Package ‘ATEHonest’

June 25, 2020

Title Honest Inference for Treatment Effects under Unconfoundedness

Version 0.1.2

Description Construct matching estimators, and optimal linear estimators, along with confidence intervals for conditional and population average treatment effects under unconfoundedness that are valid in finite samples 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 7.1.0

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(h, d, sigma2, C = 1, beta = 0.8, alpha = 0.05)
```

Arguments

`h`	The output of `ATTh`.
`d`	Vector of treatment indicators with length `n`
`sigma2`	Estimate of the conditional variance of the outcome, used to optimize the number of matches.
`C`	Lipschitz smoothness constant
`beta`	The quantile `beta` of excess length for determining performance of one-sided CIs.
`alpha`	Level of confidence interval, `1-alpha`.

References

Armstrong, T. B., and M. Kolesár (2018): Finite-Sample Optimal Estimation and Inference on Average Treatment Effects Under Unconfoundedness, Unpublished manuscript

ATTh *Homotopy for average treatment effect for the treated*

Description

Calculates optimal weights m and r for the control and treated observations as a function of δ, or equivalently μ, using the algorithm described in the appendix to Armstrong and Kolesár (2018)

Usage

```
ATTh(
  D0,
  h,
  maxiter = 50,
  check = FALSE,
  tol = .Machine$double.eps * ncol(D0) * nrow(D0)
)
```

Arguments

`D0`	matrix of distances with dimension `[n1 n0]` between untreated and treated units, where `n0` is the number of untreated units and `n1` is the number of treated units
`h`	Optionally, supply previous output of `ATTh`. If not provided, the homotopy is started at the beginning. If provided, it starts at the step where the previous call to `ATTh` ended.
`maxiter`	maximum number of steps in the homotopy. If the homotopy has less steps than `maxiter`, returns the whole solution path.
`check`	check at each step that the solution matches that in the `CVXR-package` (generic convex optimizer package).
`tol`	numerical tolerance for rounding error when finding the nearest neighbor. All observations with effective distance within `tol` of the closest are considered to be active.

Value

A list with two elements:

res A matrix with rows corresponding to steps in the homotopy, so that the maximum number of rows is `maxiter` (if homotopy started at the beginning), and columns corresponding to δ, m, r, μ, and `drop`, an indicator if an observations has been dropped from an active set, or added

m0 A vector of length `n0` of corresponding to m at the last step.

r0 A vector of length `n1` of corresponding to r at the last step.

mu A scalar corresponding to μ at the last step.

D A matrix of effective distances with dimension `[n1 n0]` at the last step.

Lam A sparse matrix of Lagrange multipliers with dimension `[n1 n0]` at the last step.

N0 A sparse matrix of nearest neighbors with dimension `[n1 n0]` at the last step.

References

Armstrong, T. B., and M. Kolesár (2018): Finite-Sample Optimal Estimation and Inference on Average Treatment Effects Under Unconfoundedness, Unpublished manuscript

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
h <- ATTh(D0, maxiter=3)
## Compute the remaining steps, checking them against CVX solution
h2 <- ATTh(D0, h=h, check=TRUE)
```

ATTMatchEstimate *Inference on the CATT using the matching estimator*

Description

Computes matching estimator and confidence intervals (CIs) for the CATT. 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 from the set in `M` according to `opt.criterion`.

Usage

```
ATTMatchEstimate(
  mp,
  sigma2,
  C = 1,
  sigma2final = sigma2,
  alpha = 0.05,
  beta = 0.8,
  opt.criterion = "RMSE"
)
```

Arguments

`mp`	Output of `ATTMatchPath`
`sigma2`	Estimate of the conditional variance of the outcome, used to optimize the number of matches.
`C`	Lipschitz smoothness constant
`sigma2final`	vector of variance estimates with length `n` for determining the standard error of the optimal estimator. In contrast, `sigma2` is used only for determining the optimal tuning parameter.
`alpha`	Level of confidence interval, `1-alpha`.

