

Package ‘kernrank’

June 26, 2018

Title Kernel Methods for Rank Data

Version 1.1.0

URL <https://github.com/YunlongJiao/kernrank>

Description This package implements kernel functions and kernel methods for analyzing rank data, typically total rankings (or permutations), interleaving and top-k partial rankings, multivariate rankings. This package is built upon and is a stable and significant extension of the R CRAN package RMallow.

Depends R (>= 3.2)

LinkingTo Rcpp

Imports combinat,
kernlab,
Rcpp

Suggests pcaPP,
mvtnorm,
caret,
testthat

License GPL-3

LazyData true

RoxygenNote 6.0.1

R topics documented:

AllKendall	2
DistanceDistribution	3
kendall_partial	4
kendall_top	5
kendall_total	6
kendall_weight	6
KendallInfo	8
LogSumExp	9
Mallows	9
MallowsCV	11
RankAggreg	13
Index	15

AllKendall	<i>All Kendall's distances between two sets of total rankings or real-valued vectors</i>
------------	--

Description

Calculates all of the Kendall's distances between two sets of total rankings or real-valued vectors.

Usage

```
AllKendall(r, seqs = NULL, data.info = NULL, use.kernel.trick = FALSE,
           kmat = NULL, type = c("type-b", "type-a"), mc = 0.25)
```

Arguments

<code>r</code>	A vector or a matrix of m_1 sequences in rows and orders of n items in cols.
<code>seqs</code>	Another vector or a matrix of m_2 sequences in rows and orders of n items in cols. By default "seqs" is set equal to "r".
<code>data.info</code>	Optional argument giving the Kendall embedding of "r", that is the result of calling KendallInfo , to facilitate computing Kendall's difference for "r" to "seqs" without exploring the kernel trick.
<code>use.kernel.trick</code>	Logical. Indicator of whether the kernel trick is explored. This is particularly interesting when the number of items to be ranked is large ($m_1, m_2 \gg n$) and will use cor.fk for fast computation. By default (set FALSE), Kendall embedding is explicitly computed; otherwise kernel trick is explored.
<code>kmat</code>	Kendall kernel matrix of dimension $m_1 \times m_2$, correlation type corresponding to "type". If given, kernel trick is explored directly.
<code>type</code>	A character string indicating the type of Kendall correlation for "kmat".
<code>mc</code>	A normalization constant default to 0.25 such that output normalized squared Euclidean distance in the feature space induced by Kendall embedding amounts exactly to Kendall distances.

Value

A matrix of dimension $m_1 \times m_2$ where entry $[i, j]$ is the distance from sequence i in "r" to sequence j in "seqs".

Note

Kernel trick is explored in the sense that "r" and "seq" are only used for checking dimensions and getting attributes but not used explicitly to compute the distance. Option "use.kernel.trick" set TRUE or FALSE may give slightly different results due to computation precision of two implementations.

Author(s)

Yunlong Jiao

References

Kendall's tau rank correlation coefficient: https://en.wikipedia.org/wiki/Kendall_rank_correlation_coefficient

Yunlong Jiao, Jean-Philippe Vert. "The Kendall and Mallows Kernels for Permutations." IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), vol. 40, no. 7, pp. 1755-1769, 2018. DOI:10.1109/TPAMI.2017.2719680

See Also

[KendallInfo](#), [cor.fk](#)

