RobASt: Implementation of optimally robust estimators

Dr. Matthias Kohl

Department of Mathematics

Fraunhofer ITWM
Kaiserslautern June 12

Outline

1. Introduction
2. Important S4 classes
3. Implementation of Algorithms
4. To-Do list

UNIVERSITÄT BAYREUTH

Outline

1 Introduction

2 Important S4 classes

3 Implementation of Algorithms

4 To-Do list

A first simple example

```
R > ## 24 determinations of copper in wholemeal flour
R > library(MASS)
R > data(chem)
R > sort(chem)

 [1]  2.20  2.20  2.40  2.40  2.50  2.70  2.80  2.90  3.03
[10]  3.03  3.10  3.37  3.40  3.40  3.40  3.50  3.60  3.70
[19]  3.70  3.70  3.70  3.77  5.28 28.95

R > library(ROptEst)

R > robEst <- roptest(x = chem,
+                     L2Fam = NormLocationScaleFamily(),
+                     neighbor = ContNeighborhood(), # default
+                     eps.lower = 0.05, eps.upper = 0.20, # rmx-estimator
+                     risk = asMSE(), # default
+                     distance = CvMDist, # default
+                     steps = 3L, useLast = TRUE)
```

How to obtain information about robust estimates

```
R > distrModOptions("show.details" = "minimal") # to reduce output
R > robEst

Evaluations of 3-step estimate:
-------------------------------
     mean          sd
  3.1649732   0.6668010
 (0.1513134) (0.1127157)

R > ## confidence intervals
R > confint(robEst, method = symmetricBias())

A[n] asymptotic (LAN-based), uniform (bias-aware)
 confidence interval:
for symmetric Bias
         2.5 %    97.5 %
mean 2.7676325 3.5623140
sd   0.3708158 0.9627862
```

Some diagnostics

```
R > ## check precision of computations
R > checkIC(pIC(robEst), out = FALSE)

maximum deviation
     1.508162e-05
```

Some diagnostic plots:

plot: plot the IC.

infoPlot: plot relative and absolute information; cf. [1].

outlyingPlotIC: outlyingness plot based on distances applied to ICs.

comparePlot: compare influence curves of AL estimators.

Move IC to another parameter value (or ideal model)

```
R > head(modifyIC(pIC(robEst)), n = 19)

1  function (L2Fam, IC)
2  {
3      ICL2Fam <- eval(CallL2Fam(IC))
4      if (is(L2Fam, "L2LocationScaleFamily") && is(distribution(L2Fam),
5          class(distribution(ICL2Fam)))) {
6          sdneu <- main(L2Fam)[2]
7          sdalt <- main(ICL2Fam)[2]
8          r <- neighborRadius(IC)
9          w <- weight(IC)
10         clip(w) <- sdneu * clip(w)/sdalt
11         cent(w) <- sdalt * cent(w)/sdneu
12         stand(w) <- sdneu^2 * stand(w)/sdalt^2
13         weight(w) <- getweight(w, neighbor = ContNeighborhood(radius = r),
14             biastype = biastype(IC), normW = normtype(IC))
15         A <- sdneu^2 * stand(IC)/sdalt^2
16         b <- sdneu * clip(IC)/sdalt
17         a <- sdneu * cent(IC)/sdalt
18         mse <- sum(diag(A))
19         Cov <- sdneu^2 * Risks(IC)$asCov/sdalt^2

R > tail(modifyIC(pIC(robEst)), n = 4)

31     else {
32         makeIC(IC, L2Fam)
33     }
34 }
```

Finite-sample correction

```
R > Infos(pIC(robEst))[2,2]

[1] "least favorable radius: 0.526"

R > ## finite-sample correction for normal location and scale
R > library(RobLox)
R > (r.fi <- finiteSampleCorrection(r = 0.526, n = length(chem)))

[1] 1.74

R > ## use argument fsCor or directly
R > robEst.fi <- roptest(x = chem, L2Fam = NormLocationScaleFamily(),
+                        eps = r.fi/sqrt(length(chem)),
+                        steps = 3L, useLast = TRUE)

R > estimate(robEst)

    mean       sd
3.164973 0.666801

R > estimate(robEst.fi)

     mean       sd
3.2840688 0.6779314
```

