{ij}^{(f)} = 1$ o.w.
  - f is interval-based: use the normalized distance
  - f is ordinal or ratio-scaled
    - compute ranks r_{if} and
    - and treat z_{if} as interval-scaled

$$d(i,j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}}$$

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

K-Means Clustering

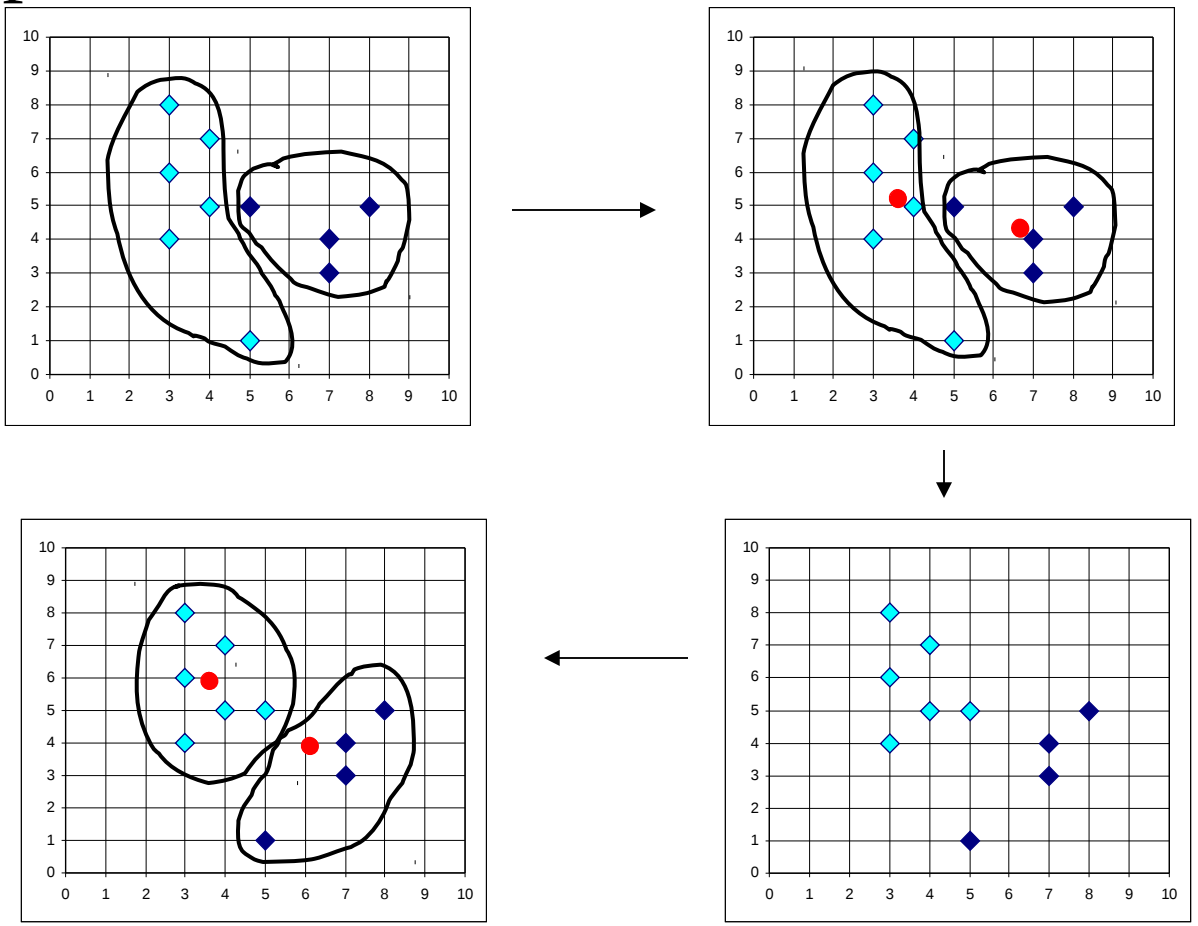
- Initially, the number of clusters must be known, or chosen, to be K say.
- The initial step is the choose a set of K instances as centres of the clusters. Often chosen such that the points are mutually “farthest apart”, in some way.
- Next, the algorithm considers each instance and assigns it to the cluster which is closest.
- The cluster centroids are recalculated either after each instance assignment, or after the whole cycle of re-assignments.
- This process is iterated.

Other K-mean Algorithm features

- Using cluster centroid to represent cluster
- Assigning data elements to the closest cluster (centre).
- Goal: Minimise the sum of the within cluster variances
- Variations of K-Means
  - Initialisation (select the number of clusters, initial partitions)
  - Updating of center
  - Hill-climbing (trying to move an object to another cluster).

The *K-Means* Clustering Method

- Example



Comments on the *K-Means* Method

- Strength
  - *Relatively efficient*: *O*(*tkn*), where *n* is # instances, c is # clusters, and *t* is # iterations. Normally, *k*, *t* << *n*.
  - Often terminates at a *local optimum*. The *global optimum* may be found using techniques such as: *simulated annealing* or *genetic algorithms*
- Weakness
  - Applicable only when *mean* is defined; what about categorical data?
  - Need to specify c, the *number* of clusters, in advance
  - Unable to handle noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*

Variations of the *K-Means* Method

- A few variants of the *k-means* which differ in
  - Selection of the initial *k* means
  - Dissimilarity calculations
  - Strategies to calculate cluster means
- Handling categorical data: *k-modes* (Huang'98)
  - Replacing means of clusters with modes
  - Using new dissimilarity measures to deal with categorical objects
  - Using a frequency-based method to update modes of clusters
  - A mixture of categorical and numerical data: *k-prototype* method

Agglomerative Hierarchical Clustering

Given a set of n instances to be clustered, and an nxn distance (or similarity) matrix, the basic process hierarchical clustering is:

- 1 Start by assigning each item to its own cluster, so that if you have N items, you now have N clusters, each containing just one item.
Let the distances (similarities) between the clusters equal the distances (similarities) between the items they contain.