Examples

```
#### Ex 1: Compute Kendall distance matrix and Mallows kernel matrix
data1 <- do.call("rbind", list(1:5, 5:1, c(3, 2, 1, 4, 5)))
data2 <- do.call("rbind", list(1:5, 5:1))

# Kendall distance matrix
s.K.d.mat <- AllKendall(data1, data2)

# Mallows kernel matrix with dispersion parameter lambda
lambda <- 0.1
M.k.kmat <- exp(-lambda * s.K.d.mat)

#### Ex 2: Why kernel trick?
r <- lapply(1:20, function(i) sample.int(1000, replace = TRUE))
r <- do.call('rbind', r)
dim(r)

# I) Without kernel trick
pt <- proc.time()
dmat1 <- AllKendall(r, use.kernel.trick = FALSE)
proc.time() - pt

# II) With kernel trick (should be much faster in this setting)
require(pcaPP)
pt <- proc.time()
dmat2 <- AllKendall(r, use.kernel.trick = TRUE)
proc.time() - pt

# NOTE: dmat1 and dmat2 may return slightly different values due to computation precision
isTRUE(all.equal(dmat1, dmat2, check.attributes = FALSE)) # May sometimes output FALSE
isTRUE(max(abs(dmat1 - dmat2)) < 1e-6) # Should always output TRUE
```

DistanceDistribution *Calculate the Kendall distance distribution in $N!$ space*

Description

This function counts the number of fully-ordered vectors at each distance in $N!$ space.

Usage

```
DistanceDistribution(N = 3)
```

Arguments

N Integer value, greater than or equal to 3.

Value

Table-like structure, where the names represent the distance from the modal sequence of each sequence in $N!$ space, and the values represent the number of sequences at that distance in the sequence space.

Note

Taken directly from R CRAN package [RMallow](#).

Author(s)

Erik Gregory

Examples

```
## Not run:

# DistanceDistribution(10)

## End(Not run)
```

kendall_partial

Kendall kernel for interleaving partial rankings

Description

Calculates Kendall kernel between interleaving partial rankings in time $O(k \log k)$, where ties (supposed few) are broken by adopting a convolution kernel averaging compatible rankings without ties.

Usage

```
kendall_partial(x, y)
```

Arguments

x, y Vector. If x is numeric, the rank vector converted from x indicate that larger values mean being preferred. NAs replace unobserved values.

Value

Kendall kernel for interleaving partial rankings, where ties (supposed few) are broken by adopting a convolution kernel averaging compatible rankings without ties.

Author(s)

Yunlong Jiao

References

Yunlong Jiao, Jean-Philippe Vert. "The Kendall and Mallows Kernels for Permutations." IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), vol. 40, no. 7, pp. 1755-1769, 2018. DOI:[10.1109/TPAMI.2017.2719680](https://doi.org/10.1109/TPAMI.2017.2719680)

Examples

```
x <- c(1.5, 0.1, NA, -4, NA)
y <- c(NA, NA, 0, 3, NA)
kendall_partial(x, y)
```

kendall_top

*Kendall kernel for top-k rankings***Description**

Calculates Kendall kernel between top-k rankings in time $O(k \log k)$, where ties (supposed few) are broken by adopting a convolution kernel averaging compatible rankings without ties.

Usage

```
kendall_top(x, y)
```

Arguments

x, y Vector. If x is numeric, the rank vector converted from x indicate that larger values mean being preferred. NAs replace unobserved values.

Value

Kendall kernel for top-k rankings, where ties (supposed few) are broken by adopting a convolution kernel averaging compatible rankings without ties.

Author(s)

Yunlong Jiao

References

Yunlong Jiao, Jean-Philippe Vert. "The Kendall and Mallows Kernels for Permutations." IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), vol. 40, no. 7, pp. 1755-1769, 2018. DOI:[10.1109/TPAMI.2017.2719680](https://doi.org/10.1109/TPAMI.2017.2719680)

Examples

```
x <- c(1.5, 0.1, NA, -4, NA)
y <- c(NA, NA, 0, 3, NA)
kendall_top(x, y)
```

kendall_total	<i>Kendall kernel for total rankings</i>
---------------	--

Description

Calculates Kendall kernel between total rankings in time $O(n \log n)$, where ties are dealt with by type-b soft version of Kendall tau correlation.

Usage

```
kendall_total(x, y)
```

Arguments

x, y	Vector. If x is numeric, the rank vector converted from x indicate that larger values mean being preferred. NAs are not allowed.
------	--

Value

Kendall kernel for total rankings, where ties are dealt with by type-b soft version of Kendall tau correlation.

