Package: gTests (via r-universe)

October 12, 2024

Version 0.2

Date 2017-12-6

Title Graph-Based Two-Sample Tests

Author Hao Chen and Jingru Zhang

Maintainer Hao Chen <hxchen@ucdavis.edu>

Depends R (>= 3.0.1)

Imports ade4

Description Four graph-based tests are provided for testing whether two samples are from the same distribution. It works for both continuous data and discrete data.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2017-12-06 22:04:04 UTC

Contents

`counts1` *A matrix representing counts in the distinct values for the two samples*

Description

This is a K by 2 matrix, where K is the number of distinct values. It specifies the counts in the K distinct values for the two samples. The data is generated from two samples with mean shift.

`counts2` *A matrix representing counts in the distinct values for the two samples*

Description

This is a K by 2 matrix, where K is the number of distinct values. It specifies the counts in the K distinct values for the two samples. The data is generated from two samples with spread difference.

`counts3` *A matrix representing counts in the distinct values for the two samples*

Description

This is a K by 2 matrix, where K is the number of distinct values. It specifies the counts in the K distinct values for the two samples. The data is generated from two samples with mean shift and spread difference.

`dfs` *Depth-first search*

Description

One starts at the root and explores as far as possible along each branch before backtracking.

Usage

```
dfs(s,visited,adj)
```

Arguments

`s` The root node.

`visited` N by 1 vector, where N is the number of nodes. This vector records whether nodes have been visited or not with 1 if visited and 0 otherwise.

`adj` N by N adjacent matrix.

See Also

`getGraph`

`ds1` *A distance matrix on the distinct values*

Description

This is a K by K matrix, which is the distance matrix on the distinct values for counts1.

`ds2` *A distance matrix on the distinct values*

Description

This is a K by K matrix, which is the distance matrix on the distinct values for counts2.

`ds3` *A distance matrix on the distinct values*

Description

This is a K by K matrix, which is the distance matrix on the distinct values for counts3.

E1 *An edge matrix representing a similarity graph*

Description

This is a matrix with the number of rows the number of edges in the similarity graph and 2 columns. Each row records the subject indices of the two edges of in the similarity graph. The subject indices of sample 1 is 1:100, and the subject indices of sample 2 is 101:250.

E2 *An edge matrix representing a similarity graph*

Description

This is a matrix with the number of rows the number of edges in the similarity graph and 2 columns. Each row records the subject indices of the two edges of in the similarity graph. The subject indices of sample 1 is 1:100, and the subject indices of sample 2 is 101:250.

E3 *An edge matrix representing a similarity graph*

Description

This is a matrix with the number of rows the number of edges in the similarity graph and 2 columns. Each row records the subject indices of the two edges of in the similarity graph. The subject indices of sample 1 is 1:100, and the subject indices of sample 2 is 101:250.

g.tests *Graph-based two-sample tests*

Description

This function provides four graph-based two-sample tests.

Usage

```
g.tests(E, sample1ID, sample2ID, test.type="all", maxtype.kappa = 1.14, perm=0)
```

Arguments

`E` An edge matrix representing a similarity graph with the number of edges in the similarity graph being the number of rows and 2 columns. Each row records the subject indices of the two ends of an edge in the similarity graph.

`sample1ID` The subject indices of sample 1.

`sample2ID` The subject indices of sample 2.

`test.type` The default value is "all", which means all four tests are performed: orignial edge-count test (Friedman and Rafsky (1979)), generalized edge-count test (Chen and Friedman (2016)), weighted edge-count test (Chen, Chen and Su (2016)) and maxtype edge-count tests (Zhang and Chen (2017)). Set this value to "original" or "o" to permform only the original edge-count test; set this value to "generalized" or "g" to perform only the generalized edge-count test; set this value to "weighted" or "w" to perform only the weighted edge-count test; and set this value to "maxtype" or "m" to perform only the maxtype edge-count tests.

`maxtype.kappa` The value of parameter(kappa) in the maxtype edge-count tests. The default value is 1.14.

