

**Original citation:**

Wade, Sara (2015) Manual for mclust.ext R package. [Online] (Unpublished)

**Permanent WRAP url:**

<http://wrap.warwick.ac.uk/71937>

**Copyright and reuse:**

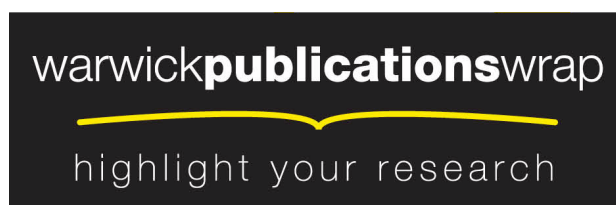
The Warwick Research Archive Portal (WRAP) makes this work of researchers of the University of Warwick available open access under the following conditions. Copyright © and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable the material made available in WRAP has been checked for eligibility before being made available.

Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

**A note on versions:**

The version presented in WRAP is the published version or, version of record, and may be cited as it appears here.

For more information, please contact the WRAP Team at: [publications@warwick.ac.uk](mailto:publications@warwick.ac.uk)



<http://wrap.warwick.ac.uk/>

# Package ‘mcclust.ext’

May 15, 2015

**Type** Package

**Title** Point estimation and credible balls for Bayesian cluster analysis

**Version** 1.0

**Date** 2015-03-24

**Author** Sara Wade

**Maintainer** Sara Wade <sara.wade@eng.cam.ac.uk>

**Description** This is an extension of the mcclust package. It provides post-processing tools for MCMC samples of partitions to summarize the posterior in Bayesian clustering models. Functions for point estimation are provided, giving a single representative clustering of the posterior. And, to characterize uncertainty in the point estimate, credible balls can be computed.

**Depends** R (>= 2.10), mcclust

**License** GPL (>= 2)

## R topics documented:

mcclust.ext-package . . . . .	1
credibleball . . . . .	3
ex1.data . . . . .	5
ex1.draw . . . . .	6
ex2.data . . . . .	7
ex2.draw . . . . .	8
galaxy.draw . . . . .	9
galaxy.fit . . . . .	10
galaxy.pred . . . . .	10
greedy . . . . .	11
minbinder.ext . . . . .	13
minVI . . . . .	15
plotpsm . . . . .	18
summary.c.estimate . . . . .	19
VI . . . . .	21

---

mcclust.ext-package    *Point estimation and credible balls for Bayesian cluster analysis*

---

## Description

This is an extension of mcclust package. It provides post-processing tools for MCMC samples of partitions to summarize the posterior in Bayesian clustering models. Functions for point estimation are provided, giving a single representative clustering of the posterior. And, to characterize uncertainty in the point estimate, credible balls can be computed.

## Details

Package: mcclust.ext  
Type: Package  
Version: 1.0  
Date: 2015-03-24  
License: GPL (>= 2)

Most important functions:

The functions `minVI` and `minbinder.ext` find a point estimate of the clustering by minimizing the posterior expected Variation of Information and Binder's loss, respectively. The function `minbinder.ext` extends `minbinder` by providing a greedy search optimization method to find the optimal clustering. The function `minVI` provides several optimization methods to find the optimal clustering. For computational reasons, the lower bound to the posterior expected Variation of Information from Jensen's inequality is minimized.

The function `credibleball` computes a credible ball around the clustering estimate to characterize uncertainty. It returns the upper vertical, lower vertical, and horizontal bounds to describe the credible ball.

The function `plotpsm` produces a heat map of the posterior similarity matrix.

## Author(s)

Sara Wade

Maintainer: Sara Wade <sara.wade@eng.cam.ac.uk>

## References

- Binder, D.A. (1978) Bayesian cluster analysis, *Biometrika* **65**, 31–38.
- Fritsch, A. and Ickstadt, K. (2009) An improved criterion for clustering based on the posterior similarity matrix, *Bayesian Analysis*, **4**, 367–391.
- Lau, J.W. and Green, P.J. (2007) Comparing clusters—an information based distance procedures, *Journal of Computational and Graphical Statistics* **16**, 526–558.

Meila, M. (2007) Bayesian model based clustering procedures, *Journal of Multivariate Analysis* **98**, 873–895.

Wade, S. and Ghahramani, Z. (2015) Bayesian cluster analysis: Point estimation and credible balls. Submitted. arXiv:1505.03339.

## See Also

[mclust](#)

## Examples

```
data(galaxy.fit)
x=data.frame(x=galaxy.fit$x)
data(galaxy.pred)
data(galaxy.draw)

# Find representative partition of posterior
# Variation of Information (minimizes lower bound to VI)
psm=comp.psm(galaxy.draw)
galaxy.VI=minVI(psm,galaxy.draw,method="all"),include.greedy=TRUE)
summary(galaxy.VI)
plot(galaxy.VI,data=x,dx=galaxy.fit$fx,xgrid=galaxy.pred$x,dxgrid=galaxy.pred$fx)
# Compute Variation of Information
VI(galaxy.VI$c1,galaxy.draw)
# Binder
galaxy.B=minbinder.ext(psm,galaxy.draw,method="all"),include.greedy=TRUE)
summary(galaxy.B)
plot(galaxy.B,data=x,dx=galaxy.fit$fx,xgrid=galaxy.pred$x,dxgrid=galaxy.pred$fx)

# Uncertainty in partition estimate
galaxy.cb=credibleball(galaxy.VI$c1[1,],galaxy.draw)
summary(galaxy.cb)
plot(galaxy.cb,data=x,dx=galaxy.fit$fx,xgrid=galaxy.pred$x,dxgrid=galaxy.pred$fx)

# Compare with uncertainty in heat map of posterior similarity matrix
plotpsm(psm)
```

---

credibleball

*Compute a Bayesian credible ball around a clustering estimate*

---

## Description

Computes a Bayesian credible ball around a clustering estimate to characterize uncertainty in the posterior, i.e. MCMC samples of clusterings.