References

Kendall rank correlation coefficient: https://en.wikipedia.org/wiki/Kendall_rank_correlation_coefficient

Yunlong Jiao, Jean-Philippe Vert. "The Kendall and Mallows Kernels for Permutations." IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), vol. 40, no. 7, pp. 1755-1769, 2018. DOI:10.1109/TPAMI.2017.2719680

Examples

```
x <- c(1.5, 0.1, 0, -4, 0)
y <- c(0, 0, 0, 3, 0)
kendall_total(x, y)
```

kendall_weight	<i>Weighted Kendall kernel for total rankings</i>
----------------	---

Description

Calculates weighted Kendall kernel between total rankings in time $O(n \log n)$, where ties (supposed few) are broken by adopting a convolution kernel averaging compatible rankings without ties.

Usage

```
kendall_weight(x, y, method = c("aken", "ken", "top", "add", "mult"),
  k = NULL, u = NULL, normalized = TRUE)
```

Arguments

<code>x, y</code>	Vector. If <code>x</code> is numeric, the rank vector converted from <code>x</code> indicate that larger values mean being preferred. NAs are not allowed.
<code>method</code>	Character. The method to perform weighted Kendall kernel. Choices include <ul style="list-style-type: none"> • <code>aken</code> denotes the Average Kendall kernel. • <code>ken</code> denotes the standard Kendall kernel. • <code>top</code> denotes the TOP-k Kendall kernel. • <code>add</code> denotes weighted Kendall kernel with ADDitive weights. • <code>mult</code> denotes weighted Kendall kernel with MULTiplicative weights.
<code>k</code>	Integer. The parameter in top-k Kendall kernel. Top-k implies ranks larger than <code>k</code> , where larger ranks mean being more preferred.
<code>u</code>	Numeric vector. The parameter in additive or multiplicative weighted Kendall kernel.
<code>normalized</code>	Logical. Whether to normalize the output kernel value. The weighted Kendall kernel elaborated in Jiao and Vert (2018) corresponds to the non-normalized version by setting <code>normalized=FALSE</code> .

Value

Weighted Kendall kernel for total rankings, where ties (supposed few) are broken by adopting a convolution kernel averaging compatible rankings without ties.

Author(s)

Yunlong Jiao

References

Yunlong Jiao, Jean-Philippe Vert. "The Weighted Kendall and High-order Kernels for Permutations." arXiv preprint arXiv:1802.08526, 2018. [arXiv:1802.08526](https://arxiv.org/abs/1802.08526)

Examples

```
x <- c(1.5, 0.1, 0, -4, 0)
y <- c(0, 0, 0, 3, 0)

# Average Kendall kernel
kendall_weight(x, y, method = "aken")

# Top-1 Kendall kernel is equiv to Standard Kendall kernel
kendall_weight(x, y, method = "ken")
kendall_weight(x, y, method = "top", k = 1)

# Additive/multiplicative weights with hyperbolic reduction factor
u <- 1/(length(x):1 + 1)
kendall_weight(x, y, method = "add", u = u)
kendall_weight(x, y, method = "mult", u = u)

# NOTE: In case of ties, standard Kendall kernel implemented by kendall_weight
# is NOT equiv to kendall_total due to different normalization technique!
# kendall_weight breaks ties by averaging over compatible full rankings:
kendall_weight(x, y, method = "ken", normalized = TRUE)
```

```
# kendall_total implements type-b of Kendall's tau that handles ties per se:
kendall_total(x, y)
```

KendallInfo

Kendall embedding of pairwise relative ordering

Description

Performs between-column comparison on a matrix of sequences denoting $\text{sign}([,i] - [,j])$ for $i < j$.

Usage

```
KendallInfo(r)
```

Arguments

r A vector or a matrix of dimension $N \times n$ with sequences in rows.