Outline

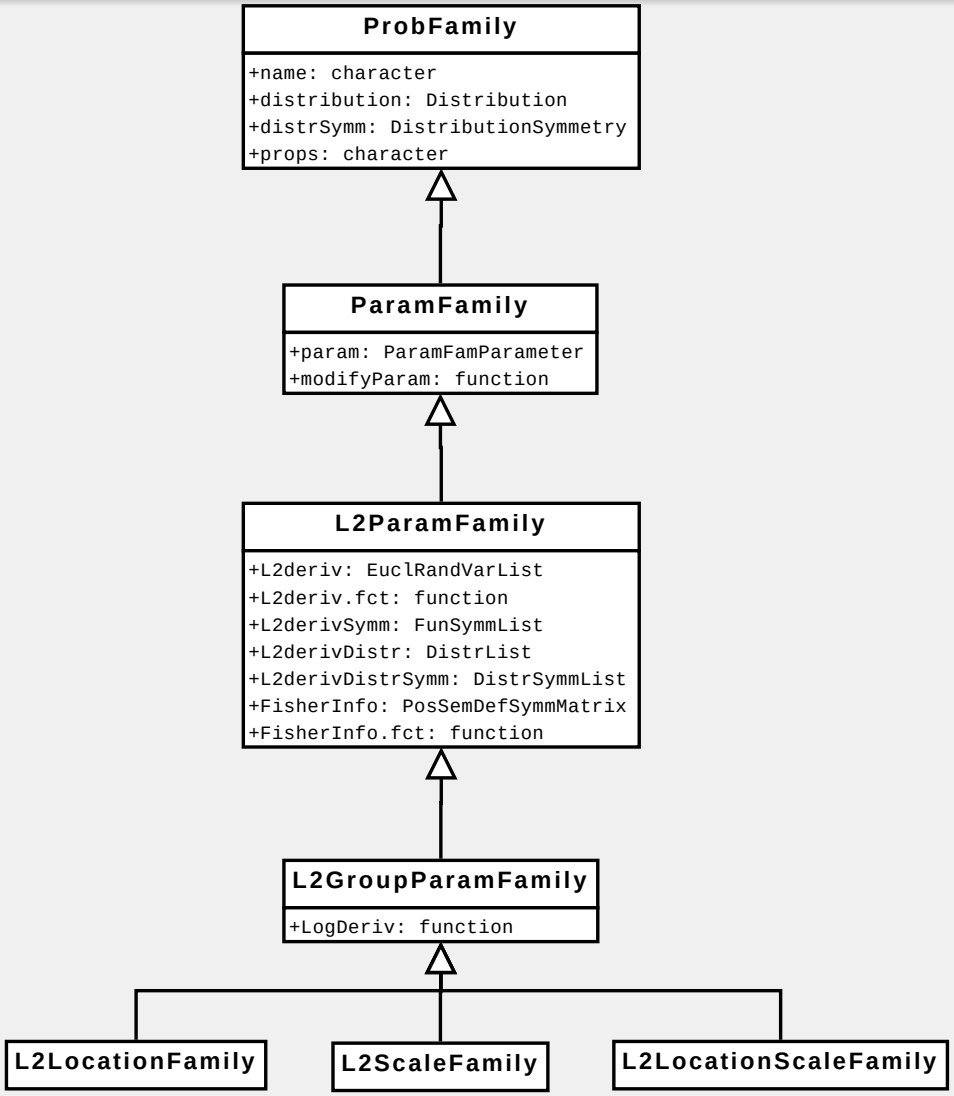
1 Introduction

2 Important S4 classes

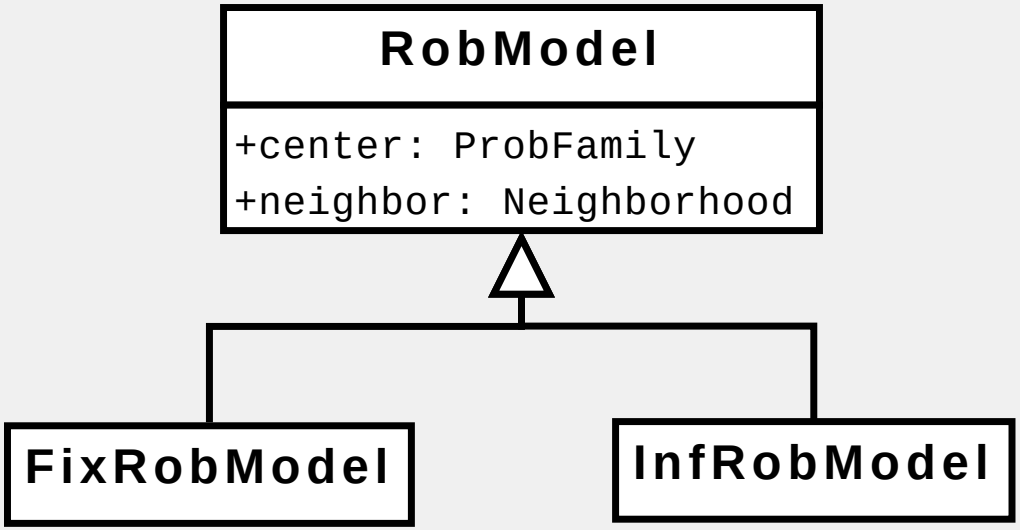
3 Implementation of Algorithms

4 To-Do list

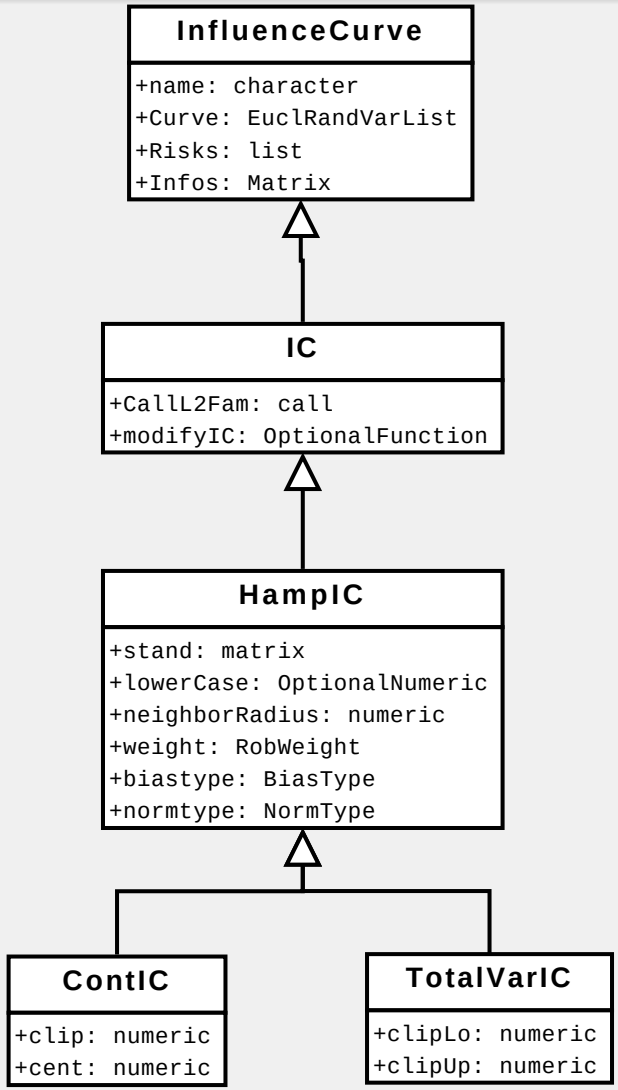
Ideal family



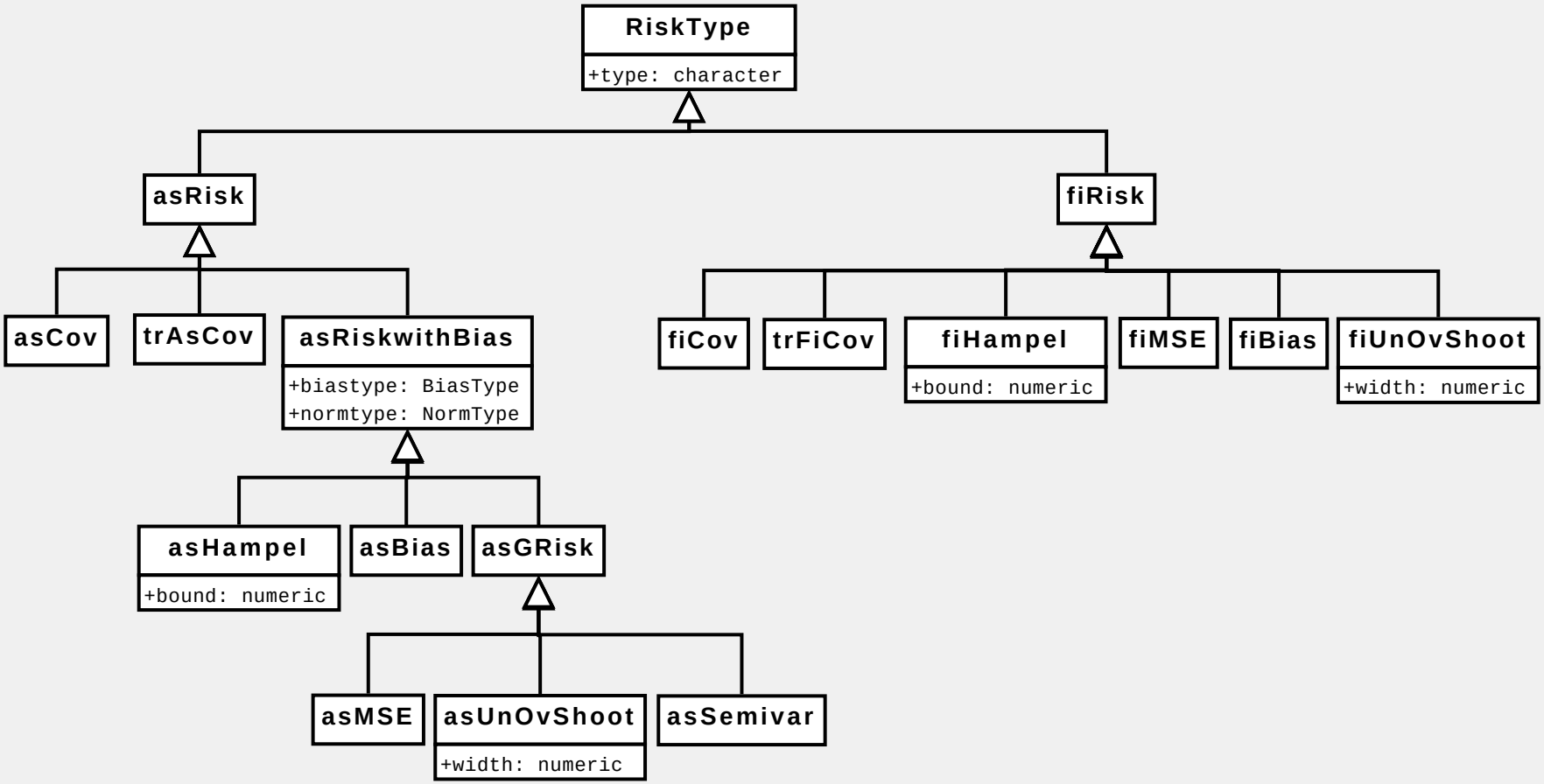
