Package ‘salso’

February 7, 2020

Type Package

Title Sequentially-Allocated Latent Structure Optimization

Version 0.1.16

Author David B. Dahl

Maintainer David B. Dahl <dahl@stat.byu.edu>

Description Point estimation for partition distributions using the sequentially-allocated latent structure optimization (SALSO) method to minimize the expectation of the Binder loss or the lower bound of the expectation of the variation of information loss. The SALSO method was presented at the workshop ``Bayesian Nonparametric Inference: Dependence Structures and their Applications'' in Oaxaca, Mexico on December 6, 2017. See <https://www.birs.ca/events/2017/5-day-workshops/17w5060/schedule>.

License MIT + file LICENSE | Apache License 2.0

Depends R (>= 3.3.0)

SystemRequirements Cargo (>= 1.31.0) for installation from sources:
see file INSTALL

Encoding UTF-8

LazyData TRUE

RoxygenNote 7.0.2

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-02-07 19:00:02 UTC

R topics documented:

bell 2
binder 2
confidence 3
dlso 4
enumerate.partitions 5
enumerate.permutations 6
iris.clusterings 7

plot.salso.confidence 7
psm 8
salso 9

Index **12**

bell *Compute the Bell Number*

Description

These functions compute the Bell number (the number of partitions of a set) or its natural logarithm.

Usage

```
bell(nItems)

lbell(nItems)
```

Arguments

nItems	The size of the set `{1,2,...,n}`.

Value

A numeric vector of length one giving the Bell number or its natural logarithm.

Examples

```
bell(12)
lbell(300)
all.equal( bell(5), exp(lbell(5)) )
```

binder *Compute a Partition Loss Function*

Description

These functions compute the expectation of the Binder loss and the lower bound of the expectation of the variation of information loss for given partitions based on the supplied pairwise similarity matrix.

Usage

```
binder(partitions, psm)

VI.lb(partitions, psm)
```

Arguments

partitions	An integer matrix of cluster labels, where each row is a partition given as cluster labels. Two items are in the same subset (i.e., cluster) if their labels are equal.
psm	A pairwise similarity matrix, i.e., n-by-n symmetric matrix whose (i,j) element gives the (estimated) probability that items i and j are in the same subset (i.e., cluster) of a partition (i.e., clustering).

Value

A numeric vector of length equal to the number of rows of partitions, where each element gives the value of the loss function.

Examples

```
probs <- psm(iris.clusterings, parallel=FALSE)
binder(iris.clusterings[1:5,], probs)
VI.lb(iris.clusterings[1:5,], probs)
```

confidence *Compute Clustering Confidence*

Description

This function computes the confidence values for n observations based on a clustering estimate and the pairwise similarity matrix.

Usage

```
confidence(estimate, psm)
```

Arguments

estimate	A vector of length n, where i and j are in the same subset (i.e., cluster) if and only if estimate[i] == estimate[j].
psm	A pairwise similarity matrix, i.e., n-by-n symmetric matrix whose (i,j) element gives the (estimated) probability that items i and j are in the same subset (i.e., cluster) of a partition (i.e., clustering).

Value

A list of the following elements:

estimate The value of the estimate argument.

psm The value of the psm argument.

confidence A numeric vector with the same length as estimate that contains the mean probability that each item is paired with all of the other items in its subset (i.e., cluster).

confidenceMatrix A matrix containing the mean confidences of items in each subset on the diagonal. In the off-diagonal elements, the mean confidence among all pairs from the two subsets. High probabilities on the diagonal and low probabilities everywhere else indicate good separability among the subsets.

exemplar A numeric vector containing the exemplar for each subset (i.e, cluster). The "exemplar" of a subset has the highest probability of being clustered with all of the other items in its subset.

order A vector giving the permutation of the original observations used in plotting.

Author(s)

David B. Dahl <dahl@stat.byu.edu>

See Also

plot.salso.confidence, salso, dlso, psm

Examples

```
# Use 'parallel=FALSE' per CRAN rules for examples but, in practice, omit this.
probs <- psm(iris.clusterings, parallel=FALSE)
est <- salso(probs, parallel=FALSE)$estimate
conf <- confidence(est, probs)
conf
```

dlso *Draws-Based Latent Structure Optimization*

Description

This function provides a point estimate for a partition distribution using the draws latent structure optimization (DLSO) method, which is also known as the least-squares clustering method (Dahl 2006). The method seeks to minimize the expectation of the Binder loss or the lower bound of the expectation of the variation of information loss by picking the minimizer among the partitions supplied by the draws argument.

Usage

```
dlso(psm, loss = c("VI.lb", "binder")[1], draws, parallel = FALSE)
```

Arguments

psm	A pairwise similarity matrix, i.e., n-by-n symmetric matrix whose (i,j) element gives the (estimated) probability that items i and j are in the same subset (i.e., cluster) of a partition (i.e., clustering).
loss	Either "VI.lb" or "binder", to indicate the desired loss function.

