R-Packages for Robust Asymptotic Statistics

Dr. Matthias Kohl

Chair for Stochastics

UNIVERSITÄT BAYREUTH

joint work with
Dr. Peter Ruckdeschel
Fraunhofer ITWM

useR! – The R User Conference 2008
Dortmund August 12

Outline

Outline

Setup I

Ideal model: L_2-differentiable parametric family of probability measures, parameter space: $\Theta \subset \mathbb{R}^k$ (open)

Estimator class: asymptotically linear estimators (ALEs) S_n

$$S_n(x_1, \dots, x_n) = \theta + \frac{1}{n}\sum_{i=1}^{n} \psi_\theta(x_i) + R_n$$

$x_1, \dots, x_n$: sample

ψ_θ: influence curve/function (IC) at $\theta \in \Theta$

R_n: asymptotically negligible remainder

E.g. as. normal M-, L-, R-, S- and MD-estimators

Setup I

Ideal model: L_2-differentiable parametric family of probability measures, parameter space: $\Theta \subset \mathbb{R}^k$ (open)

Estimator class: asymptotically linear estimators (ALEs) S_n

$$S_n(x_1, \ldots, x_n) = \theta + \frac{1}{n} \sum_{i=1}^{n} \psi_\theta(x_i) + R_n$$

$x_1, \ldots, x_n$: sample

ψ_θ: influence curve/function (IC) at $\theta \in \Theta$

R_n: asymptotically negligible remainder

E.g. as. normal M-, L-, R-, S- and MD-estimators

Setup II

Infinitesimal neighborhood: deviations (gross errors, outliers, etc.) from the ideal model P_θ of form

$$d_*(P_\theta, Q) = \frac{r}{\sqrt{n}} =: r_n \qquad Q \in \mathcal{M}_1$$

$\mathcal{M}_1$: set of all probability measures

d_*: some distance or pseudo-distance

r: radius in $[0, \sqrt{n}]$

E.g. Tukey's gross error model

$$Q = (1 - r_n)P_\theta + r_n H_n \qquad H_n \in \mathcal{M}_1$$

Optimally robust ALEs

Optimization problem:

$$\mathrm{G}\big(\mathrm{asBias}\,(\mathrm{S_n}), \mathrm{asVar}\,(\mathrm{S_n})\big) = \text{min!}$$

G: positive, convex, strictly increasing in both args

$\mathrm{asBias}\,(\mathrm{S_n})$: some function of ψ_θ (IC)

$\mathrm{asVar}\,(\mathrm{S_n})$: some function of ψ_θ (IC)

Hence: minimum is taken over all ICs ψ_θ

Optimal solutions: Rieder (1994) [3], Ruckdeschel and Rieder (2004) [10], Kohl (2005) [2]

Unknown radius: radius-minimax estimator; cf. Rieder et al. (2008) [8]

UNIVERSITÄT BAYREUTH

Optimally robust ALEs

Optimization problem:

$$\mathrm{G}\big(\mathrm{asBias}\,(\mathrm{S_n}),\mathrm{asVar}\,(\mathrm{S_n})\big) = \text{min!}$$

G: positive, convex, strictly increasing in both args

$\mathrm{asBias}\,(\mathrm{S_n})$: some function of ψ_θ (IC)

$\mathrm{asVar}\,(\mathrm{S_n})$: some function of ψ_θ (IC)

Hence: minimum is taken over all ICs ψ_θ

Optimal solutions: Rieder (1994) [3], Ruckdeschel and Rieder (2004) [10], Kohl (2005) [2]

Unknown radius: radius-minimax estimator; cf. Rieder et al. (2008) [8]

Optimally robust ALEs

Optimization problem:

$$\mathrm{G}\big(\mathrm{asBias}\,(\mathrm{S_n}), \mathrm{asVar}\,(\mathrm{S_n})\big) = \text{min!}$$

G: positive, convex, strictly increasing in both args

$\mathrm{asBias}\,(\mathrm{S_n})$: some function of ψ_θ (IC)

$\mathrm{asVar}\,(\mathrm{S_n})$: some function of ψ_θ (IC)

Hence: minimum is taken over all ICs ψ_θ

Optimal solutions: Rieder (1994) [3], Ruckdeschel and Rieder (2004) [10], Kohl (2005) [2]

Unknown radius: radius-minimax estimator; cf. Rieder et al. (2008) [8]

Optimally robust estimation

Possible steps to compute an optimally robust estimator:

1. Decide on ideal model, neighborhood and risk
2. Try to find a rough estimate for the amount $r_n \in [0,1]$ of gross errors such that $r_n \in [\underline{r_n}, \overline{r_n}]$.
3. Choose and evaluate appropriate initial estimate; e.g., Kolmogorov or Cramér von Mises MD-estimator
4. Estimate the parameter(s) of interest by means of the corresponding radius-minimax estimator (cf. Rieder et al. (2008) [8]) using a k-step ($k \geq 1$) construction.

