Data Mining Algorithms In R/Clustering/Expectation Maximization (EM)

< Data Mining Algorithms In R | Clustering

A Wikibookian suggests that *Data Mining Algorithms In R/Clustering/Expectation Maximization* be merged into this book or chapter.
Discuss whether or not this merger should happen on the discussion page.

A Wikibookian suggests that *Data Mining Algorithms In R/Clustering/Expectation Maximization (soon)* be merged into this book or chapter.
Discuss whether or not this merger should happen on the discussion page.

This chapter intends to give an overview of the technique Expectation Maximization (EM), proposed by [1] (although the technique was informally proposed in literature, as suggested by the author) in the context of R-Project environment. The first section gives an introduction of representative clustering and mixture models. The algorithm details and a case study will be presented on the second section.

The R package that will be used is the MCLUST-v3.3.2 developed by Chris Fraley and Adrian Raftery, available in CRAN repository. The MCLUST tool is a software that includes the following features: normal mixture modeling (EM); EM initialization through an hierarquical clustering approach; estimate the number of clusters based on the Bayesian Information Criteria (BIC); and displays, including uncertainty plots and dimension projections.

The information sources of this document were divided in two groups: (i) manual and guides of R and MCLUST, which includes the technical report [2] that is the base reference of this project and gives an overview of MCLUST with several examples and (ii) theoretical papers, surveys and books found in literature.

Contents [hide]

Introduction

[edit]

Clustering consists in identifying groups of entities that have characteristics in common and are cohesive and separated from each other. Interest in clustering has increased due to several applications in distinct knowledge areas. Highlighting the search for grouping of customers and products in massive datasets, document analysis in Web usage data, gene expression from microarrays and image analysis where clustering is used for segmentation.

The clustering methods can be grouped in classes. One widely used involves hierarchical clustering, which consider, initially, that each points represent one group and at each iteration it merged two groups chosen to optimize some criterion. A popular criteria, proposed by,[3] include the sum of within group sum of squares and is given by the shortest distance between the groups (single-link method).

Another typical class is based on iterative relocation, which data are moved from one group to another at each iteration. Also called as representative clustering due the use of a model, created to each cluster, that summarize the characteristics of the group elements. The most popular method in this class is the K-Means, proposed by,[4] which is based on iterative relocation with the sum of squares criterion.

In statistic and optimization problems is usual to maximize or minimize a function, and its variables in a specific space. As these optimization problems may assume several different types, each one with its own characteristics, many techniques have been developed to solve them. This techniques are very important in data mining and knowledge discovery area as it can be used as basis for most complex and powerful methods.

One of these techniques is the Maximum Likelihood and its main goal is to adjust a statistic model with a specific data set, estimating its unknown parameters so the function that can describe all the parameters in the dataset. In other words, the method will adjust some variables of a statistical model from a dataset or a known distribution, so the model can “describe” each data sample and estimate others.

It was realized that clustering can be based on probability models to cover the missing values. This provides insights into when the data should conform to the model and has led to the development of new clustering methods such as Expectation Maximization (EM) that is based on the principle of Maximum Likelihood of unobserved variables in finite mixture models.

Technique to be discussed

[edit]

The EM algorithm is an unsupervised clustering method, that is, don't require a training phase, based on mixture models. It follows an iterative approach, sub-optimal, which tries to find the parameters of the probability distribution that has the maximum likelihood of its attributes.

In general lines, the algorithm's input are the data set (x), the total number of clusters (M), the accepted error to converge (e) and the maximum number of iterations. For each iteration, first is executed the E-Step (E-xpectation), that estimates the probability of each point belongs to each cluster, followed by the M-step (M-aximization), that re-estimate the parameter vector of the probability distribution of each class. The algorithm finishes when the distribution parameters converges or reach the maximum number of iterations.