**Usage**

```
credibleball(c.star, cls.draw, c.dist = c("VI", "Binder"), alpha = 0.05)

## S3 method for class 'credibleball'
summary(object, ...)
## S3 method for class 'credibleball'
plot(x, data=NULL, dx=NULL, xgrid=NULL, dxgrid=NULL, ...)
```

**Arguments**

<code>c.star</code>	vector, a clustering estimate of the <code>length(c.star)</code> data points.
<code>cls.draw</code>	a matrix of the MCMC samples of clusterings of the <code>ncol(cls.draw)</code> data points.
<code>c.dist</code>	the distance function on clusterings to use. Should be one of "VI" or "Binder". Defaults to "VI".
<code>alpha</code>	a number in the unit interval, specifies the Bayesian confidence level of $1-\alpha$ . Defaults to <code>0.05</code> .
<code>object</code>	an object of class "credibleball".
<code>x</code>	an object of class "credibleball".
<code>data</code>	the dataset contained in a data frame with <code>ncol(cls.draw)</code> rows of data points.
<code>dx</code>	for <code>ncol(x)=1</code> , the estimated density at the observed data points.
<code>xgrid</code>	for <code>ncol(x)=1</code> , a grid of data points for density estimation.
<code>dxgrid</code>	for <code>ncol(x)=1</code> , the estimated density at the grid of data points.
<code>...</code>	other inputs to <code>summary</code> or <code>plot</code> .

**Details**

An advantage of Bayesian cluster analysis is that it provides a posterior over the entire partition space, expressing beliefs in the clustering structure given the data. The credible ball summarizes the uncertainty in the posterior around a clustering estimate `c.star` and is defined as the smallest ball around `c.star` with posterior probability at least  $1-\alpha$ . Possible distance metrics on the partition space are the Variation of Information and the N-invariant Binder's loss (Binder's loss times  $2/\text{length}(c.star)^2$ ). The posterior probability is estimated from MCMC posterior samples of clusterings.

The credible ball is summarized via the upper vertical, lower vertical, and horizontal bounds, defined, respectively, as the partitions in the credible ball with the fewest clusters that are most distant to `c.star`, with the most clusters that are most distant to `c.star`, and with the greatest distance to `c.star`.

In plots, data points are colored according to cluster membership. For `nrow(data)=1`, the data points are plotted against the density (which is estimated via a call to `density` if not provided). For `nrow(data)=2` the data points are plotted, and for `nrow(data)>2`, the data points are plotted in the space spanned by the first two principal components.

**Value**

<code>c.star</code>	vector, clustering estimate of the length( <code>c.star</code> ) data points.
<code>c.horiz</code>	A matrix of horizontal bounds of the credible ball, i.e. partitions in the credible ball with the greatest distant to <code>c.star</code> .
<code>c.uppervert</code>	A matrix of upper vertical bounds of the credible ball, i.e. partitions in the credible ball with the fewest clusters that are most distant to <code>c.star</code> .
<code>c.lowervert</code>	A matrix of lower vertical bounds of the credible ball, i.e. partitions in the credible ball with the most clusters that are most distant to <code>c.star</code> .
<code>dist.horiz</code>	the distance between <code>c.star</code> and the horizontal bounds
<code>dist.uppervert</code>	the distance between <code>c.star</code> and the upper vertical bounds
<code>dist.lowervert</code>	the distance between <code>c.star</code> and the lower vertical bounds

**Author(s)**

Sara Wade, <sara.wade@eng.cam.ac.uk>

**References**

Wade, S. and Ghahramani, Z. (2015) Bayesian cluster analysis: Point estimation and credible balls. Submitted. arXiv:1505.03339.

**See Also**

[minVI](#), [minbinder.ext](#), [maxpear](#), and [medv](#) to obtain a point estimate of clustering based on posterior MCMC samples; and [plotpsm](#) for a heat map of posterior similarity matrix.

**Examples**

```
data(galaxy.fit)
x=data.frame(x=galaxy.fit$x)
data(galaxy.pred)
data(galaxy.draw)

# Find representative partition of posterior
psm=comp.psm(galaxy.draw)
galaxy.VI=minVI(psm,galaxy.draw,method="all",include.greedy=TRUE)
summary(galaxy.VI)
plot(galaxy.VI,data=x,dx=galaxy.fit$fx,xgrid=galaxy.pred$x,dxgrid=galaxy.pred$fx)

# Uncertainty in partition estimate
galaxy.cb=credibleball(galaxy.VI$c1[,],galaxy.draw)
summary(galaxy.cb)
plot(galaxy.cb,data=x,dx=galaxy.fit$fx,xgrid=galaxy.pred$x,dxgrid=galaxy.pred$fx)

# Compare with heat map of posterior similarity matrix
plotpsm(psm)
```

---

 ex1.data

*A simulated dataset from a mixture of normals*


---

### Description

A simulated dataset from a mixture of four normals. True clusters are located at  $(\pm 2, \pm 2)$  with a standard deviation of 1.

