Package 'sdols'

October 30, 2018

Type Package

Title Summarizing Distributions of Latent Structures

Version 1.7

Date 2018-10-29

URL https://dahl.byu.edu

BugReports https://dahl.byu.edu

Description Summaries of distributions on clusterings and feature allocations are provided. Specifically, point estimates are obtained by the sequentially-allocated latent structure optimization (SALSO) algorithm to minimize squared error loss, absolute error loss, Binder loss, or the lower bound of the variation of information loss. Clustering uncertainty can be assessed with the confidence calculations and the associated plot.

Imports rscala (>= 3.2.3), commonsMath (>= 1.2), stats

Depends R (>= 3.1.0)

LazyData TRUE

License Apache License 2.0 | file LICENSE

Encoding UTF-8

RoxygenNote 6.1.0

NeedsCompilation no

Author David B. Dahl [aut, cre],
Peter Müller [aut]

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

Repository CRAN

Date/Publication 2018-10-30 11:30:06 UTC

R topics documented:

`confidence` *Compute Clustering Confidence*

Description

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

Usage

```
confidence(estimate, expectedPairwiseAllocationMatrix)
```

Arguments

`estimate`	A vector of length `n`, where `i` and `j` are in the same cluster if and only if `clustering[i] == clustering[j]`.
`expectedPairwiseAllocationMatrix`	A `n`-by-`n` symmetric matrix whose `(i,j)` elements gives the estimated expected probability that items `i` and `j` are in the same cluster.

Author(s)

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

See Also

`expectedPairwiseAllocationMatrix`, `dlso`, `salso`

Examples

```
probabilities <- expectedPairwiseAllocationMatrix(iris.clusterings)
clustering <- salso(probabilities)
conf <- confidence(clustering,probabilities)
conf
```

dlso *Perform Draws-Based Latent Structure Optimization*

Description

Among the supplied latent structures, this function picks the structure that minimizes one of various loss functions.

Usage

```
dlso(x, loss = c("squaredError", "absoluteError", "binder",
  "lowerBoundVariationOfInformation")[1], maxSize = 0,
  multicore = TRUE, expectedPairwiseAllocationMatrix = NULL)
```

Arguments

x	A collection of clusterings or feature allocations. If x is a B-by-n matrix, 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 and only if x[b,i] == x[b,j]. If x is a list of length B, each element of list represents a feature allocation using a binary matrix of n rows and an arbitrary number of columns. For feature allocation b, items i and j share m features if, for k = 1, 2, ..., the expression x[[b]][i,k] == x[[b]][j,k] == 1 is true exactly m times.
loss	One of "squaredError", "absoluteError", "binder", or "lowerBoundVariationOfInformation" to indicate the optimization should seeks to minimize squared error loss, absolute error loss, Binder loss (Binder 1978), or the lower bound of the variation of information loss (Wade & Ghahramani 2017), respectively. For clustering, the first three are equivalent. For feature allocation, only the first two are valid.
maxSize	Either zero or a positive integer. If a positive integer, the optimization is constrained to produce solutions whose number of clusters or number of features is no more than the supplied value. If zero, the size is not constrained.
multicore	Logical indicating whether computations should take advantage of multiple CPU cores.
expectedPairwiseAllocationMatrix	A n-by-n symmetric matrix whose (i,j) elements gives the estimated expected number of times that items i and j are in the same subset (i.e., cluster or feature). If NULL, it is computed from x.

Value

A clustering (as a vector of cluster labels) or a feature allocation (as a binary matrix of feature indicators).

Author(s)

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

References

Wade, S. and Ghahramani, Z. (2017). Bayesian cluster analysis: Point estimation and credible balls. Bayesian analysis.

Binder, D. (1978). Bayesian Cluster Analysis. Biometrika, 65: 31–38.

See Also

expectedPairwiseAllocationMatrix, salso

Examples

```
dlso(iris.clusterings)
dlso(USArrests.featureAllocations)
```

`expectedPairwiseAllocationMatrix`
Compute Expected Pairwise Allocation Matrix

Description

This function computes the `n`-by-`n` matrix whose `(i,j)` element gives the estimated expected number of times that `i` and `j` are in the same subset (i.e, cluster or feature). For clusterings, this is the estimated probability that items are clustered together. For feature allocations, this is the estimated expectation of the number of shared features. These estimates are based on the frequencies from the supplied, randomly-sampled clusterings or feature allocations.

Usage

```
expectedPairwiseAllocationMatrix(x)
```

Arguments

`x`	A collection of clusterings or feature allocations. If `x` is a `B`-by-`n` matrix, 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]`. If `x` is a list of length `B`, each element of list represents a feature allocation using a binary matrix of `n` rows and an arbitrary number of columns. For feature allocation `b`, items `i` and `j` share `m` features if, for `k = 1, 2, ...`, the expression `x[[b]][i,k] == x[[b]][j,k] == 1` is true exactly `m` times.

Value

A n-by-n symmetric matrix whose (i,j) elements 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.

Author(s)

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

See Also

dlso, salso

Examples

```
probabilities <- expectedPairwiseAllocationMatrix(iris.clusterings)
probabilities

expectedCounts <- expectedPairwiseAllocationMatrix(USArrests.featureAllocations)
expectedCounts
```

---

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.

See Also

iris

latentStructureFit *Compute Fit Summaries for a Latent Structure Estimate*

Description

This function computes various summaries of the fit of a clustering or feature allocation based on the expected pairwise allocation matrix.