Algorithm

[edit]

Initialization

Each class j, of M classes (or clusters), is constituted by a parameter vector (θ), composed by the mean (μj) and by the covariance matrix (P_j), which represents the features of the Gaussian probability distribution (Normal) used to characterize the observed and unobserved entities of the data set x.

$$\theta(t) = \mu_j(t),\ P_j(t),\ j\ =\ 1..M$$

On the initial instant (t = 0) the implementation can generate randomly the initial values of mean (μj) and of covariance matrix (P_j). The EM algorithm aims to approximate the parameter vector (θ) of the real distribution. Another alternative offered by MCLUST is to initialize EM with the clusters obtained by a hierarquical clustering technique.

E-Step

This step is responsible to estimate the probability of each element belong to each cluster ($P(C_j|x_k)$). Each element is composed by an attribute vector (x_k). The relevance degree of the points of each cluster is given by the likelihood of each element attribute in comparison with the attributes of the other elements of cluster C_j.

$$P(C_j|x) = \frac{|\sum_j(t)|^{-\frac{1}{2}}\ exp^{n_j}\ P_j(t)}{\sum_{k=1}^{M}\ |\sum_j(t)|^{-\frac{1}{2}}\ exp^{n_j}\ P_k(t)}$$

M-Step

This step is responsible to estimate the parameters of the probability distribution of each class for the next step. First is computed the mean (μj) of classe j obtained through the mean of all points in function of the relevance degree of each point.

$$\mu_j(t+1) = \frac{\sum_{k=1}^{N}\ P(C_j|x_k)\ x_k}{\sum_{k=1}^{N}\ P(C_j|x_k)}$$

To compute the covariance matrix for the next iteration is applied the Bayes Theorem, which implies that $P(A|B) = P(B|A) * P(A)P(B)$, based on the conditional probabilities of the class occurrence.

$$\sum_j(t+1) = \frac{\sum_{k=1}^{N}\ P(C_j|x_k)\ (x_k - \mu_j(t))\ (x_k - \mu_j(t))}{\sum_{k=1}^{N}\ P(C_j|x_k)}$$

The probability of occurrence of each class is computed through the mean of probabilities (C_j) in function of the relevance degree of each point from the class.

$$P_j(t+1) = \frac{1}{N}\sum_{k=1}^{N} P(C_j|x_k)$$

The attributes represents the parameter vector θ that characterize the probability distribution of each class that will be used in the next algorithm iteration.

Convergence Test

After each iteration is performed a convergence test which verifies if the difference of the attributes vector of an iteration to the previous iteration is smaller than an acceptable erro tolerance, given by parameter. Some implementations uses the difference between the averages of class distribution as the convergence criterion.

```
if(||θ(t + 1) − θ(t)|| < ϱ)
   stop
else
   call E-Step
```

```
end
```

The algorithm has the property of, at each step, estimate a new attribute vector that has the maximum local likelihood, not necessarily the global, what reduces the its complexity. However, depending on the dispersion of the data and on its volume, the algorithm can stop due the maximum number of iterations defined.

Implementation [edit]

Packages [edit]

The expectation-maximization in algorithm in R,[5] proposed in,[6] will use the package mclust. This package contains crucial methods for the execution of the clustering algorithm, including functions for the E-step and M-step calculation. The package manual explains all of its functions, including simple examples. This manual can be found in.[2][6]

The mclust package also provides various models for EM and also hierarchical clustering(HC), which is defined by the covariance structures. These models are presented in Table 1 and are explained in detail in.[7]

