## Package ‘TML’

**Type** Package  

**Title** Tropical Geometry Tools for Machine Learning  

**Version** 1.2.0  


**License** MIT + file LICENSE  

**Maintainer** David Barnhill <david.barnhill@nps.edu>  

**URL** [https://github.com/barnhilldave/TML](https://github.com/barnhilldave/TML)  

**BugReports** [https://github.com/barnhilldave/TML/issues](https://github.com/barnhilldave/TML/issues)  

**Depends** R (>= 3.5.0)  

**Imports** MASS, Matrix, RcppAlgos, Rfast, combinat, gtools, lpSolve, lpSolveAPI, magick, miscTools, phangorn, rcdd, rgl, ape, phytools, maps, cluster, ROCR, stringr, stats  

**Encoding** UTF-8  

**LazyData** true  

**RoxygenNote** 7.2.3  

**NeedsCompilation** no  

**Author** David Barnhill [aut, cre] (<https://orcid.org/0009-0006-2781-7434>), Ruriko Yoshida [aut], Georgios Aliatimis [aut], Keiji Miura [aut]  

**Repository** CRAN  

**Date/Publication** 2023-12-14 03:10:03 UTC
**R topics documented:**

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>bw.nn</td>
<td>3</td>
</tr>
<tr>
<td>cluster.ratio_HC</td>
<td>4</td>
</tr>
<tr>
<td>cluster.ratio_KM</td>
<td>5</td>
</tr>
<tr>
<td>convert.to.tree</td>
<td>6</td>
</tr>
<tr>
<td>draw.tpolytope</td>
<td>6</td>
</tr>
<tr>
<td>FWpoint.num.w.reg</td>
<td>7</td>
</tr>
<tr>
<td>FWpoint_numerical</td>
<td>8</td>
</tr>
<tr>
<td>HAR.TLineSeg</td>
<td>9</td>
</tr>
<tr>
<td>HAR.TLineSeg.Norm</td>
<td>10</td>
</tr>
<tr>
<td>hyper_3D</td>
<td>11</td>
</tr>
<tr>
<td>lung_fish</td>
<td>12</td>
</tr>
<tr>
<td>max_ins_ball</td>
<td>12</td>
</tr>
<tr>
<td>min_enc_ball</td>
<td>13</td>
</tr>
<tr>
<td>normaliz.tree</td>
<td>14</td>
</tr>
<tr>
<td>normalize</td>
<td>14</td>
</tr>
<tr>
<td>over_bet_HC</td>
<td>15</td>
</tr>
<tr>
<td>over_bet_KM</td>
<td>16</td>
</tr>
<tr>
<td>Points.TLineSeg</td>
<td>17</td>
</tr>
<tr>
<td>pre.pplot.pro</td>
<td>18</td>
</tr>
<tr>
<td>prob.class</td>
<td>19</td>
</tr>
<tr>
<td>project_pi</td>
<td>20</td>
</tr>
<tr>
<td>pw.trop.dist</td>
<td>20</td>
</tr>
<tr>
<td>rounding</td>
<td>21</td>
</tr>
<tr>
<td>sigmoid</td>
<td>22</td>
</tr>
<tr>
<td>Sim_points</td>
<td>23</td>
</tr>
<tr>
<td>Sim_Trees1</td>
<td>23</td>
</tr>
<tr>
<td>Sim_Trees2</td>
<td>24</td>
</tr>
<tr>
<td>tdets</td>
<td>25</td>
</tr>
<tr>
<td>TKmeans</td>
<td>25</td>
</tr>
<tr>
<td>TLineSeg</td>
<td>26</td>
</tr>
<tr>
<td>TLineSeg_min</td>
<td>27</td>
</tr>
<tr>
<td>TML</td>
<td>27</td>
</tr>
<tr>
<td>tree.to.vector</td>
<td>28</td>
</tr>
<tr>
<td>trop.dist</td>
<td>29</td>
</tr>
<tr>
<td>trop.dist.hyp_max</td>
<td>29</td>
</tr>
<tr>
<td>trop.dist.hyp_min</td>
<td>30</td>
</tr>
<tr>
<td>trop.logistic.regression</td>
<td>31</td>
</tr>
<tr>
<td>trop.tri.plot.w.pts</td>
<td>32</td>
</tr>
<tr>
<td>tropical.Gaussian</td>
<td>33</td>
</tr>
<tr>
<td>tropical.Gaussian.MH</td>
<td>34</td>
</tr>
<tr>
<td>Tropical.HC.AGNES</td>
<td>35</td>
</tr>
<tr>
<td>tropical.KDE</td>
<td>35</td>
</tr>
<tr>
<td>tropical.PCA</td>
<td>36</td>
</tr>
<tr>
<td>TropicalPolytope.extrapolation.HAR</td>
<td>38</td>
</tr>
<tr>
<td>trop_bal.vert</td>
<td>39</td>
</tr>
<tr>
<td>Trop_ball</td>
<td>40</td>
</tr>
</tbody>
</table>
bw.nn

Nearest neighbor bandwidth calculation

Description

This function finds the bandwidth for an ultrametric based on the tropical distance of the nearest point. The function provides the bandwidth input to trop.KDE and was originally used in the KDETrees package.