`beta`	The quantile `beta` of excess length for determining performance of one-sided CIs.
`opt.criterion`	One of `"RMSE"` (root mean squared error), `"OCI"` (one-sided confidence intervals), `"FLCI"` (fixed-length two-sided confidence intervals)

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

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$treated, D0, M=c(1, 2), tol=1e-12)
## Distance matrix for variance estimation
DM <- distMat(NSWexper[, 2:10], Ahalf, method="manhattan")
sigma2 <- nnvar(DM, NSWexper$treated, NSWexper$re78, J=3)
## Estimator based on a single match is better than with 2 matches for RMSE
ATTMatchEstimate(mp, mean(sigma2), C=1, sigma2final=sigma2)
```

`ATTMatchPath` *Compute the 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.

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 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`, and `att`, corresponding to the number of matches, the scaled worst-case bias, and the CATT estimate.

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`

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=1:2, tol=1e-12)
```

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

Description

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

Usage

```
ATTOptEstimate(
  op,
  sigma2,
  C = 1,
  sigma2final = sigma2,
  alpha = 0.05,
  beta = 0.8,
  opt.criterion = "RMSE"
)
```

Arguments

`op`	Output of `ATTOptPath`.
`sigma2`	Estimate of the conditional variance of the outcome, used to optimize the number of matches.
`C`	Lipschitz smoothness constant
`sigma2final`	vector of variance estimates with length `n` for determining the standard error of the optimal estimator. In contrast, `sigma2` is used only for determining the optimal tuning parameter.
`alpha`	Level of confidence interval, `1-alpha`.

`beta`	The quantile `beta` of excess length for determining performance of one-sided CIs.
`opt.criterion`	One of `"RMSE"` (root mean squared error), `"OCI"` (one-sided confidence intervals), `"FLCI"` (fixed-length two-sided confidence intervals)

Examples

```
Ahalf <- diag(c(0.15, 0.6, 2.5, 2.5, 2.5, 0.5, 0.5, 0.1, 0.1))
## Use NSW experimental subsample with 20 treated and untreated units
dt <- NSWexper[c(1:20, 426:445), ]
D0 <- distMat(dt[, 2:10], Ahalf, method="manhattan", dt$treated)
## Distance matrix for variance estimation
DM <- distMat(dt[, 2:10], Ahalf, method="manhattan")
sigma2 <- nnvar(DM, dt$treated, dt$re78, J=3)
## Compute homotopy/solution path, and the class of optimal estimators based
## on the solution path
h <- ATTh(D0, maxiter=120)
op <- ATTOptPath(h, dt$re78, dt$treated)
ATTOptEstimate(op, mean(sigma2), C=1, sigma2final=sigma2,
               opt.criterion="RMSE")
ATTOptEstimate(op, mean(sigma2), C=1, sigma2final=sigma2,
               opt.criterion="FLCI")
```

`ATTOptPath` *Compute the class of optimal linear estimators for the CATT*

Description

Computes the class of optimal linear estimators that minimize variance subject to a bound on bias, and the optimal weights. The output of this function is used by `ATTOptEstimate` for optimal estimation and inference on the CATT.

Usage

```
ATTOptPath(h, y, d)
```

Arguments

`h`	The output of `ATTh` on which to base the estimate
`y`	Outcome vector with length `n`
`d`	Vector of treatment indicators with length `n`

`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` submatrix 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))
```

`nnvar` *Nearest-neighbor variance estimator*

Description

Calculate an `n`-vector of estimates of the variance of `y` using a nearest-neighbor estimator among observations with the same treatment status `d`.

Usage

```
nnvar(DM, d, y, J = 3, tol = 0)
```

Arguments

- `DM` — distance matrix with dimension `n` by `n`.
- `d` — Vector of treatment indicators with length `n`
- `y` — Outcome vector with length `n`
- `J` — number of nearest neighbors to average over
- `tol` — numerical tolerance for determining nearest neighbors in constructing matches

Value

An `n`-vector of estimates of the variance of `y`.

Examples

```
X <- as.matrix(NSWexper[, 2:10])
DM <- distMat(X, chol(solve(cov(X))), method="euclidean")
sigma2 <- nnvar(DM, d=NSWexper$treated, y=NSWexper$re78, J=3)
```

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