Package ‘KMD’

October 12, 2022

Type Package

Title Kernel Measure of Multi-Sample Dissimilarity

Version 0.1.0

Maintainer Zhen Huang <zh2395@columbia.edu>

Description Implementations of the kernel measure of multi-sample dissimilarity (KMD) between several samples using K-nearest neighbor graphs and minimum spanning trees. The KMD measures the dissimilarity between multiple samples, based on the observations from them. It converges to the population quantity (depending on the kernel) which is between 0 and 1. A small value indicates the multiple samples are from the same distribution, and a large value indicates the corresponding distributions are different. The population quantity is 0 if and only if all distributions are the same, and 1 if and only if all distributions are mutually singular. The package also implements the tests based on KMD for H0: the M distributions are equal against H1: not all the distributions are equal. Both permutation test and asymptotic test are available. These tests are consistent against all alternatives where at least two samples have different distributions. For more details on KMD and the associated tests, see Huang, Z. and B. Sen (2022) <arXiv:2210.00634>.

License GPL-3

Encoding UTF-8

RoxygenNote 7.1.2

Depends R (>= 4.0.0), data.table

Imports RANN, proxy, mlpack, boot, igraph

NeedsCompilation no

Author Zhen Huang [aut, cre]

Repository CRAN

Date/Publication 2022-10-06 16:10:12 UTC

R topics documented:

KMD *Kernel Measure of Multi-sample Dissimilarity*

Description

Compute the kernel measure of multi-sample dissimilarity (KMD) with directed K-nearest neighbor (K-NN) graph or minimum spanning tree (MST).

Usage

```
KMD(X, Y, M = length(unique(Y)), Knn = 1, Kernel = "discrete")
```

Arguments

`X`	the data matrix (n by dx) or the distance/similarity matrix (n by n)
`Y`	a vector of length n, indicating the labels (from 1 to M) of the data
`M`	the number of possible labels
`Knn`	the number of nearest neighbors to use, or "MST"
`Kernel`	an M by M kernel matrix with row i and column j being the kernel value $k(i, j)$; or "discrete" which indicates using the discrete kernel.

Details

The kernel measure of multi-sample dissimilarity (KMD) measures the dissimilarity between multiple samples, based on the observations from them. It converges to the population quantity (depending on the kernel) which is between 0 and 1. A small value indicates the multiple samples are from the same distribution, and a large value indicates the corresponding distributions are different. The population quantity is 0 if and only if all distributions are the same, and 1 if and only if all distributions are mutually singular.

If X is an n by n matrix, it will be interpreted as a distance/similarity matrix. In such case, MST requires it to be symmetric (an undirected graph). K-NN graph does not require it to be symmetric, with the nearest neighbors of point i computed based on the i-th row, and ties broken at random. The diagonal terms (self-distances) will be ignored. If X is an n by dx data matrix, Euclidean distance will be used for computing the K-NN graph (ties broken at random) and the MST.

Value

The algorithm returns a real number which is the sample KMD and is asymptotically between 0 and 1.

See Also

`KMD_test`

Examples

```
n = 60
d = 2
set.seed(1)
X1 = matrix(runif(n*d/2),ncol = d)
X2 = matrix(runif(n*d/2),ncol = d)
X2[,1] = X2[,1] + 1
X = rbind(X1,X2)
Y = c(rep(1,n/2),rep(2,n/2))
print(KMD(X, Y, M = 2, Knn = 1, Kernel = "discrete"))
# 0.9344444. X1 and X2 are mutually singular, so the theoretical KMD is 1.
print(KMD(X, Y, M = 2, Knn = 1, Kernel = base::diag(c(1,1))))
# 0.9344444. This is essentially the same as specifying the discrete kernel above.
print(KMD(X, Y, M = 2, Knn = 2, Kernel = "discrete"))
print(KMD(X, Y, M = 2, Knn = "MST", Kernel = "discrete"))
# 0.9508333, 0.9399074. One can also use other geometric graphs (2-NN graph and MST here)
# to estimate the same theoretical quantity.
```

KMD_test *Testing based on KMD*

Description

Testing based on the kernel measure of multi-sample dissimilarity (KMD). Both permutation test and asymptotic test are available. The tests are consistent against all alternatives where at least two samples have different distributions.