Usage

bw.nn(x, prop = 0.2, tol = 1e-06)

Arguments

- **x**
  matrix; dissimilarity matrix between points in a data set
- **prop**
  proportion of observations that defines neighborhood of a point
- **tol**
  tolerance for zero bandwidth check

Value

- a vector of bandwidths for each tree (row) in x

Author(s)

Ruriko Yoshida <ryoshida@nps.edu>

References


Examples

T1 <- Sim_Trees15
T2 <- Sim_Trees25
D <- rbind(T1, T2[1,])
M <- pw.trop.dist(D, D)
bw.nn(M)
cluster.ratio_HC

Ratio of within and between tropical measures for tropical hierarchical clusters

Description

Ratio of within and between cluster tropical measures for a set hierarchical clusters

Usage

cluster.ratio_HC(A, V, method = "avg")

Arguments

A       matrix of tropical points; rows are points
V       list of clusters where each cluster is defined as a matrix
method  method to use for within cluster measure; "avg" or "max"

Value

vector of ratios for each cluster

Author(s)

David Barnhill <david.barnhill@nps.edu>

References


Examples

har<-rbind(Sim_points[1:20,],Sim_points[51:70,])
V<-Tropical.HC.AGNES(har, method="average")
inds<-V[[2]][[38]]
cluster.ratio_HC(har,inds,method='avg')
cluster.ratio_KM

Description

Ratio of within and between cluster tropical measures for k-means derived clusters

Usage

cluster.ratio_KM(A, C, method = "avg")

Arguments

A matrix of tropical points; rows are points
C number of clusters
method method to use for within cluster measure; "avg" or "max"

Value

vector of ratios for each cluster

Author(s)

David Barnhill <david.barnhill@nps.edu>

References


Examples

hars<-Sim_points
cls<-c(rep(1,50),rep(2,50),rep(3,50))
cl_pt<-cbind(hars,cls)

C<-3
cluster.ratio_KM(cl_pt,C,method='avg')
convert.to.tree Create a phylogenetic tree from an ultrametric

Description
This function constructs a phylogenetic tree from an ultrametric.

Usage
convert.to.tree(n, L, u)

Arguments
- n: is the number of leaves
- L: is a vector of labels (strings) of leaves
- u: is an ultrametric

Value
A phylogenetic tree of class phylo

Author(s)
Ruriko Yoshida <ryoshida@nps.edu>

Examples
um<-Sim_Trees21[1,]
ll <- 10
L <- LETTERS[1:10]
tr<-convert.to.tree(ll, L, um)

draw.tpolytope Draw a 2-D or 3-D tropical polytope

Description
This command draws a three dimensional tropical polytope

Usage
draw.tpolytope.3d(D, c, cc, plt = TRUE)
draw.tpolytope.2d(D, c, cc, plt = TRUE)
Arguments

D  matrix of vertices of a tropical polytope; rows are the vertices

c  string; color to render the polytope.

cc  string; color to render the vertices.

plt  logical; initiate new plot visualization or not.

Value

2-D or 3-D rendering of a tropical polytope.

Author(s)

Ruriko Yoshida <ryoshida@nps.edu>

Examples

D <- matrix(c(0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1),4,4,TRUE)
c <- 'blue'
cc <- 'red'
draw.tpolytope.3d(D,c,cc,plt=TRUE)

D <- matrix(c(0,-2,2,0,-2,5,0,2,1,0,1,-1),4,3,TRUE)
c <- 'blue'
c <- 'red'
draw.tpolytope.2d(D,c,cc,plt=TRUE)
Arguments

- **datamatrix**: matrix of dimension $N^e$, where $N$ is the number of observations which lie in $\mathbb{R}^e$
- **penalty**: positive real number; the regularization rate

Value

- vector; Fermat-Weber point approximation

Author(s)

Georgios Aliatimis <g.aliatimis@lancaster.ac.uk>

References

Aliatimis, Georgios, Ruriko Yoshida, Burak Boyaci and James A. Grant (2023). Tropical Logistic Regression on Space of Phylogenetic Trees

Examples

```r
D = matrix(c(0,0,0,0,2,5,0,3,1),3,3,TRUE)
FWpoint.num.w.reg(D,1e4) # (0,2,5/3) not ultrametric
FWpoint.num.w.reg(D,1e4) # (0,5/3,5/3) ultrametric
```

Description

Returns the Fermat-Weber point of $N$ points using a gradient based numerical method

Usage

```r
FWpoint_numerical(datamatrix)
```

Arguments

- **datamatrix**: matrix of dimension $N^e$, where $N$ is the number of observations which lie in $\mathbb{R}^e$.

Value

- Fermat-Weber point approximation (vector in $\mathbb{R}^e$)

Author(s)

Georgios Aliatimis <g.aliatimis@lancaster.ac.uk>
### References

Aliatimis, Georgios, Ruriko Yoshida, Burak Boyaci and James A. Grant (2023). Tropical Logistic Regression on Space of Phylogenetic Trees

### Examples

```r
D = matrix(c(0,0,0,0,2,5,0,3,1),3,3,TRUE)
FWpoint_numerical(D)
```

---

**HAR.TLineSeg**

*Uniformly sample from a max-plus tropical line segment*

### Description

This function uses a hit-and-run sampler to uniformly sample from a max-plus tropical line segment

### Usage

```r
HAR.TLineSeg(D1, D2)
```

### Arguments

- **D1**
  - point in the tropical projective torus
- **D2**
  - point in the tropical projective torus

### Value

point on the line segment defined by D1 and D2

### Author(s)

Ruriko Yoshida <ryoshida@nps.edu>

### References


### Examples

```r
D1 <- c(0,4,2)
D2 <- c(0,7,-1)
HAR.TLineSeg(D1, D2)
```
Description

This function samples points on a tropical line segment about a location parameter for a given scale parameter defined in terms of tropical distance.

Usage

```
HAR.TLineSeg.Norm(D1, D2, mu, stdev)
```

Arguments

- **D1**: point in the tropical projective torus
- **D2**: point in the tropical projective torus
- **mu**: location parameter
- **stdev**: scale parameter

Value

point on the line segment defined by D1 and D2 sampled about mu

Author(s)

David Barnhill <david.barnhill@nps.edu>

Examples

```
D1 <- c(0, 4, 2)
D2 <- c(0, 7, -1)
mu <- c(0, 7, 2)
stdev <- 1
HAR.TLineSeg.Norm(D1, D2, mu, stdev)
```
hyper_3D

2D or 3D rendering of max-plus or min-plus tropical hyperplane

Description
This function renders a 2D or 3D max-plus or min-plus tropical hyperplane

Usage

hyper3d_max(D, di, mi, ma, plt = FALSE)

hyper3d_min(D, di, mi, ma, plt = FALSE)

Arguments

D point in the tropical projective torus representing the apex of the hyperplane
di scalar; indicates how far the hyperplane should extend
mi scalar; minimum value on axes of the plot
ma scalar; maximum value on axes of the plot
plt logical; if true produces a new plot otherwise overlays tropical hyperplane on existing plot