Table 1: *Covariance matrix structures.*

identifier	Model	HC	EM	Distribution	Volume	Shape	Orientation
E		*	*	(univariate)	equal		
V		*	*	(univariate)	variable		
EII	λI	*	*	Spherical	equal	equal	NA
VII	$\lambda_k I$	*	*	Spherical	variable	equal	NA
EEI	λA		*	Diagonal	equal	equal	coordinate axes
VEI	$\lambda_k A$		*	Diagonal	variable	equal	coordinate axes
EVI	λA_k		*	Diagonal	equal	variable	coordinate axes
VVI	$\lambda_k A_k$		*	Diagonal	variable	variable	coordinate axes
EEE	λDAD^T	*	*	Ellipsoidal	equal	equal	equal
EEV	$\lambda D_k A D_k^T$		*	Ellipsoidal	equal	equal	variable
VEV	$\lambda_k D_k A D_k^T$		*	Ellipsoidal	variable	equal	variable
VVV	$\lambda_k D_k A_k D_k^T$	*	*	Ellipsoidal	variable	variable	variable

Executing the Algorithm [edit]

The function “em” can be used for the expectation-maximization method, as it implements the method for parameterized Gaussian Mixture Models (GMM), starting in the E-step. This function uses the following parameters:

- **model-name**: the name of the model used;
- **data**: all the collected data, which must be all numerical. If the data format is represent by a matrix, the rows will represent the samples (observations) and the columns the variables;
- **parameters**: model parameters, which can assume the following values: pro, mean, variance and Vinv, corresponding to the mixture proportion for the components of mixture, mean of each component, parameters variance and the estimate hypervolume of the data region, respectively.
- **other**: less relevant parameters which wont be described here. More details can be found in the

package manual.

After the execution, the function will return:

- **model-name**: the name of the model;
- **z**: a matrix whose the element in position [I,k] presents the conditional probability of the ith sample belongs to the kth mixture component;
- **parameters**: same as the input;
- **others**: other metrics which wont be discussed here. More details can be found in the package manual.

A simple example [edit]

In order to demonstrate how to use the R to execute the expectation-Maximization method, the following algorithm presents a simple example for a test dataset. This example can also be found in the package manual.

```
> modelName = ``EEE''
> data = iris[,-5]
> z = unmap(iris[,5])
> msEst <- mstep(modelName, data, z)
> names(msEst)
> modelName = msEst$modelName
> parameters = msEst$parameters
> em(modelName, data, parameters)
```

The first line executes the M-step so the parameters used in the em function can be generated. This function is called mstep and its inputs are model name, as “EEE”, the dataseta the iris dataset and finally, the z matrix, which contains the conditional probability of each class contains each data sample. This z matrix is generated by the unmap function.

After the M-step, the algorithm will show (line 2) the attributes of the object returned by this function. The third line will start the clustering process using some of the result of the M-step method as input.

The clustering method will return the parameters estimated in the process and the conditional probability of each sample falls in each class. These parameters include mean and variance, and this last one corresponds to the use of the mclustVariance method.

View [edit]

This section will present some examples of visualization available in MCLUST package. First will be showed a simple example of the overall process of clustering from the choice of the number of clusters, the initialization and the partitioning. Then it will be explained a didactical example using two random mixtures in comparison with two gaussian mixtures.

