Package ‘mcclust’

February 15, 2013

Type Package
Title Process an MCMC Sample of Clusterings
Version 1.0
Date 2009-05-22
Author Arno Fritsch
Depends R (>= 2.10), lpSolve
Maintainer Arno Fritsch <arno.fritsch@tu-dortmund.de>
Description Implements methods for processing a sample of (hard)
clustering, e.g. the MCMC output of a Bayesian clustering
model. Among them are methods that find a single best
clustering to represent the sample, which are based on the
posterior similarity matrix or a relabelling algorithm.
License GPL (>= 2)
LazyLoad yes
Repository CRAN
Date/Publication 2012-07-23 10:35:32
NeedsCompilation yes

R topics documented:

  mcclust-package .................................. 2
  arandi ............................................ 3
  cls.draw1.5 ...................................... 4
  cls.draw2 ....................................... 5
  cltoSim .......................................... 5
  comp.psm ........................................ 6
  maxpear ......................................... 7
mcclust-package

Process MCMC Sample of Clusterings.

Description

Implements methods for processing a sample of (hard) clusterings, e.g. the MCMC output of a Bayesian clustering model. Among them are methods that find a single best clustering to represent the sample, which are based on the posterior similarity matrix or a relabelling algorithm.

Details

- Package: mcclust
- Type: Package
- Version: 1.0
- Date: 2009-03-12
- License: GPL (>= 2)
- LazyLoad: yes

Most important functions:

- comp.psm for computing posterior similarity matrix (PSM). Based on the PSM maxpear and minbinder provide several optimization methods to find a clustering with maximal posterior expected adjusted Rand index with the true clustering or one that minimizes the posterior expectation of a loss function by Binder (1978). minbinder provides the optimization algorithm of Lau and Green.


- arandi and vi.dist compute distance functions for clusterings, the (adjusted) Rand index and the entropy-based variation of information distance.

Author(s)

Arno Fritsch

Maintainer: Arno Fritsch <arno.fritsch@tu-dortmund.de>
References


Examples

data(cls.draw2)
# sample of 500 clusterings from a Bayesian cluster model
tru.class <- rep(1:8,each=50)
# the true grouping of the observations
psm2 <- comp.psm(cls.draw2)
# posterior similarity matrix

# optimize criteria based on PSM
mbind2 <- minbinder(psm2)
mpear2 <- maxpear(psm2)

# Relabelling
k <- apply(cls.draw2,1, function(cl) length(table(cl)))
max.k <- as.numeric(names(table(k))[which.max(table(k))])
relab2 <- relabel(cls.draw2[k==max.k,])

# compare clusterings found by different methods with true grouping
arandi(mpear2$cl, tru.class)
arandi(mbind2$cl, tru.class)
arandi(relab2$cl, tru.class)

---

**arandi**  
*(Adjusted) Rand Index for Clusterings*

**Description**

Computes the adjusted or unadjusted Rand index between two clusterings/partitions of the same objects.

**Usage**

```
arandi(cl1, cl2, adjust = TRUE)
```
Arguments

- `cl1, cl2`: vectors of cluster memberships (need to have the same lengths).
- `adjust`: logical. Should index be adjusted? Defaults to TRUE.

Details

The Rand index is based on how often the two clusterings agree in the treatment of pairs of observations, where agreement means that two observations are in/not in the same cluster in both clusterings.

The adjusted Rand index adjusts for the expected number of chance agreements.

Formulas of Hubert and Arabie (1985) are used for the computation.

Author(s)

Arno Fritsch, <arno.fritsch@tu-dortmund.de>

References


See Also

- `vi.dist`

Examples

```r
cl1 <- sample(1:3, 10, replace=TRUE)
cl2 <- c(cl1[1:5], sample(1:3, 5, replace=TRUE))
arandi(cl1, cl2)
arandi(cl1, cl2, adjust=FALSE)
```

---

**cls.draw1.5**

*Sample of Clusterings from Posterior Distribution of Bayesian Cluster Model*

Description

Output of a Dirichlet process mixture model with normal components fitted to the data set `Ysim1.5`. True clusters are given by `rep(1:8, each = 50)`.

