Data Science with R

Cluster Analysis

Graham.Williams@togaware.com

Senior Director and Data Scientist, Analytics
Australian Taxation Office

Adjunct Professor, Australian National University
Fellow, Institute of Analytics Professionals of Australia

Graham.Williams@togaware.com
http://datamining.togaware.com

Overview

1. Cluster Analysis
   - Introduction
   - Requirements
2. Measuring Similarity
   - Distances
   - Data Types
3. Algorithms
   - Cluster Methods
   - K-Means

Overview

1. Cluster Analysis
   - Introduction
   - Requirements
2. Measuring Similarity
   - Distances
   - Data Types
3. Algorithms
   - Cluster Methods
   - K-Means

What is Cluster Analysis?

1. How do we understand the behaviour of an individual?
   - "Paint everyone with the same brush;"
   - Treat everyone as an individual.
2. How do we understand our fellow human beings in the world?
   - Through understanding every individual in the world?
3. We categorise, for *good or bad*, observations into groups:
   - Socio-economic groups: "the poor", "the rich";
   - Political: a lefty, a new right;
   - Racial, religious, geographical, ...;
4. We find that to understand our world we generally talk about groups, not individuals. But computers don't need to—they increasingly have the power to build an understanding of the individual, for *better or worse*—Facebook, gmail, ...

What is Cluster Analysis?

1. How do we understand the behaviour of an individual?
   - "Paint everyone with the same brush;"
   - Treat everyone as an individual.
2. How do we understand our fellow human beings in the world?
   - Through understanding every individual in the world?
3. We categorise, for *good or bad*, observations into groups:
   - Socio-economic groups: "the poor", "the rich";
   - Political: a lefty, a new right;
   - Racial, religious, geographical, ...;
4. We find that to understand our world we generally talk about groups, not individuals. But computers don't need to—they increasingly have the power to build an understanding of the individual, for *better or worse*—Facebook, gmail, ...

What is Cluster Analysis?

1. How do we understand the behaviour of an individual?
   - "Paint everyone with the same brush;"
   - Treat everyone as an individual.
2. How do we understand our fellow human beings in the world?
   - Through understanding every individual in the world?
3. We categorise, for *good or bad*, observations into groups:
   - Socio-economic groups: "the poor", "the rich";
   - Political: a lefty, a new right;
   - Racial, religious, geographical, . . . ;
4. We find that to understand our world we generally talk about groups, not individuals. But computers don't need to—they increasingly have the power to build an understanding of the individual, for *better or worse*—Facebook, gmail, ...

What is Cluster Analysis?

1. How do we understand the behaviour of an individual?
   - "Paint everyone with the same brush;"
   - Treat everyone as an individual.
2. How do we understand our fellow human beings in the world?
   - Through understanding every individual in the world?
3. We categorise, for *good or bad*, observations into groups:
   - Socio-economic groups: "the poor", "the rich";
   - Political: a lefty, a new right;
   - Racial, religious, geographical, ...;
4. We find that to understand our world we generally talk about groups, not individuals. But computers don't need to—they increasingly have the power to build an understanding of the individual, for *better or worse*—Facebook, gmail, ...

Specific Examples—Where Clusters Help

- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Spatial Data Analysis: Create thematic maps by identifying areas of similar land use (by clustering feature spaces) in an earth observation dataset
- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost
- Fraud: Identifying groups of individuals that, as a group, are very different to the other groups.
- WWW: Document classification, question categorisation, and web log data to discover similar access patterns.
- City Planning: Identify services for households according to their house type, value, and geographic location.

Specific Examples—Where Clusters Help

- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Spatial Data Analysis: Create thematic maps by identifying areas of similar land use (by clustering feature spaces) in an earth observation dataset
- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost
- Fraud: Identifying groups of individuals that, as a group, are very different to the other groups.
- WWW: Document classification, question categorisation, and web log data to discover similar access patterns.
- City Planning: Identify services for households according to their house type, value, and geographic location.

Specific Examples—Where Clusters Help

- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Spatial Data Analysis: Create thematic maps by identifying areas of similar land use (by clustering feature spaces) in an earth observation dataset
- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost
- Fraud: Identifying groups of individuals that, as a group, are very different to the other groups.
- WWW: Document classification, question categorisation, and web log data to discover similar access patterns.
- City Planning: Identify services for households according to their house type, value, and geographic location.

Specific Examples—Where Clusters Help

- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Spatial Data Analysis: Create thematic maps by identifying areas of similar land use (by clustering feature spaces) in an earth observation dataset
- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost
- Fraud: Identifying groups of individuals that, as a group, are very different to the other groups.
- WWW: Document classification, question categorisation, and web log data to discover similar access patterns.
- City Planning: Identify services for households according to their house type, value, and geographic location.

Specific Examples—Where Clusters Help

- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Spatial Data Analysis: Create thematic maps by identifying areas of similar land use (by clustering feature spaces) in an earth observation dataset
- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost
- Fraud: Identifying groups of individuals that, as a group, are very different to the other groups.
- WWW: Document classification, question categorisation, and web log data to discover similar access patterns.
- City Planning: Identify services for households according to their house type, value, and geographic location.

Specific Examples—Where Clusters Help

- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Spatial Data Analysis: Create thematic maps by identifying areas of similar land use (by clustering feature spaces) in an earth observation dataset
- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost
- Fraud: Identifying groups of individuals that, as a group, are very different to the other groups.
- WWW: Document classification, question categorisation, and web log data to discover similar access patterns.
- City Planning: Identify services for households according to their house type, value, and geographic location.

What is Cluster Analysis?

- **Cluster**: a collection of observations
  - Similar to one another within the same cluster
  - Dissimilar to the observations in other clusters
- **Cluster analysis**
  - Grouping a set of data observations into classes
- Clustering is unsupervised classification: no predefined classes—descriptive data mining.
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

What is Cluster Analysis?

- **Cluster**: a collection of observations
  - Similar to one another within the same cluster
  - Dissimilar to the observations in other clusters
- **Cluster analysis**
  - Grouping a set of data observations into classes
- Clustering is unsupervised classification: no predefined classes—descriptive data mining.
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

What is Cluster Analysis?

- **Cluster**: a collection of observations
  - Similar to one another within the same cluster
  - Dissimilar to the observations in other clusters
- **Cluster analysis**
  - Grouping a set of data observations into classes
- Clustering is unsupervised classification: no predefined classes—descriptive data mining.
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

What is Cluster Analysis?