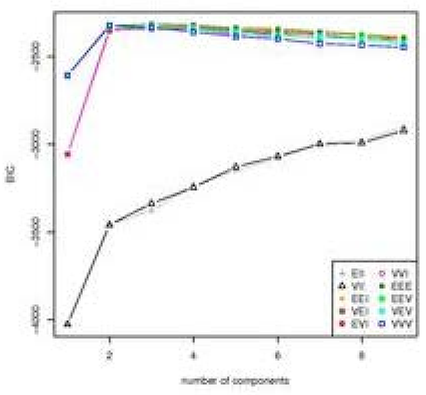
Basic Example

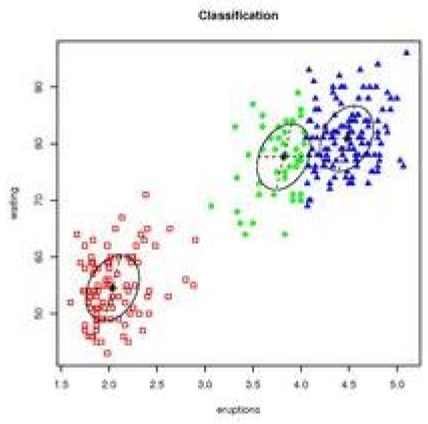
This is a simple example to show the features offered by MCLUST package. It is applied to the faithful dataset (included in R project). First the cluster analysis estimates the number of clusters that best represents this data set and also the covariance structure of the spread points. This is performed through the technique called Bayesian Information Criterion (BIC) that varies the number of cluster from 1 to 9. The BIC is the value of the maximized loglikelihood measured with a penalty for the number of parameters in the model. Then it's executed the hierarchical clustering technique (HC), which doesn't require a initialization phase. The output of the HC, that is, the cluster that each element belongs, is used to initialize the Expectation-Maximization technique (EM). After the execution of EM clustering the charts are showed below:

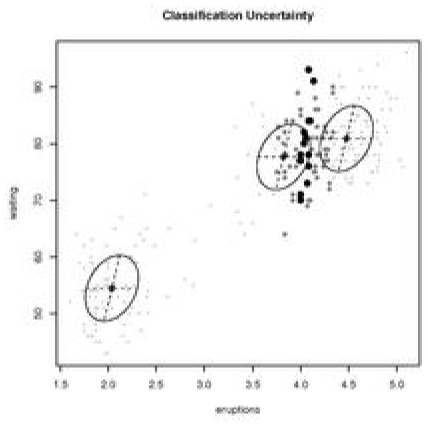
```
### basic_example.R ###
# usage: R --no-save < basic_example.R
```

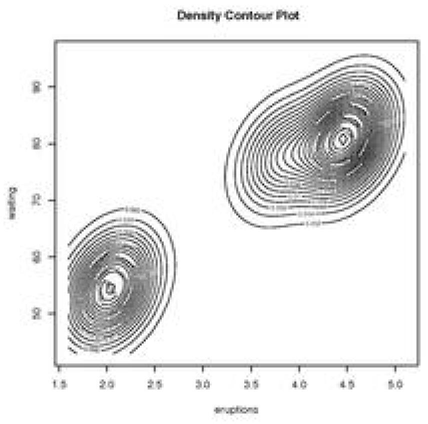
```
library(mclust)           # load mclust library
x = faithful[,1]          # get the first column of the faithful data set
y = faithful[,2]          # get the first column of the faithful data set
plot(x,y)                # plot the spread points before the clustering
model <- Mclust(faithful) # estimate the number of cluster (BIC), initialize
(HC) and clusterize (EM)
data = faithful           # get the data set
plot(model, faithful)     # plot the clustering results
```

File:Basic points.pdf









Didactical Example

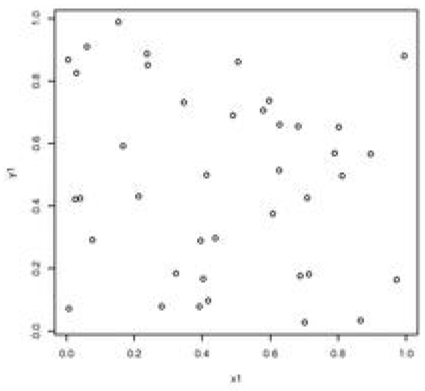
A didactical example is developed to show two distinct scenarios: (a) one that the model doesn't represents the data and (b) another that the data is conformed to the model. This example intends to show how EM behaves with noised and clean data sets.

a) Uniform random mixtures

A noised data set is generated through a uniform random function. The points spread are showed in a chart. Then the clustering analysis is executed with default parameters (i.e. varies cluster from 1 to 9). The clustering tool shows a warning with a message that the best model occurs at the min or max, in this case is the min that all points are grouped in a single cluster.

```
### random_example.R ###
# usage: R --no-save < random_example.R

library(mclust)                 # load mclust library
x1 = runif(20)                  # generate 20 random random numbers for x axis
(1st class)
y1 = runif(20)                  # generate 20 random random numbers for y axis
(1st class)
x2 = runif(20)                  # generate 20 random random numbers for x axis
(2nd class)
y2 = runif(20)                  # generate 20 random random numbers for y axis
(2nd class)
rx = range(x1,x2)               # get the axis x range
```