Usage

```
latentStructureFit(estimate, expectedPairwiseAllocationMatrix)
```

Arguments

estimate	A clustering or feature allocation. If estimate is a length n vector, it is taken to be a clustering where items i and j are in the same cluster if and only if estimate[i] == estimate[j]. If estimate is a binary matrix of n rows and an arbitrary number of columns, it is taken to be a feature allocation where items i and j share m features if, for k = 1, 2, ..., the expression estimate[i,k] == estimate[j,k] == 1 is true exactly m times.
expectedPairwiseAllocationMatrix	A n-by-n symmetric matrix whose (i,j) elements gives the estimated expected number of times that items i and j are in the same subset (i.e., cluster or feature).

Author(s)

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

See Also

expectedPairwiseAllocationMatrix, salso

Examples

```
probabilities <- expectedPairwiseAllocationMatrix(iris.clusterings)
estimate <- salso(probabilities)
latentStructureFit(estimate, probabilities)

expectedCounts <- expectedPairwiseAllocationMatrix(USArrests.featureAllocations)
estimate <- salso(expectedCounts,"featureAllocation")
latentStructureFit(estimate, expectedCounts)
```

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

Description

Functions to produce confidence plots (e.g., heatmaps of pairwise allocation probabilities) and exemplar plots.

Usage

```
## S3 method for class 'sdols.confidence'
plot(x, clustering = NULL, data = NULL,
  show.labels = length(x$clustering) <= 50, ...)
```

Arguments

`x`	An object of class `shallot.confidence`.
`clustering`	A vector of cluster labels, or `NULL`.
`data`	The data from which the distances were computed.
`show.labels`	Show the items names be shown in the plot?
`...`	Currently ignored.

Author(s)

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

See Also

`expectedPairwiseAllocationMatrix`, `dlso`, `salso`

Examples

```
probabilities <- expectedPairwiseAllocationMatrix(iris.clusterings)
clustering <- salso(probabilities)
conf <- confidence(clustering,probabilities)
plot(conf)
plot(conf,data=iris)
```

salso *Perform Sequentially-Allocated Latent Structure Optimization*

Description

This function implements the sequentially-allocated latent structure optimization (SALSO) to find a clustering or feature allocation that minimizes various loss functions. The SALSO method was presented at the workshop "Bayesian Nonparametric Inference: Dependence Structures and their Applications" in Oaxaca, Mexico on December 6, 2017.

Usage

```
salso(expectedPairwiseAllocationMatrix, structure = c("clustering",
  "featureAllocation")[1], loss = c("squaredError", "absoluteError",
  "binder", "lowerBoundVariationOfInformation")[1], nCandidates = 100,
  budgetInSeconds = 10, maxSize = 0, maxScans = 10,
  multicore = TRUE, useOldImplementation = FALSE)
```

Arguments

`expectedPairwiseAllocationMatrix`
: A n-by-n symmetric matrix whose `(i,j)` elements gives the estimated expected number of times that items `i` and `j` are in the same subset (i.e., cluster or feature).

`structure`
: Either `"clustering"` or `"featureAllocation"` to indicate the optimization seeks to produce a clustering or a feature allocation.

`loss`
: One of `"squaredError"`, `"absoluteError"`, `"binder"`, or `"lowerBoundVariationOfInformation"` to indicate the optimization should seeks to minimize squared error loss, absolute error loss, Binder loss (Binder 1978), or the lower bound of the variation of information loss (Wade & Ghahramani 2017), respectively. When `structure="clustering"`, the first three are equivalent. When `structure="featureAllocation"`, only the first two are valid.

`nCandidates`
: The (maximum) number of candidates to consider. Fewer than `nCandidates` may be considered if the time in `budgetInSeconds` is exceeded. The computational cost is linear in the number of candidates and there are rapidly diminishing returns to more candidates.

`budgetInSeconds`
: The (maximum) number of seconds to devote to the optimization. When this time is exceeded, no more candidates are considered.

`maxSize`
: Either zero or a positive integer. If a positive integer, the optimization is constrained to produce solutions whose number of clusters or number of features is no more than the supplied value. If zero, the size is not constrained. To avoid overfitting in feature allocation estimation, it is recommended that `"maxSize"` be close the mean number of features (i.e., columns) in the feature allocations that generated the `expectedPairwiseAllocationMatrix`.

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

`multicore` Logical indicating whether computations should take advantage of multiple CPU cores.

`useOldImplementation` Logical indicating whether to use the old implementation. This should be removed after sufficient testing of the new implementation.

Value

A clustering (as a vector of cluster labels) or a feature allocation (as a binary matrix of feature indicators).

Author(s)

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

References

Wade, S. and Ghahramani, Z. (2017). Bayesian cluster analysis: Point estimation and credible balls. Bayesian analysis.

Binder, D. (1978). Bayesian Cluster Analysis. Biometrika, 65: 31–38.

See Also

`expectedPairwiseAllocationMatrix`, `dlso`

Examples

```
probabilities <- expectedPairwiseAllocationMatrix(iris.clusterings)
salso(probabilities)

expectedCounts <- expectedPairwiseAllocationMatrix(USArrests.featureAllocations)
salso(expectedCounts,"featureAllocation")
```

`USArrests.featureAllocations` *Feature Allocations of the USArrests Dataset*

Description

Randomly generated feature allocations of the states in the USArrests dataset.

Usage

```
USArrests.featureAllocations
```

Format

A list of 1000 randomly generated feature allocations for the 50 states in the USArrests dataset.

See Also

`USArrests`

Index