- **Cluster**: a collection of observations
  - Similar to one another within the same cluster
  - Dissimilar to the observations in other clusters
- **Cluster analysis**
  - Grouping a set of data observations into classes
- Clustering is unsupervised classification: no predefined classes—descriptive data mining.
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

What Is Good Clustering?

- High Quality:
  - high intra-class similarity
  - low inter-class similarity
- The Quality depends on:
  - similarity measure
  - algorithm for searching
- *Depends on the opinion of the user, and the algorithm's ability to discover hidden patterns that are of interest to the user.*

What Is Good Clustering?

- High Quality:
  - high intra-class similarity
  - low inter-class similarity
- The Quality depends on:
  - similarity measure
  - algorithm for searching

- *Depends on the opinion of the user, and the algorithm's ability to discover hidden patterns that are of interest to the user.*

What Is Good Clustering?

- High Quality:
  - high intra-class similarity
  - low inter-class similarity
- The Quality depends on:
  - similarity measure
  - algorithm for searching

- *Depends on the opinion of the user, and the algorithm's ability to discover hidden patterns that are of interest to the user.*

Clustering Caveats

Clustering may not be the best way to discover interesting groups in a data set. Often visulisation methods work well, allowing the human expert to identify useful groups. However, as the data set sizes increase to millions of observations, this becomes impractical and clusters help to partition the data so that we can deal with smaller groups.

Different algorithms, and even multiple runs of the one algorithm, will deliver different clusterings.

Clustering Caveats

Clustering may not be the best way to discover interesting groups in a data set. Often visulisation methods work well, allowing the human expert to identify useful groups. However, as the data set sizes increase to millions of observations, this becomes impractical and clusters help to partition the data so that we can deal with smaller groups.

Different algorithms, and even multiple runs of the one algorithm, will deliver different clusterings.

Overview

1 Cluster Analysis
- Introduction
- Requirements

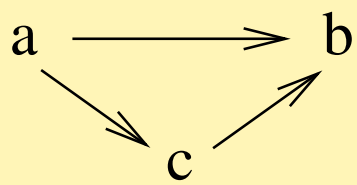
2 Measuring Similarity
- Distances
- Data Types

3 Algorithms
- Cluster Methods
- K-Means

Similarity and Dissimilarity Between Observations

- "Distance" measures the dissimilarity between two data observations $a = (a_1, a_2, \ldots, a_n)$ and $b = (b_1, b_2, \ldots, b_n)$.

- A distance measure should satisfy the following requirements:
  - $d(a, b) \geq 0$ — distance is non-negative
  - $d(a, a) = 0$ — distance to itself is 0
  - $d(a, b) = d(b, a)$ — distance is symmetric
  - $d(a, b) \leq d(a, c) + d(c, b)$ — triangular inequality



Euclidean Distance

$$d(a, b) = \sqrt{(|a_1 - b_1|^2 + |a_2 - b_2|^2 + \ldots + |a_n - b_n|^2)}$$

PLOT HERE

Manhattan Distance

$$d(a, b) = |a_1 - b_1| + |a_2 - b_2| + \ldots + |a_n - b_n|$$

PLOT HERE

Minkowski Distance

A general measure of distance:

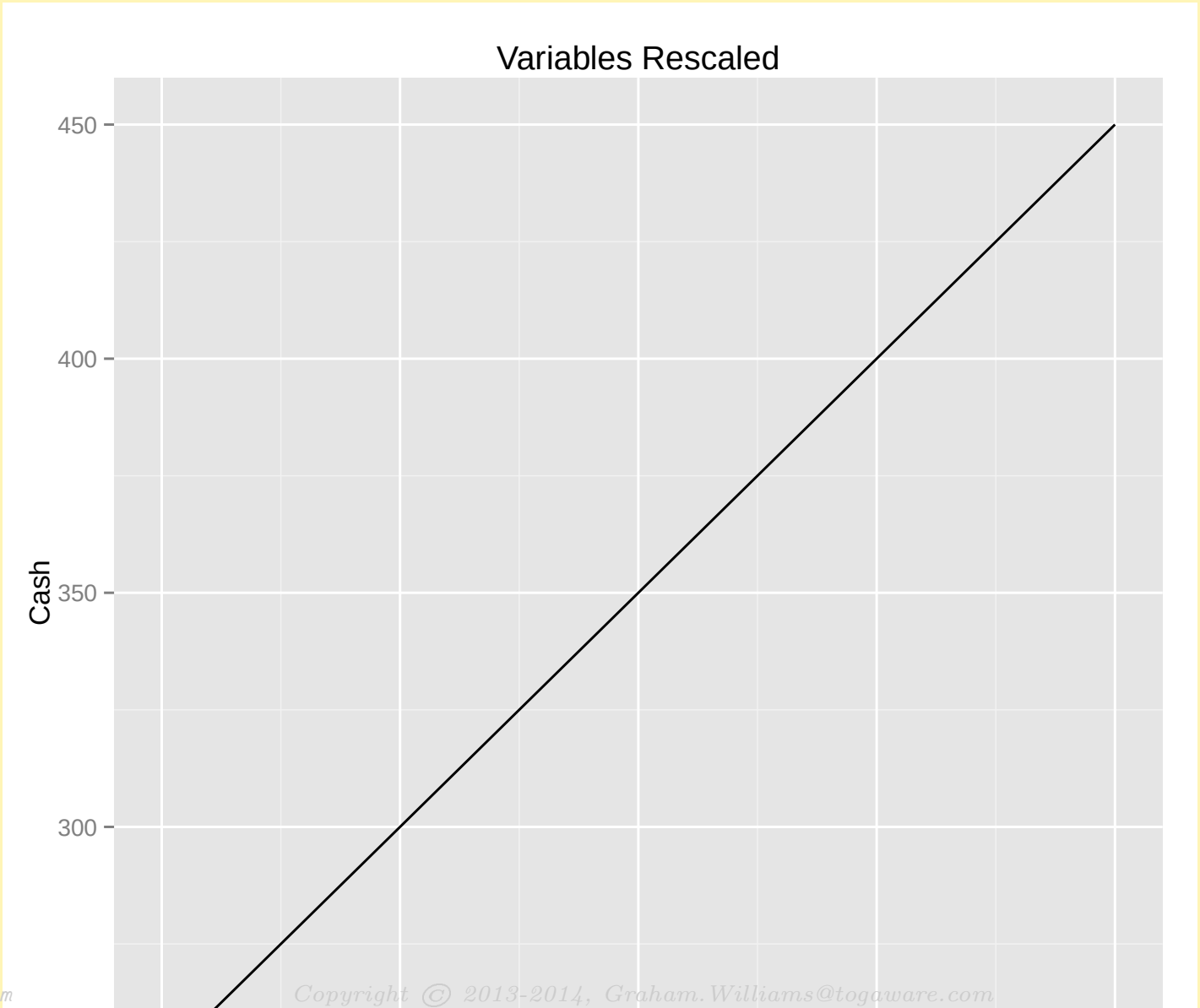
$$d(a,b) = \sqrt[q]{(|a_1 - b_1|^q + |a_2 - b_2|^q + \ldots + |a_n - b_n|^q)}$$