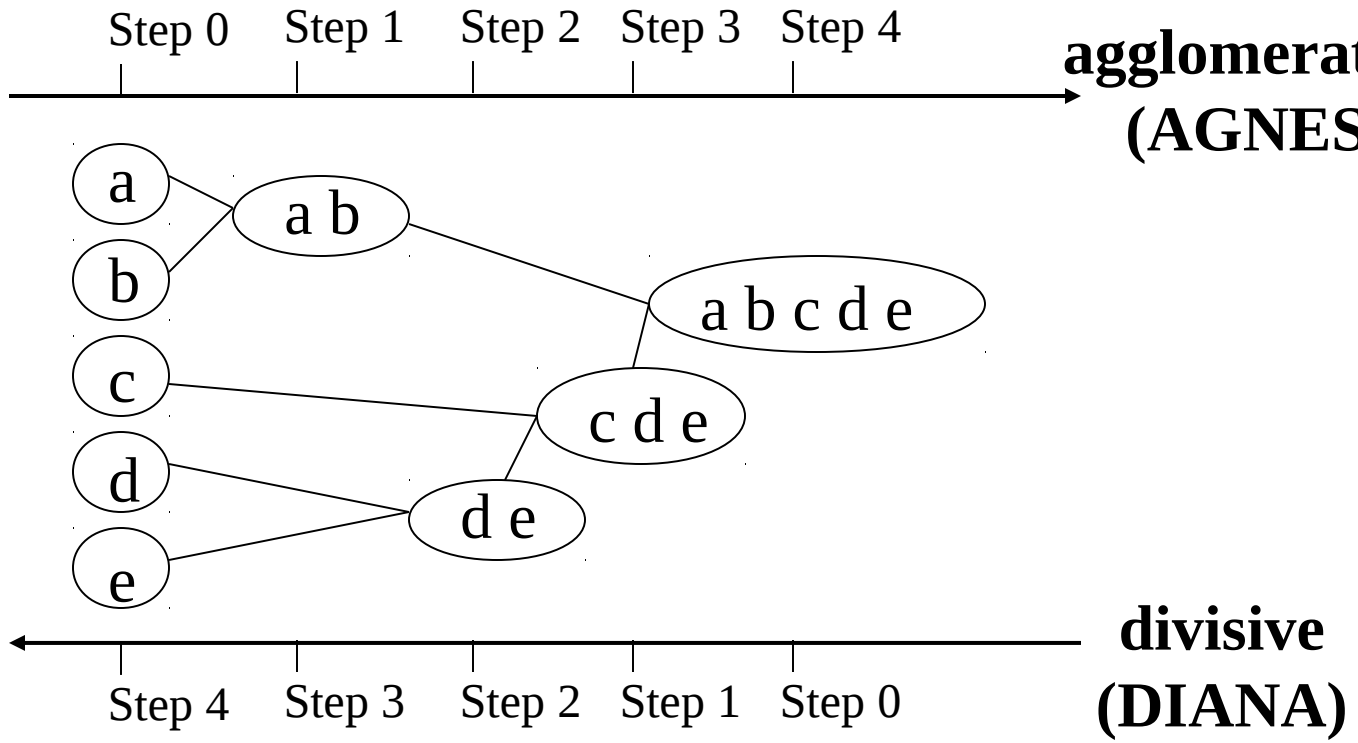
- 2 .Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one less cluster.

- 3. Compute distances (similarities) between the new clusters and each of the old clusters.

- 4. Repeat steps 2 and 3 until all items are clustered into a single cluster of size n.

Hierarchical Clustering

- Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition

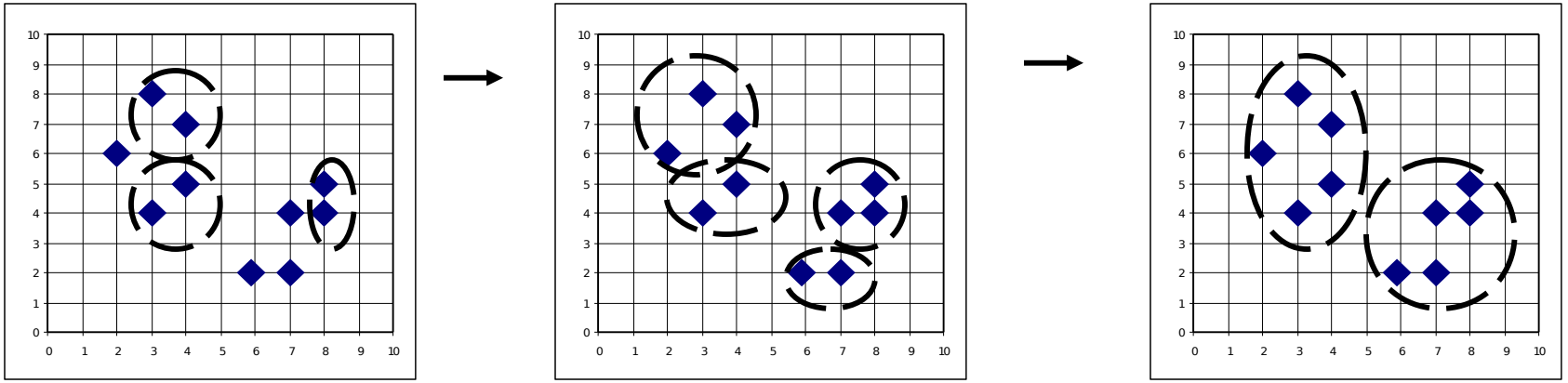


More on Hierarchical Clustering Methods

- Major weakness of agglomerative clustering methods
  - do not scale well: time complexity of at least $O(n^2)$, where n is the total number of instances
  - can never undo what was done previously
- Integration of hierarchical with distance-based clustering
  - BIRCH (1996): uses CF-tree and incrementally adjusts the quality of sub-clusters
  - CURE (1998): selects well-scattered points from the cluster and then shrinks them towards the center of the cluster by a specified fraction
  - CHAMELEON (1999): hierarchical clustering using dynamic modeling

AGNES (Agglomerative Nesting)

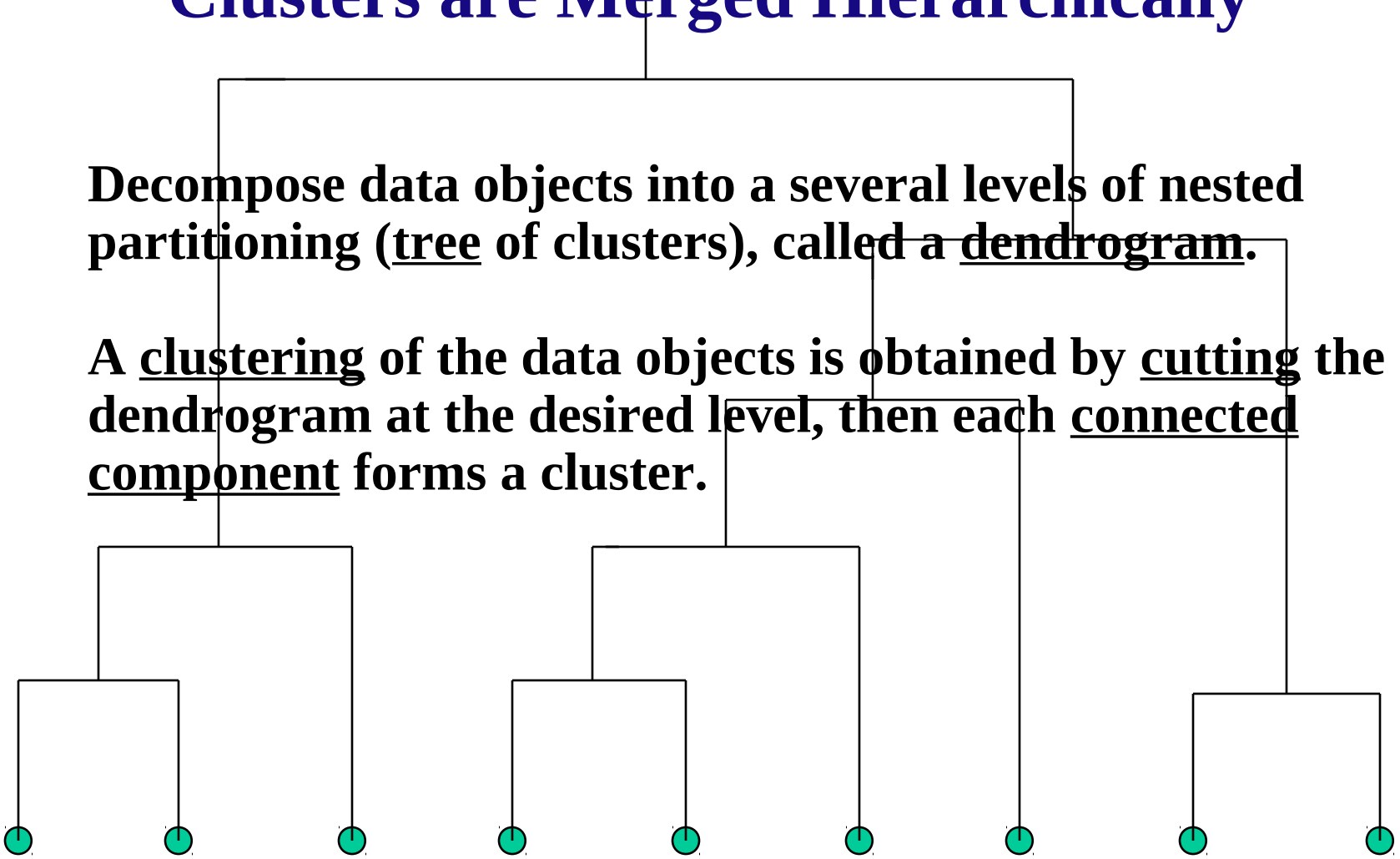
- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Use the Single-Link method and the dissimilarity matrix.
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



