Package 'shallot'

April 4, 2020

Type Package

Title Random Partition Distribution Indexed by Pairwise Information

Version 0.4.9

Date 2020-04-03

Description Implementations are provided for the models described in the paper D. B. Dahl, R. Day, J. Tsai (2017) <DOI:10.1080/01621459.2016.1165103>. The Ewens, Ewens-Pitman, Ewens attraction, Ewens-Pitman attraction, and ddCRP distributions are available for prior and posterior simulation. Posterior simulation is based on a user-supplied likelihood. Supporting functions for partition estimation and plotting are also provided.

URL https://github.com/dbdahl/shallot

BugReports https://github.com/dbdahl/shallot/issues

Imports rscala (>= 3.2.18), commonsMath (>= 1.2.5), salso (>= 0.1.16)

License Apache License 2.0 | file LICENSE

RoxygenNote 7.1.0

Encoding UTF-8

NeedsCompilation no

Author David B. Dahl [aut, cre]

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

Repository CRAN

Date/Publication 2020-04-04 08:40:02 UTC

R topics documented:

`shallot-package` *Random Partition Distribution Indexed by Pairwise Information*

Description

This package implements models described in the paper Dahl, D. B., Day, R., and Tsai, J. (2017), Random Partition Distribution Indexed by Pairwise Information, *Journal of the American Statistical Association*, 112, 721-732. The Ewens, Ewens-Pitman, Ewens attraction, Ewens-Pitman attraction, and ddCRP distributions are available for prior simulation. We hope in the future to add posterior simulation with a user-supplied likelihood. Supporting functions for partition estimation and plotting are also planned.

Author(s)

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

References

Dahl, D. B., Day, R., and Tsai, J. (2017), Random Partition Distribution Indexed by Pairwise Information, *Journal of the American Statistical Association*, 112, 721-732. <DOI:10.1080/01621459.2016.1165103>

See Also

`ewens.pitman.attraction`, `sample.partitions`

Examples

```
data <- iris[,-ncol(iris)]
truth <- as.integer(iris[,ncol(iris)])
distance <- as.dist(as.matrix(dist(scale(data))+0.001))

decay <- decay.exponential(temperature(9.0, fixed=TRUE), distance)
permutation <- permutation(n.items=nrow(data), fixed = FALSE)
attraction <- attraction(permutation, decay)
mass <- mass(1.0, fixed = TRUE)
discount <- discount(0.2, fixed = TRUE)
```

```
distribution <- ewens.pitman.attraction(mass, discount, attraction)

raw <- sample.partitions(distribution, 500, parallel=FALSE)
samples <- process.samples(raw)

library(salso)
pp <- psm(samples$labels)
est <- salso(pp)
conf <- confidence(est$estimate, pp)
plot(conf)
plot(conf,data=data)
```

`association.matrix` *Association Matrix*

Description

This function creates an association matrix for a clustering/partition. The (i,j) element of the matrix is 1 if item i and j are in the same cluster/subset and 0 otherwise.

Usage

```
association.matrix(cl)
```

Arguments

`cl`	A vector containing cluster labels for a clustering/partition.

Value

A matrix of 0s and 1s indicating whether items i and j are in the same cluster/subset.

Examples

```
cl <- rep(1:3,times=c(2,4,3))
association.matrix(cl)
```

attraction *Attraction*

Description

This function creates an attraction from a permutation and a decay in preparation for use in the ewens.attraction, ewens.pitman.attraction, and ddcrp functions. For details on each of these arguments, please see the links below.

Usage

```
attraction(permutation, decay)

## S3 method for class 'shallot.attraction'
print(x, ...)

## S3 method for class 'shallot.attraction'
as.matrix(x, ...)
```

Arguments

permutation	An object of class shallot.permutation encoding the permutation of the items.
decay	An object of class shallot.decay detailing the transformation from distances to attractions.
x	An object of class shallot.attraction.
...	Currently ignored.

Value

An object of class shallot.attraction.

Author(s)

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

References

Dahl, D. B., Day, R., and Tsai, J. (2017), Random Partition Distribution Indexed by Pairwise Information, *Journal of the American Statistical Association*, 112, 721-732. <DOI:10.1080/01621459.2016.1165103>

See Also

ddcrp, decay, ewens.attraction, ewens.pitman.attraction, permutation

Examples

```
permutation <- permutation(n.items=50,fixed=FALSE)
decay <- decay.exponential(temperature(1.0),dist(scale(USArrests)))
attraction(permutation, decay)
```

`decay.reciprocal` *Decay Functions*

Description

These functions specify the decay to map distances to attractions.

