Package 'approxmatch'

May 7, 2026

Type Package

Title Approximately Optimal Fine Balance Matching with Multiple Groups

Version 2.0

Date 2017-10-23

Author Bikram Karmakar

Maintainer Bikram Karmakar <bkarmakar@ufl.edu>

Description Tools for constructing a matched design with multiple comparison groups. Further specifications of refined covariate balance restriction and exact match on covariate can be imposed. Matches are approximately optimal in the sense that the cost of the solution is at most twice the optimal cost, Crama and Spieksma (1992) <doi:10.1016/0377-2217(92)90078-N>, Karmakar, Small and Rosenbaum (2019) <doi:10.1080/10618600.2019.1584900>.

Suggests optmatch, MASS

License MIT + file LICENSE

NeedsCompilation no

Repository CRAN

Date/Publication 2020-03-30 08:50:13 UTC

Contents

approxmatch-package *Approximately Optimal Fine Balance Matching with Multiple Groups.*

Description

Tools for constructing a matched design with multiple comparison groups. Further specifications of refined covariate balance restriction and exact match on covariate can be imposed. Matches are approximately optimal in the sense that the cost of the solution is at most twice the optimal cost, Crama and Spieksma (1992) <doi:10.1016/0377-2217(92)90078-N>, Karmakar, Small and Rosenbaum (2019) <doi:10.1080/10618600.2019.1584900>.

Details

Index of help topics:

Dodgeram	Dodge ram pk 2500 data on side airbag (SAB) usage from 1995 to 2015
approxmatch-package	Approximately Optimal Fine Balance Matching with Multiple Groups.
covbalance	Check covariate balance of a design.
kwaymatching	Create approximately optimal matched strata of multiple, at least two, groups.
multigrp_dist_struc	Construct the distance structure for the multiple groups.
nrbalancematch	The is the background function to perform matching using network optimization.

An R package for creating matched strata with multiple treatments. Default design for a stratum structure is one unit from each treatment, but, other designs can be specified. User can also fine match/ near fine match on one or more categorical covariates, e.g. sex and age group.

The main functions of the package are `kwaymatching` and `tripletmatch`. These functions take as input the distance structure of multiple groups and the grouping information to create an approximately optimal multigroup design minimizing the total distance. A distance structure can be calculated as per requirement by the `multigrp_dist_struc` function.

The algorithm used to create matched design is an approximation algorithm developed by Karmakar, Small and Rosenbaum (2019). The design built is guaranteed to be close to the optimal matched design of the specified structure.

IMPORTANT NOTE: In order to perform matching, `kwaymatching` requires the user to load the optmatch (>= 0.9-1) package separately. A manual loading is required due to software license issues. If the package is not loaded, the `kwaymatching` command will fail with an error saying the optmatch package is not present. Reference to optmatch is given below.

Author(s)

Bikram Karmakar

Maintainer: Bikram Karmakar <bkarmakar@ufl.edu>

References

Crama, Y. and Spieksma, F. C. R. (1992), Approximation algorithms for three-dimensional assignment problems with triangle inequalities, European Journal of Operational Research 60, 273–279.

Hansen, B.B. and Klopfer, S.O. (2006) Optimal full matching and related designs via network flows, JCGS 15 609–627.

Karmakar, B., Small, D. S. and Rosenbaum, P. R. (2019) Using Approximation Algorithms to Build Evidence Factors and Related Designs for Observational Studies, Journal of Computational and Graphical Statistics, 28, 698–709.

Examples

```
## See kwaymatching for usage
```

`covbalance` *Check covariate balance of a design.*

Description

For a given match, this function evaluates the balance of variables before and after matching. Balance is evaluated using standardized differences.

Usage

```
covbalance(.data, grouplabel, matches, vars, details)
```

Arguments

`.data`	A data frame or matrix containing the informations on the `vars` for which balance will be checked.
`grouplabel`	Argument describing the group structure. See the description in the documentation of `kwaymatching` function of this package.
`matches`	A character matrix describing the strata structure of the design. Standard use is the output of `tripletmatching` or `kwaytmatching` function. Each row of the matrix corresponds to a strata and each entry corresponds to rowname of `.data`.
`vars`	A character vector of the names of the variables for which balance should be checked.
`details`	Optional argument. This argument can be used to get other details on the `vars` before and after matching. A character vector of names of functions which summarizes a vector, e.g. mean.

Details

Standardized difference of the covariates between two groups is computed as difference of the means of the variable over the squared root of the average variance of the variable in the groups.