A *Dendrogram* Shows How the Clusters are Merged Hierarchically

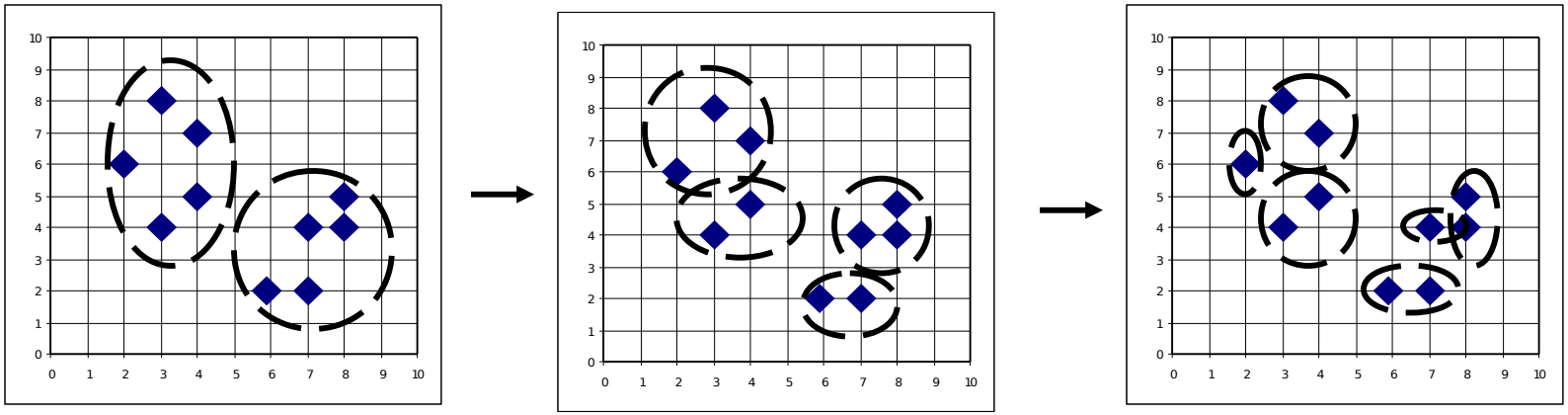
Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram.

A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.



DIANA (Divisive Analysis)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Inverse order of AGNES
- Eventually each node forms a cluster on its own



Computing Inter-Cluster Distances

- **single-link clustering** (also called the connectedness or minimum method) : we consider the distance between one cluster and another cluster to be equal to the **shortest distance** from any member of one cluster to any member of the other cluster. If the data consist of similarities, we consider the similarity between one cluster and another cluster to be equal to the greatest similarity from any member of one cluster to any member of the other cluster.

- **complete-link clustering** (also called the diameter or maximum method): we consider the distance between one cluster and another cluster to be equal to the **longest distance** from any member of one cluster to any member of the other cluster.

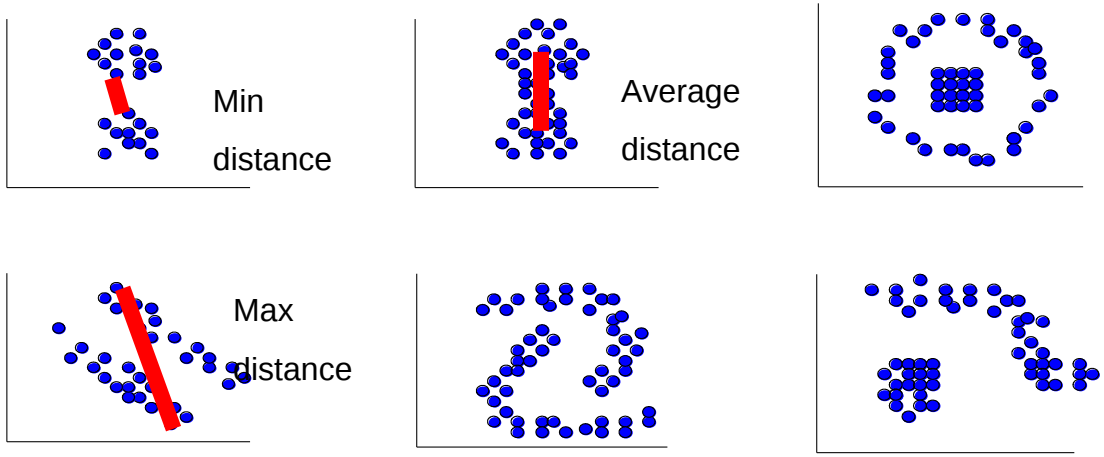
- **average-link clustering** : we consider the distance between one cluster and another cluster to be equal to the **average distance** from any member of one cluster to any member of the other cluster.

Distance Between Two Clusters

• single-link clustering (also called the connectedness or minimum method) : we consider the distance between one cluster and another cluster to be equal to the shortest distance from any member of one cluster to any member of the other cluster. If the data consist of similarities, we consider the similarity between one cluster and another cluster to be equal to the greatest similarity from any member of one cluster to any member of the other cluster.

