Package ‘RobRex’

February 19, 2015

Version 0.9
Date 2013-09-12
Title Optimally robust influence curves for regression and scale
Description Functions for the determination of optimally robust influence curves in case of linear regression with unknown scale and standard normal distributed errors where the regressor is random.
Depends R (>= 2.14.0), ROptRegTS(>= 0.9)
Author Matthias Kohl <Matthias.Kohl@stamats.de>
Maintainer Matthias Kohl <Matthias.Kohl@stamats.de>
LazyLoad yes
ByteCompile yes
License LGPL-3
Encoding latin1
URL http://robast.r-forge.r-project.org/
LastChangedDate ${LastChangedDate: 2013-09-14 14:22:48 +0200 (Sa, 14. Sep 2013) $}
LastChangedRevision ${LastChangedRevision: 714 $}
SVNRevision 696
NeedsCompilation no
Repository CRAN
Date/Publication 2013-09-14 15:40:39

R topics documented:

  rgsOptIC.AL .......................................................... 2
  rgsOptIC.ALc .......................................................... 3
  rgsOptIC.ALS .......................................................... 4
  rgsOptIC.BM .......................................................... 6
  rgsOptIC.M ............................................................ 7
  rgsOptIC.Mc ........................................................... 8
  rgsOptIC.MK ........................................................... 10
  rgsOptIC.Ms ........................................................... 11
The function `rgsOptIC.AL` computes the optimally robust IC for AL estimators in case of linear regression with unknown scale and (convex) contamination neighborhoods where the regressor is random; confer Subsubsection 7.2.1.1 of Kohl (2005).

**Usage**

```r
rgsOptIC.AL(r, K, theta, scale = 1, A.rg.start = 0, A.sc.start = 0.5,
             bUp = 1000, delta = 1e-06, itmax = 50, check = FALSE)
```

**Arguments**

- `r` non-negative real: neighborhood radius.
- `K` object of class "Distribution".
- `theta` specified regression parameter.
- `scale` specified error scale.
- `A.rg.start` positive definite and symmetric matrix: starting value for the standardizing matrix of the regression part.
- `a.sc.start` real: starting value for centering constant of the scale part.
- `A.sc.start` positive real: starting value for the standardizing constant of the scale part.
- `bUp` positive real: the upper end point of the interval to be searched for b.
- `delta` the desired accuracy (convergence tolerance).
- `itmax` the maximum number of iterations.
- `check` logical. Should constraints be checked.

**Details**

If `theta` is missing, it is set to 0. If `A.rg.start` is missing, the inverse of the second moment matrix of `K` is used. The Lagrange multipliers contained in the expression of the optimally robust IC can be accessed via the accessor functions `cent`, `clip` and `stand`.

**Value**

Object of class "ContIC"

**Author(s)**

Matthias Kohl <Matthias.Kohl@stamats.de>
rgsOptIC.ALc

References

See Also
ContIC-class

Examples

```r
K <- DiscreteDistribution(1:5) # = Unif((1,2,3,4,5))
IC1 <- rgsOptIC.ALc(r = 0.1, K = K)
checkIC(IC1)
Risks(IC1)
cent(IC1)
clip(IC1)
stand(IC1)
```

Computation of the optimally robust IC for AL estimators

Description
The function `rgsOptIC.ALc` computes the optimally robust conditionally centered IC for AL estimators in case of linear regression with unknown scale and average conditional (convex) contamination neighborhoods where the regressor is random; confer Subsubsection 7.2.1.2 of Kohl (2005).