For better understanding of the matching, `details` can be used. This argument can be used to get summaries of the variables before and after matching. For example, details = c(mean = 'mean', median = 'function(x) quantile(x, probs=.5)') given the mean and median of the variables. Only functions that give a single number summary can be used!

Currently, this function cannot be immediately used for a design with different strata sizes. One way to get around would be to fill in the smaller stratum with false units and making all the strata of equal size.

Value

A list consisting of the following elements.

`std_diff`	Standardized differences of the specified variables before and after matching for every pair of groups.
`details`	Only if `details` is provided. A list of summaries of the variables before and after matching using the functions specified by `details`.

Note

See `kwaymatching` for usage.

Author(s)

Bikram Karmakar

See Also

`tripletmatching`, `kwaymatching`

Examples

```
data(Dodgeram)

## An example strata structure
matches = as.matrix(sample(rownames(Dodgeram), 500), ncol = 5)

vars = c("AGE", "SEX.2", "IMPACT3.3", "DR_DRINK")
details = c('std_diff', 'mean', 'function(x) diff(range(x))',
                        'function(x) quantile(x, probs = .9)')
names(details) <- c('std_diff', 'mean', 'range', '90perc')

covbalance(.data=Dodgeram, grouplabel=c("NOSAB", "optSAB", "WITHSABS"),
                matches = matches, vars = vars, details)
```

Dodgeram *Dodge ram pk 2500 data on side airbag (SAB) usage from 1995 to 2015*

Description

This is an example dataset on the description of the Dodge pk 2500 cars involved in fatal crashes between 1995 to 2015.

Usage

```
data("Dodgeram")
```

Format

A data frame with 6953 observations on the following 33 variables.

`X.1` a numeric vector.

`X` a numeric vector.

`indexinSABdata` a numeric vector

`NOSAB` a numeric vector. In the earlier period when SAB was not available? 1 = yes.

`WITHoptSAB` a numeric vector. Bought SAB in the optional period? 1 = yes.

`optSAB` a numeric vector. In the optional period when SAB was available as an option? 1 = yes.

`WITHSABS` a numeric vector. In the later period when SAB was standard issue? 1 = yes.

`IMPACT3` a numeric vector. Impact code. (0 = no, 1 = right, 2 = behind, 3 = left, 4 = front, 9 = other)

`ROLLOVER1` a numeric vector. Rollover occurred? 1 = yes.

`REST_USE1` a numeric vector. Restraint used by the driver? 1 = yes, 99 = unknown type, 0 = not used.

`FRpass.REST_USE1` a numeric vector. Restraint use by the front right passenger.

`SP_LIMIT` a numeric vector. Speed Limit of the route.

`AGE` a numeric vector. Age of the driver.

`DR_DRINK` a numeric vector. Whether the driver was drinking? 1 = yes.

`FR.pass` a logical vector. Whether a front right passenger was present.

`FRpass.AGE` a numeric vector. Age of the front right passenger if present.

`SEX.2` a numeric vector. Sex of the driver. 1 = female.

`EJECTION.1` a numeric vector. Ejection of the driver. 1 = yes.

`EJECTION.2` a numeric vector. Ejection type of driver unknown. 1= yes.

`IMPACT3.1` a numeric vector. Impact from right? 1= yes.

`IMPACT3.2` a numeric vector. Impact from behind? 1= yes.

`IMPACT3.3` a numeric vector. Impact from left? 1= yes.

`IMPACT3.4` a numeric vector. Impact from front? 1= yes.

`IMPACT3.9` a numeric vector. Impact of other type? 1= yes.

`ROLLOVER1.1` a numeric vector. Rollover occurred? 1= yes.

`FIRE_EXP1.1` a numeric vector. Fire occurred in the car? 1= yes.

`REST_USE1.1` a numeric vector. Driver used restraint? 1= yes.

`REST_USE1.99` a numeric vector. Driver restraint use of unknown type? 1= yes.

`FRpass.SEX.2` a numeric vector. Sex of the front right passenger. 2 = female.

`FRpass.EJECTION.1` a numeric vector. Ejection of the front right passenger. 1 = yes.

`FRpass.EJECTION.2` a numeric vector. Ejection type of front right passenger unknown. 1= yes.

`FRpass.REST_USE1.1` a numeric vector. Front right passenger used restraint? 1= yes.

`FRpass.REST_USE1.99` a numeric vector. Front right passenger restraint use of unknown type? 1= yes.