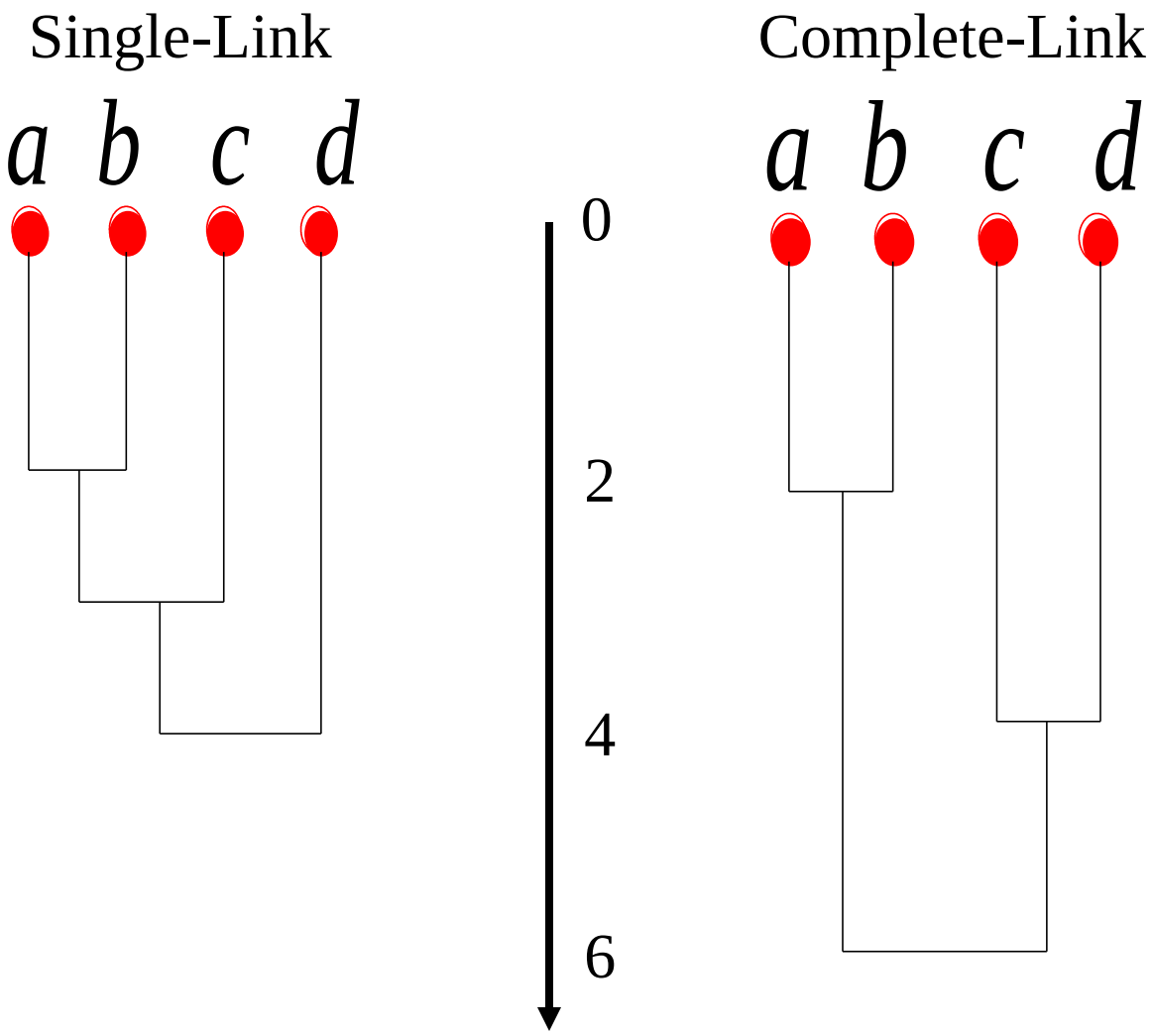
• complete-link clustering (also called the diameter or maximum method): we consider the distance between one cluster and another cluster to be equal to the longest distance from any member of one cluster to any member of the other cluster.

• average-link clustering : we consider the distance between one cluster and another cluster to be equal to the average distance from any member of one cluster to any member of the other cluster.



- ❑Single-Link Method / Nearest Neighbor
- ❑Complete-Link / Furthest Neighbor
- ❑Their Centroids.
- ❑Average of all cross-cluster pairs.

Compare Dendrograms



3.3 The K-Means Algorithm
R & G

1. Choose a value for *K*, the total number of clusters.
2. Randomly choose *K* points as cluster centers.
3. Assign the remaining instances to their closest cluster center.
4. Calculate a new cluster center for each cluster.
5. Repeat steps 3-5 until the cluster centers do not change.

K-Means: General Considerations

- Requires real-valued data.
- We must select the number of clusters present in the data.
- Works best when the clusters in the data are of approximately equal size.
- Attribute significance cannot be determined.
- Lacks explanation capabilities.

4.3 iDAV Format for Data Mining

Table 4.1 • **Credit Card Promotion Database: iDAV Format**

Income Range	**Magazine Promotion**	**Watch Promotion**	**Life Insurance Promotion**	**Credit Card Insurance**	**Sex**	**Age**
C	C	C	C	C	C	R
I	I	I	I	I	I	I
40–50K	Yes	No	No	No	Male	45
30–40K	Yes	Yes	Yes	No	Female	40
40–50K	No	No	No	No	Male	42
30–40K	Yes	Yes	Yes	Yes	Male	43
50–60K	Yes	No	Yes	No	Female	38
20–30K	No	No	No	No	Female	55
30–40K	Yes	No	Yes	Yes	Male	35
20–30K	No	Yes	No	No	Male	27
30–40K	Yes	No	No	No	Male	43
30–40K	Yes	Yes	Yes	No	Female	41
40–50K	No	Yes	Yes	No	Female	43
20–30K	No	Yes	Yes	No	Male	29
50–60K	Yes	Yes	Yes	No	Female	39
40–50K	No	Yes	No	No	Male	55
20–30K	No	No	Yes	Yes	Female	19

Table 4.2 • **Values for Attribute Usage**

Character	Usage
I	The attribute is used as an input attribute.
U	The attribute is not used.
D	The attribute is not used for classification or clustering, but attribute value summary information is displayed in all output reports.
O	The attribute is used as an output attribute. For supervised learning with ESX, exactly one categorical attribute is selected as the output attribute.

