Package 'Semblance'

January 25, 2019

Type Package

Title A Data-Driven Similarity Kernel on Probability Spaces

Version 1.1.0

Author Divyansh Agarwal <divyansh@upenn.edu>
Nancy R. Zhang <nzh@wharton.upenn.edu>

Maintainer Divyansh Agarwal <divyansh@upenn.edu>

Description We present a rank-based Mercer kernel to compute a pair-wise similarity metric corresponding to informative representation of data. We tailor the development of a kernel to encode our prior knowledge about the data distribution over a probability space. The philosophical concept behind our construction is that objects whose feature values fall on the extreme of that feature's probability mass distribution are more similar to each other, than objects whose feature values lie closer to the mean. Semblance emphasizes features whose values lie far away from the mean of their probability distribution. The kernel relies on properties empirically determined from the data and does not assume an underlying distribution. The use of feature ranks on a probability space ensures that Semblance is computational efficacious, robust to outliers, and statistically stable, thus making it widely applicable algorithm for pattern analysis. The output from the kernel is a square, symmetric matrix that gives proximity values between pairs of observations.

License GPL-2

Encoding UTF-8

LazyData true

Imports fields (>= 9.6), PerformanceAnalytics (>= 1.5.2), DescTools (>= 0.99.26), msos (>= 1.1.0)

Suggests kernlab

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-01-25 13:40:10 UTC

R topics documented:

computeSemblanceOneFeature 2
computeSemblanceOneFeature_Gini 2
makeUpperLower 3
ranksem 3
ranksem_Gini 5
repCol 6
repRow 6

Index **7**

`computeSemblanceOneFeature`

Compute semblance when there is only one feature, given as a vector x.

Description

Compute semblance when there is only one feature, given as a vector x.

Usage

```
computeSemblanceOneFeature(x)
```

Arguments

x	a vector of observations for whom a given feature has been measured or estimated

Value

a Semblance metric for only one feature measured for several observations

---

`computeSemblanceOneFeature_Gini`

Compute semblance when there is only one feature, given as a vector x, but weight the feature by its Gini coefficient. Use for data with strictly positive values.

Description

Compute semblance when there is only one feature, given as a vector x, but weight the feature by its Gini coefficient. Use for data with strictly positive values.

Usage

```
computeSemblanceOneFeature_Gini(x)
```

Arguments

x	a vector of observations for whom a given feature has been measured or estimated

Value

a Semblance metric for only one feature measured for several observations

---

`makeUpperLower` *Make the upper triangular part the same as the lower triangular part.*

Description

Make the upper triangular part the same as the lower triangular part.

Usage

```
makeUpperLower(m)
```

Arguments

m	a matrix whose upper traingular part needs to be created using the lower traingular part

Value

a matrix where the upper triangular part the same as the lower triangular part

---

`ranksem` *Compute Semblance For a Given Input Matrix or Data Frame*

Description

Kernel methods can operate in a high-dimensional, implicit feature space with low computational cost. Here, we present a rank-based Mercer kernel to compute a pair-wise similarity metric, corresponding to informative representation of data. We tailor the development of a kernel to encode our prior knowledge about the data distribution over a probability space. The philosophical concept behind our construction is that objects whose feature values fall on the extreme of that feature's probability mass distribution are more similar to each other, than objects whose feature values lie closer to the mean. This idea represents a fundamentally novel way of assessing similarity between two observations. Our kernel (henceforth called 'Semblance') naturally lends itself to the construction of a distance metric that emphasizes features whose values lie far away from the mean of their

probability distribution. Semblance relies on properties empirically determined from the data and does not assume an underlying distribution. The use of feature ranks on a probability space ensures that Semblance is computational efficacious, robust to outliers, and statistically stable, thus making it widely applicable algorithm for pattern analysis. This R package accompanies the research article "Semblance: A Data-driven Kernel Redefines the Notion of Similarity", to appear in Science Advances.

