Package ‘RobustEM’

June 5, 2015

Type  Package
Title  Robust Mixture Modeling Fitted via Spatial-EM Algorithm for
       Model-Based Clustering and Outlier Detection
Version  1.0
Date    2015-06-04
Author  Aishat Aloba, Kai Yu, Xin Dang, Yixin Chen, and Henry Bart Jr.
Maintainer  Aishat Aloba <adetokaloba@gmail.com>
Description  The Spatial-EM is a new robust EM algorithm for the finite
            mixture learning procedures. The algorithm utilizes median-
            based location and rank-based scatter estimators to replace sample
            mean and sample covariance matrix in each M step, hence enhancing
            stability and robustness of the algorithm. To understand more about
            this algorithm, read the article
            "Yu, K., Dang, X., Bart Jr, H. and Chen, Y. (2015). Robust Model-
            based Learning via Spatial-EM Algorithm. IEEE Transactions on
            Knowledge and Data Engineering. 27(6), 1670-1682.
            doi:10.1109/TKDE.2014.2373355".
License  GPL-3
Depends  R (>= 2.15.0)
Imports  mvtnorm,e1071,ggplot2,ellipse, doParallel,grid, foreach
NeedsCompilation  yes
Repository  CRAN
Date/Publication  2015-06-05 01:11:35

R topics documented:

  cluster_em .......................................................... 2
  cluster_em_outlier ............................................... 3
  cluster_plot ....................................................... 4
  confusionMatrix ................................................... 6
  errorRate .......................................................... 6
  matchCluster ....................................................... 7
Description
This function estimates the mixture model by one of three EM algorithms and clusters observations to the component in which their density is the highest.

Usage
cluster_em(x, k, method=c("reg", "rcm", "kotz"), iter_max=100)

Arguments
x
This is a matrix or data frame of observations, where rows correspond to n observations and columns correspond to d variables. Categorical variables are not allowed.
k
This is an integer specifying the number of mixture components (clusters).
method
This specifies which of the algorithms is to be used. Presently there are three algorithms that can be used. These are reg(regular-EM), rcm(spatial-EM) and kotz. Regular EM uses sample mean and sample covariance in each M step and hence is sensitive to outliers. Spatial EM utilizes median-based location and rank-based scatter estimators, hence enhancing stability and robustness. It is robust to outliers and initial values. Robustness of Kotz EM is between that of other two.
iter_max
This is a parameter maxiter. It is the maximum number of iterations of the EM algorithm. The default value is 100. If the EM algorithm has not converged at this iteration, the estimates for the 100th iteration are returned and a warning message is presented.

Details
The cluster_em function provides mixture model estimation by one of EM algorithms used and clustering of data. Observations clustered to the component in which their density is highest.

Value
The function returns a list with the following components:

clusters
A factor consisting of cluster labels of observations.
mean
The mean of each component. If there is more than one component, it forms a matrix whose kth row is the mean of kth component of the mixture model.
sigma
A list of variance estimator for each component of the model.
taul
A vector of the mixing proportion for each of the components.
cluster_em_outlier

References


Examples

```r
x1<-matrix(rnorm(2*200),ncol=2)
x2<-matrix(rnorm(2*200,2,1),ncol=2)
x<-rbind(x1,x2)
k<-2
cluster_em(x,k,"rcm")
```

---

cluster_em_outlier **Clustering and Outlier Detection Algorithm**

Description

This function estimates parameters of mixture model, provides robust clustering and identifies outliers.

Usage

```r
cluster_em_outlier(x, k, method=c("reg","rcm","kotz"),eps = 0.01,iter_max=100)
```