4.4 A Five-step Approach for Unsupervised Clustering

Step 1: Enter the Data to be Mined

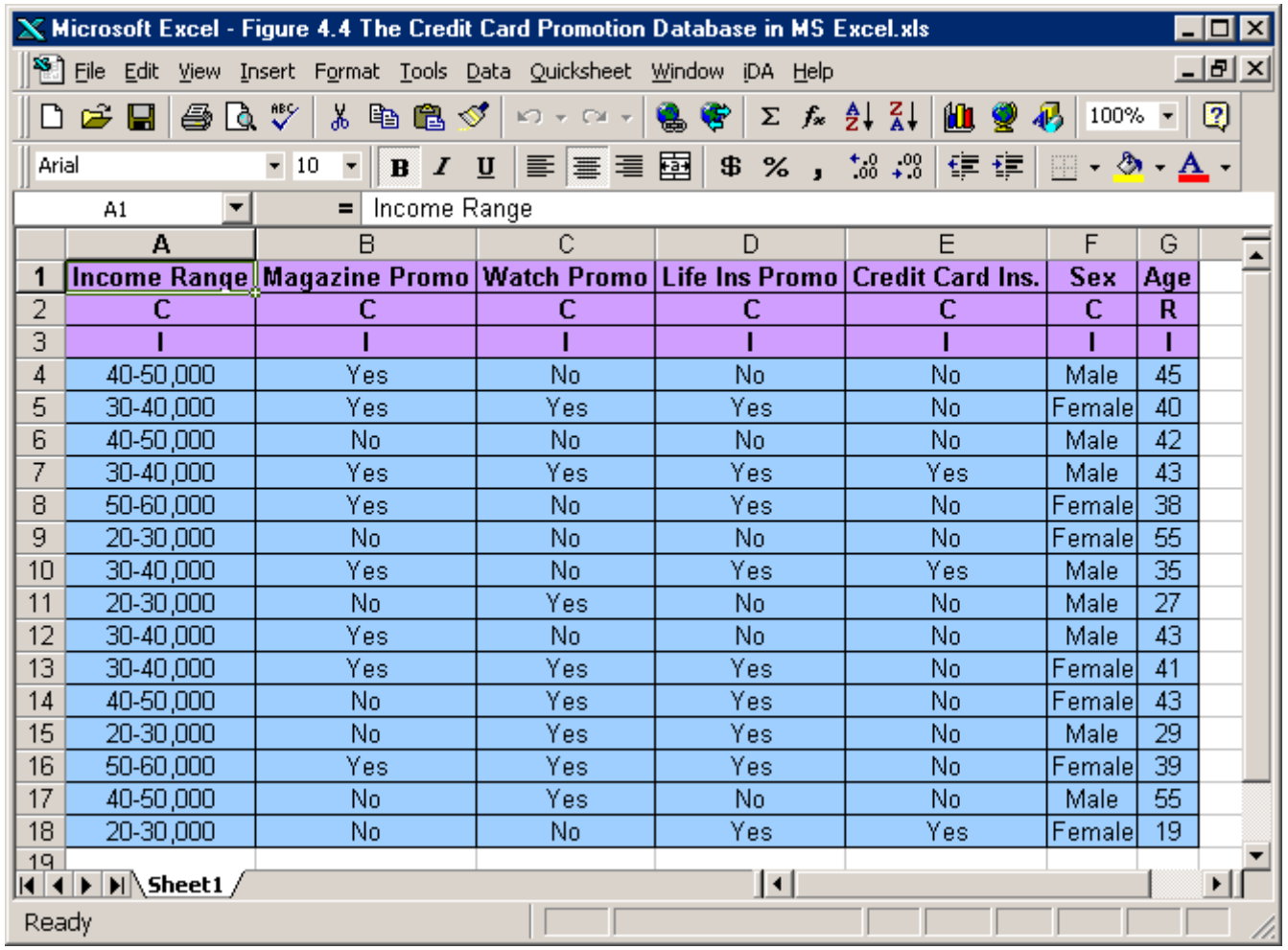
Step 2: Perform a Data Mining Session

Step 3: Read and Interpret Summary Results

Step 4: Read and Interpret Individual Class Results

Step 5: Visualize Individual Class Rules

Step 1: Enter The Data To Be Mined



	A	B	C	D	E	F	G
1	Income Range	Magazine Promo	Watch Promo	Life Ins Promo	Credit Card Ins.	Sex	Age
2	C	C	C	C	C	C	R
3	I	I	I	I	I	I	I
4	40-50,000	Yes	No	No	No	Male	45
5	30-40,000	Yes	Yes	Yes	No	Female	40
6	40-50,000	No	No	No	No	Male	42
7	30-40,000	Yes	Yes	Yes	Yes	Male	43
8	50-60,000	Yes	No	Yes	No	Female	38
9	20-30,000	No	No	No	No	Female	55
10	30-40,000	Yes	No	Yes	Yes	Male	35
11	20-30,000	No	Yes	No	No	Male	27
12	30-40,000	Yes	No	No	No	Male	43
13	30-40,000	Yes	Yes	Yes	No	Female	41
14	40-50,000	No	Yes	Yes	No	Female	43
15	20-30,000	No	Yes	Yes	No	Male	29
16	50-60,000	Yes	Yes	Yes	No	Female	39
17	40-50,000	No	Yes	No	No	Male	55
18	20-30,000	No	No	Yes	Yes	Female	19
19							