- If $q = 2$, d is the Euclidean distance.
- If $q = 1$, d is the Manhattan distance.
- A variation is the weighted distance (variables have different importance):

$$d(a,b) = \sqrt[q]{(w_1|a_1 - b_1|^q + w_2|a_2 - b_2|^q + \ldots + w_n|a_n - b_n|^q)}$$

Issue of Scale

Illustrate difference between Age and Cash on a plot.



Type of data in clustering analysis

A distance measure works well for numeric variables, but what about other types of variables?

Type types of variables include:

- Numeric, interval-scaled variables
- Binary variables
- Categorical, ordinal, and ratio variables
- Variables of mixed types

Interval-Scaled Variables

Interval-scaled variables are continuous variables of a roughly linear scale.

- The Euclidean distance or some other instance of the Minkowski distance can be used.
- Before applying the distance measure, the variables need to be normalized:
  - Variables with larger ranges (e.g., *income*) will overwhelm variables with smaller ranges (e.g., *age*):

    $\$50,000 - \$40,000 = 10,000$ versus $50 years - 40 years = 10$
  - Variation of *z-score normalisation*:

    $$v' = \frac{v - m}{s}$$

    where m is the **mean** and s is the **mean absolute deviation**

Interval-Scaled Variables

Interval-scaled variables are continuous variables of a roughly linear scale.

- The Euclidean distance or some other instance of the Minkowski distance can be used.
- Before applying the distance measure, the variables need to be normalized:
  - Variables with larger ranges (e.g., *income*) will overwhelm variables with smaller ranges (e.g., *age*):

    $\$50,000 - \$40,000 = 10,000$ versus $50 years - 40 years = 10$
  - Variation of *z-score normalisation*:

    $$v' = \frac{v - m}{s}$$

    where m is the **mean** and s is the **mean absolute deviation**

Interval-Scaled Variables

Interval-scaled variables are continuous variables of a roughly linear scale.

- The Euclidean distance or some other instance of the Minkowski distance can be used.
- Before applying the distance measure, the variables need to be normalized:
  - Variables with larger ranges (e.g., *income*) will overwhelm variables with smaller ranges (e.g., *age*):

    $\$50,000 - \$40,000 = 10,000$ versus $50 years - 40 years = 10$

  - Variation of *z-score normalisation*:

    $$v' = \frac{v - m}{s}$$

    where m is the **mean** and s is the **mean absolute deviation** (c.f. stdev: robust to outliers and retains the outliers)

Interval-Scaled Variables

Interval-scaled variables are continuous variables of a roughly linear scale.

- The Euclidean distance or some other instance of the Minkowski distance can be used.
- Before applying the distance measure, the variables need to be normalized:
  - Variables with larger ranges (e.g., *income*) will overwhelm variables with smaller ranges (e.g., *age*):

    $\$50,000 - \$40,000 = 10,000$ versus $50 \mathit{years} - 40 \mathit{years} = 10$

  - Variation of *z-score normalisation*:

$$v' = \frac{v - m}{s}$$

    where m is the **mean** and s is the **mean absolute deviation** (c.f. stdev: robust to outliers and retains the outliers)

Interval-Scaled Variables

Interval-scaled variables are continuous variables of a roughly linear scale.

- The Euclidean distance or some other instance of the Minkowski distance can be used.
- Before applying the distance measure, the variables need to be normalized:
  - Variables with larger ranges (e.g., *income*) will overwhelm variables with smaller ranges (e.g., *age*):

    $\$50,000 - \$40,000 = 10,000$ versus $50 years - 40 years = 10$

  - Variation of *z-score normalisation*:

$$v' = \frac{v - m}{s}$$

    where m is the **mean** and s is the **mean absolute deviation** (c.f. stdev: robust to outliers and retains the outliers)

Interval-Scaled Variables

Interval-scaled variables are continuous variables of a roughly linear scale.

- The Euclidean distance or some other instance of the Minkowski distance can be used.
- Before applying the distance measure, the variables need to be normalized:
  - Variables with larger ranges (e.g., *income*) will overwhelm variables with smaller ranges (e.g., *age*):

    $\$50,000 - \$40,000 = 10,000$ versus $50 \mathit{years} - 40 \mathit{years} = 10$

  - Variation of *z-score normalisation*:

$$v' = \frac{v - m}{s}$$

    where m is the **mean** and s is the **mean absolute deviation** (c.f. stdev: robust to outliers and retains the outliers)

Binary Variables

Binary variables have just two possible values: 0 and 1.

We consider as a group all of the binary variables and count for observation x_i and x_j the number of times they both have 0, 1, or $(0, 1)$ or $(1, 0)$ to build a contingency table:

	1	0	sum
1	a	b	a+b
0	c	d	c+d
sum	a+c	b+d	n

- Simple matching coefficient (symmetric variable):

$$d(x_i, x_j) = \frac{b + c}{a + b + c + d}$$

- Jaccard coefficient
  (asymmetric: 1 is more important - e.g. diseases):

$$d(x_i, x_j) = \frac{b + c}{a + b + c}$$

Binary Variables

Binary variables have just two possible values: 0 and 1.