### Usage

```
data(ex1.data)
```

### Format

1. The data points are contained in the first two columns `x1` and `x2` of length 200; the 200 data points were simulated from a mixture of four normals at locations  $(\pm 2, \pm 2)$  with a standard deviation of 1.
2. The third column `cls.true` of length 200 contains the true clustering of the 200 data points.

### Source

Wade, S. and Ghahramani, Z. (2015) Bayesian cluster analysis: Point estimation and credible balls. Submitted. arXiv:1505.03339.

### Examples

```
data(ex1.data)
x=ex1.data[,c(1,2)]
cls.true=ex1.data$cls.true
plot(x[,1],x[,2],xlab="x1",ylab="x2")
k=max(cls.true)
for(l in 2:k){
  points(x[cls.true==l,1],x[cls.true==l,2],col=l)}
```

---

 ex1.draw

*MCMC samples of Bayesian cluster model for a simulated dataset*


---

### Description

MCMC samples of clusterings from a Dirichlet process scale-location mixture model with normal components fitted to a simulated dataset, see [ex1.data](#). True clusters are located at  $(\pm 2, \pm 2)$  with a standard deviation of 1.

### Usage

```
data(ex1.draw)
```

**Format**

The matrix `ex1.draw` has 10,000 rows and 200 columns, with each row representing a MCMC posterior sample of the clustering of the 200 data points.

**Source**

Wade, S. and Ghahramani, Z. (2015) Bayesian cluster analysis: Point estimation and credible balls. Submitted. arXiv:1505.03339.

**Examples**

```
data(ex1.data)
data(ex1.draw)
x=data.frame(ex1.data[,c(1,2)])
cls.true=ex1.data$cls.true
plot(x[,1],x[,2],xlab="x1",ylab="x2")
k=max(cls.true)
for(l in 2:k){
  points(x[cls.true==l,1],x[cls.true==l,2],col=l)}

# Find representative partition of posterior
psm=comp.psm(ex1.draw)
ex1.VI=minVI(psm,ex1.draw,method="all"),include.greedy=TRUE)
summary(ex1.VI)
plot(ex1.VI,data=x)

# Uncertainty in partition estimate
ex1.cb=credibleball(ex1.VI$cl[1,],ex1.draw)
summary(ex1.cb)
plot(ex1.cb,data=x)
```

---

ex2.data

*A simulated dataset from a mixture of normals*


---

**Description**

A simulated dataset from a mixture of four normals. True clusters are located at  $(\pm 2, \pm 2)$  with a standard deviation of 1, 0.5, 1, and 1.5 in the first, second, third, and fourth quadrant respectively.

**Usage**

```
data(ex2.data)
```

**Format**

1. The data points are contained in the first two columns `x1` and `x2` of length 200; the 200 data points were simulated from a mixture of four normals at locations  $(\pm 2, \pm 2)$  with a standard deviation of 1, 0.5, 1, and 1.5 in the first, second, third, and fourth quadrant respectively.
2. The third column `cls.true` of length 200 contains the true clustering of the 200 data points.



**Source**

Wade, S. and Ghahramani, Z. (2015) Bayesian cluster analysis: Point estimation and credible balls. Submitted. arXiv:1505.03339.

**Examples**

```
data(ex2.data)
x=ex2.data[,c(1,2)]
cls.true=ex2.data$cls.true
plot(x[,1],x[,2],xlab="x1",ylab="x2")
k=max(cls.true)
for(l in 2:k){
  points(x[cls.true==l,1],x[cls.true==l,2],col=l)}
```

---

ex2.draw

---

*MCMC samples of Bayesian cluster model for a simulated dataset*


---

**Description**

MCMC samples of clusterings from a Dirichlet process scale-location mixture model with normal components fitted to a simulated dataset, see [ex2.data](#). True clusters are located at (+/- 2, +/- 2) with a standard deviation of 1, 0.5, 1, and 1.5 in the first, second, third, and fourth quadrant respectively.

**Usage**

```
data(ex2.draw)
```

**Format**

The matrix `ex2.draw` has 10,000 rows and 200 columns, with each row representing a MCMC posterior sample of the clustering of the 200 data points contained in [ex2.data](#).

**Source**

Wade, S. and Ghahramani, Z. (2015) Bayesian cluster analysis: Point estimation and credible balls. Submitted. arXiv:1505.03339.

**Examples**

```
data(ex2.data)
data(ex2.draw)
x=data.frame(ex2.data[,c(1,2)])
cls.true=ex2.data$cls.true
plot(x[,1],x[,2],xlab="x1",ylab="x2")
k=max(cls.true)
for(l in 2:k){
  points(x[cls.true==l,1],x[cls.true==l,2],col=l)}
```

```
# Find representative partition of posterior
psm=comp.psm(ex2.draw)
ex2.VI=minVI(psm,ex2.draw,method="all"),include.greedy=TRUE)
summary(ex2.VI)
plot(ex2.VI,data=x)

# Uncertainty in partition estimate
ex2.cb=credibleball(ex2.VI$cl[1,],ex2.draw)
summary(ex2.cb)
plot(ex2.cb,data=x)
```

---

galaxy.draw

*MCMC samples of a Bayesian cluster model for the galaxy dataset*


---

## Description

MCMC samples of clusterings from a Dirichlet process scale-location mixture model with normal components fitted to the [galaxies](#) dataset.