Value

2D or 3D rendering of max-plus or min-plus tropical hyperplane

Author(s)

David Barnhill <david.barnhill@nps.edu>

Examples

# 2D Example
D <- t(as.matrix(c(0,0,0)))
di<-4
mi<- -5
ma<-5
hyper3d_max(D,di,mi,ma,plt=TRUE)
hyper3d_min(D,di,mi,ma,plt=TRUE)

# 3D Example
D <- t(as.matrix(c(0,0,0,0)))
di<-4
mi<- -5
ma<-5
hyper3d_max(D,di,mi,ma,plt=TRUE)
hyper3d_min(D,di,mi,ma,plt=TRUE)
**lung_fish**  
*Phylogenetic trees based on lung fish data*

**Description**

1290 (non-equidistant) gene trees with 45 leaves originating from lung fish data in matrix form. Also we provide a vector of strings consisting of leaf labels for each species associated with the data set.

**Usage**

```r
lung_fish
lf_labels
```

**Format**

An object of class `matrix` (inherits from `array`) with 1290 rows and 45 columns.
An object of class `character` of length 10.

**Source**


**max_ins_ball**  
*Calculate the center point and radius of the maximum inscribed ball for a tropical simplex*

**Description**

This function calculates the center point and radius of the maximum inscribed ball for a tropical simplex

**Usage**

```r
max_ins_ball(A)
```

**Arguments**

- `A`  
  matrix of points defining a tropical polytope; rows are the points

**Value**

list containing the radius and center point of a maximum inscribed ball
Author(s)
David Barnhill <david.barnhill@nps.edu>

References

Examples
P <- matrix(c(0,0,0,2,5,0,3,1),3,3,TRUE)
min_enc_ball(P)

Description
This function constructs a minimum enclosing ball for a set of points defining a tropical polytope.

Usage
min_enc_ball(A)

Arguments
A matrix of points defining a tropical polytope. Rows are the points.

Value
list containing center point and radius of minimum enclosing ball of P

Author(s)
David Barnhill <david.barnhill@nps.edu>

References

Examples
P <- matrix(c(0,0,0,3,1,0,2,5),3,3,TRUE)
min_enc_ball(P)
normalize

---

**normalize.tree**  
Normalize a phylogenetic tree

**Description**  
This function normalizes the height of a phylogenetic tree

**Usage**  
```
normalize.tree(D, h = 1)
```

**Arguments**  
- `D`: numeric vector; ultrametric equidistant tree
- `h`: desired height; defaults to 1

**Value**  
normalized equidistant tree

**Author(s)**  
Ruriko Yoshida <ryoshida@nps.edu>

**Examples**
```
D <- c(4, 4, 2)
normalize.tree(D, h = 1)
```

---

**normalize**  
Normalize a point or set of points in the tropical projective torus

**Description**  
This function normalizes a point or set of points in the tropical projective torus by making the first coordinate zero

**Usage**
```
normalize.vector(D)
normalize.vectors(D)
normalize.polytope(D)
normalize.ultrametrics(D)
```
over_bet_HC

Arguments

D numeric vector in the tropical projective torus or a matrix of points in the tropical projective torus; for matrices, rows are the points

Value

a single or set of normalized points with the first coordinate zero

Author(s)

Ruriko Yoshida <ryoshida@nps.edu>

Examples

D <-c(8,4,2)
normaliz.vector(D)
P <-matrix(c(8,4,2,10,1,3,7,2,1),3,3,TRUE)
normaliz.vectors(P)
M<-matrix(c(2,2,2,3,6,4,2,4,7),3,3,TRUE)
normaliz.polytope(M)
M <- Sim_Trees15[1:3,]
normaliz.ultrametrics(M)

over_bet_HC  Tropical cluster betweenness measure for each cluster in a set of hierarchical clusters

Description

This function calculates an overall betweenness measure based on tropical distance between a set of clusters derived from tropical hierarchical clustering

Usage

over_bet_HC(A, V)

Arguments

A matrix of tropical points; rows are points with the last column representing a numbered cluster assignment

V list of clusters defined as matrices derived from agglomerative or divisive hierarchical clustering
over_bet_KM

**Value**

vector of betweenness cluster measures

**Author(s)**

David Barnhill <david.barnhill@nps.edu>

**References**


**Examples**

```r
har<-rbind(Sim_points[1:20,],Sim_points[51:70,])
V<-Tropical.HC.AGNES(har, method="average")
inds<-V[[2]][[38]]
over_bet_HC(har,inds)
```

**Description**

This function calculates an overall betweenness measure between a set of clusters derived from tropical k-means clustering.

**Usage**

```r
over_bet_KM(A, C)
```

**Arguments**

- `A`: matrix of tropical points; rows are points with the last column representing a numbered cluster assignment
- `C`: number of clusters

**Value**

betweenness cluster measure

**Author(s)**

David Barnhill <david.barnhill@nps.edu>

**References**

Points.TLineSeg

Examples

hars<-Sim_points
cls<-c(rep(1,50),rep(2,50),rep(3,50))
cl_pt<-cbind(hars,cls)

C<-3
over_bet_KM(cl_pt,C)

Points.TLineSeg Samples k equally spaced points on a max-plus tropical line segment

Description
This function calculates k equally spaced points on a tropical line segment

Usage
Points.TLineSeg(D1, D2, k = 20)

Arguments
D1 point in the tropical projective torus
D2 point in the tropical projective torus
k number of points

Value
matrix of k equally spaced points on a tropical line segment

Author(s)
Ruriko Yoshida <ryoshida@nps.edu>

Examples
D1 <-c(0,4,2)
D2 <- c(0,7,-1)
p<-Points.TLineSeg(D1, D2, k = 5)
pre.pplot.pro  

*Projections of points onto a tropical triangle*

**Description**

This function produces a matrix of points projected onto a tropical triangle defined by the column space of a matrix.