Arguments

- **x**: This is a matrix or data frame of observations, where rows correspond to observations and columns correspond to variables. Categorical variables are not allowed.
- **k**: This is an integer specifying the number of mixture components (clusters).
- **method**: This specifies which of the algorithms is to be used. Presently there are three algorithms that can be used. These are reg(regular-EM), rcm(spatial-EM) and kotz. Regular EM uses sample mean and sample covariance in each M step and hence is sensitive to outliers. Spatial EM utilizes median-based location and rank-based scatter estimators, hence enhancing stability and robustness. It is robust to outliers and initial values. Robustness of Kotz EM is between that of other two.
- **eps**: This is a value used to determine the threshold for outlier identification. (1-eps)-quantile of Chi-square distribution with degree freedom d is used as the threshold, where d is the dimension.
- **iter_max**: This is the parameter maxiter. It is the maximum number of iterations of the EM algorithm. The default value is 100. If the EM algorithm has not converged at this iteration, the estimates for the 100th iteration are returned and a warning message is presented.
Details

The cluster_em_outlier function uses the mean and covariance of each component returned by one of the algorithms specified using the method, and computes squared Mahalanobis distances (MD) of each observation with respect to each component. If its lowest MD is greater than the threshold, then the observation is identified as an outlier and grouped into the outlier cluster, otherwise, it is clustered to the component in which its MD is the lowest.

It is essential to use robust Spatial EM for outlier detection, otherwise model estimation is distorted with presence of outliers and hence the outlier detection easily suffers masking and swamping effects (false negative and false positive errors).

Value

The function returns a list with the following components:

- **clusters**: A factor consisting of cluster labels of observations. The cluster label includes a "outlier" class.
- **mean**: The mean of each component. If there is more than one component, it forms a matrix whose kth row is the mean of kth component of the mixture model.
- **sigma**: A list of variance estimator for each component of the model.
- **taul**: A vector of the mixing proportion for each of the components. The last value of the value is a estimated proportion of outliers.

References


Examples

```r
## Not run:
x1 <- matrix(rnorm(2*200), ncol=2)
x2 <- matrix(rnorm(2*200, 2, 1), ncol=2)
x3 <- matrix(c(rnorm(20, 3, 1), rnorm(20, -3, 1)), ncol=2, byrow=FALSE)
x <- rbind(x1, x2, x3)
k<-2
cluster_em_outlier(x, k, "rcm")

## End(Not run)
```

cluster_plot

*Clustering Plot*

Description

A graph in 2D used to visualize the results of the EM clustering algorithm.
**Usage**

```r
cluster_plot(x, means, covariance, clusters)
```

**Arguments**

- `x`: This is the data matrix.
- `means`: This is a matrix with rows representing the means of the mixture model.
- `covariance`: This is a list where each element represents the covariance of the components of the distribution.
- `clusters`: This is a cluster label factor indicating each observation belonging to.

**Details**

This function shows the clustering results of the EM algorithm. Ellipses in the plot represent the estimated 95 probability density contours of each of the component. When the dimension is higher than 2, a pairs plot is produced. Presently, the function only works for dimensions being between 2 and 5.

**Value**

The value returned is the model-based clustering plot(s).

**See Also**

`cluster_em`

**Examples**

```r
## Not run:
x1 <- matrix(rnorm(2*200), ncol=2)
x2 <- matrix(rnorm(2*200,2,1),ncol=2)
x <- rbind(x1,x2)
k <- 2
c1 <- cluster_em(x,k,"rcm")
mean <- c1$mean
covariance <- c1$sigma
cluster_plot(x,mean,covariance)
## End(Not run)
```
**confusionMatrix**  
*Confusion Matrix*

**Description**

Given the actual and predicted labels, this function computes the confusion matrix.

**Usage**

```r
confusionMatrix(actual, predicted)
```

**Arguments**

- `actual`: This is a list of values corresponding to the true class of the data.
- `predicted`: This is a list of values corresponding to the predicted class of the data using the EM algorithm.

**Value**

- `t`: This is the table representing the confusion matrix.

**Examples**

```r
## Not run:
confusionMatrix(c("A", "B", "A", "B", "B", "A"), c(1, 2, 2, 1, 2, 1))

## End(Not run)
```

---

**errorRate**  
*This function calculates the error rates for the different EM algorithms*

**Description**

This function uses different criteria to measure the accuracy of predicted labels with respect to the true labels.