Usage

- `data(cls.draw1.5)`
Format
matrix with 500 rows and 400 columns. Each row contains a clustering of the 400 observations.

Source

cls.draw2
Sample of Clusterings from Posterior Distribution of Bayesian Cluster Model

Description
Output of a Dirichlet process mixture model with normal components fitted to the data set Ysim2. True clusters are given by rep(1:8, each = 50).

Usage
data(cls.draw2)

Format
matrix with 500 rows and 400 columns. Each row contains a clustering of the 400 observations.

Source

cltoSim
Compute Similarity Matrix for a Clustering and vice versa

Description
A similarity matrix is a symmetric matrix whose entry \([i, j]\) is 1 if observation \(i\) and \(j\) are in the same cluster and 0 otherwise.

Usage
cltoSim(cl)
SimtoCl(Sim)
Arguments

- cl: vector of cluster memberships
- Sim: similarity matrix

Warning

`Simtocl` does not check whether `Sim` is a valid similarity matrix, e.g. that `Sim[i,j]` == 1 if `Sim[i,k]` == 1 and `Sim[j,k]` == 1.

Author(s)

Arno Fritsch, <arno.fritsch@tu-dortmund.de>

See Also

- `comp.psm` for an average similarity matrix.

Examples

```r
cl <- c(3,3,1,2,2)
(Sim <- cltoSim(cl))
Simtocl(Sim)

# not a valid similarity matrix
(Sim2 <- matrix(c(1,0,0,1,1,1,1,1,1), ncol=3))
Simtocl(Sim2) # no warning
```

Description

For a sample of clusterings of the same objects the proportion of clusterings in which observation \(i\) and \(j\) are together in a cluster is computed and a matrix containing all proportions is given out.

Usage

```r
comp.psm(cls)
```

Arguments

- cls: a matrix in which every row corresponds to a clustering of the `nrow(cls)` objects

Details

In Bayesian cluster analysis the posterior similarity matrix is a matrix whose entry \([i,j]\) contains the posterior probability that observation \(i\) and \(j\) are together in a cluster. It is estimated by the proportion of a posteriori clusterings in which \(i\) and \(j\) cluster together.
maxpear

Value
a symmetric ncol(cls)*ncol(cls) matrix

Author(s)
Arno Fritsch, <arno.fritsch@tu-dortmund.de>

See Also
c1toSim

Examples
(cl <- rbind(c(1,1,2,2),c(1,1,2,2),c(1,2,2,2),c(2,2,1,1)))
comp.psm(cl)

maxpear
Maximize/Compute Posterior Expected Adjusted Rand Index

Description
Based on a posterior similarity matrix of a sample of clusterings maxpear finds the clustering that maximizes the posterior expected Rand adjusted index (PEAR) with the true clustering, while pear computes PEAR for several provided clusterings.

Usage
maxpear(psm, cls.draw = NULL, method = c("avg", "comp", "draws", "all"), max.k = NULL)
ppear(cls, psm)

Arguments

psm
a posterior similarity matrix, usually obtained from a call to comp.psm.

cls, cls.draw
a matrix in which every row corresponds to a clustering of the ncol(cls) objects. cls.draw refers to the clusterings that have been used to compute psm, cls.draw has to be provided if method="draw" or "all".

method
the maximization method used. Should be one of "avg", "comp", "draws" or "all". The default is "avg".

max.k
integer, if method="avg" or "comp" the maximum number of clusters up to which the hierarchical clustering is cut. Defaults to ceiling(nrow(psm)/8).
Details

For method="avg" and "comp" 1- psm is used as a distance matrix for hierarchical clustering with average/complete linkage. The hierarchical clustering is cut for the cluster sizes 1:max.k and PEAR computed for these clusterings.
Method "draws" simply computes PEAR for each row of cls.draw and takes the maximum.
If method="all" all maximization methods are applied.

Value

cl clustering with maximal value of PEAR. If method="all" a matrix containing the clustering with the highest value of PEAR over all methods in the first row and the clusterings of the individual methods in the next rows.
value value of PEAR. A vector corresponding to the rows of cl if method="all".
method the maximization method used.