Usage

```
decay.reciprocal(temperature, distance)

decay.exponential(temperature, distance)

decay.subtraction(temperature, distance, multiplier = 1.01)

## S3 method for class 'shallot.decay'
print(x, ...)
```

Arguments

`temperature`	An object of class `shallot.temperature`.
`distance`	An object of class `dist`.
`multiplier`	An scalar greater than `1.0` to ensure that attractions from `decay.subtraction` are finite.
`x`	An object of class `shallot.decay`.
`...`	Currently ignored.

Details

There are currently three choices for decay functions: reciprocal, exponential, and subtraction.

The reciprocal decay maps a distance `d` to an attraction `a` as follows: `a = 1/d^t`, where `t` is the temperature.

The exponential decay maps a distance `d` to an attraction `a` as follows: `a = exp(-t*d)`, where `t` is the temperature.

The subtract decay maps a distance `d` to an attraction `a` as follows: `a = (m-d)^t`, where `t` is the temperature and `m` is the maximum distance in `distance` multiplied by the supplied *multiplier*.

Author(s)

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

References

Dahl, D. B., Day, R., and Tsai, J. (2017), Random Partition Distribution Indexed by Pairwise Information, *Journal of the American Statistical Association*, 112, 721-732. <DOI:10.1080/01621459.2016.1165103>

See Also

dist, temperature, attraction

Examples

```
temp <- temperature(1.0)
distance <- dist(scale(USArrests))
decay1 <- decay.reciprocal(temp,distance)
decay2 <- decay.exponential(temp,distance)
decay3 <- decay.subtraction(temp,distance)
```

---

default.mass *Default Mass Selection*

Description

This function selects an optimal mass value for Cluster Analysis via Random Partition Distribtuions, using the Ewens-Pitman Attraction distribution.

Usage

```
default.mass(
  mass,
  list.epam,
  dis,
  new.draws = TRUE,
  w = c(1, 1, 1),
  discount = 0,
  temp = 10,
  loss = "binder",
  n.draws = 100L,
  two.stage = TRUE,
  parallel = TRUE
)

## S3 method for class 'shallot.default.mass'
print(x, ...)
```

Arguments

mass	optional, a vector of mass values.
list.epam	optional, a list of expected pairwise allocation matrices. Each matrix in the list needs the attributes "mass" and "n.draws".
dis	a dissimilarity structure of class dist.
new.draws	logical; if TRUE then new draws are obtained at each mass value.
w	a vector of length 3 of the weights to be used in the mass.algorithm.
discount	parameter of the Ewens-Pitman Attraction distribution.
temp	temperature parameter of the Ewens-Pitman Attraction distribution.
loss	One of "binder" or "VI.lb" to indicate the optimization should seek to minimize the expectation of the Binder loss (Binder 1978) or the lower bound of the expectation of the variation of information loss (Wade & Ghahramani 2017), respectively.
n.draws	number of draws of partitions to be obtained at each mass value.
two.stage	logical; if TRUE, the two stage algorithm is implemented in mass.algorithm.
parallel	logical; if TRUE computations will take advantage multiple CPU cores.
x	An object from the default.mass function.
...	currently ignored

Details

The function draws n.draws partitions at each specified mass value. If a vector of mass values is not given, then the default of seq(0.1,10,0.2) is used for loss "VI.lb" and seq(0.1,5,0.05) used for the other loss functions.

If a list of expected pairwise allocation matrices (EPAM) is provided, additional draws at matching mass values are added to the corresponding matrix. Additionally, no new draws are needed for estimation, if a list of EPAMs is provided.

A partition/clustering estimate from each EPAM is obtained using the SALSO method in salso. The estimate given minimizes the specified loss function with respect to the EPAM.

The function then uses the mass.algorithm to select the optimal mass value for clustering estimation.

Value

An object of class shallot.default.mass. This object is a list containing a matrix of 'best' possible mass values to maximize partition confidence and minimize the variance ratio, the clustering estimate, the expected pairwise allocation matrix, parameters used for optimization and the EPA distribution, and the list of expected pairwise allocation matrices for each mass value.