**Usage**

```r
pre.pplot.pro(S, D)
```

**Arguments**

- `S`: matrix of points representing a tropical polytope; rows are the vertices
- `D`: data points in the tropical projective torus

**Value**

matrix of points representing projections of the points in `D` (row vectors) onto `S`

**Author(s)**

Ruriko Yoshida <ryoshida@nps.edu>

**Examples**

```r
s <- 3  # number of vertices. Here it is a tropical triangle
d <- 3  # dimension
N <- 100  # sample size
D <- matrix(rep(0, N*d), N, d)
D[, 1] <- rnorm(N, mean = 5, sd = 5)
D[, 2] <- rnorm(N, mean = -5, sd = 5)
D[, 3] <- rnorm(N, mean = 0, sd = 5)

index <- sample(1:N, s)
S <- D[index,]

DD <- pre.pplot.pro(S, D)
```
prob.class

Estimated probability for binary class assignment

Description

Estimates the probability that an observation x belongs to class 1.

Usage

prob.class(pars, x)

Arguments

- pars: vector of parameters, which can be decomposed as two normal vectors and two scaling parameters and has dimension 2*e+2
- x: vector of dimension e

Value

real number

Author(s)

Georgios Aliatimis <g.aliatimis@lancaster.ac.uk>

References

Aliatimis, Georgios, Ruriko Yoshida, Burak Boyaci and James A. Grant (2023). Tropical Logistic Regression on Space of Phylogenetic Trees

Examples

```r
library(ROCR)
T0 = Sim_Trees15
T1 = Sim_Trees25
D = rbind(T0,T1)
Y = c(rep(0,dim(T0)[1]), rep(1,dim(T1)[1]))
N = length(Y)
set.seed(1)
train_set = sample(N,floor(0.8 * N)) ## 80/20 train-test split
pars <- trop.logistic.regression(D[train_set,],Y[train_set], penalty=1e4)
test_set = (1:N)[-train_set]
Y.hat <- rep(0, length(test_set))
for(i in 1:length(test_set)) Y.hat[i] <- prob.class(pars, D[test_set[i],])
Logit.ROC <- performance(prediction(Y.hat, Y[test_set]), measure="tpr", x.measure="fpr")
plot(Logit.ROC, lwd = 2, main = "ROC Curve for Logistic Regression Model")
print(paste("Logit.AUC=", performance(prediction(Y.hat, Y[test_set]), measure="auc")@y.values))
```
project_pi  Project a point on the tropical projective torus onto a tropical polytope.

Description
This function projects points in the tropical projective torus onto a tropical polytope based on tropical distance.

Usage
project_pi(D_s, D)

Arguments

D_s  matrix where each row is a point defining a tropical polytope
D  point to be projected onto D_s

Value
projection of point D onto the tropical polytope defined by D_s

Author(s)
David Barnhill <david.barnhill@nps.edu>

Examples

D_s <- matrix(c(0,0,0,0,2,5,0,3,1),3,3,TRUE)
D <- c(0,7,-1)
project_pi(D_s,D)

pw.trop.dist  Constructs the dissimilarity matrix for a set of ultrametrics

Description
Constructs the dissimilarity matrix based on the tropical distance between points in a dataset

Usage
pw.trop.dist(D1, D2)

Arguments

D1  matrix of ultrametrics
D2  matrix of ultrametrics
rounding

Value
matrix; dissimilarity matrix showing the tropical pairwise distance between each point

Author(s)
Ruriko Yoshida <ryoshida@nps.edu>

References

Examples
T1<-Sim_Trees15
T2<-Sim_Trees25
D <- rbind(T1, T2[1,])
pw.trop.dist(D, D)

rounding

Remove all tentacles from a tropical simplex

Description
This function removes all tentacles from a tropical simplex. The remaining portion is a full-dimensional tropical polytope known as the trunk of the tropical polytope.

Usage
rounding(P)

Arguments
\( P \) matrix of points defining a tropical simplex. Rows are the points

Value
matrix of points defining only the full-dimensional element (the trunk) of a tropical polytope; rows are points

Author(s)
David Barnhill <david.barnhill@nps.edu>
References


Examples

P <- matrix(c(0,-1,1,0,0,0,0,1,-1),3,3,TRUE)
BP <- min_enc_ball(P)
RP <- rounding(P)
BRP <- min_enc_ball(RP)

sigmoid

Sigmoid function

Description

Returns the sigmoid function valuation

Usage

sigmoid(x)

Arguments

x real number

Value

sigmoid function value at x

Author(s)

Georgios Aliatimis <g.aliatimis@lancaster.ac.uk>

References

Aliatimis, Georgios, Ruriko Yoshida, Burak Boyaci and James A. Grant (2023). Tropical Logistic Regression on Space of Phylogenetic Trees

Examples

sigmoid(0) # 0.5
Sim**_points**

**Simulated points over the tropical projective torus**

**Description**

150 points generated using Gaussian-like Hit-and-Run sampling with three separate pairs of location and scale parameters

**Usage**

Sim**_points**

**Format**

Sim**_points**:  
A 150 x 3 matrix where each row is a point in the tropical projective torus

---

Sim**_Trees1**

**Six data sets of phylogenetic trees data simulated from the Coalescent model.**

**Description**

Six data sets of 1000 gene trees simulated from the Coalescent model based on a specified species with each data set possessing a ratio of species depth to effective population of 0.25, 0.5, 1, 2, 5, and 10.

**Usage**

Sim**_Trees1025**  
Sim**_Trees105**  
Sim**_Trees11**  
Sim**_Trees12**  
Sim**_Trees15**  
Sim**_Trees110**
Sim_Trees2

Format
An object of class matrix (inherits from array) with 1000 rows and 45 columns. An object of class matrix (inherits from array) with 1000 rows and 45 columns. An object of class matrix (inherits from array) with 1000 rows and 45 columns. An object of class matrix (inherits from array) with 1000 rows and 45 columns. An object of class matrix (inherits from array) with 1000 rows and 45 columns.

Sim_Trees2

Six data sets of phylogenetic trees data simulated from the Coalescent model.

