Package ‘GreedyEPL’

January 20, 2025

Type Package

Title Greedy Expected Posterior Loss

Version 1.2

Date 2021-08-30

Description Summarises a collection of partitions into a single optimal partition. The objective function is the expected posterior loss, and the minimisation is performed through a greedy algorithm described in Rastelli, R. and Friel, N. (2017) ``Optimal Bayesian estimators for latent variable cluster models'' <DOI:10.1007/s11222-017-9786-y>.

License GPL-3

Imports Rcpp (>= 0.12.10)

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Author Riccardo Rastelli [aut, cre]

Maintainer Riccardo Rastelli <riccardoras@gmail.com>

Repository CRAN

Date/Publication 2021-09-02 15:30:02 UTC

Contents

`GreedyEPL-package` *Greedy Expected Posterior Loss*

Description

Summarises a collection of partitions into a single optimal partition. The objective function is the expected posterior loss, and the minimisation is performed through a greedy algorithm described in Rastelli, R. and Friel, N. (2017) "Optimal Bayesian estimators for latent variable cluster models" <DOI:10.1007/s11222-017-9786-y>.

Author(s)

Riccardo Rastelli

Mantainer: Riccardo Rastelli <riccardo.rastelli@wu.ac.at>

References

Rastelli, R. and Friel, N. (2017) "Optimal Bayesian estimators for latent variable cluster models" <DOI:10.1007/s11222-017-9786-y>

`CollapseLabels` *CollapseLabels*

Description

Performs a permutation on the group labels so that, if `K` non-empty groups are present, the labels used are exactly `{1, ..., K}`.

Usage

```
CollapseLabels(decision)
```

Arguments

`decision` A vector identifying a partition. The entries must be positive integers.

Details

The labels of the clustering provided `{1,...,G}` are mapped into `{1,...,K}` (`K is less or equal than G`) based on their order of appearance in the vector provided.

Examples

```
set.seed(123)
decision <- sample(1:50, 50, TRUE)
CollapseLabels(decision = decision)
```

MinimiseEPL *MinimiseEPL*

Description

Performs greedy optimisation to find the Bayes optimal decision given a collection (sample) of partitions.

Usage

```
MinimiseEPL(sample_of_partitions, pars = list())
```

Arguments

sample_of_partitions	A TxN matrix whose rows correspond to partitions of the same N data observations.
pars	A list of additional parameters. See 'Details'.

Details

- weights: a vector of size T assigning a positive weight to each of the partitions in the sample. The vector does not have to be normalised but its entries must be positive numbers. If unspecified all weights are set to 1.
- Kup: positive integer identifying the number of groups used to generate the random initial partition. Defaults to N.
- decision_init: a vector of size N identifying the partition used to initialise the greedy algorithm. Entries must be integers greater than one. The default is sample(1:Kup, N, TRUE). Note that the number of groups in the initial partition also corresponds to the maximum number of groups allowed.
- loss_type: defines the loss function to be used in the expected posterior loss minimisation. Can be one of "VI" (Variation of Information), "B" (Binder's loss), "NVI" (Normalised Variation of Information) or "NID" (Normalised Information Distance). Defaults to "VI".

See Also

CollapseLabels.

Examples

```
set.seed(123)
N <- 10
niter <- 50
Kup <- 3
sample_of_partitions <- matrix(sample(x = 1:Kup, size = niter*N, replace = TRUE),niter,N)
loss_type = "VI"
output <- MinimiseEPL(sample_of_partitions, list(Kup = Kup, loss_type = loss_type))
```

```
output$decision
output$EPL
```

Index