We consider as a group all of the binary variables and count for observation x_i and x_j the number of times they both have 0, 1, or $(0, 1)$ or $(1, 0)$ to build a contingency table:

	1	0	sum
1	a	b	a+b
0	c	d	c+d
sum	a+c	b+d	n

- Simple matching coefficient (symmetric variable):

$$d(x_i, x_j) = \frac{b + c}{a + b + c + d}$$

- Jaccard coefficient
  (asymmetric: 1 is more important - e.g. diseases):

$$d(x_i, x_j) = \frac{b + c}{a + b + c}$$

Binary Variables

Binary variables have just two possible values: 0 and 1.

We consider as a group all of the binary variables and count for observation x_i and x_j the number of times they both have 0, 1, or $(0, 1)$ or $(1, 0)$ to build a contingency table:

	1	0	sum
1	a	b	a+b
0	c	d	c+d
sum	a+c	b+d	n

- Simple matching coefficient (symmetric variable):

$$d(x_i, x_j) = \frac{b + c}{a + b + c + d}$$

- Jaccard coefficient
  (asymmetric: 1 is more important - e.g. diseases):

$$d(x_i, x_j) = \frac{b + c}{a + b + c}$$

Binary Variables

Binary variables have just two possible values: 0 and 1.
We consider as a group all of the binary variables and count for observation x_i and x_j the number of times they both have 0, 1, or $(0, 1)$ or $(1, 0)$ to build a contingency table:

	1	0	sum
1	a	b	a+b
0	c	d	c+d
sum	a+c	b+d	n

- Simple matching coefficient (symmetric variable):

$$d(x_i, x_j) = \frac{b + c}{a + b + c + d}$$

- Jaccard coefficient
(asymmetric: 1 is more important - e.g. diseases):

$$d(x_i, x_j) = \frac{b + c}{a + b + c}$$

Binary Variables

Binary variables have just two possible values: 0 and 1.
We consider as a group all of the binary variables and count for observation x_i and x_j the number of times they both have 0, 1, or $(0,1)$ or $(1,0)$ to build a contingency table:

	1	0	sum
1	a	b	a+b
0	c	d	c+d
sum	a+c	b+d	n

- Simple matching coefficient (symmetric variable):

$$d(x_i, x_j) = \frac{b + c}{a + b + c + d}$$

- Jaccard coefficient
(asymmetric: 1 is more important - e.g. diseases):

$$d(x_i, x_j) = \frac{b + c}{a + b + c}$$

Categorical Variables

- A generalisation of the binary variable in that it can take more than 2 levels, e.g., red, yellow, blue, green.
- Method 1: Simple matching

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

where p is the number of matched categorical variables and n is the total number of variables.

- Method 2: Convert each level into a binary variable, creating many new binary variables.

Categorical Variables

- A generalisation of the binary variable in that it can take more than 2 levels, e.g., red, yellow, blue, green.
- Method 1: Simple matching

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

where p is the number of matched categorical variables and n is the total number of variables.

- Method 2: Convert each level into a binary variable, creating many new binary variables.

Categorical Variables

- A generalisation of the binary variable in that it can take more than 2 levels, e.g., red, yellow, blue, green.
- Method 1: Simple matching

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

where p is the number of matched categorical variables and n is the total number of variables.

- Method 2: Convert each level into a binary variable, creating many new binary variables.

Variables of Mixed Types

- A dataset may contain all types of variables: interval, binary, categorical.
- Use a weighted formula to combine the different normalised (to $[0, 1]$) distances, where the weights are used to express the relative importance of the variables:

$$d(x_i, x_j) = \sum_k w_k d_{ij}^{A_k}$$

where w_k is the weight of variable A_k, $d_{ij}^{A_k}$ is the dissimilarity of the ith observation and the jth observation on variable A_k. $d_{ij}^{A_k}$ is normalized to $[0, 1]$

Variables of Mixed Types

- A dataset may contain all types of variables: interval, binary, categorical.
- Use a weighted formula to combine the different normalised (to $[0, 1]$) distances, where the weights are used to express the relative importance of the variables:

$$d(x_i, x_j) = \sum_k w_k d_{ij}^{A_k}$$

where w_k is the weight of variable A_k, $d_{ij}^{A_k}$ is the dissimilarity of the ith observation and the jth observation on variable A_k. $d_{ij}^{A_k}$ is normalized to $[0, 1]$

Overview

1. Cluster Analysis
   - Introduction
   - Requirements
2. Measuring Similarity
   - Distances
   - Data Types
3. Algorithms
   - Cluster Methods
   - K-Means

Requirements of Clustering in Data Mining

- Scalability—many observations and variables
- Different variable types—limit to numerics
- Clusters with arbitrary shape
- Minimal domain knowledge required
- Can cope with noise and outliers
- Insensitive to order of input records
- High dimensionality—curse of dimensionality

Major Clustering Approaches

- Partitioning algorithms (`kmeans`, `pam`, `clara`, `fanny`): Construct various partitions and then evaluate them by some criterion. A fixed number of clusters, k, is generated. Start with an initial (perhaps random) cluster.
- Hierarchical algorithms: (`hclust`, `agnes`, `diana`) Create a hierarchical decomposition of the set of observations using some criterion
- Density-based algorithms: based on connectivity and density functions
- Grid-based algorithms: based on a multiple-level granularity structure
- Model-based algorithms: (`mclust` for mixture of Gaussians) A model is hypothesized for each of the clusters and the idea is to find the best fit of that model

Major Clustering Approaches

- Partitioning algorithms (`kmeans`, `pam`, `clara`, `fanny`): Construct various partitions and then evaluate them by some criterion. A fixed number of clusters, k, is generated. Start with an initial (perhaps random) cluster.
- Hierarchical algorithms: (`hclust`, `agnes`, `diana`) Create a hierarchical decomposition of the set of observations using some criterion
- Density-based algorithms: based on connectivity and density functions
- Grid-based algorithms: based on a multiple-level granularity structure
- Model-based algorithms: (`mclust` for mixture of Gaussians) A model is hypothesized for each of the clusters and the idea is to find the best fit of that model

Major Clustering Approaches

- Partitioning algorithms (`kmeans`, `pam`, `clara`, `fanny`): Construct various partitions and then evaluate them by some criterion. A fixed number of clusters, k, is generated. Start with an initial (perhaps random) cluster.
- Hierarchical algorithms: (`hclust`, `agnes`, `diana`) Create a hierarchical decomposition of the set of observations using some criterion
- Density-based algorithms: based on connectivity and density functions
- Grid-based algorithms: based on a multiple-level granularity structure
- Model-based algorithms: (`mclust` for mixture of Gaussians) A model is hypothesized for each of the clusters and the idea is to find the best fit of that model