`draws` A `B`-by-`n` matrix, where each of the `B` rows represents a clustering of `n` items using cluster labels. For clustering `b`, items `i` and `j` are in the same cluster if `x[b,i] == x[b,j]`.

`parallel` Should the search use all CPU cores? (Currently ignored since parallelization is not implemented.)

Value

A list of the following elements:

estimate An integer vector giving a partition encoded using cluster labels.

loss A character vector equal to the `loss` argument.

expectedLoss A numeric vector of length one giving the expected loss.

References

D. A. Binder (1978), Bayesian cluster analysis, *Biometrika* **65**, 31-38.

D. B. Dahl (2006), Model-Based Clustering for Expression Data via a Dirichlet Process Mixture Model, in *Bayesian Inference for Gene Expression and Proteomics*, Kim-Anh Do, Peter Müller, Marina Vannucci (Eds.), Cambridge University Press.

J. W. Lau and P. J. Green (2007), Bayesian model based clustering procedures, *Journal of Computational and Graphical Statistics* **16**, 526-558. D. B. Dahl, M. A. Newton (2007), Multiple Hypothesis Testing by Clustering Treatment Effects, *Journal of the American Statistical Association*, **102**, 517-526.

A. Fritsch and K. Ickstadt (2009), An improved criterion for clustering based on the posterior similarity matrix, *Bayesian Analysis*, **4**, 367-391.

S. Wade and Z. Ghahramani (2018), Bayesian cluster analysis: Point estimation and credible balls. *Bayesian Analysis*, **13:2**, 559-626.

See Also

`psm`, `confidence`, `salso`

Examples

```
dlso(draws=iris.clusterings)
```

`enumerate.partitions` *Enumerate Partitions of a Set*

Description

This function produces a matrix whose rows provide all possible partitions of the set of integers `{0,1,...,n-1}`. These partitions are provided as cluster labels, where two items are in the same subset (i.e., cluster) if their labels are equal.

Usage

```
enumerate.partitions(nItems)
```

Arguments

nItems The size of the set `{0,1,...,n-1}`, i.e., `n`.

Value

A matrix of integers, where each row is a partition encoded as a vector of cluster labels.

Examples

```
enumerate.partitions(5)
```

---

enumerate.permutations *Enumerate Permutations of Items*

Description

This function produces a matrix whose rows provide all possible permutations of the set of integers `{0,1,...,n-1}`.

Usage

```
enumerate.permutations(nItems)
```

Arguments

nItems The size of the set `{0,1,...,n-1}`, i.e., `n`.

Value

A matrix of integers, where each row is a permutation.

Examples

```
enumerate.permutations(5)
```

`iris.clusterings` *Clusterings of the Iris Data*

Description

Randomly generated clusterings of the iris dataset.

Usage

```
iris.clusterings
```

Format

A 1000-by-150 matrix of 1000 randomly generated clusterings of the 150 observations in the iris dataset.

Source

Unknown.

See Also

`iris`

`plot.salso.confidence` *Confidence and Exemplar Plotting*

Description

This function produces confidence plots (e.g., heatmaps of pairwise allocation probabilities) and exemplar plots. The "exemplar" refers to the best representative of a particular cluster. See `confidence` for further explanation.

Usage

```
## S3 method for class 'salso.confidence'
plot(
  x,
  estimate = NULL,
  data = NULL,
  showLabels = length(x$estimate) <= 50,
  ...
)
```

Arguments

x	An object returned by the confidence function.
estimate	A vector of length n, where i and j are in the same subset (i.e., cluster) if and only if estimate[i] == estimate[j].' If NULL, the x$estimate in used.
data	The data from which the distances were computed.
showLabels	Should the names of items be shown in the plot?
...	Currently ignored.

Value

NULL, invisibly.

Author(s)

David B. Dahl <dahl@stat.byu.edu>

See Also

confidence, psm, dlso, salso

Examples

```
# Use 'parallel=FALSE' per CRAN rules for examples but, in practice, omit this.
probs <- psm(iris.clusterings, parallel=FALSE)
est <- salso(probs, parallel=FALSE)$estimate
conf <- confidence(est, probs)
plot(conf)
plot(conf, data=iris)
```

psm *Compute the Pairwise Similarity Matrix*

Description

This function computes the n-by-n matrix whose (i,j) element gives the (estimated) probability that items i and j are in the same subset (i.e., cluster).

Usage

```
psm(x, parallel = TRUE)
```

Arguments

x	A B-by-n matrix, where each of the B rows represents a clustering of n items using cluster labels. For clustering b, items i and j are in the same cluster if x[b,i] == x[b,j].
parallel	Should the computation use all CPU cores?

Value

A `n`-by-`n` symmetric matrix whose `(i,j)` element gives the estimated expected number of times that items `i` and `j` are in the same subset (i.e., cluster or feature) based on the frequencies from the supplied clusterings or feature allocations.