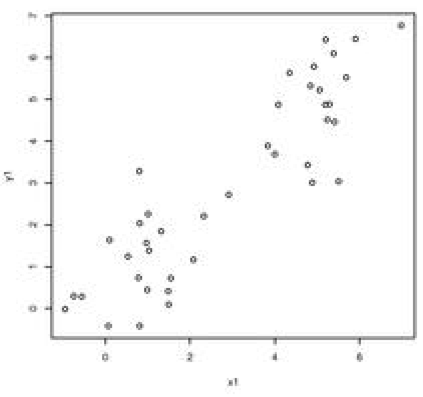
```
ry = range(y1,y2)                  # get the axis y range
plot(x1, y1, xlim=rx, ylim=ry)   # plot the first class points
points(x2, y2 )                    # plot the second class points
mix = matrix(nrow=40, ncol=2)    # create a dataframe matrix
mix[,1] = c(x1, x2)               # insert first class points into the matrix
mix[,2] = c(y1, y2)               # insert second class points into the matrix
mixclust = Mclust(mix)            # initialize EM with hierarchical clustering,
execute BIC and EM

# Warning messages:
# 1: In summary.mclustBIC(Bic, data, G = G, modelNames = modelNames) :
#    best model occurs at the min or max # of components considered
# 2: In Mclust(mix) : optimal number of clusters occurs at min choice
```

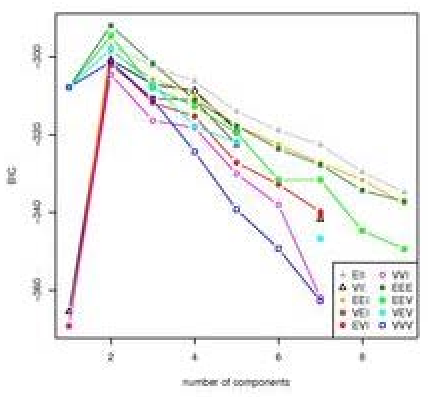


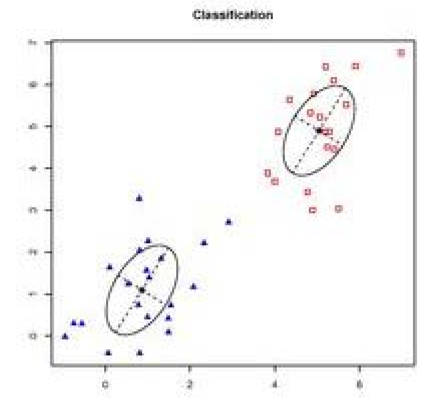
b) Two gaussian mixtures

This scenario is composed by two well separated data sets generated through a gaussian distribution function (Normal). The points are showed in the first chart. The EM clustering is applied and the results are also showed in the graphs below. As we can see, the EM clustering obtain two gaussian models that is in conformed to the data.

```
### gaussian_example.R ###
# usage: R --no-save < gaussian_example.R

library(mclust)                   # load mclust library
x1 = rnorm(n=20, mean=1, sd=1)   # get 20 normal distributed points for x axis
with mean=1 and std=1 (1st class)
y1 = rnorm(n=20, mean=1, sd=1)   # get 20 normal distributed points for x axis
with mean=1 and std=1 (2nd class)
x2 = rnorm(n=20, mean=5, sd=1)   # get 20 normal distributed points for x axis
with mean=5 and std=1 (1st class)
y2 = rnorm(n=20, mean=5, sd=1)   # get 20 normal distributed points for x axis
with mean=5 and std=1 (2nd class)
rx = range(x1,x2)                  # get the axis x range
ry = range(y1,y2)                  # get the axis y range
plot(x1, y1, xlim=rx, ylim=ry)   # plot the first class points
points(x2, y2)                    # plot the second class points
mix = matrix(nrow=40, ncol=2)    # create a dataframe matrix
mix[,1] = c(x1, x2)               # insert first class points into the matrix
mix[,2] = c(y1, y2)               # insert second class points into the matrix
mixclust = Mclust(mix)            # initialize EM with hierarchical clustering,
execute BIC and EM
plot(mixclust, data = mix)       # plot the two distinct clusters found
```