Major Clustering Approaches

- Partitioning algorithms (`kmeans`, `pam`, `clara`, `fanny`): Construct various partitions and then evaluate them by some criterion. A fixed number of clusters, k, is generated. Start with an initial (perhaps random) cluster.
- Hierarchical algorithms: (`hclust`, `agnes`, `diana`) Create a hierarchical decomposition of the set of observations using some criterion
- Density-based algorithms: based on connectivity and density functions
- Grid-based algorithms: based on a multiple-level granularity structure
- Model-based algorithms: (`mclust` for mixture of Gaussians) A model is hypothesized for each of the clusters and the idea is to find the best fit of that model

Major Clustering Approaches

- Partitioning algorithms (`kmeans`, `pam`, `clara`, `fanny`): Construct various partitions and then evaluate them by some criterion. A fixed number of clusters, k, is generated. Start with an initial (perhaps random) cluster.
- Hierarchical algorithms: (`hclust`, `agnes`, `diana`) Create a hierarchical decomposition of the set of observations using some criterion
- Density-based algorithms: based on connectivity and density functions
- Grid-based algorithms: based on a multiple-level granularity structure
- Model-based algorithms: (`mclust` for mixture of Gaussians) A model is hypothesized for each of the clusters and the idea is to find the best fit of that model

Basic Partitioning Algorithm

- Partition dataset D of n observations into k clusters
  - Given k, find k clusters that optimises partitioning criterion
  - Global optimal: exhaustively enumerate all partitions

- Heuristic methods: k-means and k-medoids algorithms
  - k-means: Each cluster represented by center of the cluster
  - k-medoids or PAM (partition around medoids): Each cluster represented by one of the observations in the cluster

Basic Partitioning Algorithm

- Partition dataset D of n observations into k clusters
  - Given k, find k clusters that optimises partitioning criterion
  - Global optimal: exhaustively enumerate all partitions

- Heuristic methods: k-means and k-medoids algorithms
  - k-means: Each cluster represented by center of the cluster
  - k-medoids or PAM (partition around medoids): Each cluster represented by one of the observations in the cluster

The K-Means Clustering Method

- Given k, the k-means algorithm is implemented in 4 steps:
  1. Partition observations into k random nonempty subsets
  2. Compute seed points as the centroids of the clusters of the current partition. The centroid is the center (mean point) of the cluster.
  3. Assign each observation to the cluster with the nearest seed point.
  4. Go back to Step 2 unless no observations change clustering.

The K-Means Clustering Method

- Given k, the k-means algorithm is implemented in 4 steps:
  1. Partition observations into k random nonempty subsets
  2. Compute seed points as the centroids of the clusters of the current partition. The centroid is the center (mean point) of the cluster.
  3. Assign each observation to the cluster with the nearest seed point.
  4. Go back to Step 2 unless no observations change clustering.

The K-Means Clustering Method

- Given k, the k-means algorithm is implemented in 4 steps:
  1. Partition observations into k random nonempty subsets
  2. Compute seed points as the centroids of the clusters of the current partition. The centroid is the center (mean point) of the cluster.
  3. Assign each observation to the cluster with the nearest seed point.
  4. Go back to Step 2 unless no observations change clustering.

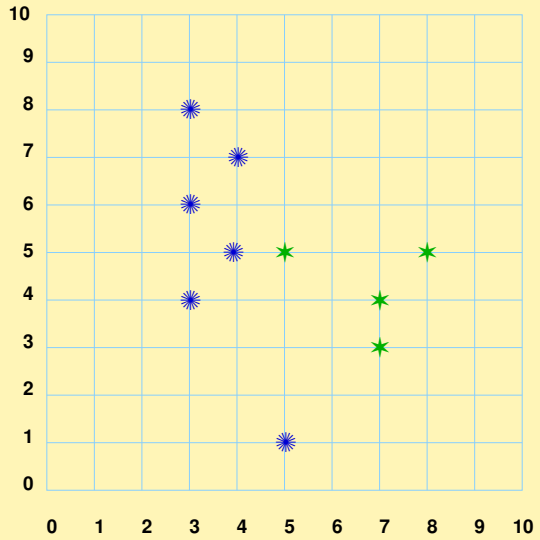
The K-Means Clustering Method

- Given k, the k-means algorithm is implemented in 4 steps:
  1. Partition observations into k random nonempty subsets
  2. Compute seed points as the centroids of the clusters of the current partition. The centroid is the center (mean point) of the cluster.
  3. Assign each observation to the cluster with the nearest seed point.
  4. Go back to Step 2 unless no observations change clustering.

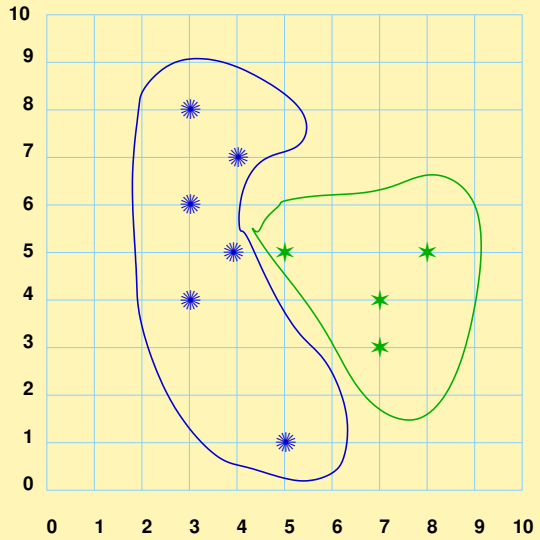
The K-Means Clustering Method

- Given k, the k-means algorithm is implemented in 4 steps:
  1. Partition observations into k random nonempty subsets
  2. Compute seed points as the centroids of the clusters of the current partition. The centroid is the center (mean point) of the cluster.
  3. Assign each observation to the cluster with the nearest seed point.
  4. Go back to Step 2 unless no observations change clustering.