Examples

```
dim(iris.clusterings)
# Use 'parallel=FALSE' per CRAN rules for examples but, in practice, omit this.
probs <- psm(iris.clusterings, parallel=FALSE)
dim(probs)
probs[1:6, 1:6]
```

---

salso *Sequentially-Allocated Latent Structure Optimization*

Description

This function provides a point estimate for a partition distribution using the sequentially-allocated latent structure optimization (SALSO) method. The method seeks to minimize the expectation of the Binder loss or the lower bound of the expectation of the variation of information loss. The SALSO method was presented at the workshop "Bayesian Nonparametric Inference: Dependence Structures and their Applications" in Oaxaca, Mexico on December 6, 2017. See <https://www.birs.ca/events/2017/5-day-workshops/17w5060/schedule>.

Usage

```
salso(
  psm,
  loss = c("VI.lb", "binder")[1],
  maxSize = 0,
  batchSize = 100,
  seconds = Inf,
  maxScans = 10,
  probExplorationProbAtZero = 0.5,
  probExplorationShape = 0.5,
  probExplorationRate = 50,
  parallel = TRUE
)
```

Arguments

`psm`	A pairwise similarity matrix, i.e., `n`-by-`n` symmetric matrix whose `(i,j)` element gives the (estimated) probability that items `i` and `j` are in the same subset (i.e., cluster) of a partition (i.e., clustering).
`loss`	Either `"VI.lb"` or `"binder"`, to indicate the desired loss function.

`maxSize` The maximum number of subsets (i.e, clusters). The optimization is constrained to produce solutions whose number of subsets is no more than the supplied value. If zero, the size is not constrained.

`batchSize` The number of permutations to consider per batch (although the actual number of permutations per batch is a multiple of the number of cores when `parallel=TRUE`). Batches are sequentially performed until the most recent batch does not lead to a better result. Therefore, at least two batches are performed (unless the `seconds` threshold is exceeded.)

`seconds` A time threshold in seconds after which the function will be curtailed (with a warning) instead of performing another batch of permutations. Note that the function could take considerably longer because the threshold is only checked after each batch is completed.

`maxScans` The maximum number of reallocation scans after the initial allocation. The actual number of scans may be less than `maxScans` since the method stops if the result does not change between scans.

`probExplorationProbAtZero`
The probability of the point mass at zero for the spike-and-slab distribution of the probability of exploration, i.e. the probability of picking the second best micro-optimization (instead of the best). This probability is randomly sampled for (and constant within) each permutation.

`probExplorationShape`
The shape of the gamma distribution for the slab in the spike-and-slab distribution of the probability of exploration.

`probExplorationRate`
The rate of the gamma distribution for the slab in the spike-and-slab distribution of the probability of exploration.

`parallel` Should the search use all CPU cores?

Value

A list of the following elements:

estimate An integer vector giving a partition encoded using cluster labels.

loss A character vector equal to the `loss` argument.

expectedLoss A numeric vector of length one giving the expected loss.

nScans An integer vector giving the number of scans used to arrive at the supplied estimate.

probExploration The probability of picking the second best micro-optimization (instead of the best) for the permutation yielding the supplied estimate.

nPermutations An integer giving the number of permutations actually performed.

batchSize An integer giving the number of permutations per batch.

curtailed A logical indicating whether the search was cut short because the time exceeded the threshold.

References

D. A. Binder (1978), Bayesian cluster analysis, *Biometrika* **65**, 31-38.

D. B. Dahl (2006), Model-Based Clustering for Expression Data via a Dirichlet Process Mixture Model, in *Bayesian Inference for Gene Expression and Proteomics*, Kim-Anh Do, Peter Müller, Marina Vannucci (Eds.), Cambridge University Press.

J. W. Lau and P. J. Green (2007), Bayesian model based clustering procedures, *Journal of Computational and Graphical Statistics* **16**, 526-558. D. B. Dahl, M. A. Newton (2007), Multiple Hypothesis Testing by Clustering Treatment Effects, *Journal of the American Statistical Association*, **102**, 517-526.

A. Fritsch and K. Ickstadt (2009), An improved criterion for clustering based on the posterior similarity matrix, *Bayesian Analysis*, **4**, 367-391.

S. Wade and Z. Ghahramani (2018), Bayesian cluster analysis: Point estimation and credible balls. *Bayesian Analysis*, **13:2**, 559-626.

See Also

`psm`, `confidence`, `dlso`

Examples

```
# Use 'parallel=FALSE' per CRAN rules for examples but, in practice, omit this.
probs <- psm(iris.clusterings, parallel=FALSE)
salso(probs, parallel=FALSE)
```

Index

∗Topic **datasets**
iris.clusterings, 7

bell, 2
binder, 2

confidence, 3, *5*, *7*, *8*, *11*

dlso, *4*, 4, *8*, *11*

enumerate.partitions, 5
enumerate.permutations, 6

iris, *7*
iris.clusterings, 7

lbell *(*bell*)*, 2

plot.salso.confidence, *4*, 7
psm, *4*, *5*, *8*, 8, *11*

salso, *4*, *5*, *8*, 9

VI.lb *(*binder*)*, 2