Figure 4.4 The Credit Card Promotion Database

Step 2: Perform A Data Mining Session

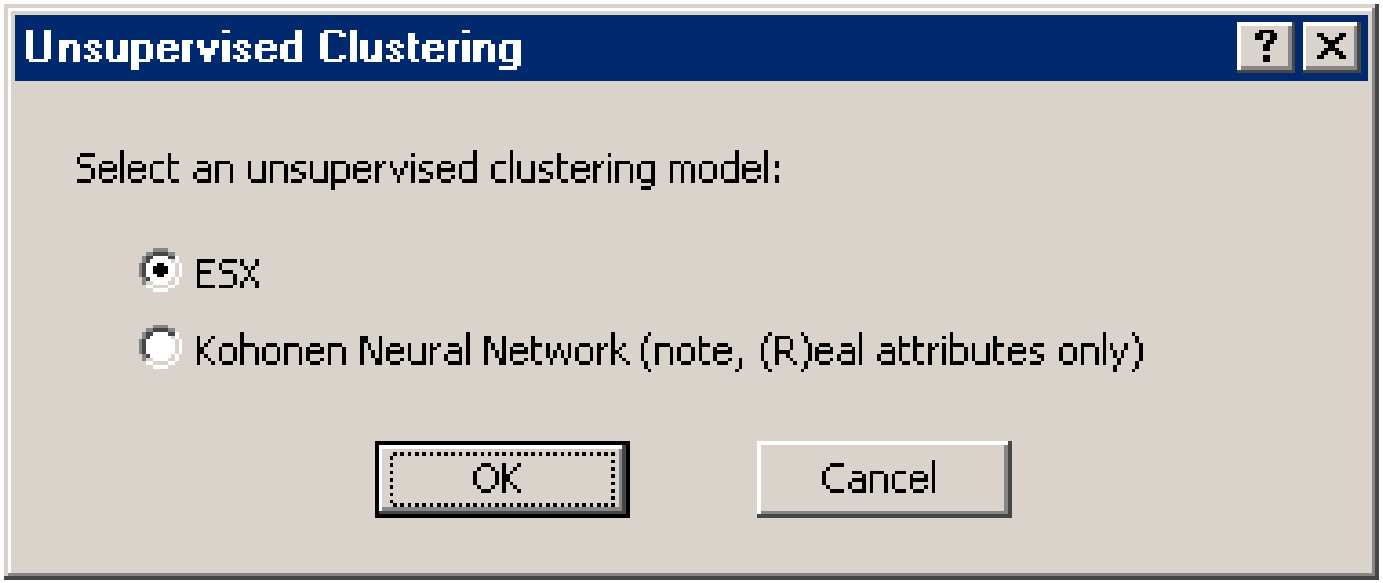


Figure 4.5 Unsupervised settings for ESX

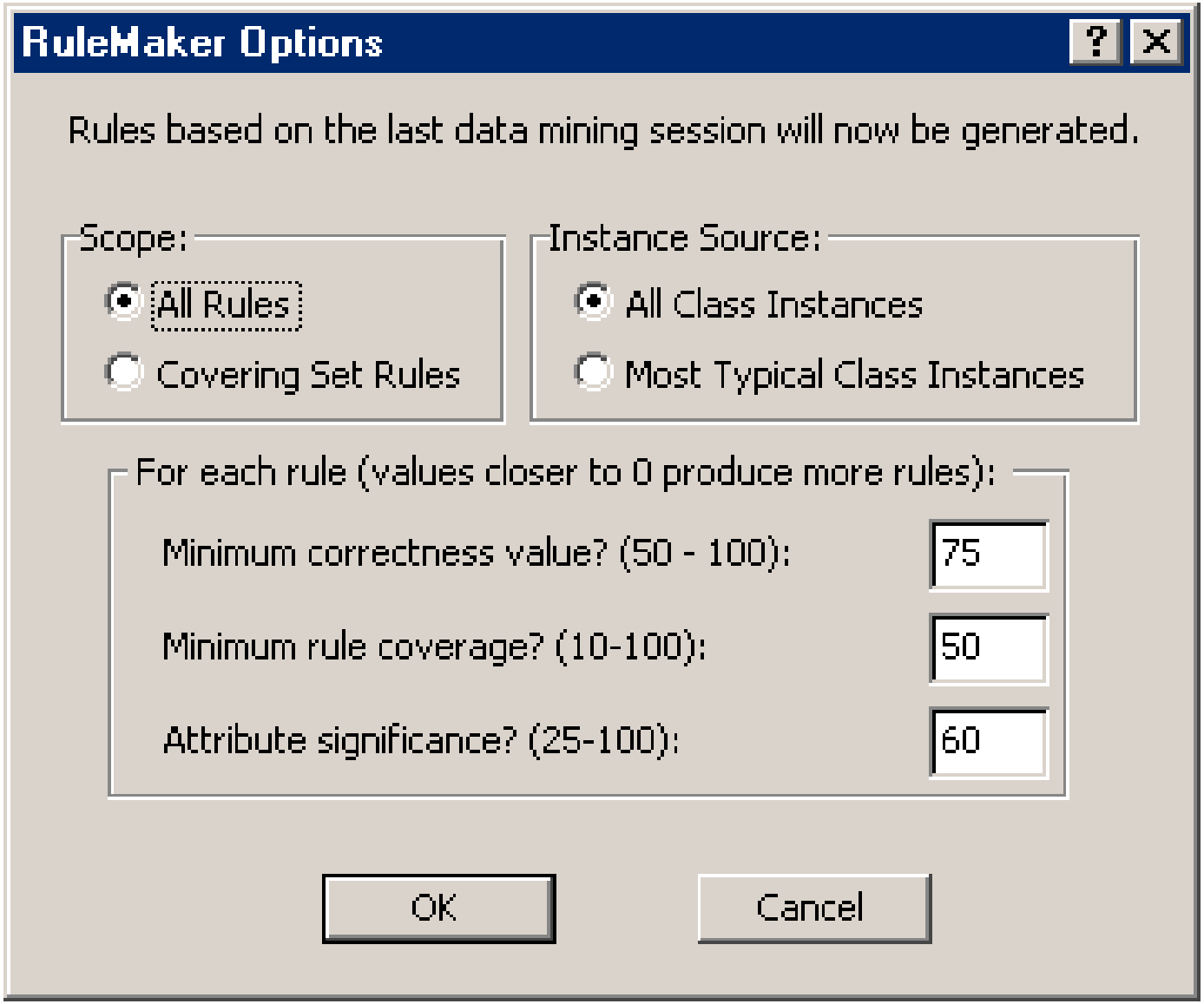


Figure 4.6 RuleMaker options

Step 3: Read and Interpret Summary Results

- Class Resemblance Scores
- Domain Resemblance Score
- Domain Predictability

Microsoft Excel - Figure 4.8 Summary statistics for the Acme credit card promotion database.xls

File Edit View Insert Format Tools Data Quicksheet Window iDA Help

Arial 10 B I U 100%

Q27 =

	A	B	C	D	E
1	**CLASS RESEMBLANCE STATISTICS**				
2		Class 1	Class 2	Class 3	**Domain**
3	**Res. Score:**	0.81	0.529	0.66	0.46
4	**No. of Inst.**	3	5	7	15
5	**Cluster Quality:**	0.75	0.14	0.43	
6					
7	**DOMAIN STATISTICS FOR CATEGORICAL ATTRIBUTES**				
8	**Number of Classes:**	3			
9	**Domain Res. Score:**	0.46			
10					
11	**Categorical Attribute Summary:**	**Name**	**Value**	**Frequency**	**Predictability**
12		Income Range	"40-50,000"	4	0.27
13			"20-30,000"	4	0.27
14			"30-40,000"	5	0.33
15			"50-60,000"	2	0.13
16					
17		Magazine Promo	Yes	8	0.53
18			No	7	0.47
19					
20		Watch Promo	No	7	0.47
21			Yes	8	0.53
22					
23		Life Ins Promo	No	6	0.40