## Usage

```
data(galaxy.draw)
```

## Format

The matrix `galaxy.draw` has 10,000 rows and 82 columns, with each row representing a MCMC posterior sample of the clustering of the 82 data points.

## Source

Roeder, K. (1990) Density estimation with confidence sets exemplified by superclusters and voids in the galaxies, *Journal of the American Statistical Association*, 85: 617-624.

Wade, S. and Ghahramani, Z. (2015) Bayesian cluster analysis: Point estimation and credible balls. Submitted. arXiv:1505.03339.

## Examples

```
data(galaxy.fit)
x=data.frame(x=galaxy.fit$x)
data(galaxy.pred)
data(galaxy.draw)

# Find representative partition of posterior
psm=comp.psm(galaxy.draw)
galaxy.VI=minVI(psm,galaxy.draw,method="all"),include.greedy=TRUE)
summary(galaxy.VI)
plot(galaxy.VI,data=x,dx=galaxy.fit$x,xgrid=galaxy.pred$x,dxgrid=galaxy.pred$x)

# Uncertainty in partition estimate
```

```
galaxy.cb=credibleball(galaxy.VI$c1[,],galaxy.draw)
summary(galaxy.cb)
plot(galaxy.cb,data=x,dx=galaxy.fit$fx,xgrid=galaxy.pred$x,dxgrid=galaxy.pred$fx)
```

---

galaxy.fit	<i>Fitted density values from a Dirichlet process mixture model for the galaxy dataset</i>
------------	--

---

### Description

Fitted density values of a Dirichlet process scale-location mixture model with normal components fitted to the [galaxies](#) dataset.

### Usage

```
data(galaxy.fit)
```

### Format

1. The data points are contained in the first column `x` of length 82, see [galaxies](#) for more information.
2. The second column `fx` of length 82 contains the density estimate from the Dirichlet process mixture at each data point.

### Source

Roeder, K. (1990) Density estimation with confidence sets exemplified by superclusters and voids in the galaxies, *Journal of the American Statistical Association*, 85: 617-624.

Wade, S. and Ghahramani, Z. (2015) Bayesian cluster analysis: Point estimation and credible balls. Submitted. arXiv:1505.03339.

### Examples

```
data(galaxy.fit)
x=galaxy.fit$x
fx=galaxy.fit$fx
plot(x,fx,xlab="x",ylab="f(x)")
```

---

galaxy.pred	<i>Predicted density values from a Dirichlet process mixture model for the galaxy dataset</i>
-------------	---

---

**Description**

Predicted density values at a grid of new data points from a Dirichlet process scale-location mixture model with normal components fitted to the [galaxies](#) dataset.

**Usage**

```
data(galaxy.pred)
```

**Format**

1. The column `x` of length 141 contains a grid of new data points from 5 to 40 by 0.25.
2. The column `fx` of length 141 contains the density estimate from the Dirichlet process mixture at each new data point.

**Source**

Roeder, K. (1990) Density estimation with confidence sets exemplified by superclusters and voids in the galaxies, *Journal of the American Statistical Association*, 85: 617-624.

Wade, S. and Ghahramani, Z. (2015) Bayesian cluster analysis: Point estimation and credible balls. Submitted. arXiv:1505.03339.

**Examples**

```
data(galaxy.fit)
x=galaxy.fit$x
fx=galaxy.fit$fx
data(galaxy.pred)
xgrid=galaxy.pred$x
fxgrid=galaxy.pred$fx
plot(xgrid,fxgrid,xlab="x",ylab="f(x)",type="l")
points(x,fx)
```

---

greedy	<i>Optimizes the posterior expected loss with the greedy search algorithm</i>
--------	---

---

**Description**

Finds a representative partition of the posterior by minimizing the posterior expected loss with possible loss function of Binder's loss, the Variation of Information, and the modified Variation of Information through a greedy search algorithm.

**Usage**

```
greedy(psm, cls.draw = NULL, loss = NULL, start.cl = NULL, maxiter = NULL, L = NULL, suppress.comment = FALSE)
```

**Arguments**

<code>psm</code>	a posterior similarity matrix, which can be obtained from MCMC samples of clusterings through a call to <code>comp.psm</code> .
<code>cls.draw</code>	a matrix of the MCMC samples of clusterings of the <code>ncol(cls)</code> data points that have been used to compute <code>psm</code> . Note: <code>cls.draw</code> has to be provided if <code>loss="VI"</code> .
<code>loss</code>	the loss function used. Should be one of <code>"Binder"</code> , <code>"VI"</code> , or <code>"VI.lb"</code> . Defaults to <code>"VI.lb"</code> .
<code>start.cl</code>	clustering used as starting point. If <code>NULL</code> <code>start.cl= 1:nrow(psm)</code> is used.
<code>maxiter</code>	integer, maximum number of iterations. Defaults to <code>2*nrow(psm)</code> .
<code>L</code>	integer, specifies the number of local partitions considered at each iteration. Defaults to <code>2*nrow(psm)</code> .
<code>suppress.comment</code>	logical, for <code>method="greedy"</code> , prints a description of the current state (iteration number, number of clusters, posterior expected loss) at each iteration if set to <code>FALSE</code> . Defaults to <code>TRUE</code> .