Usage

```
KMD_test(
  X,
  Y,
  M = length(unique(Y)),
  Knn = ceiling(length(Y)/10),
  Kernel = "discrete",
  Permutation = TRUE,
  B = 500
)
```

Arguments

X	the data matrix (n by dx) or the distance/similarity matrix (n by n)
Y	a vector of length n, indicating the labels (from 1 to M) of the data
M	the number of possible labels
Knn	the number of nearest neighbors to use, or "MST"
Kernel	an M by M kernel matrix with row i and column j being the kernel value $k(i, j)$; or "discrete" which indicates using the discrete kernel.

`Permutation`	TRUE or FALSE; whether to perform permutation test or the asymptotic test.
`B`	the number of permutations to perform, only used for permutation test.

Details

The kernel measure of multi-sample dissimilarity (KMD) measures the dissimilarity between multiple samples using geometric graphs such as K-nearest neighbor (K-NN) graph and minimum spanning tree (MST), based on the observations from them. A small value indicates the multiple samples are from the same distribution, and a large value indicates the corresponding distributions are different. The test rejects the null hypothesis that all samples are from the same distribution for large value of sample KMD. The permutation test returns the p-value given by (sum(KMD_i >= KMD_0) + 1)/(B + 1), where KMD_i is the KMD computed after a random permutation on the Y labels, and B is the total number of permutations that have been performed. The asymptotic test first normalizes the KMD by the square root of the permutation variance, and then returns the p-value given by: P(N(0,1) > normalized KMD).

If X is an n by n matrix, it will be interpreted as a distance/similarity matrix. In such case, MST requires it to be symmetric (an undirected graph). K-NN graph does not require it to be symmetric, with the nearest neighbors of point i computed based on the i-th row, and ties broken at random. The diagonal terms (self-distances) will be ignored. If X is an n by dx data matrix, Euclidean distance will be used for computing the K-NN graph (ties broken at random) and the MST.

Value

If Permutation == TRUE, permutation test is performed and the algorithm returns a p-value for testing H0: the M distributions are equal against H1: not all the distributions are equal. If Permutation == FALSE, asymptotic test is performed and a 1 by 2 matrix: (z value, p-value) is returned.

See Also

`KMD`

Examples

```
d = 2
set.seed(1)
X1 = matrix(rnorm(100*d), nrow = 100, ncol = d)
X2 = matrix(rnorm(100*d,sd=sqrt(1.5)), nrow = 100, ncol = d)
X3 = matrix(rnorm(100*d,sd=sqrt(2)), nrow = 100, ncol = d)
X = rbind(X1,X2,X3)
Y = c(rep(1,100),rep(2,100),rep(3,100))
print(KMD_test(X, Y, M = 3, Knn = 1, Kernel = "discrete"))
# A small p-value since the three distributions are not the same.
print(KMD_test(X, Y, M = 3, Knn = 1, Kernel = "discrete", Permutation = FALSE))
# p-value of the asymptotic test is similar to that of the permutation test
print(KMD_test(X, Y, M = 3, Knn = 1, Kernel = diag(c(10,1,1))))
# p-value is improved by using a different kernel
print(KMD_test(X, Y, M = 3, Knn = 30, Kernel = "discrete"))
# The suggested choice Knn = 0.1n yields a very small p-value.
print(KMD_test(X, Y, M = 3, Knn = "MST", Kernel = "discrete"))
# One can also use the MST.
```

```
print(KMD_test(X, Y, M = 3, Knn = 2, Kernel = "discrete"))
# MST has similar performance as 2-NN, which is between 1-NN and 30-NN

# Check null distribution of the z values
ni = 100
n = 3*ni
d = 2
Null_KMD = function(id){
  set.seed(id)
  X = matrix(rnorm(n*d), nrow = n, ncol = d)
  Y = c(rep(1,ni),rep(2,ni),rep(3,ni))
 return(KMD_test(X, Y, M = 3, Knn = "MST", Kernel = "discrete", Permutation = FALSE)[1,1])
}
hist(sapply(1:500, Null_KMD), breaks = c(-Inf,seq(-5,5,length=50),Inf), freq = FALSE,
  xlim = c(-4,4), ylim = c(0,0.5), main = expression(paste(n[i]," = 100")),
  xlab = expression(paste("normalized ",hat(eta))))
lines(seq(-5,5,length=1000),dnorm(seq(-5,5,length=1000)),col="red")
# The histogram of the normalized KMD is close to that of a standard normal distribution.
```

Index