Package 'ATEHonest'

November 5, 2020

Title Honest Inference for Treatment Effects under Unconfoundedness

Version 0.1.2

Description Construct matching estimators, and optimal linear estimators, along with honest confidence intervals for conditional and population average treatment effects under unconfoundedness under the assumption that the regression function satisfies a Lipschitz constraint.

Depends R (>= 4.0.0)

License GPL-3

Imports Matrix,
igraph,
MASS,
stats,
methods

Suggests spelling,
testthat,
CVXR,
lpSolve,
knitr,
rmarkdown

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

VignetteBuilder knitr

Language en-US

BugReports https://github.com/kolesarm/ATEHonest/issues

URL https://github.com/kolesarm/ATEHonest

R topics documented:

`ATTEffBounds` *Efficiency bounds for confidence intervals*

Description

Computes the asymptotic efficiency of two-sided fixed-length confidence intervals at smooth functions, as well as the efficiency of one-sided confidence intervals that optimize a given `beta` quantile of excess length, using the formula described in Appendix A of Armstrong and Kolesár (2018)

Usage

```
ATTEffBounds(op, C = 1, beta = 0.8, alpha = 0.05, sigma2, J = 3, DM)
```

Arguments

`op`	The output of `ATTOptPath`.
`C`	Lipschitz smoothness constant
`beta`	quantile `beta` of excess length for determining performance of one-sided CIs.
`alpha`	determines confidence level, `1-alpha`.
`sigma2`	estimate of the conditional variance of the outcome (assuming homoskedasticity). If not supplied, use homoskedastic variance estimate based on a nearest neighbor variance estimator.
`J`	number of nearest neighbors to use when estimating `sigma2`.
`DM`	distance matrix with dimension `n` by `n` to determine nearest neighbors when when estimating `sigma2`.

Value

A list with two elements, `onesided` and `twosided`, for one- and two-sided efficiency.

References

Armstrong, T. B., and M. Kolesár (2018): Finite-Sample Optimal Estimation and Inference on Average Treatment Effects Under Unconfoundedness, https://arxiv.org/abs/1712.04594

Examples

```
## Use NSW experimental subsample with 25 treated and untreated units
dt <- NSWexper[c(1:25, 421:445), ]
Ahalf <- diag(c(0.15, 0.6, 2.5, 2.5, 2.5, 0.5, 0.5, 0.1, 0.1))
D0 <- distMat(dt[, 2:10], Ahalf, method="manhattan", dt$treated)
## Distance matrix for variance estimation
DM <- distMat(dt[, 2:10], Ahalf, method="manhattan")
## Compute the solution path, first 50 steps will be sufficient
op <- ATTOptPath(dt$re78, dt$treated, D0, maxsteps=50)
ATTEffBounds(op, C=1, DM=DM)
```

ATTMatchEstimate *Inference on the CATT and the PATT using the matching estimator*

Description

Computes the matching estimator and confidence intervals (CIs) for the CATT and the PATT. If `ATTMatchPath` used a single `M`, the estimator and CIs are based on a matching estimator with this number of matches. Otherwise, optimize the number of matches according to `opt.criterion`.

Usage

```
ATTMatchEstimate(
  mp,
  C = 1,
  opt.criterion = "RMSE",
  sigma2init,
  sigma2,
  mvar,
  DM,
  alpha = 0.05,
  beta = 0.8,
  J = 3
)
```

Arguments

`mp`	output of `ATTMatchPath`
`C`	Lipschitz smoothness constant
`opt.criterion`	criterion to optimize. One of `"RMSE"` (root mean squared error), `"OCI"` (one-sided confidence intervals), `"FLCI"` (fixed-length two-sided confidence intervals)
`sigma2init`	estimate of the conditional variance of the outcome, used to optimize the number of matches. If not supplied, use homoskedastic variance estimate based on a nearest neighbor variance estimator.

sigma2	vector of variance estimates with length n for determining the conditional standard error of the optimal estimator. In contrast, sigma2init is used only for determining the optimal tuning parameter. If not supplied, use the nearest neighbor variance estimator.
mvar	Marginal variance estimate (variance of the CATT) used to construct CIs for the PATT. If not supplied use the matching estimator of Abadie and Imbens (2006).
DM	distance matrix with dimension n by n to determine nearest neighbors when when estimating sigma2init and sigma2.
alpha	determines confidence level, 1-alpha.
beta	quantile beta of excess length for determining performance of one-sided CIs.
J	number of nearest neighbors to use when estimating sigma2init and sigma2.