Author(s)

Arno Fritsch, <arno.fritsch@tu-dortmund.de>

References


See Also

comp.psm for computing posterior similarity matrix, minbinder, medv, relabel for other possibilities for processing a sample of clusterings.

Examples

data(cls.draw1.5)
# sample of 500 clusterings from a Bayesian cluster model
tru.class <- rep(1:8,each=50)
# the true grouping of the observations
psm1.5 <- comp.psm(cls.draw1.5)
mpear1.5 <- maxpear(psm1.5)
table(mpear1.5$cl, tru.class)

# Does hierarchical clustering with Ward's method lead
# to a better value of PEAR?
hclust.ward <- hclust(as.dist(1-psm1.5), method="ward")
cls.ward <- t(apply(matrix(1:20),1, function(k) cutree(hclust.ward,k=k)))
ward1.5 <- pear(cls.ward, psm1.5)
max(ward1.5) > mpear1.5$value
Description

Based on a posterior similarity matrix of a sample of clusterings medv obtains a clustering by using 1-psm as distance matrix for hierarchical clustering with complete linkage. The dendrogram is cut at a value h close to 1.

Usage

medv(psm, h=0.99)

Arguments

psm a posterior similarity matrix, usually obtained from a call to comp.psm.

h The height at which the dendrogram is cut.

Value

vector of cluster memberships.

Author(s)

Arno Fritsch, <arno.fritsch@tu-dortmund.de>

References


See Also

comp.psm for computing posterior similarity matrix, maxpear, minbinder, relabel for other possibilities for processing a sample of clusterings.

Examples

data(cls.draw1.5)
# sample of 500 clusterings from a Bayesian cluster model
tru.class <- rep(1:8,each=50)
# the true grouping of the observations
psm1.5 <- comp.psm(cls.draw1.5)
medv1.5 <- medv(psm1.5)
table(medv1.5, tru.class)
minbinder  

Minimize/Compute Posterior Expectation of Binders Loss Function

Description

Based on a posterior similarity matrix of a sample of clusterings minbinder finds the clustering that minimizes the posterior expectation of Binders loss function, while binder computes the posterior expected loss for several provided clusterings.

Usage

minbinder(psm, cls.draw = NULL, method = c("avg", "comp", "draws", "laugreen","all"), max.k = NULL, include.lg = FALSE, start.cl = NULL, tol = 0.001)

binder(cls,psm)

laugreen(psm, start.cl, tol=0.001)

Arguments

- **psm**: a posterior similarity matrix, usually obtained from a call to comp.psm.
- **cls, cls.draw**: a matrix in which every row corresponds to a clustering of the ncol(cls) objects. cls.draw refers to the clusterings that have been used to compute psm, cls.draw has to be provided if method="draw" or "all".
- **method**: the maximization method used. Should be one of "avg", "comp", "draws", "laugreen" or "all". The default is "avg".
- **max.k**: integer, if method="avg" or "comp" the maximum number of clusters up to which the hierarchical clustering is cut. Defaults to ceiling(nrow(psm)/4).
- **include.lg**: logical, should method "laugreen" be included when method="all"? Defaults to FALSE.
- **start.cl**: clustering used as starting point for method="laugreen". If NULL start.cl = 1:nrow(psm) is used.
- **tol**: convergence tolerance for method="laugreen".

Details

The posterior expected loss is the sum of the absolute differences of the indicator function of observation $i$ and $j$ clustering together and the posterior probability that they are in one cluster.

For method="avg" and "comp" 1-psm is used as a distance matrix for hierarchical clustering with average/complete linkage. The hierarchical clustering is cut for the cluster sizes 1:max.k and the posterior expected loss is computed for these clusterings.

Method "draws" simply computes the posterior expected loss for each row of cls.draw and takes the minimum.

Method "laugreen" implements the algorithm of Lau and Green (2007), which is based on binary
integer programming. Since the method can take some time to converge it is only used if explicitly
demanded with method="laugreen" or method="all" and include.lg=TRUE. If method="all"
al minimization methods except "laugreen" are applied.