Optimally robust estimation

Possible steps to compute an optimally robust estimator:

1. Decide on ideal model, neighborhood and risk
2. Try to find a rough estimate for the amount $\mathrm{r_n} \in [0, 1]$ of gross errors such that $\mathrm{r_n} \in [\underline{\mathrm{r_n}}, \overline{\mathrm{r_n}}]$.
3. Choose and evaluate appropriate initial estimate; e.g., Kolmogorov or Cramér von Mises MD-estimator
4. Estimate the parameter(s) of interest by means of the corresponding radius-minimax estimator (cf. Rieder et al. (2008) [8]) using a k-step ($\mathrm{k} \geq 1$) construction.

Optimally robust estimation

Possible steps to compute an optimally robust estimator:

1. Decide on ideal model, neighborhood and risk
2. Try to find a rough estimate for the amount $\mathrm{r_n} \in [0, 1]$ of gross errors such that $\mathrm{r_n} \in [\underline{\mathrm{r_n}}, \overline{\mathrm{r_n}}]$.
3. Choose and evaluate appropriate initial estimate; e.g., Kolmogorov or Cramér von Mises MD-estimator
4. Estimate the parameter(s) of interest by means of the corresponding radius-minimax estimator (cf. Rieder et al. (2008) [8]) using a k-step ($\mathrm{k} \geq 1$) construction.

Optimally robust estimation

Possible steps to compute an optimally robust estimator:

1. Decide on ideal model, neighborhood and risk
2. Try to find a rough estimate for the amount $\mathrm{r_n} \in [0,1]$ of gross errors such that $\mathrm{r_n} \in [\underline{\mathrm{r_n}}, \overline{\mathrm{r_n}}]$.
3. Choose and evaluate appropriate initial estimate; e.g., Kolmogorov or Cramér von Mises MD-estimator
4. Estimate the parameter(s) of interest by means of the corresponding radius-minimax estimator (cf. Rieder et al. (2008) [8]) using a k-step ($\mathrm{k} \geq 1$) construction.

Outline

Some examples

- Normal (Gaussian): location and scale
- Binomial: probability of success
- Poisson: positive mean
- Gamma: shape and scale
- Gumbel: location and scale
- all smoothly parameterized exponential families of full rank

- Approach also works for other smoothly parametrized families!

Some examples

- Normal (Gaussian): location and scale
- Binomial: probability of success
- Poisson: positive mean
- Gamma: shape and scale
- Gumbel: location and scale
- all smoothly parameterized exponential families of full rank
- Approach also works for other smoothly parametrized families!

Some examples

- Normal (Gaussian): location and scale
- Binomial: probability of success
- Poisson: positive mean
- Gamma: shape and scale
- Gumbel: location and scale
- all smoothly parameterized exponential families of full rank

- Approach also works for other smoothly parametrized families!

Basic R-Packages

- distr: S4-classes for distributions.
- distrEx: Functionals on distributions.
- RandVar: S4-classes and methods for random variables.
- distrMod: S4-classes for parametric families of probability measures, minimum distance (MD) estimators.
- RobAStBase: S4-classes for ICs and infinitesimal neighborhoods.

cf. Ruckdeschel et al. (2006) [9], Kohl (2005) [2], `http://r-forge.r-project.org/projects/distr/`,

`http://r-forge.r-project.org/projects/robast/`

Basic R-Packages

distr: S4-classes for distributions.

distrEx: Functionals on distributions.

RandVar: S4-classes and methods for random variables.

distrMod: S4-classes for parametric families of probability measures, minimum distance (MD) estimators.

RobAStBase: S4-classes for ICs and infinitesimal neighborhoods.

cf. Ruckdeschel et al. (2006) [9], Kohl (2005) [2], http://r-forge.r-project.org/projects/distr/,

http://r-forge.r-project.org/projects/robast/

R-Packages for optimally robust estimation

Devel version 0.6 (version 0.5 on CRAN)

ROptEst: Optimally robust estimation for L2 differentiable parametric families.

RobLox: Optimally robust estimation for normal (Gaussian) location and scale (optimized for speed).

cf. Ruckdeschel et al. (2006) [9], Kohl (2005) [2], http://r-forge.r-project.org/projects/distr/,

http://r-forge.r-project.org/projects/robast/

R-Packages for optimally robust estimation

Devel version 0.6 (version 0.5 on CRAN)

ROptEst: Optimally robust estimation for L2 differentiable parametric families.