**Details**

This function is called by `minVI` and `minbinder.ext` to optimize the posterior expected loss via a greedy search algorithm. Possible loss functions include Binder's loss (`"Binder"`) and the Variation of Information (`"VI"`). As computation of the posterior expected Variation of Information is expensive, a third option (`"VI.lb"`) is to minimize a modified Variation of Information by swapping the log and expectation. From Jensen's inequality, this can be viewed as minimizing a lower bound to the posterior expected Variation of Information.

At each iteration of the algorithm, we consider the `L` closest ancestors or descendants and move in the direction of minimum posterior expected; the distance is measured by Binder's loss or the Variation of Information, depending on the choice of `loss`. We recommend trying different starting locations `cl.start` and values of `l` that control the amount of local exploration. A description of the algorithm at every iteration is printed if `suppress.comment=FALSE`.

**Value**

<code>cl</code>	clustering with minimal value of expected loss.
<code>value</code>	value of posterior expected loss.
<code>iter.greedy</code>	the number of iterations the method needed to converge.

**Author(s)**

Sara Wade, <sara.wade@eng.cam.ac.uk>

## References

Wade, S. and Ghahramani, Z. (2015) Bayesian cluster analysis: Point estimation and credible balls. Submitted. arXiv:1505.03339.

## See Also

[minVI](#) or [minbinder.ext](#) which call [greedy](#) to find the point estimate that minimizes the posterior expected loss.

## Examples

```
data(ex1.data)
x=ex1.data[,c(1,2)]
cls.true=ex1.data$cls.true
plot(x[,1],x[,2],xlab="x1",ylab="x2")
k=max(cls.true)
for(l in 2:k){
  points(x[cls.true==1,1],x[cls.true==1,2],col=1)}

# Find representative partition of posterior
data(ex1.draw)
psm=comp.psm(ex1.draw)
ex1.VI=minVI(psm,method="greedy"),suppress.comment=FALSE)
summary(ex1.VI)
# Different initialization
ex1.VI.v2=minVI(psm,method="greedy"),suppress.comment=FALSE,start.cl=ex1.draw[nrow(ex1.draw),])
summary(ex1.VI.v2)
```

---

minbinder.ext

*Minimize the posterior expected Binder's loss*

---

## Description

Finds a representative partition of the posterior by minimizing the posterior expected Binder's loss.

## Usage

```
minbinder.ext(psm, cls.draw = NULL, method = c("avg", "comp", "draws", "laugreen", "greedy", "all"),
  max.k = NULL, include.lg = FALSE, include.greedy = FALSE, start.cl.lg = NULL,
  start.cl.greedy = NULL, tol = 0.001, maxiter = NULL, l = NULL, suppress.comment = TRUE)
```

## Arguments

<code>psm</code>	a posterior similarity matrix, which can be obtained from MCMC samples of clusterings through a call to <code>comp.psm</code> .
<code>cls.draw</code>	a matrix of the MCMC samples of clusterings of the <code>ncol(cls)</code> data points that have been used to compute <code>psm</code> . Note: <code>cls.draw</code> has to be provided if <code>method="draw"</code> or <code>"all"</code> .
<code>method</code>	the optimization method used. Should be one of <code>"avg"</code> , <code>"comp"</code> , <code>"draws"</code> , <code>"laugreen"</code> , <code>"greedy"</code> or <code>"all"</code> . Defaults to <code>"avg"</code> .
<code>max.k</code>	integer, if <code>method="avg"</code> or <code>"comp"</code> the maximum number of clusters up to which the hierarchical clustering is cut. Defaults to <code>ceiling(nrow(psm)/4)</code> .
<code>include.lg</code>	logical, should method <code>"laugreen"</code> be included when <code>method="all"</code> ? Defaults to <code>FALSE</code> .
<code>include.greedy</code>	logical, should method <code>"greedy"</code> be included when <code>method="all"</code> ? Defaults to <code>FALSE</code> .
<code>start.cl.lg</code>	clustering used as starting point for <code>method="laugreen"</code> . If <code>NULL</code> <code>start.cl= 1:nrow(psm)</code> is used.
<code>start.cl.greedy</code>	clustering used as starting point for <code>method="greedy"</code> . If <code>NULL</code> <code>start.cl= 1:nrow(psm)</code> is used.
<code>tol</code>	convergence tolerance for <code>method="laugreen"</code> .
<code>maxiter</code>	integer, maximum number of iterations for <code>method="greedy"</code> . Defaults to <code>2*nrow(psm)</code> .
<code>l</code>	integer, specifies the number of local partitions considered at each iteration for <code>method="greedy"</code> . Defaults to <code>2*nrow(psm)</code> .
<code>suppress.comment</code>	logical, for <code>method="greedy"</code> , prints a description of the current state (iteration number, number of clusters, posterior expected loss) at each iteration if set to <code>FALSE</code> . Defaults to <code>TRUE</code> .

## Details

This functions extends `minbinder` by implementing the greedy search algorithm to minimize the posterior expected Binder's loss.

Binder's loss counts the number of disagreements in all possible pairs of data points. The value returned is the posterior expected N-invariant Binder's loss, which is defined by multiplying Binder's loss times 2 and dividing by  $N^2$ ,  $N$  representing the sample size, and is so-called because it only depends on the sample size through the proportion of data points in each cluster intersection.