Robust model



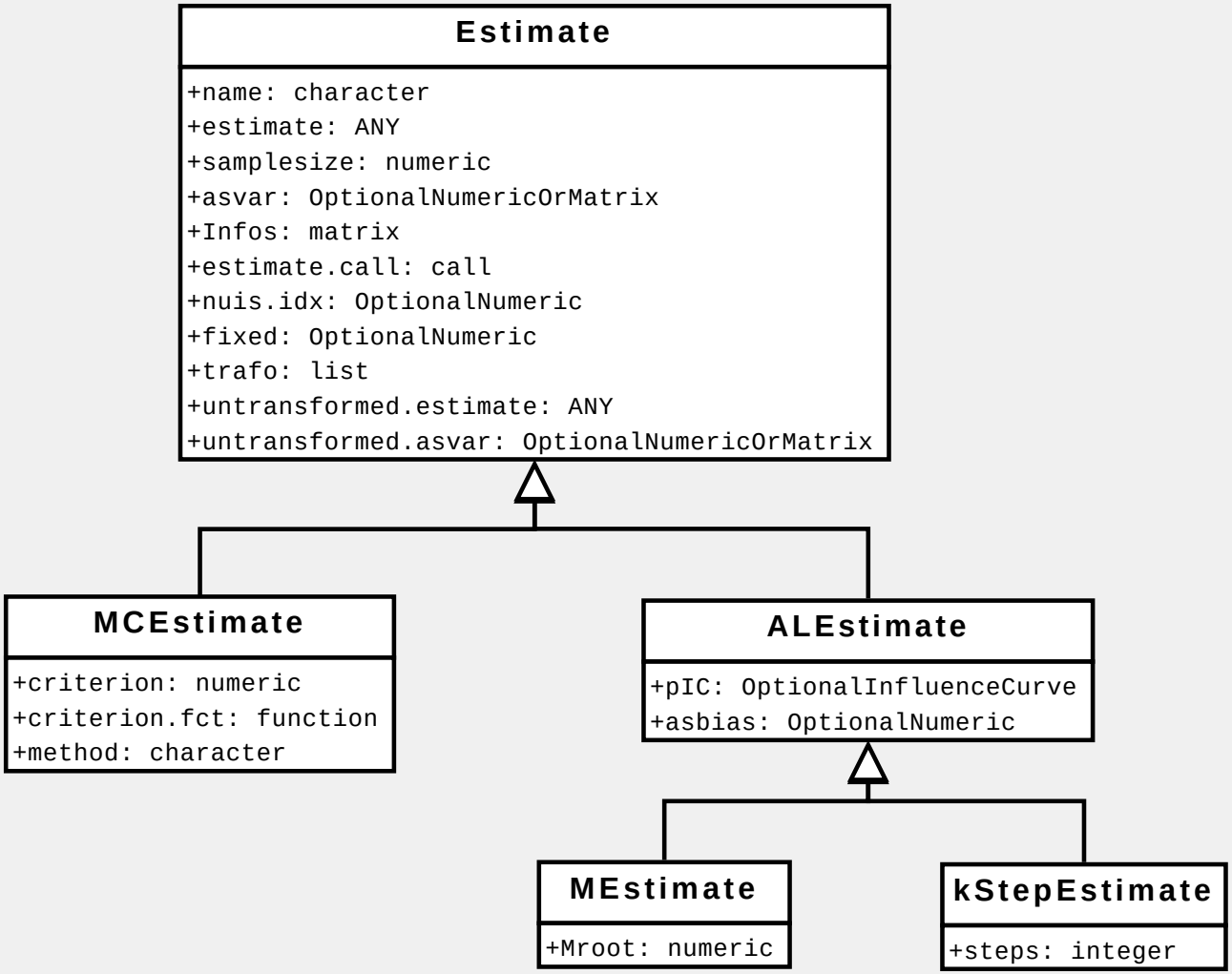
Influence curve



Risk type


RiskType
+type: character
asRisk
fiRisk
asCov
trAsCov
asRiskwithBias
+biastype: BiasType
+normtype: NormType
fiCov
trFiCov
fiHampel
+bound: numeric
fiMSE
fiBias
fiUnOvShoot
+width: numeric
asHampel
+bound: numeric
asBias
asGRisk
asMSE
asUnOvShoot
+width: numeric
asSemivar

Estimate


Estimate
+name: character
+estimate: ANY
+samplesize: numeric
+asvar: OptionalNumericOrMatrix
+Infos: matrix
+estimate.call: call
+nuis.idx: OptionalNumeric
+fixed: OptionalNumeric
+trafo: list
+untransformed.estimate: ANY
+untransformed.asvar: OptionalNumericOrMatrix
MCEstimate
+criterion: numeric
+criterion.fct: function
+method: character
ALEstimate
+pIC: OptionalInfluenceCurve
+asbias: OptionalNumeric
MEstimate
+Mroot: numeric
kStepEstimate
+steps: integer