Sheet1 Sheet2 Sheet3

Ready

Figure 4.8 Summery statistics for the Acme cred

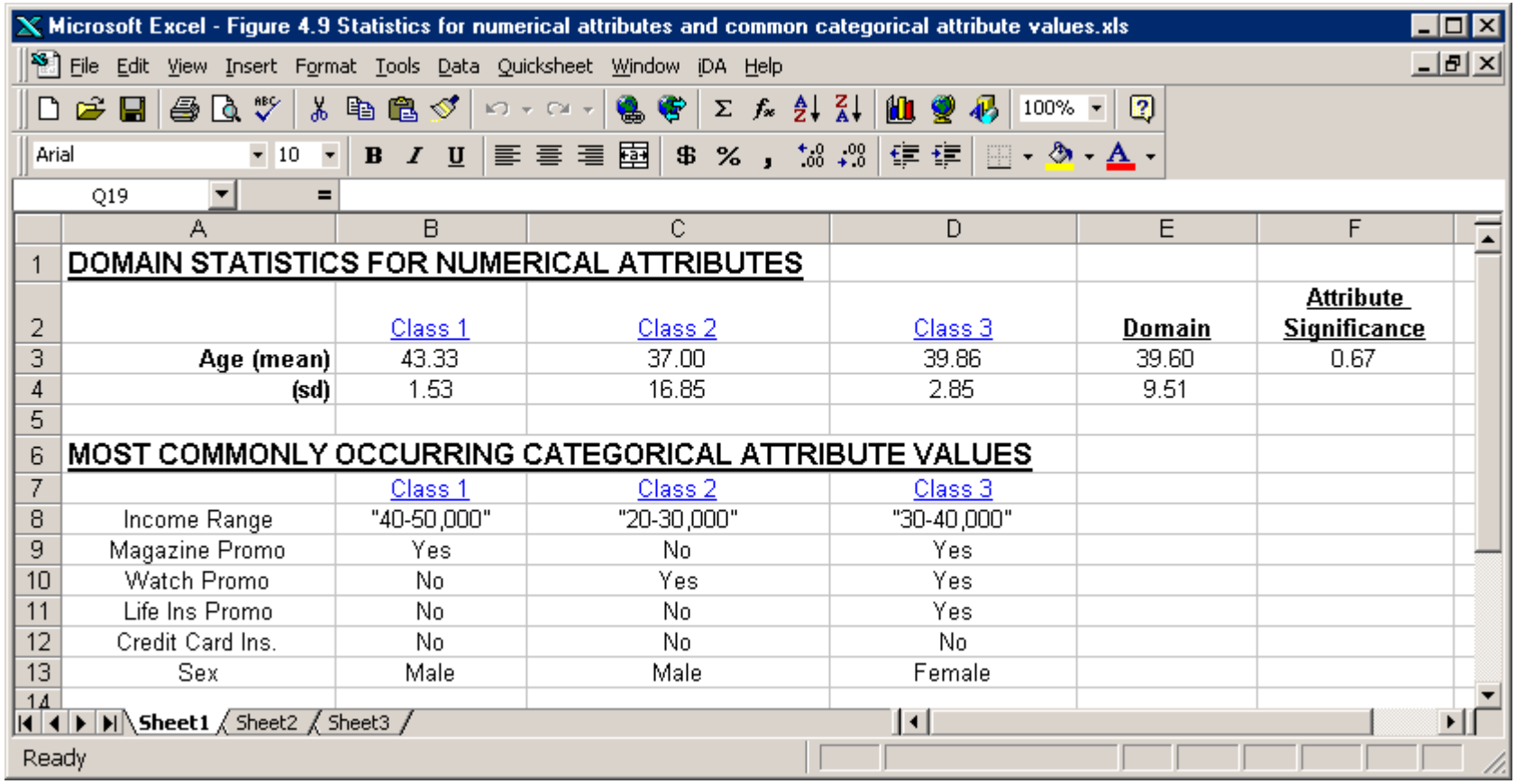
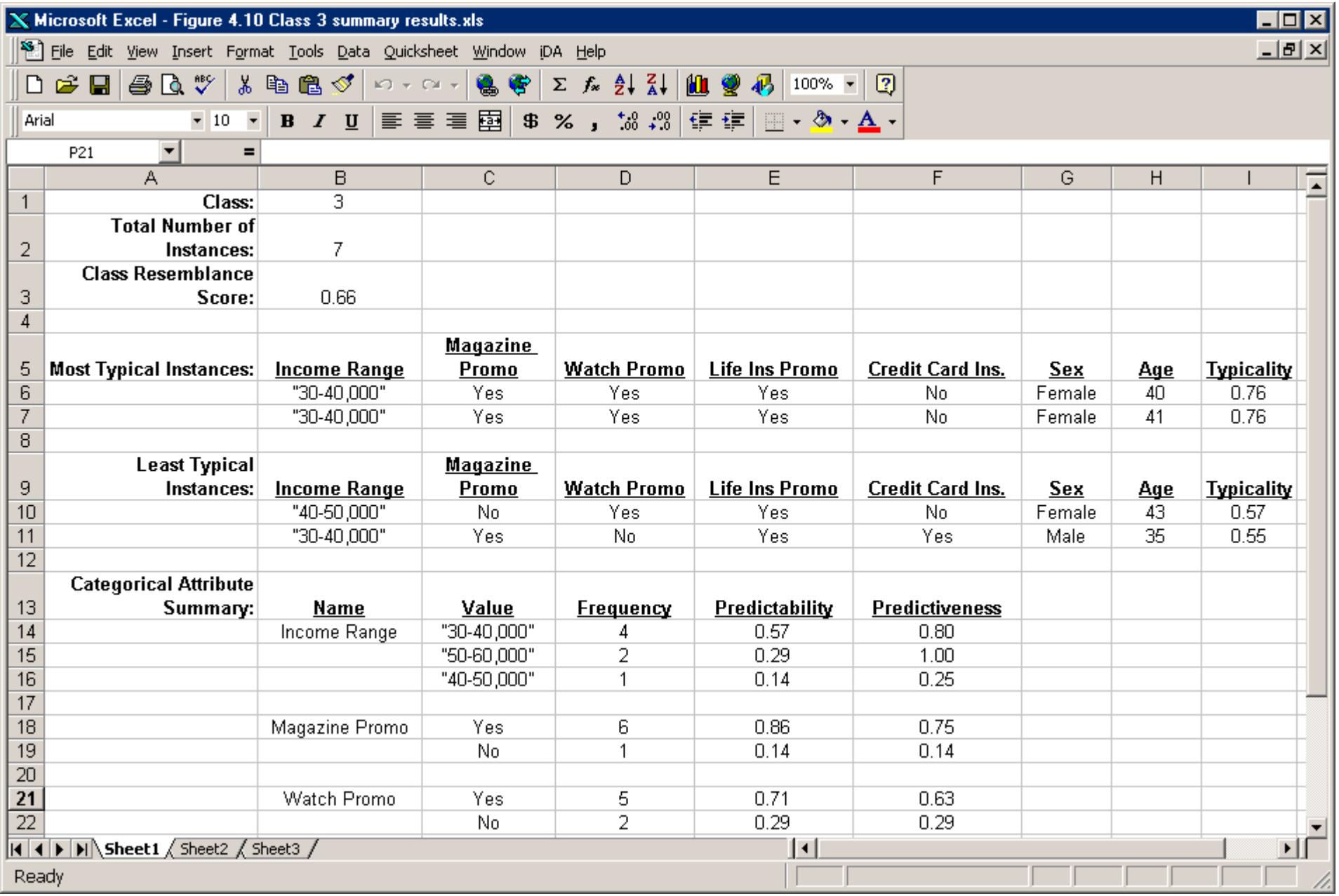


Figure 4.9 Statistics for numerical attributes and

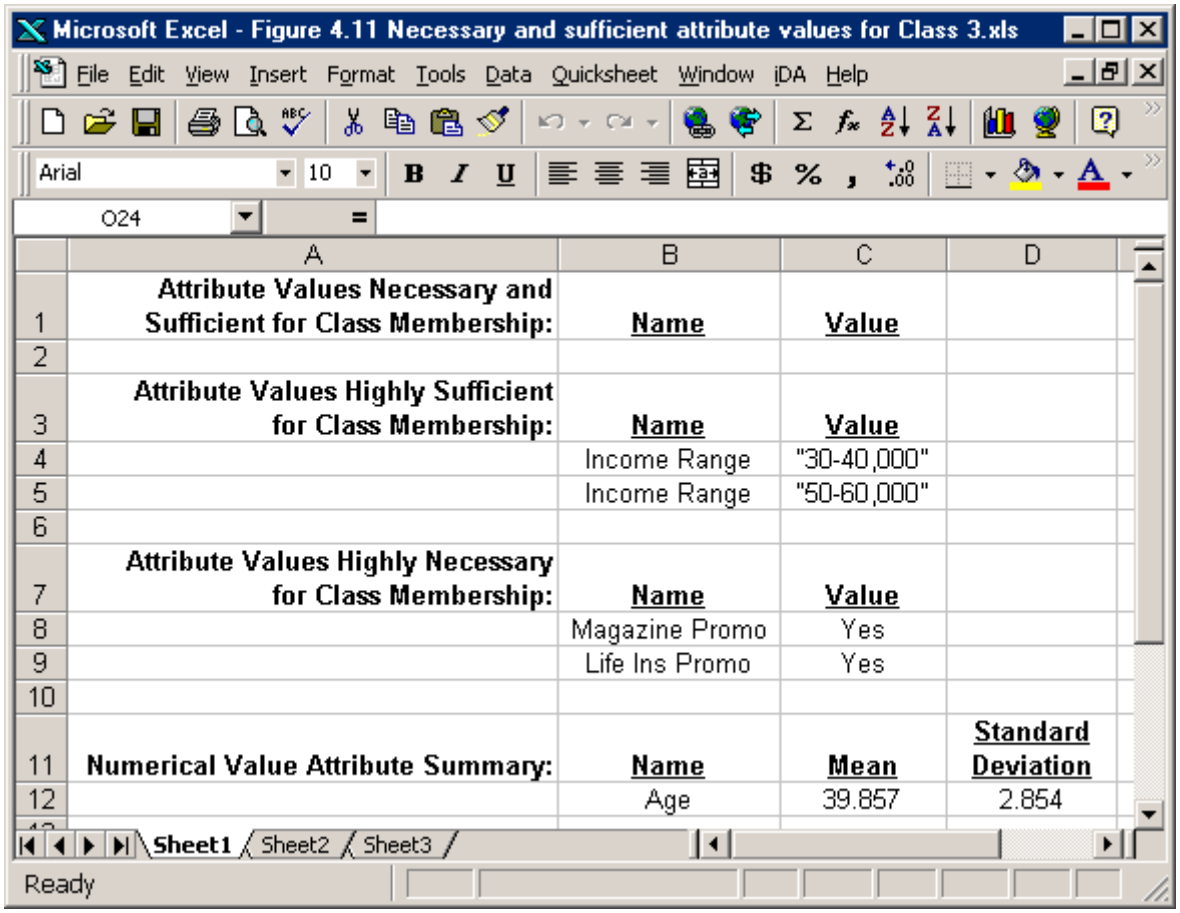
Step 4: Read and Interpret Individual Class Results