Description
Six data sets of 1000 gene trees simulated from the Coalescent model based on a specified species with each data set possessing a ratio of species depth to effective population of 0.25, 0.5, 1, 2, 5, and 10.

Usage
Sim_Trees2025
Sim_Trees205
Sim_Trees21
Sim_Trees22
Sim_Trees25
Sim_Trees210

Format
An object of class matrix (inherits from array) with 1000 rows and 45 columns. An object of class matrix (inherits from array) with 1000 rows and 45 columns. An object of class matrix (inherits from array) with 1000 rows and 45 columns. An object of class matrix (inherits from array) with 1000 rows and 45 columns. An object of class matrix (inherits from array) with 1000 rows and 45 columns.
tdets

Calculate the tropical determinant of a square matrix.

Description
This function calculates the tropical determinant (or singularity) of a square matrix

Usage
tdets(P)

Arguments
P matrix of points defining a tropical polytope. Rows are the points

Value
list containing the value of the determinant and reordered matrix P

Author(s)
David Barnhill <david.barnhill@nps.edu>

Examples
P<-matrix(c(0,0,0,0,2,5,0,3,1),3,3,TRUE)
tdets(P)

TKmeans
K-means clustering over the tropical projective torus

Description
This function performs k-means clustering over the tropical projective torus

Usage
TKmeans(A, C, M)

Arguments
A matrix of points defining a tropical polytope; rows are the tropical points
C number of clusters
M maximum number of iterations of algorithm to find cluster centroids
TLineSeg

Construct a max-plus tropical line segment between two points

Description

This function constructs a max-plus tropical line segment between two points

Usage

TLineSeg(D1, D2)

Arguments

D1 point in the tropical projective torus
D2 point in the tropical projective torus

Value

list of points defining the tropical line segment

Author(s)

Ruriko Yoshida <ryoshida@nps.edu>
TLineSeg_min

Examples

D1 <- c(0, 4, 2)
D2 <- c(0, 7, -1)
TLineSeg(D1, D2)

Description

This function constructs a min-plus tropical line segment between two points.

Usage

TLineSeg_min(D1, D2)

Arguments

D1 point in the tropical projective torus
D2 point in the tropical projective torus

Value

list of points defining the tropical line segment

Author(s)

David Barnhill <david.barnhill@nps.edu>

Examples

D1 <- c(0, 4, 2)
D2 <- c(0, 7, -1)
TLineSeg_min(D1, D2)

TML

Tropical Machine Learning in R

Description

TML provides a suite of tools for machine learning application on data over the tropical semiring
tree.to.vector

Phylogenetic tree to vector

Description

A tree is converted to a vector of pairwise distances between leaves. Distance between leaves is defined as the cophenetic distance between them. Normalization is applied so that the maximum distance in the vector output is 1.

Usage

tree.to.vector(tree, normalization = TRUE)

Arguments

- tree: phylogenetic tree
- normalization: logical; normalize the tree if TRUE

Value

vector of pairwise distances in $\mathbb{R}^{m \choose 2}$, where $m$ is the number of leaves

Author(s)

Georgios Aliatimis <g.aliatimis@lancaster.ac.uk>

References

Aliatimis, Georgios, Ruriko Yoshida, Burak Boyaci, James A. Grant (2023). Tropical Logistic Regression on Space of Phylogenetic Trees

Examples

tree <- ape::read.tree(text='((A:1, B:1):2, (C:1, D:1):2);')
tree.to.vector(tree)
**trop.dist**

*Compute the tropical distance*

**Description**

This function computes the tropical distance between two points in the tropical projective torus.

**Usage**

```r
trop.dist(D1, D2)
```

**Arguments**

- **D1**: point in the tropical projective torus
- **D2**: point in the tropical projective torus

**Value**

tropical distance between D1 and D2

**Author(s)**

Ruriko Yoshida <ryoshida@nps.edu>

**Examples**

```r
D1 <- c(0, 4, 2)
D2 <- c(0, 7, -1)
trop.dist(D1, D2)
```

---

**trop.dist.hyp_max**

*Calculate the tropical distance to a max-tropical hyperplane*

**Description**

Calculate the tropical distance to a max-tropical hyperplane.

**Usage**

```r
-trop.dist.hyp_max(0, x0)
```

**Arguments**

- **0**: normal vector of a tropical hyperplane; numeric vector
- **x0**: point of interest; numeric vector
trop.dist.hyp_min

Value
tropical distance to max-plus tropical hyperplane

Author(s)
David Barnhill <david.barnhill@nps.edu>

Examples

```r
O <- c(0,-1,-1)
x0 <- c(0,-2,-8)
trop.dist.hyp_min(O,x0)
```

Description
Calculate the tropical distance to a min-tropical hyperplane

Usage
trop.dist.hyp_min(O, x0)

Arguments
- `O`: normal vector of a tropical hyperplane; numeric vector
- `x0`: point of interest; numeric vector

Value
tropical distance to min-plus tropical hyperplane

Author(s)
David Barnhill <david.barnhill@nps.edu>

Examples

```r
O <- c(0,-1,-1)
x0 <- c(0,-2,-8)
trop.dist.hyp_min(O,x0)
```
trop.logistic.regression

Tropical Logistic Regression

Description
Performs tropical logistic regression, by finding the optimal statistical parameters for the training dataset \((D, Y)\), where \(D\) is the matrix of covariates and \(Y\) is the binary response vector.

Usage
```
trop.logistic.regression(D, Y, penalty = 0, model_type = "two_species")
```

Arguments
- **D**: matrix of dimension \(N \times e\), where \(N\) is the number of observations which lie in \(\mathbb{R}^e\).
- **Y**: binary vector with dimension \(N\), with each component corresponding to an observation.
- **penalty**: scalar; positive real number.
- **model_type**: string; options are "two-species" (default), "one-species", "general".

Value
vector; optimal model parameters (two normal vectors and two scaling factors).