Value

Returns an object of class "ATTEstimate". An object of class "ATTEstimate" is a list containing the following components:

e Data frame with columns TODO

k weights TODO

mp TODO

References

Abadie, A. and G. W. Imbens (2006): "Large sample properties of matching estimators for average treatment effects," Econometrica, 74, 235–267.

Armstrong, T. B., and M. Kolesár (2018): Finite-Sample Optimal Estimation and Inference on Average Treatment Effects Under Unconfoundedness, https://arxiv.org/abs/1712.04594

Examples

```
Ahalf <- diag(c(0.15, 0.6, 2.5, 2.5, 2.5, 0.5, 0.5, 0.1, 0.1))
D0 <- distMat(NSWexper[, 2:10], Ahalf, method="manhattan", NSWexper$treated)
mp <- ATTMatchPath(NSWexper$re78, NSWexper$treated, D0, M=c(1, 2), tol=1e-12)
## Distance matrix for variance estimation
DM <- distMat(NSWexper[, 2:10], Ahalf, method="manhattan")
## Estimator based on a single match is better than with 2 matches for RMSE
ATTMatchEstimate(mp, C=1, DM=DM)
```

ATTMatchPath *Matching estimator for the ATT*

Description

Computes the matching estimator and the matching weights for a range of matches M. The output of this function is used as an input for ATTMatchEstimate for inference on the CATT and the PATT.

Usage

```
ATTMatchPath(y, d, D0, M = 1:25, tol = 1e-12)
```

Arguments

`y`	outcome vector with length `n`.
`d`	vector of treatment indicators with length `n`.
`D0`	matrix of distances with dimension `n1` by `n0` between untreated and treated units, where `n0` is the number of untreated units and `n1` is the number of treated units.
`M`	a vector of integers determining the number of matches. If `Inf`, then use the simple difference in means estimator.
`tol`	numerical tolerance for determining nearest neighbors in constructing matches

Value

List with the following components:

ep A data frame with columns `M`, `maxbias`, `att`, and `lindw` corresponding to the number of matches, the scaled worst-case bias, the ATT estimate, and the largest Lindeberg weight.

K A matrix where each row `j` corresponds to the linear weights k used to form the matching estimator with `M[j]` matches.

d Vector of treatment indicators, as supplied by `d`

y Vector of outcomes, as supplied by `y`

tol The tolerance parameter `tol`, as supplied by `tol`

D0 The distance matrix, as supplied by `D0`

Examples

```
## Construct distance matrix
Ahalf <- diag(c(0.15, 0.6, 2.5, 2.5, 2.5, 0.5, 0.5, 0.1, 0.1))
D0 <- distMat(NSWexper[, 2:10], Ahalf, method="manhattan", NSWexper$treated)
mp <- ATTMatchPath(NSWexper$re78, NSWexper$treated, D0, M=1:2, tol=1e-12)
```

`ATTOptEstimate` *Optimal estimation and inference for the CATT and the PATT*

Description

Computes the estimator and confidence intervals (CIs) for the CATT and the PATT. The tuning parameter δ is chosen to optimize `opt.criterion` criterion.

Usage

```
ATTOptEstimate(
  op,
  C = 1,
  opt.criterion = "RMSE",
  sigma2init,
  sigma2,
  mvar,
  DM,
  alpha = 0.05,
  beta = 0.8,
  J = 3,
  extrasteps = 50
)
```

Arguments

`op`	Output of `ATTOptPath`.
`C`	Lipschitz smoothness constant
`opt.criterion`	criterion to optimize. One of `"RMSE"` (root mean squared error), `"OCI"` (one-sided confidence intervals), `"FLCI"` (fixed-length two-sided confidence intervals)
`sigma2init`	estimate of the conditional variance of the outcome, used to choose the optimal smoothing parameter δ. If not supplied, use homoskedastic variance estimate based on a nearest neighbor variance estimator.
`sigma2`	vector of variance estimates with length `n` for determining the conditional standard error of the optimal estimator. In contrast, `sigma2init` is used only for determining the optimal tuning parameter. If not supplied, use the nearest neighbor variance estimator.
`mvar`	Marginal variance estimate (variance of the CATT) used to construct CIs for the PATT. If not supplied use the matching estimator of Abadie and Imbens (2006).
`DM`	distance matrix with dimension `n` by `n` to determine nearest neighbors when when estimating `sigma2init` and `sigma2`.
`alpha`	determines confidence level, `1-alpha`.
`beta`	quantile `beta` of excess length for determining performance of one-sided CIs.
`J`	number of nearest neighbors to use when estimating `sigma2init` and `sigma2`.
`extrasteps`	If the optimal smoothing parameter δ is attained at the end of the solution path, compute additional `extrasteps` steps in the solution path and estimate it again. If `extrasteps==0`, then do not compute extra steps.