`perm` The number of permutations performed to calculate the p-value of the test. The default value is 0, which means the permutation is not performed and only approximate p-value based on asymptotic theory is provided. Doing permutation could be time consuming, so be cautious if you want to set this value to be larger than 10,000.

Value

`test.statistic` The test statistic.

`pval.approx` The approximated p-value based on asymptotic theory.

`pval.perm` The permutation p-value when argument ‘perm‘ is positive.

References

Friedman J. and Rafsky L. Multivariate generalizations of the WaldWolfowitz and Smirnov two-sample tests. The Annals of Statistics, 7(4):697-717, 1979.

Chen, H. and Friedman, J. H. A new graph-based two-sample test for multivariate and object data. Journal of the American Statistical Association, 2016.

Chen, H., Chen, X. and Su, Y. A weighted edge-count two sample test for multivariate and object data. Journal of the American Statistical Association, 2017.

Zhang, J. and Chen, H. Graph-based two-sample tests for discrete data.

Examples

```
# the "example" data contains three similarity graphs represted in the matrix form: E1, E2, E3.
data(example)

# E1 is an edge matrix representing a similarity graph.
# It is constructed on two samples with mean difference.
# Sample 1 indices: 1:100; sample 2 indices: 101:250.
```

```
g.tests(E1, 1:100, 101:250)

# E2 is an edge matrix representing a similarity graph.
# It is constructed on two samples with variance difference.
# Sample 1 indices: 1:100; sample 2 indices: 101:250.
g.tests(E2, 1:100, 101:250)

# E3 is an edge matrix representing a similarity graph.
# It is constructed on two samples with mean and variance difference.
# Sample 1 indices: 1:100; sample 2 indices: 101:250.
g.tests(E3, 1:100, 101:250)

## Uncomment the following line to get permutation p-value with 200 permutations.
# g.tests(E1, 1:100, 101:250, perm=200)
```

---

g.tests_discrete *Graph-based two-sample tests for discrete data*

Description

This function provides four graph-based two-sample tests for discrete data.

Usage

```
g.tests_discrete(E, counts, test.type = "all", maxtype.kappa = 1.14, perm = 0)
```

Arguments

E	An edge matrix representing a similarity graph on the distinct values with the number of edges in the similarity graph being the number of rows and 2 columns. Each row records the subject indices of the two ends of an edge in the similarity graph.
counts	A K by 2 matrix, where K is the number of distinct values. It specifies the counts in the K distinct values for the two samples.
test.type	The default value is "all", which means all four tests are performed: the orignial edge-count test (Chen and Zhang (2013)), extension of the generalized edge-count test (Chen and Friedman (2016)), extension of the weighted edge-count test (Chen, Chen and Su (2016)) and extension of the maxtype edge-count tests (Zhang and Chen (2017)). Set this value to "original" or "o" to permform only the original edge-count test; set this value to "generalized" or "g" to perform only extension of the generalized edge-count test; set this value to "weighted" or "w" to perform only extension of the weighted edge-count test; and set this value to "maxtype" or "m" to perform only extension of the maxtype edge-count tests.
maxtype.kappa	The value of parameter(kappa) in the extension of the maxtype edge-count tests. The default value is 1.14.

`perm` The number of permutations performed to calculate the p-value of the test. The default value is 0, which means the permutation is not performed and only approximate p-value based on asymptotic theory is provided. Doing permutation could be time consuming, so be cautious if you want to set this value to be larger than 10,000.

Value

`test.statistic_a`
The test statistic using ‘average‘ method to construct the graph.

`test.statistic_u`
The test statistic using ‘union‘ method to construct the graph.

`pval.approx_a` Using ‘average‘ method to construct the graph, the approximated p-value based on asymptotic theory.

`pval.approx_u` Using ‘union‘ method to construct the graph, the approximated p-value based on asymptotic theory.

`pval.perm_a` Using ‘average‘ method to construct the graph, the permutation p-value when argument ‘perm‘ is positive.

`pval.perm_u` Using ‘union‘ method to construct the graph, the permutation p-value when argument ‘perm‘ is positive.