Usage

```
ranksem(X)
```

Arguments

X	a matrix X with n observations and m features, whose Semblance Gram Matrix is to be computed

Value

The resultant Gram Matrix after applying Semblance kernel to the input

Examples

```
# Simulation Example when the user inputs a matrix with single-cell gene expression data
ngenes = 10
ncells = 10
nclust = 2
mu=c(100, 0) #mean in cluster 1, cluster 2 for informative genes
sigma=c(0.01, 1) #stdev in cluster 1, cluster 2 for informative genes
size.rare.clust = 0.1
prop.info.genes = 0.2
n.info.genes=round(prop.info.genes*ngenes)
n.clust1.cells = round(ncells*size.rare.clust)
mu1=c(rep(mu[1]*sigma[2], n.info.genes), rep(0, ngenes-n.info.genes))
mu2=c(rep(mu[2]*sigma[2], n.info.genes), rep(0, ngenes-n.info.genes))
sig1=c(rep(sigma[1], n.info.genes), rep(1, ngenes-n.info.genes))
sig2=c(rep(sigma[2], n.info.genes), rep(1, ngenes-n.info.genes))
X=matrix(ncol=ngenes, nrow=ncells, data=0)
for(i in 1:n.clust1.cells){
  X[i,] = rnorm(ngenes, mean=mu1, sd=sig1)
}
for(i in (n.clust1.cells+1):ncells){
  X[i,] = rnorm(ngenes, mean=mu2, sd=sig2)
}
#Compute kernels/distances
rks=ranksem(X)
```

ranksem_Gini *Compute Gini-weighted Semblance*

Description

Compute Gini-weighted Semblance

Usage

```
ranksem_Gini(X)
```

Arguments

X	a matrix X with n observations and m features, whose Semblance Gram Matrix is to be computed. While computing this Gram Matrix, each feature is weighed by the Gini index for efficient feature selection.

Value

The resultant Gini-weighted Gram Matrix after applying Semblance kernel to the input

Examples

```
# Simulation Example when the user inputs a matrix with single-cell gene expression data
ngenes = 10
ncells = 10
nclust = 2
mu=c(5, 1) #mean in cluster 1, cluster 2 for informative genes
sigma=c(2, 1) #stdev in cluster 1, cluster 2 for informative genes
size.rare.clust = 0.2
prop.info.genes = 0.2
n.info.genes=round(prop.info.genes*ngenes)
n.clust1.cells = round(ncells*size.rare.clust)
mu1=c(rep(mu[1]*sigma[2], n.info.genes), rep(0, ngenes-n.info.genes))
mu2=c(rep(mu[2]*sigma[2], n.info.genes), rep(0, ngenes-n.info.genes))
sig1=c(rep(sigma[1], n.info.genes), rep(1, ngenes-n.info.genes))
sig2=c(rep(sigma[2], n.info.genes), rep(1, ngenes-n.info.genes))
X=matrix(ncol=ngenes, nrow=ncells, data=0)
for(i in 1:n.clust1.cells){
  X[i,] = rnorm(ngenes, mean=mu1, sd=sig1)
}
for(i in (n.clust1.cells+1):ncells){
  X[i,] = rnorm(ngenes, mean=mu2, sd=sig2)
}
Noise <- matrix(rnorm(prod(dim(X)), mean=2, sd=0.4), nrow = 10)
X = X + Noise
#Compute kernels/distances
rks=ranksem_Gini(X)
```

repCol *Make a matrix by repeating vector v into n columns*

Description

Make a matrix by repeating vector v into n columns

Usage

```
repCol(v, n)
```

Arguments

v	a vector to be operated on
n	number of columns the vector will be repeated over

Value

a matrix with repeated columns

repRow *Make a matrix by repeating vector v into n rows*

Description

Make a matrix by repeating vector v into n rows

Usage

```
repRow(v, n)
```

Arguments

v	a vector to be operated on
n	number of rows the vector will be repeated over

Value

a matrix with repeated rows

Index

computeSemblanceOneFeature, 2
computeSemblanceOneFeature_Gini, 2

makeUpperLower, 3

ranksem, 3
ranksem_Gini, 5
repCol, 6
repRow, 6