The function `minbinder` is called for optimization methods `method="avg"`, `"comp"`, `method="draws"`, and `"laugreen"`.

Method `"greedy"` implements a greedy search algorithm, where at each iteration, we consider the `l` closest ancestors or descendants and move in the direction of minimum posterior expected loss with the N-invariant Binder's loss as the distance. We recommend trying different starting locations `cl.start` and values of `l` that control the amount of local exploration. Depending on the starting location and `l`, the method can take some time to converge, thus it is only included in `method="all"` if `include.greedy=TRUE`. If `method="all"`, the starting location `cl.start` defaults to the best clustering found by the other methods. A description of the algorithm at every iteration is printed

if `suppress.comment=FALSE`. If `method="all"` all minimization methods except "laugreen" and "greedy" are applied by default.

### Value

<code>cl</code>	clustering with minimal value of expected loss. If <code>method="all"</code> a matrix containing the clustering with the smallest value of the expected loss over all methods in the first row and the clusterings of the individual methods in the next rows.
<code>value</code>	value of posterior expected loss. A vector corresponding to the rows of <code>cl</code> if <code>method="all"</code> .
<code>method</code>	the optimization method used.
<code>iter.greedy</code>	if <code>method="greedy"</code> or <code>method="all"</code> and <code>include.greedy=T</code> the number of iterations the method needed to converge.
<code>iter.lg</code>	if <code>method="laugreen"</code> or <code>method="all"</code> and <code>include.lg=T</code> the number of iterations the method needed to converge.

### Author(s)

Sara Wade, <sara.wade@eng.cam.ac.uk>

### References

- Binder, D.A. (1978) Bayesian cluster analysis, *Biometrika* **65**, 31–38.
- Fritsch, A. and Ickstadt, K. (2009) An improved criterion for clustering based on the posterior similarity matrix, *Bayesian Analysis*, **4**,367–391.
- Lau, J.W. and Green, P.J. (2007) Bayesian model based clustering procedures, *Journal of Computational and Graphical Statistics* **16**, 526–558.
- Wade, S. and Ghahramani, Z. (2015) Bayesian cluster analysis: Point estimation and credible balls. Submitted. arXiv:1505.03339.

### See Also

[summary.c.estimate](#) and [plot.c.estimate](#) to summarize and plot the resulting output from [minVI](#) or [minbinder.ext](#); [comp.psm](#) for computing posterior similarity matrix; [maxpear](#), [minVI](#), and [medv](#) for other point estimates of clustering based on posterior; and [credibleball](#) to compute credible ball characterizing uncertainty around the point estimate.

### Examples

```
data(ex2.data)
x=data.frame(ex2.data[,c(1,2)])
cls.true=ex2.data$cls.true
plot(x[,1],x[,2],xlab="x1",ylab="x2")
k=max(cls.true)
for(l in 2:k){
  points(x[cls.true==l,1],x[cls.true==l,2],col=l)}
```



```
# Find representative partition of posterior
data(ex2.draw)
psm=comp.psm(ex2.draw)
ex2.B=minbinder.ext(psm,ex2.draw,method="all"),include.greedy=TRUE)
summary(ex2.B)
plot(ex2.B,data=x)

# Compare with VI
ex2.VI=minVI(psm,ex2.draw,method="all"),include.greedy=TRUE)
summary(ex2.VI)
plot(ex2.VI,data=x)
```

---

minVI

---

*Minimize the posterior expected Variation of Information*


---

## Description

Finds a representative partition of the posterior by minimizing the lower bound to the posterior expected Variation of Information from Jensen's Inequality.

## Usage

```
minVI(psm, cls.draw=NULL, method=c("avg","comp","draws","greedy","all"),
      max.k=NULL, include.greedy=FALSE, start.cl=NULL, maxiter=NULL,
      l=NULL, suppress.comment=TRUE)
```

## Arguments

psm	a posterior similarity matrix, which can be obtained from MCMC samples of clusterings through a call to <code>comp.psm</code> .
cls.draw	a matrix of the MCMC samples of clusterings of the <code>ncol(cls)</code> data points that have been used to compute <code>psm</code> . Note: <code>cls.draw</code> has to be provided if <code>method="draw"</code> or <code>"all"</code> .
method	the optimization method used. Should be one of <code>"avg"</code> , <code>"comp"</code> , <code>"draws"</code> , <code>"greedy"</code> or <code>"all"</code> . Defaults to <code>"avg"</code> .
max.k	integer, if <code>method="avg"</code> or <code>"comp"</code> the maximum number of clusters up to which the hierarchical clustering is cut. Defaults to <code>ceiling(nrow(psm)/4)</code> .
include.greedy	logical, should method <code>"greedy"</code> be included when <code>method="all"</code> ? Defaults to <code>FALSE</code> .
start.cl	clustering used as starting point for <code>method="greedy"</code> . If <code>NULL</code> <code>start.cl= 1:nrow(psm)</code> is used.
maxiter	integer, maximum number of iterations for <code>method="greedy"</code> . Defaults to <code>2*nrow(psm)</code> .
l	integer, specifies the number of local partitions considered at each iteration for <code>method="greedy"</code> . Defaults to <code>2*nrow(psm)</code> .
suppress.comment	logical, for <code>method="greedy"</code> , prints a description of the current state (iteration number, number of clusters, posterior expected loss) at each iteration if set to <code>FALSE</code> . Defaults to <code>TRUE</code> .