- Class Predictability is a within-class measure.
- Class Predictiveness is a between-class measure.



	A	B	C	D	E	F	G	H	I
1	**Class:**	3							
2	**Total Number of Instances:**	7							
3	**Class Resemblance Score:**	0.66							
4									
5	**Most Typical Instances:**	**Income Range**	**Magazine Promo**	**Watch Promo**	**Life Ins Promo**	**Credit Card Ins.**	**Sex**	**Age**	**Typicality**
6		"30-40,000"	Yes	Yes	Yes	No	Female	40	0.76
7		"30-40,000"	Yes	Yes	Yes	No	Female	41	0.76
8									
9	**Least Typical Instances:**	**Income Range**	**Magazine Promo**	**Watch Promo**	**Life Ins Promo**	**Credit Card Ins.**	**Sex**	**Age**	**Typicality**
10		"40-50,000"	No	Yes	Yes	No	Female	43	0.57
11		"30-40,000"	Yes	No	Yes	Yes	Male	35	0.55
12									
13	**Categorical Attribute Summary:**	**Name**	**Value**	**Frequency**	**Predictability**	**Predictiveness**			
14		Income Range	"30-40,000"	4	0.57	0.80			
15			"50-60,000"	2	0.29	1.00			
16			"40-50,000"	1	0.14	0.25			
17									
18		Magazine Promo	Yes	6	0.86	0.75			
19			No	1	0.14	0.14			
20									
21		Watch Promo	Yes	5	0.71	0.63			
22			No	2	0.29	0.29			

Figure 4.10 Class 3 summary results



	A	B	C	D
1	**Attribute Values Necessary and Sufficient for Class Membership:**	**Name**	**Value**	
2				
3	**Attribute Values Highly Sufficient for Class Membership:**	**Name**	**Value**	
4		Income Range	"30-40,000"	
5		Income Range	"50-60,000"	
6				
7	**Attribute Values Highly Necessary for Class Membership:**	**Name**	**Value**	
8		Magazine Promo	Yes	
9		Life Ins Promo	Yes	
10				
11	**Numerical Value Attribute Summary:**	**Name**	**Mean**	**Standard Deviation**
12		Age	39.857	2.854

Figure 4.11 Necessary and sufficient attribute va

Step 5: Visualize Individual Class Rules

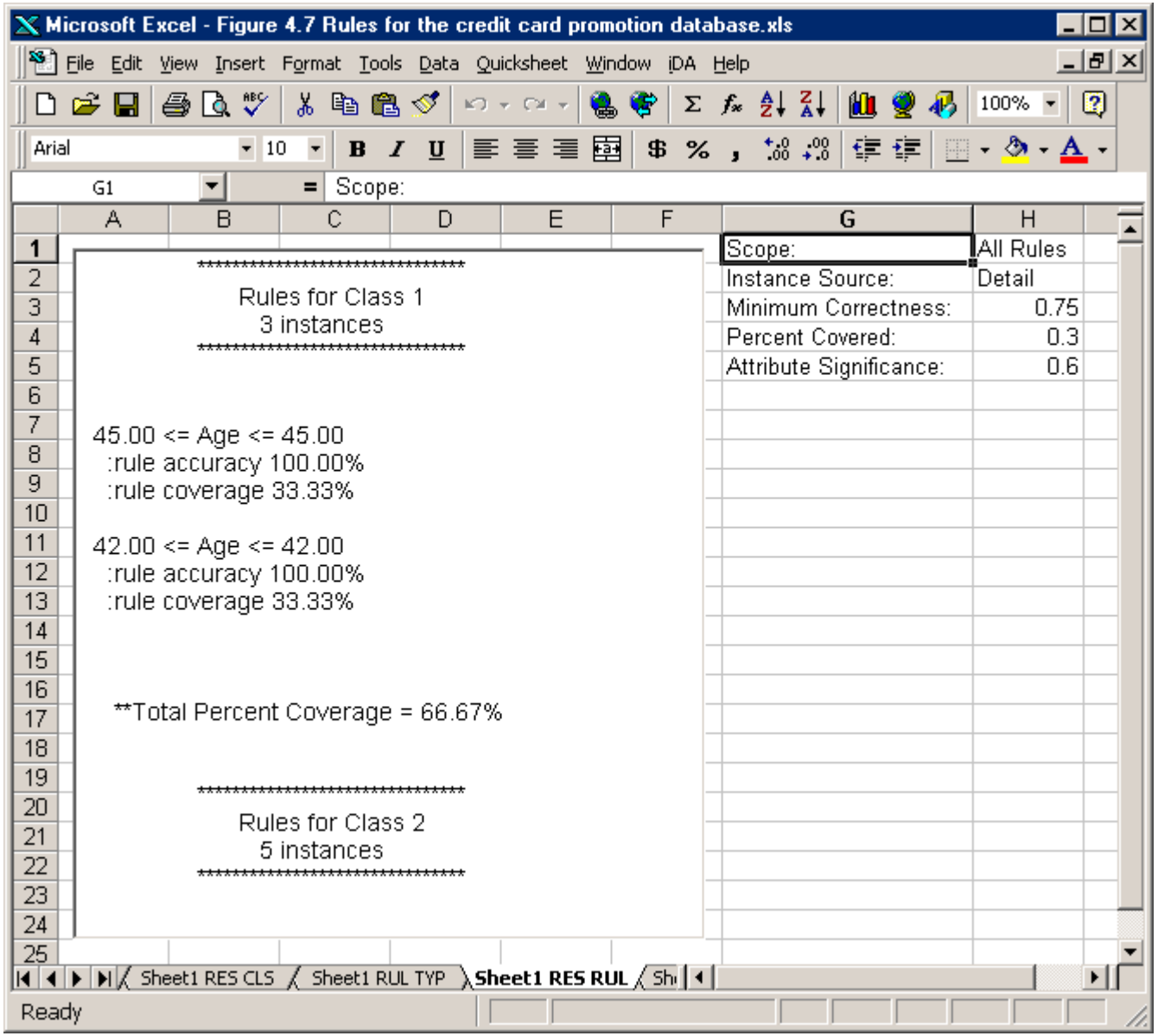


Figure 4.7 Rules for the credit card promotion da