Value

Returns an object of class `"ATTEstimate"`. An object of class `"ATTEstimate"` is a list containing the following components:

e Data frame with columns TODO

k weights TODO

res TODO

op TODO

References

Armstrong, T. B., and M. Kolesár (2018): Finite-Sample Optimal Estimation and Inference on Average Treatment Effects Under Unconfoundedness, https://arxiv.org/abs/1712.04594

Examples

```
## Use NSW experimental subsample with 25 treated and untreated units
dt <- NSWexper[c(1:25, 421:445), ]
Ahalf <- diag(c(0.15, 0.6, 2.5, 2.5, 2.5, 0.5, 0.5, 0.1, 0.1))
D0 <- distMat(dt[, 2:10], Ahalf, method="manhattan", dt$treated)
## Distance matrix for variance estimation
DM <- distMat(dt[, 2:10], Ahalf, method="manhattan")
## Compute the solution path, first 50 steps will be sufficient
op <- ATTOptPath(dt$re78, dt$treated, D0, maxsteps=50)
ATTOptEstimate(op, C=1, DM=DM, opt.criterion="RMSE")
ATTOptEstimate(op, C=1, DM=DM, opt.criterion="FLCI")
```

ATTOptPath — *Class of optimal linear estimators for the ATT*

Description

Use a LASSO-like algorithm to compute the solution path $\{\hat{L}_\delta : \delta > 0\}$ tracing out the class of optimal linear estimators that minimize variance subject to a bound on bias. The output of this function is used by `ATTOptEstimate` for optimal estimation and inference on the CATT and PATT

Usage

```
ATTOptPath(y, d, D0, maxsteps = 50, tol, path = NULL, check = FALSE)
```

Arguments

`y`	outcome vector with length `n`.
`d`	vector of treatment indicators with length `n`.
`D0`	matrix of distances with dimension `n1` by `n0` between untreated and treated units, where `n0` is the number of untreated units and `n1` is the number of treated units.
`maxsteps`	maximum number of steps in the solution path. If the full solution path is shorter than `maxsteps`, compute the whole path.
`tol`	numerical tolerance for rounding error when finding the nearest neighbors. All observations with effective distance within `tol` of the closest are considered to be active.

path
: Optionally, supply previous output of `ATTOptPath`. If not provided, the path is started at the beginning (at $\delta = 0$). If provided, it starts at the step where the previous call to `ATTOptPath` ended.

check
: check at each step that the solution matches that obtained by direct optimization using `CVXR-package` (generic convex optimizer package).

Value

A list with the following elements:

y Output vector, as supplied by `y`

d Vector of treatment indicators, as supplied by `d`

D0 Matrix of distances, as supplied by `D0`

res A matrix with rows corresponding to steps in the solution path, so that the maximum number of rows is `maxsteps`, and columns corresponding to the state variables δ, m, r, μ, and `drop`.

K Matrix of weights k associated with the optimal estimator at each step

ep A data frame with columns `delta`, `omega`, `maxbias`, and `att`, corresponding to δ, $\omega(\delta)$, the scaled worst-case bias, and the ATT estimate.

The remaining elements are state variables at the last step of the solution path (see Appendix A in Armstrong and Kolesár (2018) for details and notation):

m0 A vector of length `n0` of corresponding to m.

r0 A vector of length `n1` of corresponding to r.

mu A scalar corresponding to the Lagrange multiplier μ.

D A matrix of effective distances with dimension `n1` by `n0`.

Lam A sparse matrix of Lagrange multipliers Λ with dimension `n1` by `n0`.

N0 A sparse matrix of effective nearest neighbors with dimension `n1` by `n0`.

drop An indicator if an observation has been dropped from an active set, or added.