References

Friedman J. and Rafsky L. Multivariate generalizations of the WaldWolfowitz and Smirnov two-sample tests. The Annals of Statistics, 7(4):697-717, 1979.

Chen, H. and Zhang, N. R. Graph-based tests for two-sample comparisons of categorical data. Statistica Sinica, 2013.

Chen, H. and Friedman, J. H. A new graph-based two-sample test for multivariate and object data. Journal of the American Statistical Association, 2016.

Chen, H., Chen, X. and Su, Y. A weighted edge-count two sample test for multivariate and object data. Journal of the American Statistical Association, 2017.

Zhang, J. and Chen, H. Graph-based two-sample tests for discrete data.

Examples

```
# the "example_discrete" data contains three two-sample counts data
# represted in the matrix form: counts1, counts2, counts3
# and the corresponding distance matrix on the distinct values: ds1, ds2, ds3.
data(example_discrete)

# counts1 is a K by 2 matrix, where K is the number of distinct values.
# It specifies the counts in the K distinct values for the two samples.
# ds1 is the corresponding distance matrix on the distinct values.
# The data is generated from two samples with mean shift.
Knnl = 3
E1 = getGraph(counts1, ds1, Knnl, graph = "nnlink")
g.tests_discrete(E1, counts1)
```

```
# counts2 is a K by 2 matrix, where K is the number of distinct values.
# It specifies the counts in the K distinct values for the two samples.
# ds2 is the corresponding distance matrix on the distinct values.
# The data is generated from two samples with spread difference.
Kmst = 6
E2 = getGraph(counts2, ds2, Kmst, graph = "mstree")
g.tests_discrete(E2, counts2)

# counts3 is a K by 2 matrix, where K is the number of distinct values.
# It specifies the counts in the K distinct values for the two samples.
# ds3 is the corresponding distance matrix on the distinct values.
# The data is generated from two samples with mean shift and spread difference.
Knnl = 3
E3 = getGraph(counts3, ds3, Knnl, graph = "nnlink")
g.tests_discrete(E3, counts3)

## Uncomment the following line to get permutation p-value with 200 permutations.
# Knnl = 3
# E1 = getGraph(counts1, ds1, Knnl, graph = "nnlink")
# g.tests_discrete(E1, counts1, test.type = "all", maxtype.kappa = 1.31, perm = 300)
```

getComdist *Get distance between two components*

Description

This function calculates the distance between two components.

Usage

```
getComdist(g1,g2,distance)
```

Arguments

`g1` The distinct values in Component 1.

`g2` The distinct values in Component 2.

`distance` A K by K matrix, which is the distance matrix on the distinct values and K is the number of distinct values with at least one observation in either group.

See Also

getGraph

getGraph *Construct similarity graph*

Description

This function provides two methods to construct the similarity graph.

Usage

```
getGraph(counts, mydist, K, graph.type = "mstree")
```

Arguments

`counts`	A K by 2 matrix, where K is the number of distinct values. It specifies the counts in the K distinct values for the two samples.
`mydist`	A K by K matrix, which is the distance matrix on the distinct values.
`K`	Set the value of k in "k-MST" or "k-NNL" to construct the similarity graph.
`graph.type`	Specify the type of the constructing graph. The default value is "mstree", which means constructing the minimal spanning tree as the similarity graph. Set this value to "nnlink" to construct the similarity graph by the nearest neighbor link method.

Value

`E`	An edge matrix representing a similarity graph on the distinct values with the number of edges in the similarity graph being the number of rows and 2 columns. Each row records the subject indices of the two ends of an edge in the similarity graph.

See Also

`g.tests_discrete`

getMV_discrete *Get intermediate results for g.tests_discrete function*

Description

This function calculates means and variances of R1 and R2 quantities using ‘average‘ method and ‘union‘ method to construct the graph.

Usage

```
getMV_discrete(E,vmat)
```

Arguments

`E` An edge matrix representing a similarity graph on the distinct values with the number of edges in the similarity graph being the number of rows and 2 columns. Each row records the subject indices of the two ends of an edge in the similarity graph.