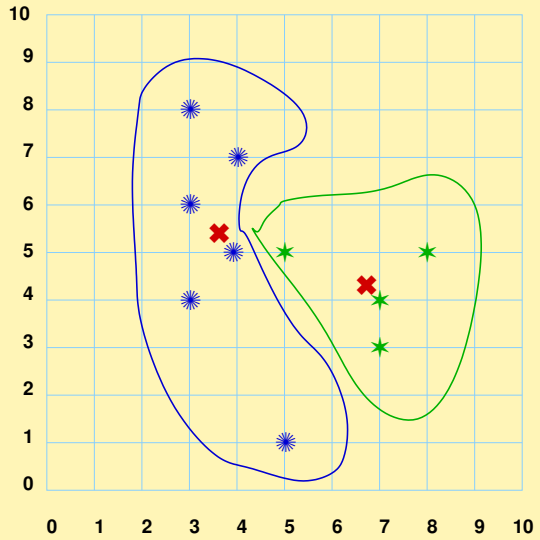
The K-Means Clustering Method



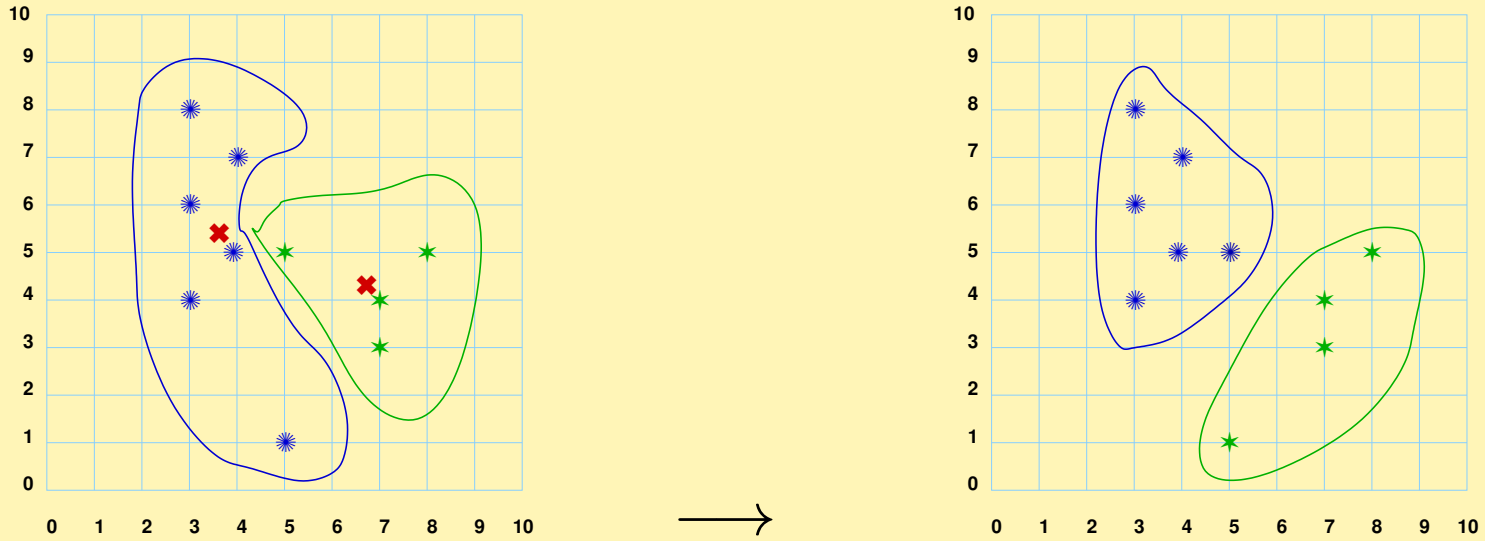
The K-Means Clustering Method



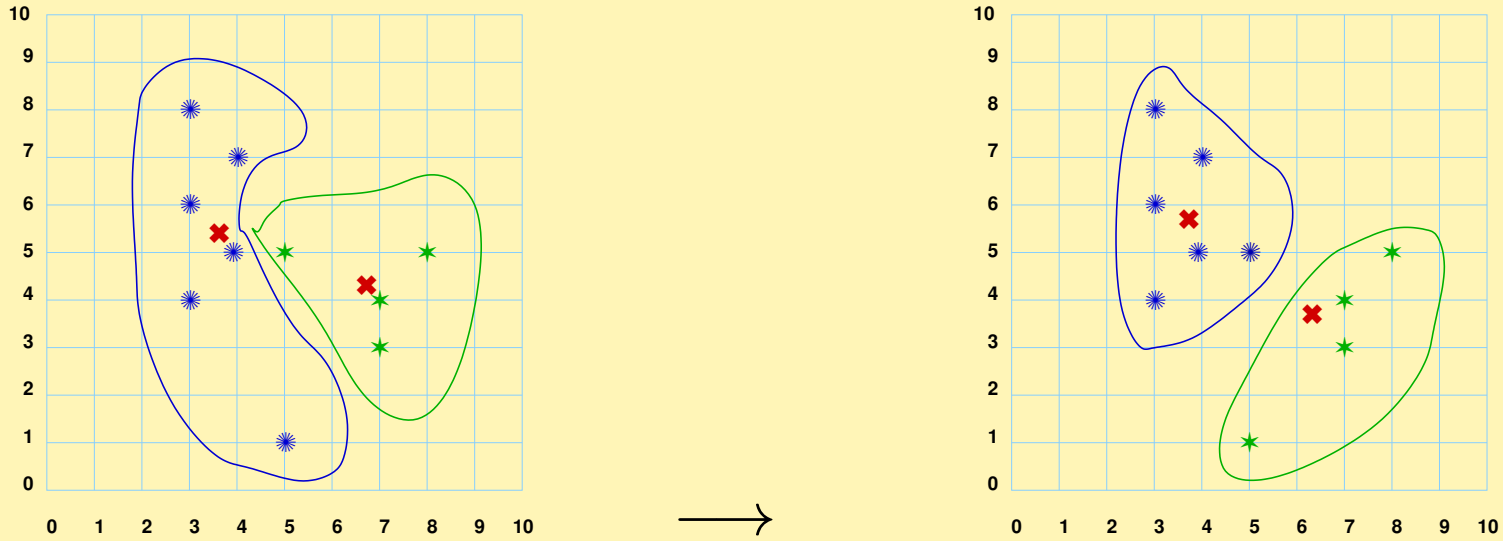
The K-Means Clustering Method



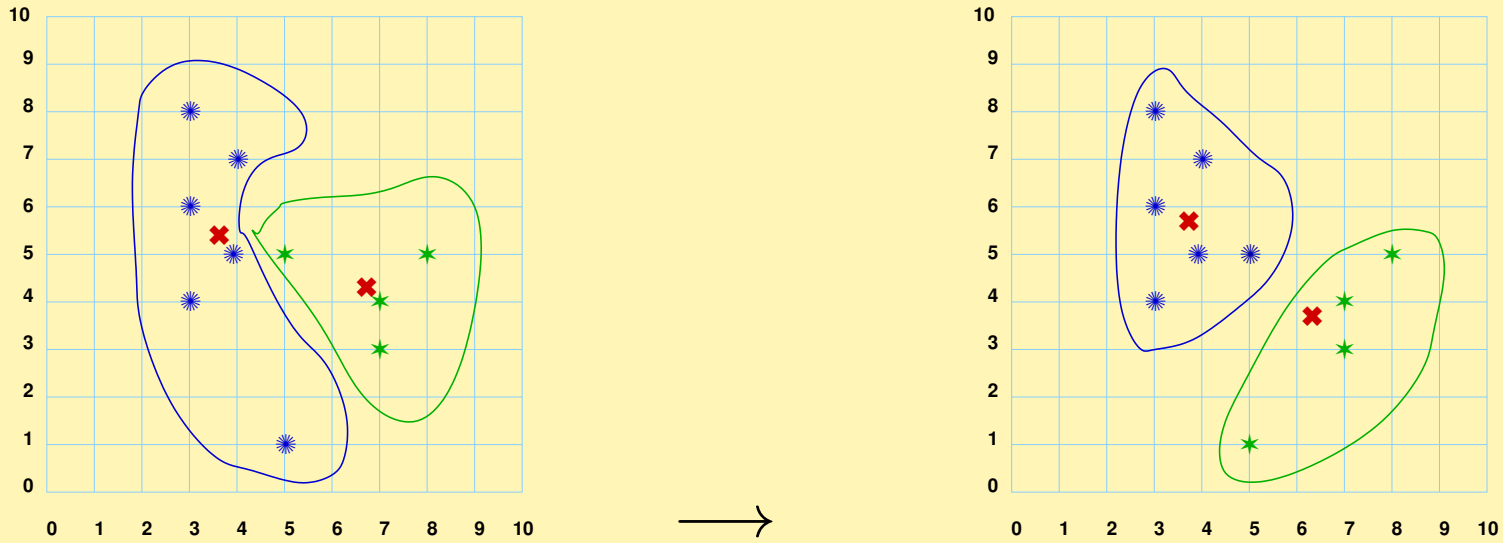
The K-Means Clustering Method



The K-Means Clustering Method



