Optimal Bias Robust Regression Psi and Rho

Kjell Konis, University of Washington

R. Douglas Martin, University of Washington

August 15 2020, Revised June 20, 2021

Abstract

This paper is focused on detailed aspects of the loss function rho and its derivative psi for an optimal bias robust regression method that minimizes the maximum asymptotic bias subject to a constraint on normal distribution efficiency. The analytic form of the psi function was discovered by Yohai and Zamar (1997) and further studied by Svarc et al. (2002), but the analytic form of the rho function was not known. Furthermore, the psi function has a curious feature of being equal to zero on an interval around the origin which forms a psi function "flat spot" whose length decreses with increasing normal distribution regression estimator efficiency, and is hardly noticeable in a plot of the psi function for 95% normal distribution efficieny. This paper focuses on the following aspects the optimal bias robust regression estimator: (1) The discovery of an analytic form for the rho function; (2) A computational problem posed by the psi function flat spot and a solution to the problem using a slighly modified psi function that entails little loss of bias robustness; (3) A suite of functions available in an R package for computing the optimal psi and rho functions, and the modified psi and rho functions, for a specified normal distribution efficiency. We briefly discuss the implementation of the optimal and modified optimal estimator in the RobStatTM R package available on CRAN.

Keywords: Robust regression, bias robustness, efficiency.

JEL Classification: C13, C61

1 Introduction

The routine use of robust regression has been substantially hampered by the existence of too many distinctly different types of such estimators, including least–trimmed–squares (LTS), bounded–influence estimators, S-estimators, M-estimators and the computational MM-estimator variant of M-estimators, e.g., as discussed in Maronna et al. (2019). Furthermore, even the most popular family of regression M-estimators has far too many variants based on different choices of the rho/psi/weight functions. For example, the SAS software product offers 10 M-estimator variants identified by the names: *Andrews*, *Bisquare*, *Cauchy*, *Fair*, *Hampel*, *Huber*, *Logistic*, *Median*, *Talworth*, *Welsch*.[1] This reflects the fact that there has been no consensus on just which robust regression estimator one should use, even within the family of regression M–estimators. We believe that the choice of a robust regression estimator should be based on the strengths of its theoretical properties. Among all the robust regression estimators available today, only two estimators have the property that they minimizes an asymptotic bias measure over a Tukey-Huber family of distributions, namely the Yohai and Zamar (1997) estimator, which we refer in this paper as the *Opt* estimator, and the Svarc et al. (2002) estimator. In this paper we focus on the *Opt* estimator, and begin by providing the definition of its psi function and introducing the previously unknown mathematical form of the corresponding rho function defined as the integral of the psi function. We

[1] See Table 80.5 in the SAS PROC ROBUST REG document at: `https://support.sas.com\T1\guilsinglrightonlinedoc\T1\guilsinglrightstat\T1\guilsinglrightrregPDF`

then point out a computational deficiency of the *Opt* psi function, and introduce a slightly modified optimal estimator *mOpt* that avoids the Opt computational deficiency of the *Opt* psi function. Along the way we introduce a family of R functions for computing the Opt and mOpt rho functions and psi functions, and the weight functions obtained from the psi functions, for any desired normal distribution efficiency in an allowable range.

The following Section 2 provides the mathematical definistions of the Opt and mOpt estimator rho, psi and weight functions, and their evaluation for user specified normal distribution efficienies.

2 Optimal Psi and Rho Functions

In this section we begin by discussing the family of optimal bias robust regression estimating equation psi function $\psi(x)$ discovered by Yohai and Zamar (1997), whose analytic form and related discussed appear in Maronna et al. (2019). Then we discuss the analytic form of the rho function $\rho(x)$ whose derivative is $\psi(x)$. We also briefly discuss the weight function obtained from the psi function, and the role of the weight function in the numerical optimization for obtaining the robust regression estimates.