## Details

The Variation of Information between two clusterings is defined as the sum of the entropies minus two times the mutual information. Computation of the posterior expected Variation of Information can be expensive, as it requires a Monte Carlo estimate. We consider a modified posterior expected Variation of Information, obtained by swapping the log and expectation, which is much more computationally efficient as it only depends on the posterior through the posterior similarity matrix. From Jensen's inequality, the problem can be viewed as minimizing a lower bound to the posterior expected loss.

We provide several optimization methods. For `method="avg"` and `"comp"`, the search is restricted to the clusterings obtained from a hierarchical clustering with average/complete linkage and `1-psm` as a distance matrix (the clusterings with number of clusters `1:max.k` are considered).

Method `"draws"` restricts the search to the clusterings sampled in the MCMC algorithm.

Method `"greedy"` implements a greedy search algorithm, where at each iteration, we consider the `l` closest ancestors or descendants and move in the direction of minimum posterior expected loss with the VI distance. We recommend trying different starting locations `cl.start` and values of `l` that control the amount of local exploration. Depending on the starting location and `l`, the method can take some time to converge, thus it is only included in `method="all"` if `include.greedy=TRUE`. If `method="all"`, the starting location `cl.start` defaults to the best clustering found by the other methods. A description of the algorithm at every iteration is printed if `suppress.comment=FALSE`. If `method="all"` all minimization methods except `"greedy"` are applied by default.

## Value

<code>cl</code>	clustering with minimal value of expected loss. If <code>method="all"</code> a matrix containing the clustering with the smallest value of the expected loss over all methods in the first row and the clusterings of the individual methods in the next rows.
<code>value</code>	value of posterior expected loss. A vector corresponding to the rows of <code>cl</code> if <code>method="all"</code> .
<code>method</code>	the optimization method used.
<code>iter.greedy</code>	if <code>method="greedy"</code> or <code>method="all"</code> and <code>include.greedy=T</code> the number of iterations the method needed to converge.

## Author(s)

Sara Wade, <sara.wade@eng.cam.ac.uk>

## References

- Meila, M. (2007) Bayesian model based clustering procedures, *Journal of Multivariate Analysis* **98**, 873–895.
- Wade, S. and Ghahramani, Z. (2015) Bayesian cluster analysis: Point estimation and credible balls. Submitted. arXiv:1505.03339

**See Also**

[summary.c.estimate](#) and [plot.c.estimate](#) to summarize and plot the resulting output from [minVI](#) or [minbinder.ext](#); [VI](#) or [VI.lb](#) for computing the posterior expected Variation of Information or the modified version from swapping the log and expectation; [comp.psm](#) for computing posterior similarity matrix; [maxpear](#), [minbinder.ext](#), and [medv](#) for other point estimates of clustering based on posterior; and [credibleball](#) to compute credible ball characterizing uncertainty around the point estimate.

**Examples**

```
data(ex2.data)
x=data.frame(ex2.data[,c(1,2)])
cls.true=ex2.data$cls.true
plot(x[,1],x[,2],xlab="x1",ylab="x2")
k=max(cls.true)
for(l in 2:k){
  points(x[cls.true==l,1],x[cls.true==l,2],col=1)}

# Find representative partition of posterior
data(ex2.draw)
psm=comp.psm(ex2.draw)
ex2.VI=minVI(psm,ex2.draw,method="all"),include.greedy=TRUE)
summary(ex2.VI)
plot(ex2.VI,data=x)

# Compare with Binder
ex2.B=minbinder.ext(psm,ex2.draw,method="all"),include.greedy=TRUE)
summary(ex2.B)
plot(ex2.B,data=x)
```

---

plotpsm

---

*Plot a heat map of the posterior similarity matrix*


---

**Description**

Produces a heat map of the posterior similarity matrix with data points reordered by hierarchical clustering.

**Usage**

```
plotpsm(psm, method = "complete", ...)
```

**Arguments**

psm	a posterior similarity matrix, which can be obtained from MCMC samples of clusterings through a call to <code>comp.psm</code> .
method	the agglomeration method to be used in hierarchical clustering. Defaults to "complete". See <code>hclust</code> .
...	other inputs to <code>image</code> .

**Details**

Produces a heatmap of the posterior similarity matrix with red representing high posterior probability of one and white representing low posterior probability of zero. Data points are first reordered by hierarchical clustering to increasing legibility.

**Value**

Produces a heatmap of the posterior similarity matrix.

**Author(s)**

Sara Wade, <sara.wade@eng.cam.ac.uk>

**See Also**

[comp.psm](#) for computing posterior similarity matrix; [hclust](#) for hierarchical clustering; and [credibleball](#) for an alternative representation of uncertainty in the posterior on clusterings.

**Examples**

```
data(ex1.data)
x=ex1.data[,c(1,2)]
cls.true=ex1.data$cls.true
plot(x[,1],x[,2],xlab="x1",ylab="x2")
k=max(cls.true)
for(l in 2:k){
  points(x[cls.true==l,1],x[cls.true==l,2],col=l)}

# Heat map to represent posterior uncertainty
data(ex1.draw)
psm=comp.psm(ex1.draw)
plotpsm(psm)
```

---

summary.c.estimate	<i>Summarize and plot the estimate of the partition</i>
--------------------	---

---

**Description**

Summarizes and plots the estimate of the partition in a Bayesian cluster analysis model.