Value

A matrix of dimension $N \times \text{choose}(n, 2)$ with entry values $-1/1/0$ representing pairwise comparisons of vector values for each row. Specifically, a -1 value denotes that there is an increase between the two columns, 1 a decrease, and 0 indicates that the column values are identical in the same row.

Note

A matrix with one row is returned if the input "r" is a vector.

Author(s)

Yunlong Jiao

References

Yunlong Jiao, Jean-Philippe Vert. "The Kendall and Mallows Kernels for Permutations." IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), vol. 40, no. 7, pp. 1755-1769, 2018. DOI:[10.1109/TPAMI.2017.2719680](https://doi.org/10.1109/TPAMI.2017.2719680)

Examples

```
r <- do.call('rbind', combinat::permn(1:5))
KendallInfo(r)
```

LogSumExp

Computing Log-Sum-Exp with a common trick

Description

Computes log-sum-exp of a series of (typically large in absolute value) numbers with a more accurate computational trick typically useful for small values.

Usage

```
LogSumExp(x, byrow = TRUE, bycol = !byrow)
```

Arguments

x	A vector or a matrix of numerics (typicall very small).
byrow	Logical. Computes by rows if a matrix "x" is provided.
bycol	Logical. Computes by cols if a matrix "x" is provided.

Value

Log-Sum-Exp of the numbers in the vector "x", that is $\log(\sum(\exp(x)))$, or row-/column-wise Log-Sum-Exp of the numbers in a matrix "x".

Author(s)

Yunlong Jiao

References

Computing Log-Sum-Exp: <https://hips.seas.harvard.edu/blog/2013/01/09/computing-log-sum-exp/>

Examples

```
x <- c(-1000, -999, -1000)
LogSumExp(x)
```

Mallows

Fits a Mallows mixture model to ranking data

Description

Fits the Mallows mixture model to total rankings, using EM algorithm, for clustering permutations.

Usage

```
Mallows(datas, G, weights = NULL, iter = 100, iterin = iter,
  tol = 0.001, logsumexp.trick = TRUE, seed = 47631439,
  key = c("copelandMallows", "bruteMallows", "bordaMallows", "kernelMallows",
    "kernelMallows_Exh", "kernelGaussian", "copelandMallows_Eqlam",
    "bruteMallows_Eqlam", "bordaMallows_Eqlam", "kernelMallows_Eqlam",
    "kernelMallows_Exh_Eqlam", "kernelGaussian_Eqlam"), exhkey = "_Exh",
  eqlamkey = "_Eqlam")
```

Arguments

<code>datas</code>	Matrix of dimension $N \times n$ with sequences in rows.
<code>G</code>	Number of modes, 2 or greater.
<code>weights</code>	Numeric vector of length N denoting frequencies of each permutation observed. Each observation is observed once by default. Notably it must not contain 0 and should be of equal length with <code>nrow(datas)</code> .
<code>iter</code>	Maximum number of iterations for EM algorithm.
<code>iterin</code>	Maximum number of iterations for alternate optimization between centers and λ . Effective only when performing kernel Mallows with exhaustive optimization.
<code>tol</code>	Stopping precision.
<code>logsumexp.trick</code>	Logical. Whether or not to use log-sum-exp trick to compute log-likelihood.
<code>seed</code>	Seed index for reproducible results when optimization is performed. Set to <code>NULL</code> to disable the action.
<code>key</code>	A character string defining the type of Mallows mixture model to perform: <ul style="list-style-type: none"> • <code>copelandMallows</code> denotes original Mallows mixture model with cluster centers found by Copeland's method • <code>bruteMallows</code> denotes original Mallows mixture model with cluster centers found by brute-force search for optimal Kemeny consensus (not applicable for large n) • <code>bordaMallows</code> denotes original Mallows mixture model with cluster centers as Borda count • <code>kernelMallows</code> denotes kernel version of Mallows mixture model with cluster centers as the barycenter in Euclidean space induced by Kendall embedding • <code>kernelGaussian</code> denotes Gaussian mixture model in the Euclidean space induced by Kendall embedding
<code>exhkey</code>	DO NOT CHANGE. A character string. If it greps successfully in "key", an alternate optimization between centers and λ . Effective only when performing "kernelMallows" with exhaustive optimization.
<code>eqlamkey</code>	DO NOT CHANGE. A character string. If it greps successfully in "key", the dispersion parameters (or λ) are constrained to be equal for all clusters; otherwise no constraints on λ .