Value

cl clustering with minimal value of expected loss. If method="all" a matrix contain-
ing the clustering with the smallest value of the expected loss over all meth-
ods in the first row and the clusterings of the individual methods in the next
rows.

dec value value of posterior expected loss. A vector corresponding to the rows of cl if
method="all".

method the maximization method used.
iter.lg if method="laugreen" the number of iterations the method needed to converge.

Author(s)

Arno Fritsch, <arno.fritsch@tu-dortmund.de>

References

similarity matrix, Bayesian Analysis, accepted.
 Lau, J.W. and Green, P.J. (2007) Bayesian model based clustering procedures, Journal of Compu-

See Also

comp.psm for computing posterior similarity matrix, maxpear, medv, relabel for other possibilities
for processing a sample of clusterings. lp for the linear programming.

Examples

data(cls.draw2)
# sample of 500 clusterings from a Bayesian cluster model
tru.class <- rep(1:8,each=50)
# the true grouping of the observations
psm2 <- comp.psm(cls.draw2)
mbind2 <- minbinder(psm2)
table(mbind2$cl, tru.class)

# Does hierachical clustering with Ward's method lead
# to a lower value of Binders loss?
# hclust.ward <- hclust(as.dist(1-psm2), method="ward")
cls.ward <- t(apply(matrix(1:20),1, function(k) cutree(hclust.ward,k=k)))
ward2 <- binder(cls.ward, psm2)
min(ward2) < mbind2$value
# Method laugreen is applied to 4 randomly selected observations
ind <- sample(1:400, 4)
mbind.lg <- minbinder(psm2[ind, ind], cl1s.draw2[,ind], method="all",
                      include.lg=TRUE)
mbind.lg$value

norm.label

## Norm Labelling of a Clustering

### Description
Cluster labels of a clusterings are replaced by 1:length(table(cl)).

### Usage
norm.label(cl)

### Arguments
- **cl**
  - vector of cluster memberships

### Value
the clustering with normed labels.

### Author(s)
Arno Fritsch, <arno.fritsch@tu-dortmund.de>

### See Also
relabel for labelling a sample of clusterings the same way

### Examples
```r
(cl <- sample(c(13,12,34), 13, replace=TRUE))
norm.label(cl)

(cl <- sample(c("a","b","f31"), 13, replace=TRUE))
norm.label(cl)
```
**Stephens’ Relabelling Algorithm for Clusterings**

**Description**

For a sample of clusterings in which corresponding clusters have different labels the algorithm attempts to bring the clusterings to a unique labelling.

**Usage**

```
relabel(cls, print.loss = TRUE)
```

**Arguments**

- `cls`: a matrix in which every row corresponds to a clustering of the `ncol(cls)` objects.
- `print.loss`: logical, should current value of loss function be printed after each iteration? Defaults to `TRUE`.

**Details**

The algorithm minimizes the loss function

\[
\sum_{m=1}^{M} \sum_{i=1}^{n} \sum_{j=1}^{K} - \log \hat{p}_{ij} \cdot I_{\{z_i^{(m)} = j\}}
\]

over the `M` clusterings, `n` observations and `K` clusters, where `\hat{p}_{ij}` is the estimated probability that observation `i` belongs to cluster `j` and `z_i^{(m)}` indicates to which cluster observation `i` belongs in clustering `m`. `I_{\{\}}` is an indicator function.

Minimization is achieved by iterating the estimation of `\hat{p}_{ij}` over all clusterings and the minimization of the loss function in each clustering by permuting the cluster labels. The latter is done by linear programming.

**Value**

- `cls`: the input `cls` with unified labelling.
- `P`: an `n x K` matrix, where entry `[i, j]` contains the estimated probability that observation `i` belongs to cluster `j`.
- `loss.val`: value of the loss function.
- `cl`: vector of cluster memberships that have the highest probabilities `\hat{p}_{ij}`.