Details

Derived from fatal accidents in the USA between 1995 and 2015 recorded by Fatality Analysis Reporting System (FARS) of NHTSA. Consists of only model years more than 1985. Sample is corrected for selection bias in FARS. For further detail of the variables see the code book of FARS.

Source

Fatality Analysis Reporting System of NHTSA, DOT, USA. https://www.nhtsa.gov/research-data/fatality-analysis-reporting-system-fars

Examples

```
data(Dodgeram)
## maybe str(Dodgeram) ; plot(Dodgeram) ...
```

---

kwaymatching *Create approximately optimal matched strata of multiple, at least two, groups.*

Description

This function takes as input a distance structure and grouping labels of units to create an strata structure of the units that minimizes the sum distance. It also provides features for near fine balance on one or more nominal variables and exact match on a nominal variable.

Usage

```
kwaymatching(distmat, grouplabel, design, indexgroup = 1, .data, finebalanceVars,
                        exactmatchon, ordering, reorder = FALSE, verbose = TRUE)

tripletmatching(distmat, grouplabel, design, indexgroup = 1, .data, finebalanceVars,
                        exactmatchon)
```

Arguments

distmat	An output of multigrp_dist_struc. This a named list of matrices. Each list element correspond to the distance matrix between two groups. If group labels are 1:k. Then the elements are named '1-2', '1-3',$\cdots$, 'i-j', $\cdots$. For matching of three groups k=3. If groups are labeled by some character string then list elements are named accordingly. For example, if the groups are 'grpA', 'grpB', 'grpC'; the elements are 'grpA-grpB', 'grpA-grpC', 'grpB-grpB'. Each element of the list is a numeric matrix of distances between units of the corresponding groups with number of rows equal to the size of the first group, and number of columns the size of the second group. The units are identified by their names in row or column names of the list elements.
grouplabel	This argument is used to provide information about the group structure of the units. There are a few options on how this information can be provided. 1) A numeric vector or a categorical vector. By providing a numeric vector which labels each the units to corresponding groups. 2) A matrix or data frame of dummy variables with number of groups as the number of rows (one for each group) and number of units as the number of columns. 3) A character vector of variable names. If .data is provided then only name of the variable which contains the information about the grouping can be provided. If the grouping information is encoded in dummy variable, one can also just provide the names of the dummy variables. If either the first or the second kind of information is provided, it is expected that the unit are identifiable from the names (in the first case) or rownames (in the second case) of grouplabel.
design	A vector of positive integers specifying the design. If not provided then the default is one unit from each group.
indexgroup	The number or the name of the group to be considered as the index group. The design size of the index group should be 1. Further, for each non-index group the size of the group should be at least the product of design size of that group and size of the index group.
.data	Optional argument but recommended. The data frame or matrix of the dataset. The units are recognized by rownames. This is used when further design structure of near fine balance and exact match on certain variable is imposed.
finebalanceVars	An optional character vector of names of the columns of .data on which the matching will be near fine balanced.
exactmatchon	Name of the column of .data on which matching will be exact. (optional)
ordering	Optional vector of size the number of groups, specifying the order in which groups will be matched sequentially.
reorder	Optional logical argument for whether matching will be done after permuting the groups randomly.
verbose	A logical argument. If true some details about the implementation may be prompted when running.

Details

Only required arguments are distmat, grouplabel, design and indexgroup. `.data` is suggested but not required, in which case units are assumed to be labeled 1:length(grouplabel).

Argument `.data` must be provided if structure of fine balance and/or exact match is imposed by arguments `finebalanceVars` and/or `exactmatchon`.

IMPORTANT NOTE: In order to perform matching, `kwaymatching` requires the user to load the optmatch (>= 0.9-1) package separately. The manual loading is required due to software license issues. If the package is not loaded the `kwaymatching` command will fail with an error saying the optmatch package is not present. Reference to optmatch is given below.

Value

A list consisting of the following two elements.

`matches`	A character matrix of size, size of the indexgroup$\times$ `sum(design)`. Each row corresponds to a strata and cell values are the units in the strata.
`cost`	The cost of the final matching calculated as the sum of the average distances between each pair of groups within every strata, and summing over all strata.

Note

For theoretical guarantee it is expected that the distance structure satisfies triangle inequality. If triangle inequality is satisfied than the cost of the solution of this algorithm is at most twice that of the optimal match. Note, for more than two groups the problem of finding the optimal solution is NP-hard.