File:Gaussian density.pdf File:Gaussian uncertainty.pdf

Case Study

[edit]

Scenario

[edit]

The scenario to be analized is composed by a sample data set available in the MCLUST package named "wreath". A clustering analysis is performed with more details, applied to a scenario composed by 14 point groups, that exceeds the maximum number of clusters allowed by the default MCLUST parameters. The clustering technique is executed two times: (i) the first based on the default MCLUST, (ii) with customized parameters.

Input data

[edit]

The input of the case study is the data set wreath provided by the MCLUST package. This data set consists in a 14 point group showed on the next figure, which can be modeled with Spherical or Ellipsoide that take into account the orientation of the data due its rotation.

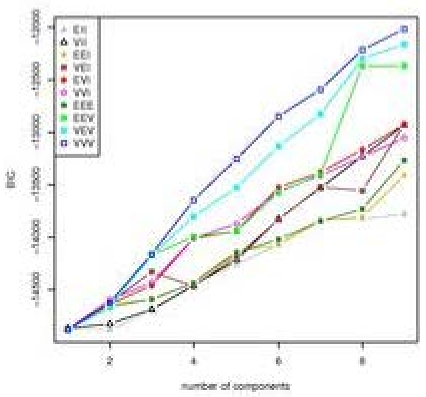
```
### case_input.R ###
# usage: R --no-save < case_default.R

plot(wreath[,1],wreath[,2])
```

Execution

[edit]

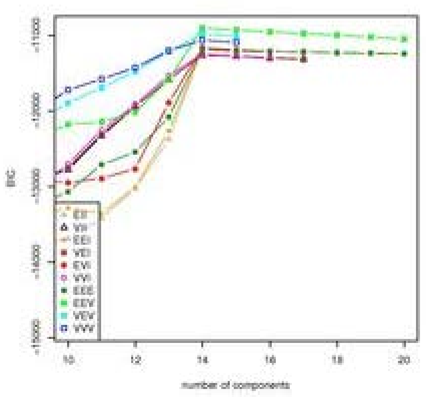
The clustering is executed two times. The first one is based on the default parameters given by the MCLUST tool, that varies the number of cluster from 1 to 9, which is smaller than the necessary to fit the case study data set. The estimation of the number of clusters is showed in a graphic that varies the number of clusters and compute the Bayesian Informatin Criterion (BIC) for each value. We can see that the BIC, using the default parameters, is divergent while is expected find a peak followed by a decrease that indicates the best number of clusters.

```
### case_default.R ###
# usage: R --no-save < case_default.R

library(mclust)

wreathBIC <- mclustBIC(wreath)
plot(wreathBIC, legendArgs = list(x = "topleft"))
```



Then the number of cluster is customized, modified to varies from 1 to 20 as showed below. The BIC technique will give the best number of clusters, in this case 14 clusters and the coefficient structure that have the properties of this data set, that is the EEV, which means that the clusters has similar shape, similar volumes but variable orientation. After executes the BIC method again we can see that 14 clusters, indicated by the peak on the graphic, is the number of cluster that present the maximum likelihood for this data.

```
### case_customized.R ###
# usage: R --no-save < case_customized.R

library(mclust)
wreathDefault <- mclustBIC(wreath)
wreathCustomize <- mclustBIC(wreath, G = 1:20, x = wreathDefault)
plot(wreathCustomize, G = 10:20, legendArgs = list(x = "bottomleft"))
summary(wreathCustomize, wreath)
```