See Also

Other Default Mass Selection: mass.algorithm(), partition.confidence(), variance.ratio()

ewens *Partition Distributions*

Description

These functions specify the Ewens, Ewens-Pitman, Ewens attraction, Ewens-Pitman attraction, and ddCRP distributions which would then be used in the `sample.partitions` `function`.

Usage

```
ewens(mass, n.items, names = paste0("c", 1:n.items))

## S3 method for class 'shallot.distribution.ewens'
print(x, ...)

ewens.pitman(mass, discount, n.items, names = paste0("c", 1:n.items))

## S3 method for class 'shallot.distribution.ewensPitman'
print(x, ...)

ewens.attraction(mass, attraction)

## S3 method for class 'shallot.distribution.ewensAttraction'
print(x, ...)

ewens.pitman.attraction(mass, discount, attraction)

## S3 method for class 'shallot.distribution.ewensPitmanAttraction'
print(x, ...)

ddcrp(mass, attraction)

## S3 method for class 'shallot.distribution.ddcrp'
print(x, ...)
```

Arguments

`mass`	An object of class `shallot.mass`.
`n.items`	An integer containing the number of items to partition.
`names`	A character vector containing the names of the items. The default names are of the form "c1", "c2", etc.
`x`	An object of class `shallot.distribution`.
`...`	Currently ignored.
`discount`	An object of class `shallot.discount`.
`attraction`	An object of class `shallot.attraction`.

Value

An object of class `shallot.distribution`.

Author(s)

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

References

Dahl, D. B., Day, R., and Tsai, J. (2017), Random Partition Distribution Indexed by Pairwise Information, *Journal of the American Statistical Association*, 112, 721-732. <DOI:10.1080/01621459.2016.1165103>

See Also

`mass`, `discount`, `attraction`, `sample.partitions`

Examples

```
pd1 <- ewens(mass(1),50)

decay <- decay.exponential(temperature(1.0),dist(scale(USArrests)))
attraction <- attraction(permutation(n.items=50,fixed=FALSE), decay)
pd2 <- ewens.pitman.attraction(mass(1), discount(0.05), attraction)

pd3 <- ddcrp(mass(1), attraction)
```

`mass` *Mass, Discount, and Temperature Parameters*

Description

These functions set the mass, discount, and temperature parameters and, in the case of them being random, specify the parameters of their distribution.

Usage

```
mass(..., fixed = TRUE)

## S3 method for class 'shallot.mass'
print(x, ...)

discount(..., fixed = TRUE)

## S3 method for class 'shallot.discount'
print(x, ...)
```

```
temperature(..., fixed = TRUE)

## S3 method for class 'shallot.temperature'
print(x, ...)
```

Arguments

...	A number greater than 0.0 representing the value of the mass, discount, or temperaure parameters. Or, in the case of them being random, a vector of two numbers representing either: i. the shape and rate parameters of the gamma distribution for the mass or temperature, or ii. the shape parameters of the beta distribution for the discount. This argument is currently ignored for the associated print functions.
fixed	If TRUE, the parameter is fixed. If FALSE, the parameter value is samples from either: i. a gamma distribution for the mass or temperature, or ii. a beta distribution for the discount.
x	An object from the mass, discount, or temperature functions.

Details

If no parameters are specified, the mass parameter defaults to 1.2, the discount parameter defaults to 0.05, the temperature parameter defaults to 3.0. If the mass parameter is random, the default shape and rate parameters of the gamma distribution are 2.5 and 2, respectively. If the discount parameter is random, the default shape parameters of the beta distribution are 1.0 and 1.0. If the temperature parameter is random, the default shape and rate parameters of the gamma distribution are 2 and 0.5, respectively.

Value

An object of class shallot.mass, shallot.discount, or shallot.temperature.

Author(s)

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

Examples

```
mass()
mass(1.0)
mass(1.4, fixed=FALSE)
mass(0.5, 1, fixed=FALSE)
discount()
discount(0.2)
discount(1, 3, fixed=FALSE)
temperature()
temperature(2)
temperature(2, 4, fixed=FALSE)
```

`mass.algorithm` *Mass Selection Algorithm*

Description

This function selects the optimal mass value for Cluster Analysis via Random Partition distributions using the Ewens-Pitman attraction distribution.