References

Armstrong, T. B., and M. Kolesár (2018): Finite-Sample Optimal Estimation and Inference on Average Treatment Effects Under Unconfoundedness, https://arxiv.org/abs/1712.04594

Examples

```
x0 <- c(0, 1, 2, 3)
x1 <- c(1, 4, 5)
d <- c(rep(FALSE, length(x0)), rep(TRUE, length(x1)))
D0 <- distMat(c(x0, x1), d=d)
## Compute first three steps
p1 <- ATTOptPath(d, d, D0, maxsteps=3)
## Compute the remaining steps, checking them against CVX solution
ATTOptPath(path=p1, maxsteps=50, check=TRUE)
```

cv *Critical values for inference based on a biased Gaussian estimator.*

Description

Critical value $cv_{1-\alpha}(B)$ such that the confidence interval $X \pm cv_{1-\alpha}(B)$ will have coverage $1-\alpha$, where X is normally distributed with variance equal to 1 and bias bounded by B in absolute value.

Usage

```
cv(B, alpha = 0.05)
```

Arguments

B	Maximum bias, a non-negative vector.
alpha	Scalar between 0 and 1 determining the confidence level, 1-alpha

Value

Critical value $cv_{1-\alpha}(B)$

Examples

```
# 90% critical value:
cv(B = 1, alpha = 0.1)
# 95% critical values
cv(B = c(0, 1, 3), alpha = 0.05)
```

distMat *Matrix of distances between observations*

Description

Compute a matrix of distances between n observations using the distance measure in method.

Usage

```
distMat(X, Ahalf = diag(NCOL(X)), method = "euclidean", d = NULL, p = 2)
```

Arguments

`X`	Design matrix of covariates with dimension `n` by `p`, or else a vector of length `n` if there is a single covariate.
`Ahalf`	Weight matrix with dimension `p` by `p` so that the distances are computed between `Ahalf %*% X[i,]`.
`method`	the distance measure to be used. This must be one of `"euclidean"`, `"maximum"`, `"manhattan"`, `"canberra"`, `"binary"` or `"minkowski"`. Any unambiguous substring can be given.
`d`	Vector of treatment indicators with length `n`. If supplied, return the `n1` by `n0` sub-matrix corresponding to distances between treated and untreated observations. Otherwise return the full `n` by `n` matrix
`p`	The power of the Minkowski distance.

Value

Matrix of distances with dimension `n` by `n` or else `n1` by `n0`

Examples

```
## 4 units, unit 1 and 3 are treated.
distMat(X=c(1, 2, 3, 4), d=c(TRUE, FALSE, TRUE, FALSE))
```

`NSW` *Dataset from Dehejia and Wahba (1999)*

Description

Subset of National Supported Work and PSID data from Dehejia and Wahba (1999).

Usage

```
NSW
```

Format

A data frame with 2,675 observations (2,490 controls from PSID, and 185 treated individuals from NSW) and 11 variables.

treated Treatment indicator

age Age in years

education Year of education

black Indicator for black

hispanic Indicator for Hispanic

married Indicator for married

re74 Earnings in 1974 (in thousands of dollars)

re75 Earnings in 1975 (in thousands of dollars)

re78 Earnings in 1978 (in thousands of dollars)

ue74 Indicator for zero earnings in 1974

ue75 Indicator for zero earnings in 1975

Source

Rajeev Dehejia's website, http://users.nber.org/~rdehejia/nswdata2.html

References

Dehejia, R., and Wahba, S. (1999), "Causal Effects in Nonexperimental Studies: Reevaluating the Evaluation of Training Programs," Journal of the American Statistical Association, 94 (448), 1053-1062.

`NSWexper` *Experimental dataset from Dehejia and Wahba (1999)*

Description

National Supported Work data from Dehejia and Wahba (1999).

Usage

```
NSWexper
```

Format

A data frame with 445 observations (185 treated and 260 controls) and 11 variables.

treated Treatment indicator

age Age in years

education Year of education

black Indicator for black

hispanic Indicator for Hispanic

married Indicator for married

re74 Earnings in 1974 (in thousands of dollars)

re75 Earnings in 1975 (in thousands of dollars)

re78 Earnings in 1978 (in thousands of dollars)

ue74 Indicator for zero earnings in 1974

ue75 Indicator for zero earnings in 1975

Source

Rajeev Dehejia's website, http://users.nber.org/~rdehejia/nswdata2.html

References

Dehejia, R., and Wahba, S. (1999), "Causal Effects in Nonexperimental Studies: Reevaluating the Evaluation of Training Programs," Journal of the American Statistical Association, 94 (448), 1053-1062.

Index