RobLox: Optimally robust estimation for normal (Gaussian) location and scale (optimized for speed).

cf. Ruckdeschel et al. (2006) [9], Kohl (2005) [2], http://r-forge.r-project.org/projects/distr/,

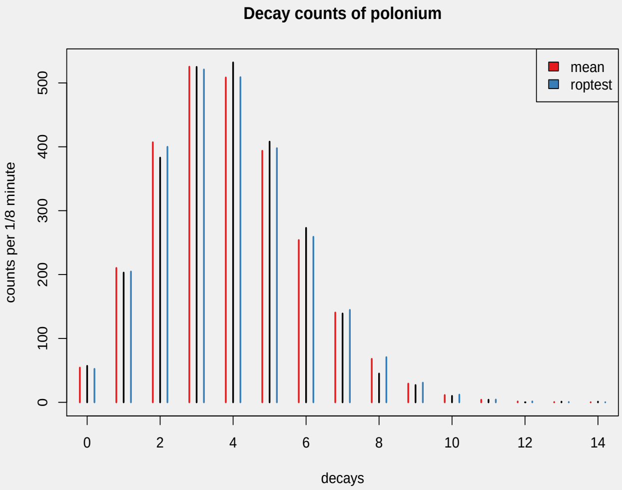
http://r-forge.r-project.org/projects/robast/

Example 1: Poisson

Decay counts of polonium by Rutherford and Geiger (1910); cf. Feller (1968)[1]

```
R > table(x)

x
  0   1   2   3   4   5   6   7   8   9  10  11  13  14
 57 203 383 525 532 408 273 139  45  27  10   4   1   1

R > ## ML-estimate
R > mean(x)

[1] 3.871549

R > ## or with package distrMod
R > MLest <- MLEstimator(x, PoisFamily(), interval = c(0, 10))
R > estimate(MLest)

  lambda
3.871547

R > ## Optimally robust 3-step estimate from package ROptEst (version 0.6.0)
R > ## takes about 4 sec (Centrino Duo 1.66 GHz)
R > ROest <- roptest(x, PoisFamily(), eps.upper = 0.05, interval = c(0, 10), steps = 3)
R > estimate(ROest)

  lambda
3.907973
```

Example 1: Poisson - comparison of results

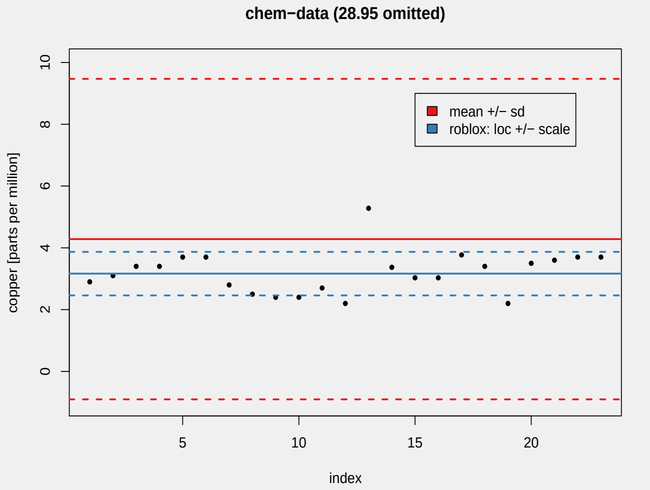


Example 2: Normal location and scale

Copper in wholemeal flour; cf. MASS [4]

```
R > chem

 [1]  2.90  3.10  3.40  3.40  3.70  3.70  2.80  2.50  2.40  2.40  2.70  2.20
[13]  5.28  3.37  3.03  3.03 28.95  3.77  3.40  2.20  3.50  3.60  3.70  3.70

R > ## ML-estimate (mean and sd) from package distrMod
R > MLest <- MLEstimator(chem, NormLocationScaleFamily())

R > ## median and MAD
R > initial.est <- c(median(chem), mad(chem))

R > ## Optimally robust 3-step estimate from package ROptEst (version 0.6.0)
R > ## takes about 80 sec (Centrino Duo 1.66 GHz)
R > ROest1 <- roptest(chem, NormLocationScaleFamily(), eps.upper = 0.05, steps = 3,
+                     initial.est = initial.est)

R > ## Use package RobLox (version 0.6.0) which is optimized for speed!
R > ## takes about 0.12 sec (Centrino Duo 1.66 GHz)
R > ROest2 <- roblox(chem, eps.upper = 0.05, k = 3, returnIC = TRUE)
```