Value

List.

key	Character string indicating the type of Mallows mixture model performed
R	List of length "G" of cluster centers, each entry being a permutation of length "n" if original Mallows mixture model is performed, or a numeric vector of length $\text{choose}(n, 2)$ if kernel version is performed
p	Numeric vector of length "G" representing the proportion probability of each cluster
lambda	Numeric vector of length "G" representing the dispersion parameters of each cluster
datas	A copy of "datas" on which the Mallows mixture model is fitted, combined with "weights", fuzzy assignment membership probability "z", distances to centers in "R"
min.like	Numeric vector of length "iter" representing fitted likelihood values at each iteration

Author(s)

Yunlong Jiao

References

Thomas Brendan Murphy, Donal Martin. "Mixtures of distance-based models for ranking data." Computational Statistics & Data Analysis, vol. 41, no. 3, pp. 645-655, 2003. DOI:[10.1016/S0167-9473\(02\)00165-2](https://doi.org/10.1016/S0167-9473(02)00165-2)

Yunlong Jiao, Jean-Philippe Vert. "The Kendall and Mallows Kernels for Permutations." IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), vol. 40, no. 7, pp. 1755-1769, 2018. DOI:[10.1109/TPAMI.2017.2719680](https://doi.org/10.1109/TPAMI.2017.2719680)

See Also[MallowsCV](#)**Examples**

```

datas <- do.call('rbind', combinat::permn(1:5))
G <- 3
weights <- runif(nrow(datas))

# Fit Mallows mixture model
model <- Mallows(datas, G, weights, key = 'bordaMallows')
str(model)

```

MallowsCV

*Compute cross-validated likelihood for Mallows mixture models***Description**

Assess model performance by cross-validated (CV) Mallows likelihood. Do NOT run for large number of ranked alternatives "n".

Usage

```
MallowsCV(datas, G, weights = NULL, ..., seed = 26921332, nfolds = 5,
           nrepeats = 10, ntry = 3, logsumexp.trick = TRUE)
```

Arguments

<code>datas</code>	Matrix of dimension $N \times n$ with sequences in rows.
<code>G</code>	Number of modes, 2 or greater.
<code>weights</code>	Integer vector of length N denoting frequencies of each permutation observed. Each observation is observed once by default. Notably it must not contain 0 and should be of equal length with <code>nrow(datas)</code> .
<code>...</code>	Arguments passed to Mallows .
<code>seed</code>	Seed index for reproducible results when creating splits of data for CV. Set to <code>NULL</code> to disable the action.
<code>nfolds</code>	<code>nfolds</code> -fold CV created each time.
<code>nrepeats</code>	CV repeated <code>nrepeats</code> times.
<code>ntry</code>	Number of random initializations to restart for each CV run. The best fit returning max likelihood is reported.
<code>logsumexp.trick</code>	Logical. Whether or not to use log-sum-exp trick to compute log-likelihood.

Value

List of length `nfolds` \times `nrepeats`, each entry being the result on each fold containing:

<code>...</code>	See output of Mallows
<code>cv.loglik</code>	Likelihood value assessed against test fold while the mixture model is trained on the training fold

Note

CV split is done by partitioning "weights" so that "weights" must be integers.