Author(s)

Bikram Karmakar

References

Karmakar, B., Small, D. S. and Rosenbaum, P. R. (2019) Using Approximation Algorithms to Build Evidence Factors and Related Designs for Observational Studies, Journal of Computational and Graphical Statistics, 28, 698–709.

Examples

```
## USAGE 1
## Not run:
library(optmatch)
## User is required to install and load the optmatch package separately,

data(Dodgeram) # dodge ram pk 2500

grouplabel = 2*Dodgeram$WITHSABS + 1*Dodgeram$NOSAB + 3*Dodgeram$optSAB
```

```
# distance components consists of log propensity
# distance and rank based Mahalanobis distance.
components <- list(prop = c("AGE", "IMPACT3"), mahal = c("SEX.2", "AGE", "FIRE_EXP1.1"))
wgts <- c(10, 5)
distmat <- multigrp_dist_struc(Dodgeram, grouplabel = grouplabel, components, wgts)

# Matching
design = c(1,1,3) # 3 units from the optional period, 1 each from other periods
indexgroup = 2
res = tripletmatching(distmat = distmat, grouplabel = grouplabel, design = design,
                            indexgroup = indexgroup)

# covariance balance
details = 'mean'
covbalance(Dodgeram, grouplabel=c("NOSAB", "optSAB", "WITHSABS"), matches = res,
                          vars = c("AGE", "SEX.2", "IMPACT3.3", "DR_DRINK"), details)

## End(Not run)

## USAGE 2
## Not run:
library(optmatch)
## User is required to install and load the optmatch package separately,

data(Dodgeram)

# Example distance structure
components <- list(prop = c("AGE", "SEX.2", "FR.pass", "REST_USE1", "ROLLOVER1",
            "IMPACT3", "SP_LIMIT", "DR_DRINK", "FIRE_EXP1.1"),
            mahal = c("SEX.2", "AGE", "SP_LIMIT", "DR_DRINK"),
            mahal = c("IMPACT3", "REST_USE1"))
wgts <- c(5, 8, 20)

distmat <- multigrp_dist_struc(Dodgeram, grouplabel = c("NOSAB", "optSAB", "WITHSABS"),
                                    components, wgts)

# Matching with fine balance and exact match
indexgroup = "WITHSABS"
finebalanceVars = c("ROLLOVER1.1", "FIRE_EXP1.1")
exactmatchon = "FR.pass"

res = tripletmatching(distmat = distmat, grouplabel = c("NOSAB", "optSAB", "WITHSABS"),
                design = c(3,3,1), indexgroup = indexgroup, .data = Dodgeram,
                finebalanceVars = finebalanceVars, exactmatchon = exactmatchon)

# covariance balance
vars = c("AGE", "SEX.2", "IMPACT3.3", "DR_DRINK")
details = c('std_diff', 'mean', 'function(x) diff(range(x))',
                          'function(x) quantile(x, probs = .9)')
names(details) <- c('std_diff', 'mean', 'range', '90perc')

covbalance(.data=Dodgeram, grouplabel=c("NOSAB", "optSAB", "WITHSABS"),
                 matches = res, vars = vars, details)
```

```
## End(Not run)
```

`multigrp_dist_struc` *Construct the distance structure for the multiple groups.*

Description

This function can be used to calculate the distance structure for multiple groups. The output of this function can be feed into the argument `distmat` of the main functions `kwaymatching` and `tripletmatching`.

Usage

```
multigrp_dist_struc(.data, grouplabel, components, wgts)
```

Arguments

`.data`	The data frame or matrix of the dataset.
`grouplabel`	The information on the group structure of the units. See description of `kwaymatching` for details on the argument.
`components`	A list specifying the components of the distance structure. Each element of the list is a character vector of column names of the .data on which 'distance' will be calculated. The element names specify the function to be used to calculate to distance of two groups. Element named 'prop' indicates the propensity distance where the propensity is calculated from the specified variable. Element named 'mahal' or 'Mahalanobis' for rank based Mahalanobis distance. User can spacify their own distance function. For example, a function `myDist` should be a function of two arguments: a logical vector of the first group indicator and a data matrix. It should return a numeric matrix of size number of units of first group$\times$number of units of second group. See details for an example.
`wgts`	A non-negative numeric vector of weights of the components.

Details

This function can be used to get distance structure suitable for creating the distances between the units of the groups.

For an example of the kind of user defined distance function that can be used see `smahal` below.

Value

A list describing the distance structure. For detail see the description of the argument `distmat` in the function `kwaymatching`.