`vmat` A K by 2 matrix, where K is the number of distinct values with at least one observation in either group. It specifies the counts in the K distinct values for the two samples.

See Also

`g.tests_discrete`

`getR1R2` *Get intermediate results for g.tests function*

Description

This function calculates R1 and R2 quantities.

Usage

```
getR1R2(E, G1)
```

Arguments

`E` A matrix with the number of rows the number of edges in the similarity graph and 2 columns. Each row records the subject indices of the two ends of an edge in the similarity graph.

`G1` The subject indices of sample 1.

See Also

`g.tests`

getR1R2_discrete *Get intermediate results for g.tests_discrete function*

Description

This function calculates R1 and R2 quantities using 'average' method and 'union' method to construct the graph.

Usage

```
getR1R2_discrete(E,vmat)
```

Arguments

E	An edge matrix representing a similarity graph on the distinct values with the number of edges in the similarity graph being the number of rows and 2 columns. Each row records the subject indices of the two ends of an edge in the similarity graph.
vmat	A K by 2 matrix, where K is the number of distinct values with at least one observation in either group. It specifies the counts in the K distinct values for the two samples.

See Also

g.tests_discrete

gTests *Graph-Based Two-Sample Tests*

Description

This package includes four graph-based two-sample tests under the continuous setting and the discrete setting.

Author(s)

Hao Chen and Jingru Zhang

Maintainer: Hao Chen (hxchen@ucdavis.edu)

References

Friedman J. and Rafsky L. (1979). Multivariate generalizations of the WaldWolfowitz and Smirnov two-sample tests. The Annals of Statistics 7(4):697-717.

Chen, H. and Zhang, N. R. (2013). Graph-based tests for two-sample comparisons of categorical data. Statistica Sinica 23:1479-1503.

Chen, H. and Friedman, J. H. (2017). A new graph-based two-sample test for multivariate and object data. Journal of the American Statistical Association, 112:517, 397-409.

Chen, H., Chen, X. and Su, Y. (2017). A weighted edge-count two sample test for multivariate and object data. Journal of the American Statistical Association.

Zhang, J. and Chen, H. (2017). Graph-based two-sample tests for discrete data. arXiv:1711.04349

See Also

`g.tests` `g.tests_discrete` `getGraph`

`nnlink` *Construct similarity graph by 1-NNL*

Description

This function provides the edges of the similarity graph constructed by 1-NNL.

Usage

```
nnlink(distance)
```

Arguments

`distance` A K by K matrix, which is the distance matrix on the distinct values and K is the number of distinct values with at least one observation in either group.

Value

`E` An edge matrix representing a similarity graph on the distinct values with the number of edges in the similarity graph being the number of rows and 2 columns. Each row records the subject indices of the two ends of an edge in the similarity graph.

See Also

`getGraph`

nnlink_Com *Get components by nearest neighbor link algorithm*

Description

This function obtains components based on the nearest neighbor link algorithm.

Usage

```
nnlink_Com(distance)
```

Arguments

`distance`	A K by K matrix, which is the distance matrix on the distinct values and K is the number of distinct values with at least one observation in either group.

See Also

`getGraph`

nnlink_K *Construct similarity graph by k-NNL*

Description

This function provides the edges of the similarity graph constructed by k-NNL.

Usage

```
nnlink_K(distance,K)
```

Arguments

`distance`	A K by K matrix, which is the distance matrix on the distinct values and K is the number of distinct values with at least one observation in either group.
`K`	Set the value of k in "k-NNL" to construct the similarity graph.

Value

`E`	An edge matrix representing a similarity graph on the distinct values with the number of edges in the similarity graph being the number of rows and 2 columns. Each row records the subject indices of the two ends of an edge in the similarity graph.

See Also

`getGraph`

`permute_discrete` *Generate a permutation for two discrete data groups*

Description

This function permutes the observations maintaining the two sample sizes unchaged.

Usage

```
permute_discrete(vmat)
```

Arguments

`vmat`	A K by 2 matrix, where K is the number of distinct values with at least one observation in either group. It specifies the counts in the K distinct values for the two samples.

See Also

`g.tests_discrete`

Index