Usage

```
mass.algorithm(mass, pc, vr, n, w = c(1, 1, 1), two.stage = TRUE)
```

Arguments

`mass`	a vector of mass values
`pc`	a vector of partition confidences for the partition estimates at the corresponding mass values
`vr`	a vector of variance ratios for the partition estimates at the corresponding mass values
`n`	a vector of the number of subsets in the partition estimates at the correpsonding mass values
`w`	a vector of length 3 specifying the weights of `pc`, `vr`, and `n`
`two.stage`	logical; if `TRUE`, the two stage algorithm is implemented

Details

The `mass.algorithm` function is used internally in the `default.mass` function.

The default value for `w` is `c(1,1,1)`.

The general algorithm is as follows:

1. Rank the partition confidence (`pc`) and variance ratio (`vr`). Select the `mass_i` value which minimizes the weigthed sum of $w_1 pc_i + w_2 vr_i + w_3 n_i$.

The two stage algorithm proceeds as follows:

1. Rank the partition confidence (`pc`) and variance ratio (`vr`). For each number of clusters `n` select the index which minimizes the weigthed sum of $w_1 pc_i + w_2 vr_i$.
2. Rerank the `pc` and `vr` of the selected indices and select the `mass_i` value which minimizes the weigthed sum of $w_1 pc_i + w_2 vr_i + w_3 n_i$ from among the selected indices.

Value

A matrix containing the 'best' `mass` value and corresponding values for `pc`, `vr`, and `n`. The matrix also contains the mass values for the partitions estimate with more one more and one less subset that the selected mass value.

See Also

Other Default Mass Selection: `default.mass()`, `partition.confidence()`, `variance.ratio()`

`nsubsets.random` *Number of Subsets*

Description

These functions either sample the number of subsets for supported partition distributions or computes probabilities, means, and variances of these distributions.

Usage

```
nsubsets.random(x, n.samples)

nsubsets.probability(x, n.subsets)

nsubsets.average(x)

nsubsets.variance(x)
```

Arguments

`x`	An object of class `shallot.distribution`.
`n.samples`	An integer containing the number of samples.
`n.subsets`	An integer containing the number of subsets.

Value

The `nsubsets.random` function returns a vector of random samples of the number of subsets in the distribution *x*.

The `nsubsets.probability` function returns the probability that the number of subsets is *n.subsets* in the distribution *x*. Depending on the number of items and the value of *n.subsets*, this function can be computationally intensive.

The `nsubsets.average` and `nsubsets.variance` functions return the mean and variances, respectively, of the number of subsets in the distribution *x*.

Author(s)

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

References

Dahl, D. B., Day, R., and Tsai, J. (2017), Random Partition Distribution Indexed by Pairwise Information, *Journal of the American Statistical Association*, 112, 721-732. <DOI:10.1080/01621459.2016.1165103>

See Also

partition.distribution

Examples

```
pd <- ewens.pitman.attraction(
        mass(1),
        discount(0.05),
        attraction(permutation(n.items=50,fixed=FALSE),
          decay.exponential(temperature(1.0),dist(scale(USArrests)))))
mean(nsubsets.random(pd,1000))
nsubsets.average(pd)

pde <- ewens(mass(1),50)
nsubsets.variance(pde)
nsubsets.probability(pde,4)
```

---

partition.confidence *Partition Confidence*

Description

This function calculates the partition confidence of a partition estimate from the corresponding expected pairwise allocation matrix (EPAM).

Usage

```
partition.confidence(x, y)
```

Arguments

x	If y is not specified then x must be an object of class salso.confidence. Otherwise, x is a vector of cluster labels and y is an expected pairwise allocation matrix.
y	If y is not specified then x must be an object of class salso.confidence. Otherwise, x is a vector of cluster labels and y is an expected pairwise allocation matrix.

Details

The partition.confidence takes as input an object of class salso.confidence and then calculates the partition confidence from the expected pairwise allocation matrix.

The partition confidence is the average values of the EPAM for items that are clustered together. Items which are in their own subset do not contribute to partition confidence.

Value

A vector of partition confidences.

See Also

Other Default Mass Selection: `default.mass()`, `mass.algorithm()`, `variance.ratio()`

Examples

```
x <- rep(c(1,2,3), times=c(2,3,5))
y <- diag(10)
y[upper.tri(y)] <- runif(45)
partition.confidence(x,y)
```

---

partition.pmf *Obtain the Probability Mass Function of a Partition Distribution*

Description

This function returns the probability mass function (pmf) of a partition distribution.

