Package 'cpt'

October 12, 2022

Title Classification Permutation Test

Description Non-parametric test for equality of multivariate distributions. Trains a classifier to classify (multivariate) observations as coming from one of several distributions. If the classifier is able to classify the observations better than would be expected by chance (using permutation inference), then the null hypothesis that the distributions are equal is rejected.

Version 1.0.2

Date 2018-10-30

Imports MASS, nnet, randomForest, glmnet

Author Johann Gagnon-Bartsch <johanngb@umich.edu>

Maintainer Johann Gagnon-Bartsch <johanngb@umich.edu>

License GPL

URL http://dept.stat.lsa.umich.edu/~johanngb

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2018-10-30 17:40:03 UTC

R topics documented:

cpt-package *Classification Permutation Test*

Description

Description: Non-parametric test for equality of multivariate distributions. Trains a classifier to classify (multivariate) observations as coming from one of several distributions. If the classifier is able to classify the observations better than would be expected by chance (using permutation inference), then the null hypothesis that the distributions are equal is rejected.

Details

Package:	cpt
Type:	Package
Version:	1.0.2
Date:	2018-10-30
License:	GPL
LazyLoad:	yes

Author(s)

Johann Gagnon-Bartsch <johanngb@umich.edu>

References

The Classification Permutation Test: A Nonparametric Test for Equality of Multivariate Distributions. Gagnon-Bartsch and Shem-Tov, 2016. Available at: https://arxiv.org/abs/1611.06408.

See Also

cpt

cpt *Classification Permutation Test*

Description

Non-parametric test for equality of multivariate distributions. Trains a classifier to classify (multivariate) observations as coming from one of several distributions. If the classifier is able to classify the observations better than would be expected by chance (using permutation inference), then the null hypothesis that the distributions are equal is rejected.

Usage

```
cpt(Z, T, leaveout = 0, class.methods = "forest",
    metric = "probability", ensemble.metric="mean.prob",
    paired=FALSE, perm.N = 1000, leaveout.N=100,
    comb.methods=c(class.methods, "ensemble"),
    comb.method="fisher")
```

Arguments

`Z`	The data. An n by p matrix, where n is the number of observations, and p is the number of covariates.
`T`	The treatment variable. Is converted to a factor.
`leaveout`	The number of observations from each treatment group to include in the test set. If 0, no data is left out and the in-sample test statistic is used. (See note below.) If an integer greater than or equal to 1, the number of observations from each treatment group to leave out. Values between 0 and 1 are converted to `ceiling(min(table(T))*leaveout`.
`class.methods`	A character vector of the different classification methods to use. Can be "lda", "logistic", "logistic2", "glmnet", "glmnet2", or "forest". The "logistic2" and "glmnet2" classifiers include all two-way interactions in the model.
`metric`	Which test statistic to use. Can be "rate" (classification accuracy rate), "mse", or "probability". The default value ("probability") is recommended.
`ensemble.metric`	Which test statistic to use for an ensemble classifier composed of all of the individual classifiers. Can be "vote", "mean.mse", or "mean.prob". The default value ("mean.prob") is recommended.
`paired`	Do a paired permutation test. The data Z must be ordered such that the first observation with T==1 is paired with the first observation with T==2, the second observation with T==1 is paired with the second observation with T==2, etc. This can be accomplished by either letting the first n/2 rows be the treatment observations, and last n/2 rows being the control observations (in the same order), or by using the first two rows for the first pair, the second two rows for the second pair, etc.
`perm.N`	The number of permutations.
`leaveout.N`	The number of training set / test set iterations. In each iteration, a random test set is generated. Thus, test sets will typically overlap somewhat. There is one exception: If leaveout = 1 and leaveout.N = n, then a traditional leave-one-out procedure is used (each obersvation is left out exactly once).
`comb.methods`	Which of the classifiers to include in the combined, overall p-value. Can be any subset of the classifiers specified in `class.methods` in addition to "ensemble" for the ensemble classifier.
`comb.method`	The method for combining p-values from the individual classifiers in order to produce an overall p-value. The default ("fisher") is recommended. The other possible option is "min" which uses the minimum p-value. Note that in both cases, the combined p-value itself is not returned; rather, the combined p-value is treated as a test statistic, which is itself then subject to permutation analysis; what is returned is the resulting p-value from this analysis.

Value

A list containing

pval	The overall p-value, after combining results from the individual classifiers.
teststat	The observed test statistics of the individual classifiers.
nulldist	The permutation distributions of the individual classifiers.
pvals	The p-values of the individual classifiers.

Note

In the special case that the classifier is "forest", the metric is "rate", and "leaveout" is 0, the out-of-bag classification accuracy rate is used rather than the true in-sample classification accuracy rate.

Author(s)

Johann Gagnon-Bartsch <johanngb@umich.edu>

References

The Classification Permutation Test: A Nonparametric Test for Equality of Multivariate Distributions. Gagnon-Bartsch and Shem-Tov, 2016. Available at: https://arxiv.org/abs/1611.06408.

Examples

```
## Create some simulated data
n = 50  # 50 observations
p = 5   # 5 covariates
T = rep(c(0,1),each=25)     # Two groups, 25 observations each
Z = matrix(rnorm(n*p),n,p)  # Random data (null is true)

## Run CPT
cpt.results = cpt(Z, T, class.methods="lda")
print(cpt.results$pval)

## False Null
Z[1:25, 1] = Z[1:25, 1] + 1  # Now the null is false
cpt.results = cpt(Z, T, class.methods="lda")
print(cpt.results$pval)
```

Index