Usage

```r
rgsOptIC.ALc(r, K, theta, scale = 1, A.rg.start, a.sc.start, A.sc.start = 0.5, 
bUp = 1000, delta = 1e-06, itmax = 50, check = FALSE)
```

Arguments

- `r` non-negative real: neighborhood radius.
- `K` object of class "DiscreteDistribution" or object of class "DiscreteMVDistribution".
- `theta` specified regression parameter.
- `scale` specified error scale.
- `A.rg.start` positive definite and symmetric matrix: starting value for the standardizing matrix of the regression part.
- `a.sc.start` real vector: starting values for centering function of the scale part.
- `A.sc.start` positive real: starting value for the standardizing constant of the scale part.
- `bUp` positive real: the upper end point of the interval to be searched for b.
- `delta` the desired accuracy (convergence tolerance).
- `itmax` the maximum number of iterations.
- `check` logical. Should constraints be checked.
Details

If theta is missing, it is set to 0. If A.rm.start is missing, the inverse of the second moment matrix of K is used. In case a.sc.start is missing, it is set to a null vector with length of the support of K.

Value

Object of class "Av1CondContIC"

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

References


See Also

Av1CondContIC-class

Examples

k <- DiscreteDistribution(1:5) # = Unif((1,2,3,4,5))
IC1 <- rgsOptIC.ALs(r = 0.1, K = K)
checkIC(IC1)
Risks(IC1)

rgsOptIC.ALs Computation of the optimally robust IC for ALs estimators

Description

The function rgsOptIC.ALs computes the optimally robust IC for ALs estimators in case of linear regression with unknown scale and (convex) contamination neighborhoods where the regressor is random; confer Subsection 7.3.1 of Kohl (2005).

Usage

rgsOptIC.ALs(r, K, A.rm.start, b.rm.Up = 1000, delta = 1e-06, itmax = 50, check = FALSE)
Arguments

- **r**: non-negative real: neighborhood radius.
- **K**: object of class "Distribution".
- **A.rg.start**: positive definite and symmetric matrix: starting value for the standardizing matrix of the regression part.
- **b.rg.Up**: positive real: the upper end point of the interval to be searched for b.rg.
- **delta**: the desired accuracy (convergence tolerance).
- **itmax**: the maximum number of iterations.
- **check**: logical. Should constraints be checked.

Details

If `A.rg.start` is missing, the inverse of the second moment matrix of `K` is used.

Value

Object of class "ContIC"

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

References


See Also

ContIC-class

Examples

```r
# code takes some time
# Not run:
K <- DiscreteDistribution(1:5) # = Unif((1,2,3,4,5))
ICI <- rgsOptIC.Als(r = 0.1, K = K)
checkIC(ICI)
Risks(ICI)
Infos(ICI)

# End(Not run)
```
Computation of the optimally robust IC for BM estimators

Description

The function \texttt{rgsOptIC.BM} computes the optimally robust IC for BM estimators in case of linear regression with unknown scale and (convex) contamination neighborhoods where the regressor is random. These estimators were proposed by Bednarski and Mueller (2001); confer also Subsection 7.3.3 of Kohl (2005).

Usage

\begin{verbatim}
rgsOptIC.BM(r, K, b_rg.start = 2.5, b_sc.0.x.start, delta = 1e-06,
             MAX = 100, itmax = 1000)
\end{verbatim}

Arguments

- \texttt{r} non-negative real: neighborhood radius.
- \texttt{K} object of class "DiscreteDistribution"
- \texttt{b_rg.start} positive real: starting value for \(b_{rg}\).
- \texttt{b_sc.0.x.start} positive real: starting value for \(b_{sc,0,x}\).
- \texttt{delta} the desired accuracy (convergence tolerance).
- \texttt{itmax} the maximum number of iterations.
- \texttt{MAX} if \(b_{loc}\) or \(b_{sc,0}\) are beyond the admitted values, \texttt{MAX} is returned.

Details

The computation of the optimally robust IC for BM estimators is based on \texttt{optim} where \texttt{MAX} is used to control the constraints on \(b_{rg}\) and \(b_{sc,0,x}\).