Example 2: Normal location and scale



Example 3: Affymetrix gene expression data

Extract log-PM (perfect match) data from a HG U133+ 2.0 array

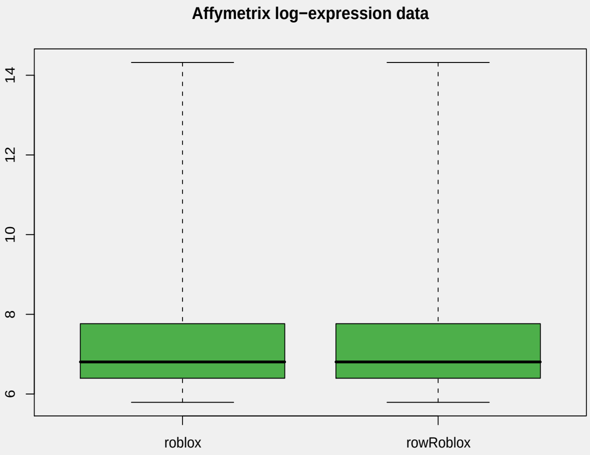
```
R > library(MAQCsubsetAFX)
R > data(refA)
R > ex.data <- refA[,1]
R > CDFINFO <- getCdfInfo(ex.data)
R > ids <- featureNames(ex.data)
R > INDEX <- sapply(ids, get, envir = CDFINFO)
R > NROW <- unlist(lapply(INDEX, nrow))
R > table(NROW)

NROW
    8     9    10    11    13    14    15    16    20    69
    5     1     6 54130     4     4     2   482    40     1

R > rawData <- intensity(ex.data)
R > fun <- function(INDEX, x) log2(x[INDEX[,1], ])
R > logPM <- lapply(INDEX, fun, x = rawData)
```

Example 3: Affymetrix gene expression data

Optimally robust estimation of location and scale for each Affymetrix ID via roblox and rowRoblox

```
R > ## takes about 17 minutes (Centrino Duo 1.66 GHz)
R > ROest1 <- lapply(logPM, function(x) estimate(roblox(x)))

R > ## takes about 1.3 sec (Centrino Duo 1.66 GHz)
R > nr <- as.integer(names(table(NROW)))
R > ROest2 <- matrix(NA, ncol = 2, nrow = length(NROW))
R > for(k in nr){
+     ind <- which(NROW == k)
+     temp <- do.call(rbind, logPM[ind])
+     ROest2[ind, 1:2] <- estimate(rowRoblox(temp))
+ }

R > ## maximum deviation roblox vs. rowRoblox: location
R > max(abs(unlist(ROest1)[seq(1, 2*54675-1, 2)] - ROest2[,1]))

[1] 5.640855e-06

R > ROest12 <- unlist(ROest1)[seq(2, 2*54675, 2)]

R > ## maximum deviation roblox vs. rowRoblox: scale
R > max(abs(unlist(ROest1)[seq(2, 2*54675, 2)] - ROest2[,2]))

[1] 2.591696e-06
```

Example 3: Affymetrix gene expression data



Outline

Regression-Type Models

Devel version 0.6 (version 0.5 on CRAN)

ROptRegTS: Optimally robust estimation for regression and time series models.

RobRex: Optimally robust estimation for linear regression with normal errors.

Kohl (2005) [2], http://r-forge.r-project.org/projects/robast/

Regression-Type Models

Devel version 0.6 (version 0.5 on CRAN)

ROptRegTS: Optimally robust estimation for regression and time series models.

RobRex: Optimally robust estimation for linear regression with normal errors.

Kohl (2005) [2], http://r-forge.r-project.org/projects/robast/

Current developments

- Confidence intervals
- Diagnostic plots
- Simpler user interfaces for regression models

Thank you!

Current developments

- Confidence intervals
- Diagnostic plots
- Simpler user interfaces for regression models

Thank you!

Bibliography I

M. Feller (1968).
An introduction to probability theory and its applications. I.
John Wiley and Sons.

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

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

Venables, W. N. and Ripley, B. D. (2002).
Modern Applied Statistics with S.
Fourth edition. Springer.

L. Gatto (2008).
MAQCsubsetAFX: MAQC data subset for the Affymetrix platform.
R package version 1.0.0. http://www.slashhome.be/MAQCsubsetAFX.php

L. Gautier, L. Cope, B.M. Bolstad, and R.A. Irizarry (2004).
affy—analysis of Affymetrix GeneChip data at the probe level.
Bioinformatics 20, 3 (Feb. 2004), 307-315.

R. Gentleman, V.J. Carey, D.M. Bates et al. (2004)
Bioconductor: Open software development for computational biology and bioinformatics.
Genome Biology, Vol. 5, R80

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.

P. Ruckdeschel, M. Kohl, T. Stabla and F. Camphausen (2006).
S4 classes for distributions.
R-News 6(2):2–6.

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

Rutherford, E. and Geiger, H. (1910).
The Probability Variations in the Distribution of alpha Particles.
Philosophical Magazine 20:698–704.

R Development Core Team (2008).
R: A language and environment for statistical computing.
R Foundation for Statistical Computing, Vienna, Austria.

ISBN 3-900051-07-0, http://www.R-project.org.