**Warning**

The algorithm assumes that the number of clusters `K` is fixed. If this is not the case `K` is taken to be the most common number of clusters. Clusterings with other numbers of clusters are discarded and a warning is issued.
Note

The implementation is a variant of the algorithm of Stephens which is originally applied to draws of parameters for each observation, not to cluster labels.

Author(s)

Arno Fritsch, <arno.fritsch@tu-dortmund.de>

References


See Also

lp.transport for the linear programming, maxpear, minbinder, medv for other possibilities of processing a sample of clusterings.

Examples

(c1 <- rbind(c(1,1,2,2),c(1,1,2,2),c(1,2,2,2),c(2,2,1,1)))
# group 2 in clustering 4 corresponds to group 1 in clustering 1-3.
c1.relab <- relabel(c1)
c1.relab$rels

---

**vi.dist**

*Variation of Information Distance for Clusterings*

Description

Computes the ’variation of information’ distance of Meila (2007) between two clusterings/partitions of the same objects.

Usage

```
vi.dist(c11, c12, parts = FALSE, base = 2)
```

Arguments

- `c11,c12`: vectors of cluster memberships (need to have the same lengths).
- `parts`: logical; should the two conditional entropies also be returned?
- `base`: base of logarithm used for computation of entropy and mutual information.

Details

The variation of information distance is the sum of the two conditional entropies of one clustering given the other. For details see Meila (2007).
Value

The VI distance. If parts=TRUE the two conditional entropies are appended.

Author(s)

Arno Fritsch, <arno.fritsch@tu-dortmund.de>

References


See Also

arandi

Examples

```r
cl1 <- sample(1:3,1/zero.noslash,replace=TRUE)
c12 <- c(cl1[1:5], sample(1:3,5,replace=TRUE))
vi.dist(cl1,c12)
vi.dist(cl1,c12, parts=TRUE)
```

---

Ysim1.5

### Simulated 3-dimensional Normal Data Containing 8 Clusters

Description

Cluster means are given by the 8 possible values of $(\pm 1.5, \pm 1.5, \pm 1.5)$ to which standard normal noise was added. True clusters are given by `rep(1:8,each =50)`.

Usage

data(Ysim1.5)

Format

matrix with 400 rows and 3 columns.

Source

Simulated by

$1.5 \times \text{matrix(c(rep(c(1,1,1),50), rep(c(1,1,-1),50), rep(c(1,-1,1),50), rep(c(-1,1,1),50), rep(c(-1,1,-1),50), rep(c(-1,-1,1),50), rep(c(-1,-1,-1),50), rep(c(1,-1,-1),50), rep(c(1,-1,1),50))))}$

References

Description

Cluster means are given by the 8 possible values of \((\pm 2, \pm 2, \pm 2)\) to which standard normal noise was added. True clusters are given by `rep(1:8, each = 50)`.

Usage

data(Ysim2)

Format

matrix with 400 rows and 3 columns.

Source

Simulated by

\[
2 \times \text{matrix}(\text{rep}(c(1,1), 50), \text{rep}(c(1,1,-1), 50), \text{rep}(c(-1,1,1), 50), \text{rep}(c(1,1), 50), \text{rep}(c(1,1), 50))
\]

References

Index

*Topic **cluster**
arandi, 3
cltreSim, 5
comp.psm, 6
maxpear, 7
medv, 9
minbinder, 10
norm.label, 12
relabel, 13
vi.dist, 14

*Topic **datasets**
cls.draw1.5, 4
cls.draw2, 5
Ysim1.5, 15
Ysim2, 16

*Topic **optimize**
maxpear, 7
minbinder, 10

*Topic **package**
mcclust-package, 2

arandi, 3, 15
binder (minbinder), 10
cls.draw1.5, 4
cls.draw2, 5
cltreSim, 5, 7
comp.psm, 6, 6, 8, 9, 11

laugreen (minbinder), 10
lp, 11
lp.transport, 14

maxpear, 7, 9, 11, 14
mcclust (mcclust-package), 2
mcclust-package, 2
medv, 8, 9, 11, 14
minbinder, 8, 9, 10, 14

norm.label, 12