Author(s)
Georgios Aliatimis <g.aliatimis@lancaster.ac.uk>

References

Examples
```
library(ROCR)
T0 = Sim_Trees15
T1 = Sim_Trees25
D = rbind(T0,T1)
Y = c(rep(0,dim(T0)[1]), rep(1,dim(T1)[1]))
N = length(Y)
set.seed(1)
train_set = sample(N,floor(0.8 * N)) ## 80/20 train-test split
pars <- trop.logistic.regression(D[train_set,],Y[train_set], penalty=1e4)
test_set = (1:N)[-train_set]
```
```r
Y.hat <- rep(0, length(test_set))
for(i in 1:length(test_set)) Y.hat[i] <- prob.class(pars, D[test_set[i],])
Logit.ROC <- performance(prediction(Y.hat, Y[test_set]), measure="tpr", x.measure="fpr")
plot(Logit.ROC, lwd = 2, main = "ROC Curve for Logistic Regression Model")
print(paste("Logit.AUC=" , performance(prediction(Y.hat, Y[test_set]), measure="auc")@y.values))
```

trop.tri.plot.w.pts

**Plotting PCA-derived tropical triangles**

**Description**

This function conducts tropical PCA to find the best fit tropical triangle given data defined in the tropical projective torus. It employs the vertex HAR with extrapolation sampler to sample points to determine the vertices of the tropical triangle.

**Usage**

trop.tri.plot.w.pts(S, D)

**Arguments**

- **S**: initial set of vertices for the tropical triangle
- **D**: matrix of data where each row is an observation in the tropical projective torus

**Value**

rendering of tropical triangle saved to current directory

**Author(s)**

Ruriko Yoshida <ryoshida@nps.edu>

**Examples**

```r
s <- 3  # number of vertices. Here it is a tropical triangle
d <- 3  ## dimension
N <- 100 ## sample size
V <- matrix(c(100, 0, 0, 0, 100, 0, 0, 100, 0, 0, 100, -100, 0, 0, 0, -100, 0, 0, -100), 6, 3, TRUE)
D <- matrix(rep(0, N*d), N, d)
D[, 1] <- rnorm(N, mean = 5, sd = 5)
D[, 2] <- rnorm(N, mean = -5, sd = 5)
D[, 3] <- rnorm(N, mean = 0, sd = 5)
index <- sample(1:N, s)
S <- D[index,]
res <- tropical.PCA.Polytope(S, D, V, I = 1000, 50)
DD <- pre.pplot.pro(res[[2]], res[[3]])
trop.tri.plot.w.pts(normaliz.ultrametrics(res[[2]]), DD)
```
tropical.Gaussian  Tropical Gaussian sampling about a center of mass

Description

This function is a Gaussian-like HAR sampler about a center of mass denoted by a location parameter with scale parameter in terms of the tropical distance.

Usage

\[
tropical.gaussian(D_s, x0, I = 1, M, S)
\]

Arguments

- \(D_s\): matrix of vertices of a tropical simplex; each row is a vertex
- \(x0\): initial point for sampler, numeric vector
- \(I\): number of states in Markov chain
- \(M\): location parameter; numeric vector indicating centroid
- \(S\): scale parameter; in terms of tropical distance

Value

next sampled point from the tropical polytope

Author(s)

David Barnhill <david.barnhill@nps.edu>

Examples

\[
D_s <- \text{matrix(c(0,0,0,10,0,0,0,10),3,3,TRUE)}
\]
\[
x0 <- c(0,0,0)
\]
\[
M <- c(0,5,5)
\]
\[
S <- 2
\]
\[
tropical.gaussian(D_s, x0, I = 50,M,S)
\]
tropical.Gaussian.MH  Gaussian-like sampling using Metropolis filter

Description
This function samples points on a tropical line segment about a location parameter for a given scale parameter defined in terms of tropical distance.

Usage

trop.Gaussian.MH(D, x0, mu, s, n, I = 50)
trop.Gaussian.MH.square(D, x0, mu, s, n, I = 50)

Arguments

D  matrix of vertices of a tropical polytope; each row is a vertex
x0  initial point for sampler, numeric vector
mu  location parameter; numeric vector
s  scale parameter; scalar
n  number of points to sample
I  states in Markov chain

Value
matrix of n sampled points where each point is a row

Author(s)
David Barnhill <david.barnhill@nps.edu>

References

Examples

D <- matrix(c(0,0,0,0,10,0,0,0,10),3,3,TRUE)
x0 <- c(0,0,0)
mu <- c(0,5,5)
s <- 1
n <- 10
trop.Gaussian.MH(D, x0, mu, s, n, I = 50)
trop.Gaussian.MH.square(D, x0, mu, s, n, I = 50)
Tropical.HC.AGNES

**Description**
This function performs agglomerative (AGNES) hierarchical clustering over the space of ultrametrics defining the space of equidistant trees.

**Usage**
Tropical.HC.AGNES(D, method = "average")

**Arguments**
- `D`: matrix of points defining a tropical polytope. Rows are the tropical points
- `method`: linkage method: "average", "min", or "max"

**Value**
list of distances in when merges occur; list of indices of points in each cluster

**Author(s)**
David Barnhill <david.barnhill@nps.edu>

**References**

**Examples**
P <- Sim_points
Tropical.HC.AGNES(P, method="average")

tropical.KDE

**Description**
This function calculates a non-parametric density estimate of a tree over the space of phylogenetic trees on m leaves. It mimics classical kernel density estimation by using a Gaussian kernel in conjunction with tropical distance.

**Usage**
tropical.KDE(D, n, sigma, h = 2)
tropical.PCA

Tropical principal component analysis (PCA) on over tropical projective torus

Description

This function conducts tropical PCA to find the best fit tropical triangle given data defined in the tropical projective torus. It employs the vertex HAR with extrapolation sampler to sample points to determine the vertices of the tropical triangle.

Usage

tropical.PCA.Polytope(S, D, V, I = 1, k)
tropical.PCA

Arguments

- **S**: initial set of vertices for the tropical triangle
- **D**: matrix of data where each row is an observation in the tropical projective torus
- **V**: matrix of vertices defining a polytope encompassing D
- **I**: number of iterations to perform
- **k**: number of iterations for the HAR sampler

Value

- list with the sum of residuals

Author(s)

Ruriko Yoshida <ryoshida@nps.edu>

References

- Page, Robert and others (2020), Tropical principal component analysis on the space of phylogenetic trees, Bioinformatics, Volume 36, Issue 17, Pages 4590–4598.