Value

Object of class "CondIC"

Author(s)

Matthias Kohl \texttt{<Matthias.Kohl@stamats.de>}

References


rgsOptIC.M

See Also

CondIC-class

Examples

## code takes some time
## Not run:
K <- DiscreteDistribution(1:5) # = Unif((1,2,3,4,5))
IC1 <- rgsOptIC.M(r = 0.1, K = K)
checkIC(IC1)
Risks(IC1)

## End(Not run)

---

**rgsOptIC.M** *Computation of the optimally robust IC for M estimators*

Description

The function `rgsOptIC.M` computes the optimally robust IC for M estimators in case of linear regression with unknown scale and (convex) contamination neighborhoods where the regressor is random; confer Subsubsection 7.2.2.1 of Kohl (2005).

Usage

```r
gsOptIC.M(r, K, A.start, gg.start = 0.6, a1.start = -0.25,
  a3.start = 0.25, B.start, bup = 1000, delta = 1e-05,
  MAX = 100, itmax = 1000, check = FALSE)
```

Arguments

- `r` : non-negative real: neighborhood radius.
- `K` : object of class "Distribution".
- `A.start` : positive definite and symmetric matrix: starting value for the standardizing matrix of the regression part.
- `gg.start` : positive real: starting value for the standardizing constant \( \gamma \) of the scale part.
- `a1.start` : real: starting value for Lagrange multiplier \( \alpha_1 \).
- `a3.start` : real: starting value for Lagrange multiplier \( \alpha_3 \).
- `B.start` : symmetric matrix: starting value for Lagrange multiplier B.
- `bup` : positive real: the upper end point of the interval to be searched for b.
- `delta` : the desired accuracy (convergence tolerance).
- `MAX` : if \( A \) or \( \gamma \) are beyond the admitted values, MAX is returned.
- `itmax` : the maximum number of iterations.
- `check` : logical. Should constraints be checked.
Details
The computation of the optimally robust IC for M estimators is based on \texttt{optim} where \texttt{MAX} is used to control the constraints on A and $\gamma$.

Value
Object of class "IC"

Author(s)
Matthias Kohl <Matthias.Kohl@stamats.de>

References

See Also
IC-class

Examples
```r
## code takes some time
## Not run:
K <- DiscreteDistribution(1:5) # = Unif([1,2,3,4,5])
IC1 <- rgsOptIC.Mc(r = 0.1, K = K)
checkIC(IC1)
Risks(IC1)
## End(Not run)
```

\texttt{rgsOptIC.Mc} \hspace{1cm} \textit{Computation of the optimally robust IC for Mc estimators}

Description
The function \texttt{rgsOptIC.Mc} computes the optimally robust conditionally centered IC for Mc estimators in case of linear regression with unknown scale and average conditional (convex) contamination neighborhoods where the regressor is random; confer Subsubsection 7.2.2.2 of Kohl (2005).

Usage
\begin{verbatim}
rgsOptIC.Mc(r, K, ggLo = 0.5, ggUp = 1, a1.x.start, a3.start = 0.25,
bUp = 1000, delta = 1e-05, itmax = 1000, check = FALSE)
\end{verbatim}
rgsOptIC.Mc

Arguments

- **r**: non-negative real: neighborhood radius.
- **K**: object of class "DiscreteDistribution"
- **ggLo**: positive real: the lower end point of the interval to be searched for $\gamma$.
- **ggUp**: positive real: the upper end point of the interval to be searched for $\gamma$.
- **a1.x.start**: real: starting value for the Lagrange multiplier function $\alpha_1(x)$.
- **a3.start**: real: starting value for Lagrange multiplier $\alpha_3$.
- **bUp**: positive real: the upper end point of the interval to be searched for b.
- **delta**: the desired accuracy (convergence tolerance).
- **itmax**: the maximum number of iterations.
- **check**: logical. Should constraints be checked.

Value

Object of class "CondIC"

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

References


See Also

CondIC-class

Examples

```r
## code takes some time
## Not run:
K <- DiscreteDistribution(1:5) # = Unif([1,2,3,4,5])
IC1 <- rgsOptIC.Mc(r = 0.1, K = K)
checkIC(IC1)
 Risks(IC1)
## End(Not run)
```
Description

The function `rgsOptIC.MK` computes the optimally robust IC for MK estimators in case of linear regression with unknown scale and (convex) contamination neighborhoods where the regressor is random; confer Subsubsection 7.2.2.1 of Kohl (2005).