Outline

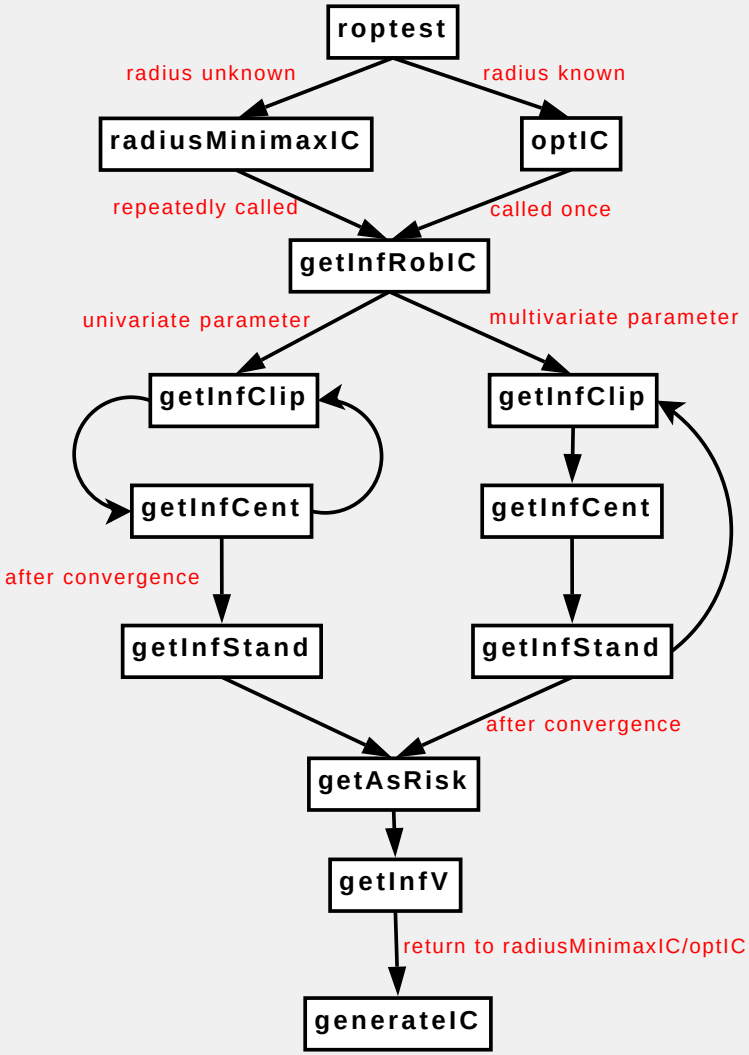
1 Introduction

2 Important S4 classes

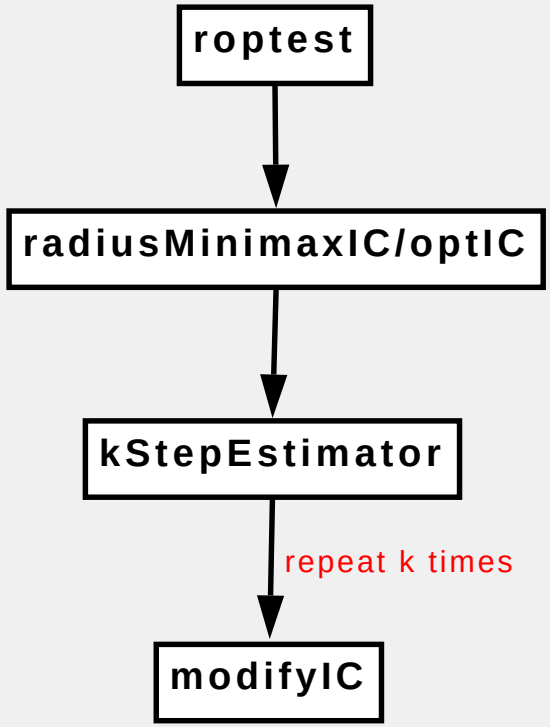
3 Implementation of Algorithms

4 To-Do list

Computation of optimally robust influence curves



Computation of k-step estimator



Note: modifyIC is slot of class IC!

Outline

1 Introduction

2 Important S4 classes

3 Implementation of Algorithms

4 To-Do list

To-Do list

- Can implementation of L2-derivative be improved?
- Move symmetry information to RandVar and Distribution classes?

Bibliography I


M. Kohl (2005).
Numerical Contributions to the Asymptotic Theory of Robustness.
Dissertation. University of Bayreuth.


H. Rieder (1994).
Robust Asymptotic Statistics.
Springer.


Kohl, M. (2008).
RobLox: Optimally robust influence curves for location and scale. *R package version 0.7*, URL http://robast.r-forge.r-project.org


Kohl, M., and Ruckdeschel, P. (2008a).
RandVar: Implementation of random variables. *R package version 0.7*, URL http://robast.r-forge.r-project.org


Kohl, M. and Ruckdeschel, P. (2008b).
RobAStBase: Robust Asymptotic Statistics. *R package version 0.7*, URL http://robast.r-forge.r-project.org


Kohl, M. and Ruckdeschel, P. (2008c).
ROptEst: Optimally robust estimation. *R package version 0.7*, URL http://robast.r-forge.r-project.org


M. Kohl, P. Ruckdeschel, and H. Rieder (2009).
Infinitesimally Robust Estimation in General Smoothly Parametrized Models.
Submitted. http://arxiv.org/abs/0901.3531

UNIVERSITÄT BAYREUTH

Bibliography II


H. Rieder, M. Kohl, and P. Ruckdeschel (2008).
The Costs of not Knowing the Radius.
Statistical Methods and Application 17(1):13–40.


Ruckdeschel, P. (2005).
Optimally One-Sided Bounded Influence Curves.
Mathematical Methods in Statistics 14(1), 105–131.


P. Ruckdeschel and H. Rieder (2004).
Optimal influence curves for general loss functions.
Stat. Decis. 22:201–223.


R Development Core Team (2009).
R: A language and environment for statistical computing.

R Foundation for Statistical Computing, Vienna, Austria. `http://www.R-project.org`.