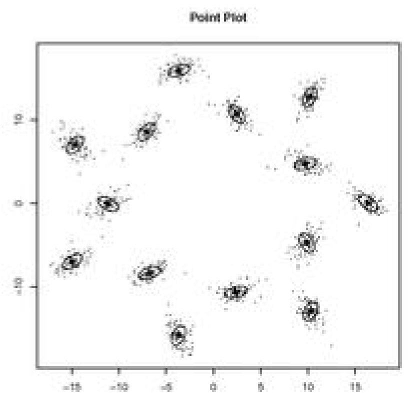


Output

[edit]

The output of this clustering is analysed is obtained using the method mclust2Dplot depicted in the next figure. It was used the density visualization. The clusters found are characterized by 14 models wich have the distribution of an ellipsoides, with different orientation, beein in conformed with the data. The method summary can be executed later analyse other aspects of the clustering result.

```
### case_output.R ###
# usage: R --no-save < case_customized.R

library(mclust)
data(wreath)
wreathBIC <- mclustBIC(wreath)
wreathBIC <- mclustBIC(wreath, G = 1:20, x = wreathBIC)
wreathModel <- summary(wreathBIC, data = wreath)
mclust2Dplot(data = wreath, what = "density", identify = TRUE, parameters =
wreathModel$parameters, z = wreathModel$z)
```



Analysis

[edit]

We can see that the mixture models created to represent the point are in conformation to the data set. On this case, the groups doesn't have an intersection between them, so all points were classified to the right group. The cluster orientation allows the method to find a better Ellipsoide to represent those points.

We conclude that the EM clustering technique, despite the dependence of the number of clusters and the initialization phase, is an efficient method that produces good results for several scenarios of data dispersion. The use of BIC to estimate the number of clusters and of the hierarchical clustering (HC) (which doesn't depend of the number of clusters) to initialize the clusters improves the quality of the results.

References

[edit]

[1] [2] [3] [4] [5] [6] [7] [8] [9]

1. ↑ *a b* Georey J. Mclachlan and Thriyambakam Krishnan. The EM Algorithm and Extensions. Wiley-Interscience, 1 edition, November 1996.
2. ↑ *a b c* C. Fraley and A. E. Raftery. MCLUST version 3 for R: Normal mixture modeling and model-based clustering. Technical Report 504, University of Washington, Department of Statistics, September 2006.
3. ↑ *a b* Ward, J. H., "Hierarquical Grouping to Optimize an Objective Function." Journal of the American Statistical Association. 58. 234-244. 1963.
4. ↑ *a b* MacQueen. J. Some Methods for Classification and Analysis of Multivariate Observations. in Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability. Vol 1. eds. L. M. L. Cam and J. Neyman, Berkeley, CA: University of California Press, pp. 281-297. 1967.
5. ↑ *a b* John M. Chambers. R language definition. http://cran.r-project.org/doc/manuals/R-

lang.html.

6. ↑ a b c Chris Fraley and Adrian E. Raftery. Bayesian regularization for normal mixture estimation and model-based clustering. J. Classif., 24(2):155-181, 2007.
7. ↑ a b C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis and density estimation. Journal of the American Statistical Association, 97:611-631, 2002.
8. ↑ Leslie Burkholder. Monty hall and bayes's theorem, 2000.
9. ↑ A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society. Series B (Methodological), 39(1):1{38, 1977.

Category: Data Mining Algorithms In R

What do you think of this page?

Please take a moment to rate this page below. Your feedback is valuable and helps us improve our website.

Reliability: (unsure) **Completeness:** (unsure) **Neutrality:** (unsure)
Presentation: (unsure) Submit