The K-Means Clustering Method



No observations changed clusters - **stop**.

Comments on K-Means

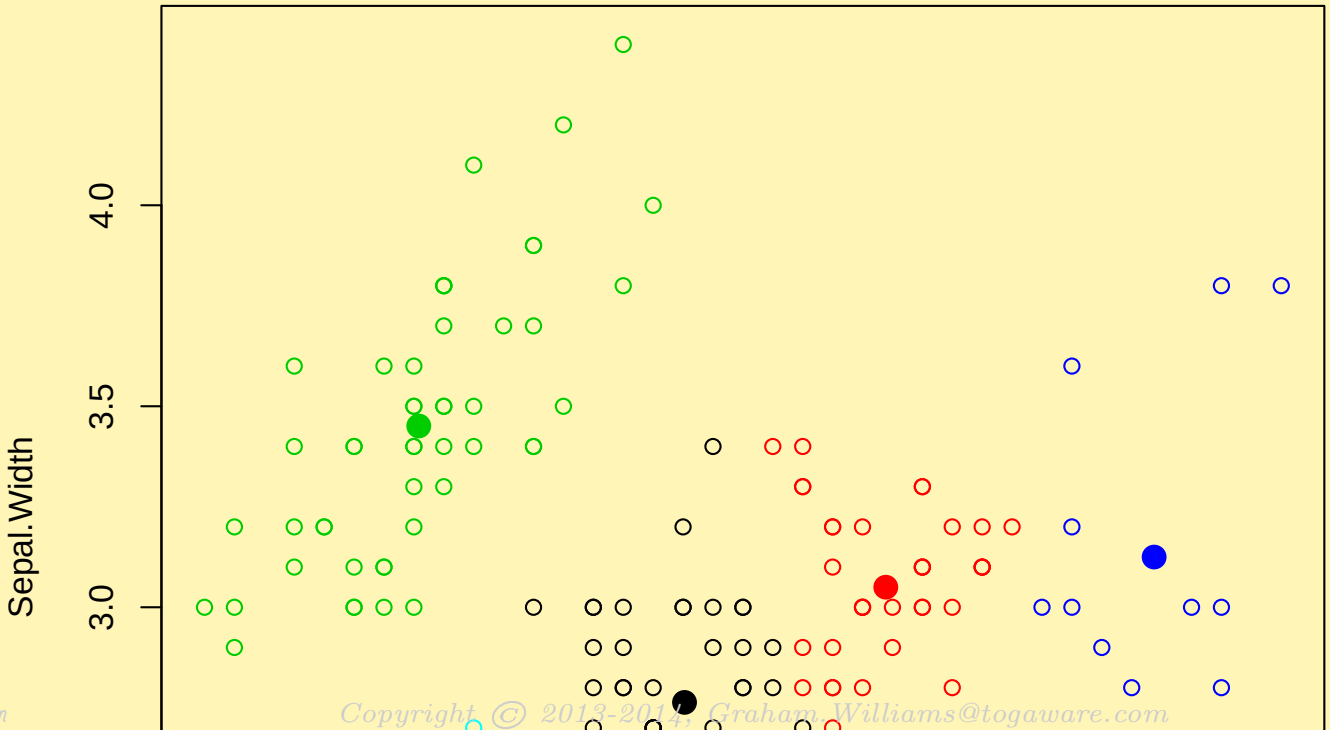
- Strengths
  - Relatively efficient: $O(tkn)$, where n is the number observations, k is the number of clusters, and t is the number iterations. Normally, k, $t \ll n$.