2.1 The Optimal Psi Functions

The family of optimal robust psi functions is given by

$$\psi_a(x) = \text{sign}(x)\left(|x| - \frac{a}{\phi(x)}\right)^+ = \begin{cases} x - \text{sign}(x)\dfrac{a}{\phi(x)} & |x| \in (\text{lower}, \text{upper}) \\ \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where the tuning constant $a > 0$ controls the support of $\psi_a(x)$, and correspondingly controls the bias versus normal distribution efficiency trade-off of the estimator. The support interval for negative x is a reflection about the origin of the support interval for positive x. Note that for the limiting case $a = 0$ the optimal estimator reduces to $\psi_0(x) = x$, and the estimator is the normal distribution MLE which has 100% efficiency, but is totally lacking in robustness.

At first glance one wonders for what range of positive a will a support interval (lower, upper) exist, where the support interval is the set of positive values of x such $g(x) = x\phi(x) - a$ is positive. Note that $g(x) = -a$ for $x = 0$ and for $lim_{x\to\infty} g(x)$, and that

$$g'(x) = (1 - x^2)\phi(x)$$

from which it follows that $g(x)$ is increasing for $x < 1$, and decreasing for $x > 1$, with the following maximum at $x = 1$:

$$g(1) = \frac{1}{\sqrt{(2\pi}} exp(-1/2) - a = 0.2419707245 - a$$

where the first term on the right-hand side has been computed to 10 significant digits. For a non-degenerate support interval (lower, upper) to exist, it must be that $0 < a < 0.2419707245$, and for such values of a the values of "lower" and "upper" will be the two roots of the function $x\phi(x) - a$.

The normal distribution efficiency of the optimal estimator is

$$\text{EFF}(\psi_a, N) = V^{-1}(\psi_a, N) \tag{2}$$

where

$$V(\psi_a, N) = \frac{E_N\left(\psi_a^2\right)}{E_N^2\left(\psi_a'\right)} \tag{3}$$

is the asymptotic variance of the optimal estimator at a standard normal distribution N. The numerator and denominator of $V(\psi_a, N)$ need to be evaluated by numerical integration in order to obtain EFF, and the values "lower" and "upper" of the support interval are needed for such integration.

For any allowable value of a, the function `OptPsiSupportIntervalFromConst(a)` computes the support interval values "lower" and "upper". For example:

```
OptPsiSupportIntervalFromConst(0.05)
```

```
## [1] 0.1263356 2.4360509
```

The function `ComputeEfficiencyFromConst_Opt(a)` uses `OptPsiSupportIntervalFromConst()` in computing the normal distribution efficiency of the optimal psi for any allowable value of a. For example:

```
computeEfficiencyFromConst_Opt(0.05)
```

```
## [1] 0.8301284
```

Use of the abovve two functions for $a = 0.20, 0.10, 0.05, 0.03, 0.01, 0.005$ results in the Table 1 values.

Table 1: Optimal Psi Support Intervals and Efficiencies Versus a

a	lower	upper	Eff
0.200000	0.600306	1.464030	0.344
0.100000	0.259228	2.050151	0.680
0.050000	0.126336	2.436051	0.830
0.030000	0.075413	2.672285	0.893
0.010000	0.025074	3.104545	0.961
0.005000	0.012534	3.342482	0.980

One typically want to use a robust regression estimator that has a specified normal distribution effiicency, with 0.95 being a typical efficiency, and so the function `ComputeConstFromEfficiency_Opt(eff)` allows one to compute the value of the constant a for any achievable efficiency value `eff`. For example:

```
computeConstFromEfficiency_Opt(0.95)


## [1] 0.01317965
```

And the following code line confirms that above value of a does indeed yield an efficiency of 95%:

```
computeEfficiencyFromConst_Opt(0.01317965)


## [1] 0.95
```

Use of `computeConstFromEfficiency_Opt()` and `OptPsiSupportIntervalFromConst()` for normal distribution efficiencies of 85%, 90%, 95%, 99% produces the results are displayed in Table 2.