Usage

```
partition.pmf(x)
```

Arguments

x	An object of class `shallot.distribution` obtained, for example, from the `ewens.pitman.attraction` function.

Value

A function that takes a partition (as a vector in cluster label notation) and returns the probability — or, if `log=TRUE`, the log of the probability — of the supplied partition.

Author(s)

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

Examples

```
## Not run:
example(shallot)

## End(Not run)
```

permutation *Permutation*

Description

These function define a permutation for subsequent use.

Usage

```
permutation(..., n.items = NULL, fixed = TRUE)

## S3 method for class 'shallot.permutation'
print(x, ...)
```

Arguments

`...`	For the function `permutation`, a permutation of the integers 1, 2,... n, where n is the length of the vector. For the function `print.shallot.permutation`, this is ignored.
`n.items`	An optional argument provided instead of `...` to request a random partition. The argument `fixed` must be `FALSE`.
`fixed`	Should the permutation be fixed?
`x`	An object of class `shallot.permutation`.

Details

A valid permutation of length n is an integer vector of length n containing each integer 1, 2,... n only once.

Value

An object of class `shallot.permutation`.

Author(s)

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

References

Dahl, D. B., Day, R., and Tsai, J. (2017), Random Partition Distribution Indexed by Pairwise Information, *Journal of the American Statistical Association*, 112, 721-732. <DOI:10.1080/01621459.2016.1165103>

See Also

`attraction`

Examples

```
## Demonstrate permutation.
permutation(c(3, 1, 2, 5, 4))
permutation(c(3, 1, 2, 5, 4), fixed=FALSE)
permutation(n.items=5, fixed=FALSE)
```

`process.samples` *Process Sampled Partitions*

Description

This function extracts the partitions from the results of the `sample.partitions` function.

Usage

```
process.samples(x)
```

Arguments

`x`	An object from the `sample.partitions` function.

Details

This function extracts the sampled partitions from the results of the `sample.partitions` function.

Value

A list containing a matrix of cluster labels in which each row represents a clusterings. The list also contains sampled model parameters if `sample.parameter` is not `NULL`.

Author(s)

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

See Also

`sample.partitions`

Examples

```
## Not run:
example(shallot)

## End(Not run)
```

`sample.partitions` *Sample Partitions from Partition Distributions*

Description

This function samples partitions from the Ewens, Ewens-Pitman, Ewens attraction, Ewens-Pitman attraction, and ddCRP distributions.

Usage

```
sample.partitions(x, n.draws, parallel = TRUE)
```

Arguments

`x`	An object of class `shallot.distribution` obtained, for example, from the `ewens.pitman.attraction` function.
`n.draws`	An integer representing the desired number of samples. Due to parallelization, slightly more samples may be returned.
`parallel`	Should sampling be done in parallel by simultaneously using all CPU cores?

Value

An object of class `shallot.samples.raw` which can be subsequently be used in `process.samples`.

Note

If this function is interrupted by the user, the computation engine will be broken and subsequent calls to package functions may fail until a new session is started.

Author(s)

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

See Also

`partition.distribution`, `process.samples`

Examples

```
## Not run:
example(shallot)

## End(Not run)
```

variance.ratio *Variance Ratio*

Description

This function calculates the variance of the expected pairwise allocation matrix (EPAM) within clusters/subsets over the total variance of the expected pairwise allocation matrix.

Usage

```
variance.ratio(x, y)
```

Arguments

x, y	If y is not specified then x must be an object of class salso.confidence. Otherwise, x is a vector of cluster labels and y is an expected pairwise allocation matrix.

Details

The variance.ratio function takes as input an object of class salso.confidence and calculates the variance ratio for the estimated partition from the corresponding expected pairwise allocation matrix (EPAM).

The variance ratio is the weighted average of the within cluster variances of the EPAM, weighted by the number of pairwise EPAM values per cluster, over the total variance of the EPAM.

Value

A vector of variance ratios.

See Also

Other Default Mass Selection: default.mass(), mass.algorithm(), partition.confidence()

Examples

```
x <- rep(c(1,2,3), times=c(2,3,5))
y <- diag(10)
y[upper.tri(y)] <- runif(45)
variance.ratio(x,y)
```

Index