Examples

```r
s <- 3  # number of vertices. Here it is a tropical triangle
d <- 3  # dimension
N <- 100  # sample size
V <- matrix(c(100, 0, 0, 0, 0, 100, 0, 0, 0, 100, -100, 0, 0, 0, -100, 0, 0, 0, -100), 6, 3, TRUE)
D <- matrix(rep(0, N*d), N, d)
D[, 1] <- rnorm(N, mean = 5, sd = 5)
D[, 2] <- rnorm(N, mean = -5, sd = 5)
D[, 3] <- rnorm(N, mean = 0, sd = 5)
index <- sample(1:N, s)
S <- D[index,]
DD <- pre.pplot.pro(S, D)
for(i in 1:N)
    DD[i, ] <- normaliz.vector(DD[i, ])
res <- tropical.PCA.Polytope(S, D, V, I = 1000, 50)
DD <- pre.pplot.pro(res[[2]], res[[3]])
trop.tri.plot.w.pts(normaliz.ultrametrics(res[[2]]), DD)
```
TropicalPolytope.extrapolation.HAR

Description

This function samples points uniformly the space defined by a tropical simplex

Usage

TropicalPolytope.extrapolation.HAR(D_s, x0, I = 1)

Arguments

D_s  
matrix of vertices of a tropical simplex; each row is a vertex

x0  
initial point for sampler, numeric vector

I  
number of states in Markov chain

Value

next sampled point from the tropical polytope

Author(s)

David Barnhill <david.barnhill@nps.edu>

References


Examples

D_s <- matrix(c(0,0,0,10,0,0,0,10),3,3,TRUE)
x0 <- c(0,0,0)
TropicalPolytope.extrapolation.HAR(D_s, x0, I = 50)
### trop_bal.vert

**Calculate the minimum generating vertex set of a tropical ball**

**Description**

This function calculates the coordinates of the minimum or entire vertex set of a tropical ball given a center point.

**Usage**

```r
trop_bal.vert(x, d, al = FALSE)
```

**Arguments**

- `x`: matrix where each row is a point defining a tropical polytope
- `d`: radius of the tropical ball in terms of tropical distance
- `al`: logical; TRUE or FALSE to determine whether to enumerate all vertices of the tropical ball

**Value**

matrix of normalized tropical points defining the tropical ball. Rows are the points

**Author(s)**

David Barnhill <david.barnhill@nps.edu>

**References**


**Examples**

```r
x <- c(0, 3, 7, 5)
d <- 2
trop_bal.vert(x, d)
trop_bal.vert(x, d, al = TRUE)
```
**Trop_ball**

*Visualize a Tropical ball in 2D or 3D*

**Description**

This function constructs a visualization of a 2D or 3D tropical ball.

**Usage**

```r
Trop_ball(v, d, a = 1, cls = "black", cent.col = "black", fil = TRUE, plt = TRUE, bord = "black")
```

**Arguments**

- `v`: center of tropical ball; numeric vector of length 3 or 4
- `d`: radius of tropical ball
- `a`: shading level; 1 is opaque
- `cls`: string indicating color of interior of ball
- `cent.col`: string indicating color of center point
- `fil`: logical for 3D plots; if TRUE 2D facets of 3D ball fill in color of cls parameter
- `plt`: logical; indicates plot a new object; defaults to TRUE; if FALSE, overlays the ball on existing plot
- `bord`: string indicating color of border of ball (only for 2D plots)

**Value**

2D or 3D visualization of tropical ball

**Author(s)**

David Barnhill <david.barnhill@nps.edu>
Examples

\begin{verbatim}
v <-c(0,0,0)
d <- 2
Trop_ball(v,d,a=.1,cls='white',cent.col='black',fil=TRUE,plt=TRUE,bord='black')
v <-c(0,0,0,0)
d <- 2
Trop_ball(v,d,a=1,cls='red',cent.col='black',fil=FALSE,plt=TRUE)
\end{verbatim}

Description

This function calculates the Fermat-Weber point for a tropical polytope

Usage

\texttt{Trop_FW(A)}

Arguments

\begin{itemize}
\item \texttt{A} \hspace{1cm} matrix with normalized tropical points as rows
\end{itemize}

Value

numeric vector providing the tropical Fermat-Weber point for the tropical polytope

Author(s)

David Barnhill <david.barnhill@nps.edu>

References


Examples

\begin{verbatim}
P <-matrix(c(0,0,0,0,2,5,0,3),3,3,TRUE)
Trop_FW(P)
\end{verbatim}
Trop_Volume  

*Estimate the volume of a tropical polytope*

**Description**

This function uses tropical HAR with a uniform target distribution to estimate the volume of a tropical polytope.

**Usage**

```
Trop_Volume(B, P, x0, S, I, R)
```

**Arguments**

- **B**: matrix of points defining a minimum enclosing ball for a polytope P; rows are the points.
- **P**: matrix of points defining a tropical polytope; rows are the points.
- **x0**: initial point used for the HAR sampler.
- **S**: number of points to sample from the minimum enclosing ball.
- **I**: number of iterations for the HAR sampler.
- **R**: radius of the minimum enclosing tropical ball.

**Value**

list containing ratio of points falling in P; volume of the tropical ball; volume estimate of P

**Author(s)**

David Barnhill <david.barnhill@nps.edu>

**References**


**Examples**

```
P <- matrix(c(0,0,0,3,1,0,2,5),3,3,TRUE)  
BR<-min_enc_ball(P)  
B<-trop_bal.vert(BR[[1]],BR[[2]])  
x0<-c(0, 1.5, .4)  
S<-200  
I<-50  
R<-BR[[2]]  
Trop_Volume(B,P,x0,S,I,R)
```
trop_wi_dist

**Description**

This function calculates a within cluster measure by measuring the pairwise tropical distance between points in the cluster.