**Usage**

```r
terrorRate(actual, predicted, beta=1)
```

**Arguments**

- `actual`: This is a vector that represents the actual true class labels.
- `predicted`: This is the predicted cluster returned by the EM algorithm.
- `beta`: This is a parameter of beta used for the F-measure. The default is 1 which correspondes to the balanced F-measure.
**matchCluster**

**Value**

- **FPR** Represents the computed False Positive Rate
- **FNR** Represents the computed False Negative Rate
- **RI** Represents the computed Rand Index
- **F** Represents the computed F-Measure

**Warning**

This function is dependent on the true and predicted clusters having the same levels.

**See Also**

matchCluster

**Examples**

```r
## Not run:
true <- c("a","b","a","a","a")
predicted <- c("a","b","b","a","a")
errorRate(true, predicted)

## End(Not run)
```

**Description**

This function returns the best correspondence between two label vectors.

**Usage**

`matchCluster(actual, predicted)`

**Arguments**

- **actual** This is a vector that represents the actual cluster of the data
- **predicted** This is the predicted cluster returned by the EM algorithm

**Value**

- **component** This returns the corresponding cluster label with respect to the true cluster label provided

**Warning**

To use this function, you must have a vector returned by one of the EM algorithms provided
normalizedMI

See Also

**cluster_em**

Examples

```r
## Not run:
actual <- c("a", "b", "a", "a", "a")
predicted <- c(1, 2, 2, 1, 1)
purity(actual, predicted)

## End(Not run)
```

---

**normalizedMI** | *Computes the normalized mutual information*

### Description

This uses the normalized mutual information to evaluate how well the EM algorithm performs with respect to the data provided.

### Usage

```r
normalizedMI(trueLabel, predictedLabel)
```

### Arguments

- `trueLabel` This is a vector that represents the actual cluster labels of the data
- `predictedLabel` This is the predicted cluster labels returned by the EM algorithm

### Value

- `normalizedMI` This value represents the efficiency of the algorithm. The higher the value, the greater the efficiency of the algorithm

### Warning

This function is dependent on the true and predicted clusters having the same levels.

### References


### See Also

**matchCluster**
**Examples**

```r
## Not run:
true<-c("a","b","a","a","a")
predicted<-c("a","b","b","a","a")
normalizedMI(true,predicted)

## End(Not run)
```

**Description**

The function selects the number of components in the distribution using a variation of the Bayesian Information Criterion (BIC). The trimmed BIC uses a trimmed likelihood and a complexity penalty term to optimally determine the number of mixture components. It uses a range of values as number of components and returns the value that gives the maximum trimmed BIC.

**Usage**

```r
trimmed_bic(data, alpha, end, method=c("reg","rcm","kotz"),iter_max=100)
```

**Arguments**

- **data**
  This is a matrix or data frame of observations, where rows correspond to n observations and columns correspond to d variables. Categorical variables are not allowed.

- **alpha**
  This is the trimming percentage in the calculation of the trimmed BIC. The alpha value ranges from 0 to 0.5. alpha = 0 corresponds the conventional BIC.

- **end**
  This is an integer value that represents the maximum number of components in the mixture models considered. The minimum number components is always set to be 2.

- **method**
  This specifies which of the algorithms is to be used. Presently there are three algorithms that can be used. These are reg(regular-EM),rcm(spatial-EM), kotz(kotz-EM).

- **iter_max**
  This is a parameter maxiter. It is the maximum number of iterations of the EM algorithm. The default value is 100. If the EM algorithm has not converged at this iteration, the parameters for the 100th iteration is returned.

**Value**

- **bic**
  A list containing the BIC computed in the range.

- **k**
  The optimal number of components selected.

**References**

Examples

```r
## Not run:
x1 <- matrix(rnorm(2*200),nrow=2)
x2 <- matrix(rnorm(2*200,2,1),nrow=2)
data <- rbind(x1,x2)
epsilon <- 0.5
end <- 3
iter_max <- 50
trimmed_bic(data,epsilon,end,"rcm",50)

## End(Not run)
```
Index

cluster_em, 2, 5, 8
cluster_em_outlier, 3
cluster_plot, 4
confusionMatrix, 6

errorRate, 6

matchCluster, 7, 7, 8

normalizedMI, 8

trimmed_bic, 9