Author(s)

Bikram Karmakar

See Also

kwaymatching, tripletmatching

Examples

```
data(Dodgeram)

# Example distance structure
components <- list(prop = c("AGE", "SEX.2", "FR.pass", "REST_USE1", "ROLLOVER1",
            "IMPACT3", "SP_LIMIT", "DR_DRINK", "FIRE_EXP1.1"),
            mahal = c("SEX.2", "AGE", "SP_LIMIT", "DR_DRINK"),
            mahal = c("IMPACT3", "REST_USE1"))
wgts <- c(5, 8, 20)

distmat <- multigrp_dist_struc(Dodgeram,
            grouplabel = c("NOSAB","optSAB","WITHSABS"), components, wgts)

## Propensity score caliper can be implemented mannually

distmat <- multigrp_dist_struc(Dodgeram,
            grouplabel = c("NOSAB","optSAB","WITHSABS"),
            list(mahal = c("SEX.2", "AGE", "SP_LIMIT", "DR_DRINK"),
                mahal = c("IMPACT3", "REST_USE1")), wgts=c(2, 5))
distmat_prop <- multigrp_dist_struc(Dodgeram,
                grouplabel = c("NOSAB", "optSAB", "WITHSABS"),
              list(prop = c("AGE", "SEX.2", "FR.pass", "REST_USE1", "ROLLOVER1",
            "IMPACT3", "SP_LIMIT", "DR_DRINK", "FIRE_EXP1.1")), 1)

## Distance structure with caliper
for(i in 1:length(distmat))
distmat[[i]][distmat_prop[[i]]>.2] <- 100*max(distmat[[i]])



## An example function for argument detail.

smahal <-  function(z,X){
        X<-as.matrix(X)
n<-dim(X)[1]
rownames(X)<-1:n
k<-dim(X)[2]
m<-sum(z)
for (j in 1:k) X[,j]<-rank(X[,j])
cv<-cov(X)
vuntied<-var(1:n)
rat<-sqrt(vuntied/diag(cv))
cv<-diag(rat)
out<-matrix(NA,m,n-m)
```

```
Xc<-X[z==0,,drop=FALSE]
Xt<-X[z==1,,drop=FALSE]
rownames(out)<-rownames(X)[z==1]
colnames(out)<-rownames(X)[z==0]
#library(MASS)
icov<-ginv(cv)
for (i in 1:m)
out[i,]<-mahalanobis(Xc,Xt[i,],icov,inverted=T)
sqrt(out)
}
```

`nrbalancematch` *The is the background function to perform matching using network optimization.*

Description

This function takes as input the information on two groups, treated and control, along with the data structure between the units of these two groups to create a matching. Custom specification of the categorization of the units for near fine balanced design and balance constraints can be provided.

Usage

```
nrbalancematch(cardata.fil, trt_labs, ctrl_labs, stratify, extmatch = NA,
distmat, balanceValues, nmatch = 1)
```

Arguments

`cardata.fil`	A data frame or matrix of the dataset. The units are identified using the rownames of the data.
`trt_labs`	A logical vector of same length as nrow(cardata.fil) for the treated units.
`ctrl_labs`	A logical vector of same length as nrow(cardata.fil) for the control units.
`stratify`	A character vector of same length as nrow(cardata.fil) specifyting the categorization of the units for fine balancing.
`extmatch`	The name of the column of cardata.fil on which design will be exactly matched. This is optional.
`distmat`	A numeric matrix of distances of size the number of treated units $\times$ number of control units. The rows and columns should be named by the unit names.
`balanceValues`	A named integer vector of balance values of the categories.
`nmatch`	Number of control units to be matched to each treated units. A positive integer.

Details

Do not use this function directly unless you are certain of the usage.

IMPORTANT NOTE: In order to perform matching, `kwaymatching` requires the user to load the optmatch (>= 0.9-1) package separately. The manual loading is required due to software license issues. If the package is not loaded the `nrbalancematch` command will fail with an error saying the optmatch package is not present. Reference to optmatch is given below.

Value

A matrix of two columns first column for the treated unit and the second column for the control unit. Units in the first column will be repeated `nmatch` times.

Author(s)

Bikram Karmakar

References

Hansen, B.B. and Klopfer, S.O. (2006) Optimal full matching and related designs via network flows, JCGS 15 609-627.

Karmakar, B., Small, D. S. and Rosenbaum, P. R. (2019) Using Approximation Algorithms to Build Evidence Factors and Related Designs for Observational Studies, Journal of Computational and Graphical Statistics, 28, 698-709.

See Also

`tripletmatching`, `kwaymatching`

Index