**Usage**

```
## S3 method for class 'c.estimate'
summary(object, ...)
## S3 method for class 'c.estimate'
plot(x,data=NULL,dx=NULL,xgrid=NULL,dxgrid=NULL,...)
```

**Arguments**

object	an object of class "c.estimate", i.e. a clustering estimate .
x	an object of class "c.estimate", i.e. a clustering estimate .
data	the dataset contained in a data.frame with ncol(x\$c1) rows of data points.
dx	for ncol(data)=1, the estimated density at the observed data points.
xgrid	for ncol(data)=1, a grid of data points for density estimation.
dxgrid	for ncol(data)=1, the estimated density at the grid of data points.
...	other inputs to summary or plot.

**Details**

Summarizes and plots the clustering estimates returned by the functions `minVI` and `minbinder.ext`. In plots, data points are colored according to cluster membership. For `nrow(x)=1`, the data points are plotted against the density (which is estimated via a call to `density` if not provided). For `nrow(x)=2` the data points are plotted, and for `nrow(x)>2`, the data points are plotted in the space spanned by the first two principal components.

**Value**

method	the optimization method used to obtain the point estimate.
k	(a vector of) the number of clusters in the point estimate. Returns a vector if <code>n.c&gt;0</code> .
n.c	the number of point estimates in the object.
t	a list of length <code>n.c</code> of the table(s) with cluster sizes.
value	(a vector of) the posterior expected loss. Returns a vector if <code>n.c&gt;0</code> .

**Author(s)**

Sara Wade, <sara.wade@eng.cam.ac.uk>

**References**

Wade, S. and Ghahramani, Z. (2015) Bayesian cluster analysis: Point estimation and credible balls. Submitted. arXiv:1505.03339.

**See Also**

[minVI](#) and [minbinder.ext](#)

**Examples**

```
data(galaxy.draw)
data(galaxy.fit)
data(galaxy.pred)
x=data.frame(x=galaxy.fit$x)

# Find representative partition of posterior
```

```

psm=comp.psm(galaxy.draw)
galaxy.VI=minVI(psm,galaxy.draw,method="all"),include.greedy=TRUE)
summary(galaxy.VI)
plot(galaxy.VI,data=x,dx=galaxy.fit$fx,xgrid=galaxy.pred$x,dxgrid=galaxy.pred$fx)
galaxy.B=minbinder.ext(psm,galaxy.draw,method="all"),include.greedy=TRUE)
summary(galaxy.B)
plot(galaxy.B,data=x,dx=galaxy.fit$fx,xgrid=galaxy.pred$x,dxgrid=galaxy.pred$fx)

```

---

VI *Compute the posterior expected Variation of Information or the modified version from swapping log and expectation*

---

### Description

Based on MCMC samples of partitions, computes the posterior expected Variation of Information or the modified Variation of Information which switches the log and expectation.

### Usage

```
VI(cls, cls.draw)
```

```
VI.lb(cls, psm)
```

### Arguments

<code>cls</code>	a matrix of partitions where the posterior expected (modified) Variation of Information is to be evaluated. Each row corresponds to a clustering of <code>ncol(cls)</code> data points.
<code>cls.draw</code>	a matrix of MCMC samples of partitions. Each row corresponds to a clustering of <code>ncol(cls)</code> data points.
<code>psm</code>	a posterior similarity matrix, which can be obtained from MCMC samples of clusterings through a call to <code>comp.psm</code> .

### Details

The Variation of Information (VI) between two clusterings is defined as the sum of the entropies minus two times the mutual information. Computation of the posterior expected VI can be expensive, as it requires a Monte Carlo estimate. The modified posterior expected VI, obtained by swapping the log and expectation, is much more computationally efficient as it only depends on the posterior through the posterior similarity matrix. From Jensen's inequality, the problem of finding the optimal partition which minimizing the posterior expected modified VI can be viewed as minimizing a lower bound to the posterior expected VI.

### Value

vector of length `nrow(cls)` of the posterior expected (modified) VI.

**Author(s)**

Sara Wade, <sara.wade@eng.cam.ac.uk>

**References**

Meila, M. (2007) Bayesian model based clustering procedures, *Journal of Multivariate Analysis* **98**, 873–895.

Wade, S. and Ghahramani, Z. (2015) Bayesian cluster analysis: Point estimation and credible balls. Submitted. arXiv:1505.03339.

**See Also**

[minVI](#) which locates the partition that minimizes the posterior expected modified VI.

**Examples**

```
data(ex2.data)
x=data.frame(ex2.data[,c(1,2)])
cls.true=ex2.data$cls.true
plot(x[,1],x[,2],xlab="x1",ylab="x2")
k=max(cls.true)
for(l in 2:k){
points(x[cls.true==l,1],x[cls.true==l,2],col=1)}

# Find representative partition of posterior
data(ex2.draw)
psm=comp.psm(ex2.draw)
ex2.VI=minVI(psm,ex2.draw,method="all"),include.greedy=TRUE)
summary(ex2.VI)

# Compute posterior expected VI for each partition
VI(ex2.VI$c1,ex2.draw)
```