Comments on K-Means

- Weaknesses
  - Often terminates at a local optimum. The global optimum may be found using techniques such as: deterministic annealing and genetic algorithms
  - Applicable only when the mean is defined—what about categorical data?
  - Need to specify k, the number of clusters, in advance. There are algorithms that can help with this.
  - Unable to handle noisy data and outliers.
  - Not suitable for non-convex clusters.

KMeans in R

```
k <- 5
set.seed(42)
iris.kmeans <- kmeans(iris[1:2], k)
plot(iris[1:2], col=iris.kmeans$cluster)
points(iris.kmeans$centers, pch=19, cex=1.5, col=1:k)
```



KMeans Iterations

Experiment with the iterations at this week's tutorial:

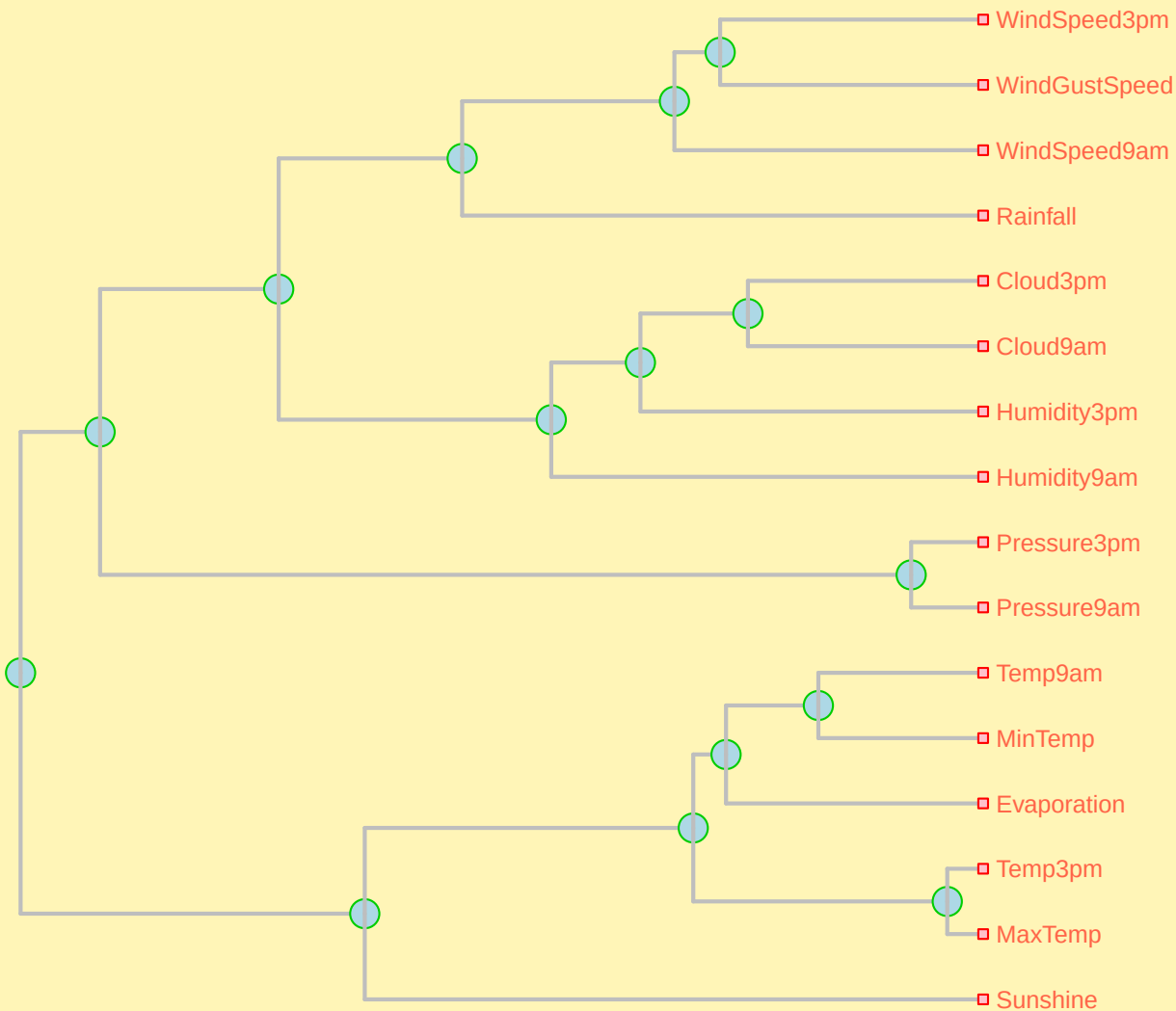
```
k <- 5
set.seed(42)

iris.kmeans <- kmeans(iris[1:2], k, iter.max=1)
plot(iris[1:2], col=iris.kmeans$cluster)
points(iris.kmeans$centers, pch=19, cex=1.5, col=1:k)

x11() # or windows() to start a new plot
iris.kmeans <- kmeans(iris[1:2], k, iter.max=2)
plot(iris[1:2], col=iris.kmeans$cluster)
points(iris.kmeans$centers, pch=19, cex=1.5, col=1:k)

x11() # or windows() to start a new plot
iris.kmeans <- kmeans(iris[1:2], k, iter.max=3)
plot(iris[1:2], col=iris.kmeans$cluster)
points(iris.kmeans$centers, pch=19, cex=1.5, col=1:k)
```

Rattle: Hierarchical Variable Cluster



Overview

1. Cluster Analysis
   - Introduction
   - Requirements
2. Measuring Similarity
   - Distances
   - Data Types
3. Algorithms
   - Cluster Methods
   - K-Means

Modelling Framework

Language Set of means

Measure Minimise intra cluster distance,
maximise inter cluster distance

Search Random assignment, calculate mean, reassign, ...

Summary

- Cluster analysis is unsupervised learning.
- Useful for partitioning a very large population, perhaps for data mining each sub-population separately.
- Often more effective under expert guidance.

Summary

- Cluster analysis is unsupervised learning.
- Useful for partitioning a very large population, perhaps for data mining each sub-population separately.
- Often more effective under expert guidance.

Summary

- Cluster analysis is unsupervised learning.
- Useful for partitioning a very large population, perhaps for data mining each sub-population separately.
- Often more effective under expert guidance.

Summary

- Cluster analysis is unsupervised learning.
- Useful for partitioning a very large population, perhaps for data mining each sub-population separately.
- Often more effective under expert guidance.