**Usage**

```r
trop_wi_dist(D1, method = "avg")
```

**Arguments**

- `D1`: matrix of tropical points; rows are points
- `method`: metric to measure; "avg" is the average pairwise tropical distance; "max" is the maximum pairwise tropical distance

**Value**

within cluster measure

**Author(s)**

David Barnhill <david.barnhill@nps.edu>

**References**


**Examples**

```r
D<-Sim_points
avg.m<-trop_wi_dist(D, method='avg')
max.m<-trop_wi_dist(D, method='avg')
```
**Ultrametrics.HAR**

*Hit-and-Run Sampler for the space of ultrametrics*

**Description**

This sampler samples a point in the space of ultrametrics where each point represents an equidistant tree on $n$ leaves.

**Usage**

```r
Ultrametrics.HAR(x0, n, I = 1, h = 1)
```

**Arguments**

- **x0**: an equidistant tree defined as ultrametric
- **n**: number of leaves for the equidistant tree
- **I**: number of states in the Markov chain
- **h**: height of phylogenetic tree

**Value**

point in the space of ultrametrics over $n$ leaves

**Author(s)**

Ruriko Yoshida <ryoshida@nps.edu>

**References**


**Examples**

```r
x0 <- Sim_Trees15[1,]
n<-10

Ultrametrics.HAR(x0, n, I = 50, h = 1)
```
vector.to.equidistant.tree

Vector to equidistant tree

Description
A vector of pairwise distances is used to reconstruct the corresponding equidistant tree

Usage
vector.to.equidistant.tree(omega)

Arguments
omega
vector of pairwise distances in $\mathbb{R}^{\binom{m}{2}}$, where $m$ is the number of leaves

Value
equidistant phylogenetic tree

Author(s)
Georgios Aliatimis <g.aliatimis@lancaster.ac.uk>

References
Aliatimis, Georgios, Ruriko Yoshida, Burak Boyaci and James A. Grant (2023). Tropical Logistic Regression on Space of Phylogenetic Trees

Examples
omega = c(1/3,1,1,1,1,1/3)
tree = vector.to.equidistant.tree(omega)
plot(tree)
Index

* 1000
  Sim_Trees1, 23
  Sim_Trees2, 24

* 45
  Sim_Trees1, 23
  Sim_Trees2, 24

* A
  Sim_Trees1, 23
  Sim_Trees2, 24

* an
  Sim_Trees1, 23
  Sim_Trees2, 24

* a
  Sim_Trees1, 23
  Sim_Trees2, 24

* datasets
  lung_fish, 12
  Sim_points, 23
  Sim_Trees1, 23
  Sim_Trees2, 24

* dataset
  lung_fish, 12

* each
  Sim_Trees1, 23
  Sim_Trees2, 24

* gene
  Sim_Trees1, 23
  Sim_Trees2, 24

* is
  Sim_Trees1, 23
  Sim_Trees2, 24

* lungfish
  lung_fish, 12

* matrix
  Sim_Trees1, 23
  Sim_Trees2, 24

* phylogenetic
  Sim_Trees1, 23
  Sim_Trees2, 24

* representing
  Sim_Trees1, 23
  Sim_Trees2, 24

* row
  Sim_Trees1, 23
  Sim_Trees2, 24

* simulated
  Sim_Trees1, 23
  Sim_Trees2, 24

* tree
  Sim_Trees1, 23
  Sim_Trees2, 24

* ultrametric
  Sim_Trees1, 23
  Sim_Trees2, 24

* where
  Sim_Trees1, 23
  Sim_Trees2, 24

* x
  Sim_Trees1, 23
  Sim_Trees2, 24

bw.nn, 3

cluster.ratio_HC, 4
cluster.ratio_KM, 5
convert.to.tree, 6
draw.tpolytope, 6

FWpoint.num.w.reg, 7
FWpoint_numerical, 8

HAR.TLineSeg, 9
HAR.TLineSeg.Norm, 10
hyper3d_max (hyper_3D), 11
hyper3d_min (hyper_3D), 11
hyper_3D, 11

If_labels(lung_fish), 12
lung_fish, 12
INDEX

max_ins_ball, 12
min_enc_ball, 13
normaliz.polytope (normalize), 14
normaliz.tree, 14
normaliz.ultrametrics (normalize), 14
normaliz.vector (normalize), 14
normaliz.vectors (normalize), 14
normalize, 14
over_bet_HC, 15
over_bet_KM, 16
Points.TLineSeg, 17
pre.pplot.pro, 18
prob.class, 19
project_pi, 20
pw.trop.dist, 20
rounding, 21
sigmoid, 22
Sim_points, 23
Sim_Trees1, 23
Sim_Trees1025 (Sim_Trees1), 23
Sim_Trees105 (Sim_Trees1), 23
Sim_Trees11 (Sim_Trees1), 23
Sim_Trees110 (Sim_Trees1), 23
Sim_Trees12 (Sim_Trees1), 23
Sim_Trees15 (Sim_Trees1), 23
Sim_Trees2, 24
Sim_Trees2025 (Sim_Trees2), 24
Sim_Trees205 (Sim_Trees2), 24
Sim_Trees21 (Sim_Trees2), 24
Sim_Trees210 (Sim_Trees2), 24
Sim_Trees22 (Sim_Trees2), 24
Sim_Trees25 (Sim_Trees2), 24
tdets, 25
TKmeans, 25
TLineSeg, 26
TLineSeg_min, 27
TML, 27
TML-package (TML), 27
tree.to.vector, 28
trop.dist, 29
trop.dist.hyp_max, 29
trop.dist.hyp_min, 30
trop.Gaussian.MH (tropical.Gaussian.MH), 34
trop.logistic.regression, 31
trop.tri.plot.w.pts, 32
trop_bal.vert, 39
Trop_ball, 40
Trop_FW, 41
Trop_Volume, 42
trop.wi_dist, 43
tropical.Gaussian, 33
tropical.Gaussian.MH, 34
Tropical.HC.AGNEs, 35
tropical.KDE, 35
tropical.PCA, 36
TropicalPolytope.extrapolation.HAR, 38
Ultrametrics.HAR, 44
vector.to.equidistant.tree, 45