Usage

```r
rgsOptIC.MK(r, K, ggLo = 0.5, ggUp = 1, a1.start = -0.25, a3.start = 0.25, 
             B.start, bUp = 1000, delta = 1e-06, itmax = 1000, check = FALSE)
```

Arguments

- `r` non-negative real: neighborhood radius.
- `K` object of class "Distribution".
- `ggLo` positive real: the lower end point of the interval to be searched for $\gamma$.
- `ggUp` positive real: the upper end point of the interval to be searched for $\gamma$.
- `a1.start` real: starting value for Lagrange multiplier $\alpha_1$.
- `a3.start` real: starting value for Lagrange multiplier $\alpha_3$.
- `B.start` symmetric matrix: starting value for Lagrange multiplier B.
- `bUp` positive real: the upper end point of the interval to be searched for b.
- `delta` the desired accuracy (convergence tolerance).
- `itmax` the maximum number of iterations.
- `check` logical. Should constraints be checked.

Value

Object of class "IC"

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

References


See Also

IC-class
Examples

```r
## code takes some time
## Not run:
K <- DiscreteDistribution(1:5) # = Unif([1,2,3,4,5])
IC1 <- rgsOptIC.Ms(r = 0.1, K = K)
checkIC(IC1)
Risks(IC1)

## End(Not run)
```

**Computation of the optimally robust IC for Ms estimators**

**Description**

The function `rgsOptIC.Ms` computes the optimally robust conditionally centered IC for Ms estimators in case of linear regression with unknown scale and average conditional (convex) contamination neighborhoods where the regressor is random; confer Subsection 7.3.2 of Kohl (2005).

**Usage**

```r
rgsOptIC.Ms(r, K, a1.x.start, a3.start = 0.25, b.sc.start = 1.5,
            bUp = 1000, ggLo = 0.5, ggUp = 1, delta = 1e-06,
            itmax = 1000, check = FALSE)
```

**Arguments**

- `r` non-negative real: neighborhood radius.
- `K` object of class "DiscreteDistribution"
- `ggLo` positive real: the lower end point of the interval to be searched for \( \gamma \).
- `ggUp` positive real: the upper end point of the interval to be searched for \( \gamma \).
- `a1.x.start` real: starting value for the Lagrange multiplier function \( \alpha_1(x) \).
- `a3.start` real: starting value for Lagrange multiplier \( \alpha_3 \).
- `b.sc.start` positive real: starting value for the clipping bound \( b_{sc} \).
- `bUp` positive real: the upper end point of the interval to be searched for b.
- `delta` the desired accuracy (convergence tolerance).
- `itmax` the maximum number of iterations.
- `check` logical. Should constraints be checked.

**Value**

Object of class "CondIC"
Author(s)
Matthias Kohl <Matthias.Kohl@stamats.de>

References

See Also
CondIC-class

Examples
```r
## code takes some time
## Not run:
K <- DiscreteDistribution(1:5) # = Unif(c(1,2,3,4,5))
IC1 <- rgsOptIC.Ms(r = 0.1, K = K)
checkIC(IC1)
Risks(IC1)

## End(Not run)
```
Index

*Topic robust
  rgsOptIC.AL, 2
  rgsOptIC.ALC, 3
  rgsOptIC.ALS, 4
  rgsOptIC.BM, 6
  rgsOptIC.M, 7
  rgsOptIC.Mc, 8
  rgsOptIC.MK, 10
  rgsOptIC.Ms, 11
  rgsOptIC.AL, 2
  rgsOptIC.ALC, 3
  rgsOptIC.ALS, 4
  rgsOptIC.BM, 6
  rgsOptIC.M, 7
  rgsOptIC.Mc, 8
  rgsOptIC.MK, 10
  rgsOptIC.Ms, 11