Author(s)

Yunlong Jiao

References

Thomas Brendan Murphy, Donal Martin. "Mixtures of distance-based models for ranking data." Computational Statistics & Data Analysis, vol. 41, no. 3, pp. 645-655, 2003. DOI:[10.1016/S0167-9473\(02\)00165-2](#)

Yunlong Jiao, Jean-Philippe Vert. "The Kendall and Mallows Kernels for Permutations." IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), vol. 40, no. 7, pp. 1755-1769, 2018. DOI:[10.1109/TPAMI.2017.2719680](#)

See Also

[Mallows](#)

Examples

```

datas <- do.call('rbind', combinat::permn(1:5))
G <- 3
weights <- rbinom(nrow(datas), 100, 0.5) # positive integers

# Cross validate Mallows mixture model
cv.model <- MallowsCV(datas, G, weights, key = 'bordaMallows', nfolds = 3, nrepeats = 1)
# Averaged cv.loglik over all CV folds
mean(apply(cv.model, function(model) model$cv.loglik))

```

RankAggreg

*Common ranking aggregation methods for permutations***Description**

Used to update modal sequences of each cluster in the EM algorithm when fitting Mallows mixture models.

Usage

```
RankAggreg(r, z = NULL, infos = NULL, perm = NULL, key = c("borda",
  "copeland", "brute"))
```

Arguments

- | | |
|-------|---|
| r | A vector or a matrix of sequences in rows. |
| z | A vector of weights/frequencies of observations or a matrix of probability of cluster membership for each sequence and each cluster. Set by default a constant vector of 1. |
| infos | The result of calling KendallInfo . Optional for speeding up computation. |
| perm | A matrix of full permutations for brute-force search of optimal Kemeny consensus. Only effective when "key" equals to "brute". |
| key | A character string indicating the ranking aggregation method to find centers. <ul style="list-style-type: none"> • borda denotes the Borda count • copeland denotes the Copeland's aggregated ranking • brute denotes the optimal Kemeny consensus found by brute-force search |

Value

List of length 1 if "z" is a vector, or `ncol(z)` if "z" is a matrix, each entry being the modal sequence in each cluster.

Author(s)

Yunlong Jiao

See Also

[KendallInfo](#)

Examples

```
r <- do.call("rbind", list(1:5, 5:1, c(2,4,1,5,3)))  
RankAggreg(r, key = "borda") # Borda count for sequences in "r"
```

Index

*Topic **Clustering**

Mallows, [9](#)

MallowsCV, [11](#)

*Topic **Distance**

AllKendall, [2](#)

DistanceDistribution, [3](#)

*Topic **Embedding**

KendallInfo, [8](#)

*Topic **Kendall**

AllKendall, [2](#)

DistanceDistribution, [3](#)

kendall_partial, [4](#)

kendall_top, [5](#)

kendall_total, [6](#)

kendall_weight, [6](#)

KendallInfo, [8](#)

*Topic **Kernel**

kendall_partial, [4](#)

kendall_top, [5](#)

kendall_total, [6](#)

kendall_weight, [6](#)

*Topic **MallowsMixture**

Mallows, [9](#)

MallowsCV, [11](#)

*Topic **PartialRanking**

kendall_partial, [4](#)

*Topic **RankAggregation**

RankAggreg, [13](#)

*Topic **TopkRanking**

kendall_top, [5](#)

*Topic **TotalRanking**

AllKendall, [2](#)

kendall_total, [6](#)

kendall_weight, [6](#)

RankAggreg, [13](#)

*Topic **Weighted**

kendall_weight, [6](#)

AllKendall, [2](#)

cor.fk, [2](#), [3](#)

DistanceDistribution, [3](#)

kendall_partial, [4](#)

kendall_top, [5](#)

kendall_total, [6](#)

kendall_weight, [6](#)

KendallInfo, [2](#), [3](#), [8](#), [13](#)

LogSumExp, [9](#)

Mallows, [9](#), [12](#)

MallowsCV, [11](#), [11](#)

RankAggreg, [13](#)