Table 2: Optimal Psi a and Support Interval vs Efficiency

Efficiency	a	lower	upper
0.85	0.043579	0.109897	2.502638
0.90	0.027902	0.070112	2.703592
0.95	0.013180	0.033055	3.003281
0.99	0.002449	0.006138	3.567972

The function `psi_Opt` can be used to plot the optimal $\psi_a(x)$ function. The results for efficiencies 85%, 90%, 95%, and 99%, are shown in Figure 1.[2]

[2]It should be noted that the `optimalRhoPsi` package contains a utility function `computeTuningPsi_Opt()` that takes either the constant a or an efficiency value eff as input, and computes a vector named `cc` with six components, where the first three components `cc[1]`, `cc[2]`, `cc[3]`are the value of a and the support values "lower" and "upper", that are used by `psi_Opt()`. The other three values `cc[1]`, `cc[2]`, `cc[3]` are used later for other purposes.

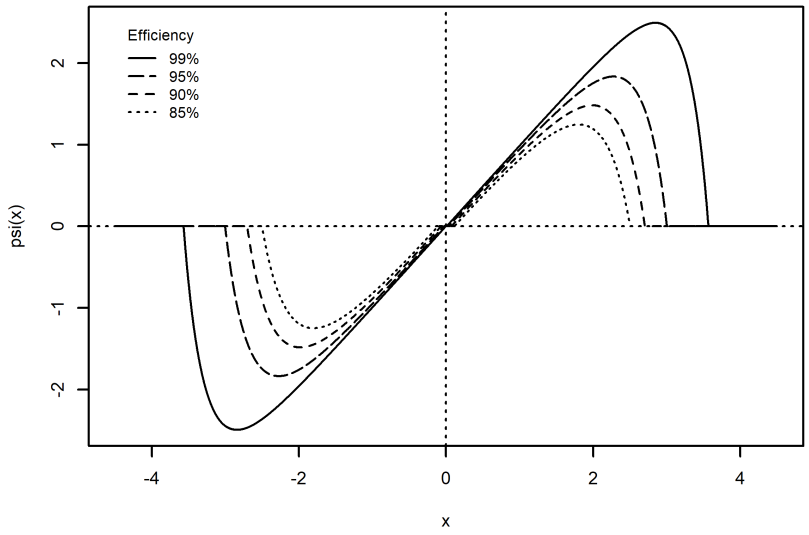

Figure 1: Optimal Psi Functions for Various Normal Distribution Efficiencies

It is not easy to discern in Figure 1, but for each efficiency there is a very small interval [−lower, lower] around the origin where $\psi(x) = 0$ for x in that interval, and we note that the value of "lower" decreases with increasing normal distribution efficiency.

2.2 The Optimal Rho Function

In order to compute an MM-estimator, due to Yohai (1987) and also described in Maronna et al. (2019), we need an analytic form for the the loss function $\rho_a(x)$ obtained by integrating $\psi_a(x)$. The integral of the first term in the right-hand side of the expression for $\psi_a(x)$ gives $x^2/2$, but it was not obvious that the second term had an analytic form. Thus, Yohai and Zamar (1997) suggested to integrate a polynomial approximation of $\psi_a(x)$ to get an analytic form for $\rho_a(x)$, and provided an example of such a polynomial approximation.

It turns out that the second term has a usable analytic form described below. Since $\rho_a(x)$ is symmetric, it suffices to consider only $x \geq 0$. We write the integral form of $\rho_a(x)$ as:

$$\rho_a(x) = \int_0^x \psi_a(u)\, du = \begin{cases} 0 & 0 \leq x \leq \text{lower} \\ \Psi_a(x) - \Psi_a(\text{lower}) & \text{lower} < x < \text{upper} \\ \Psi_a(\text{upper}) - \Psi_a(\text{lower}) & x \geq \text{upper}. \end{cases} \tag{4}$$

The antiderivative $\Psi_a(x)$ of $\psi_a(x)$ on the support interval is given by the analytic form

$$\Psi_a(x) = \frac{1}{2}x^2 - a\pi \operatorname{erfi}\left(\frac{x}{\sqrt{2}}\right) \tag{5}$$

where "erfi" is the *imaginary error function*. See for example Zhang et al. (1996). Furthermore, the evaluation of the erfi function is available in the `pracma` R package due to Borchers (2017).

The function `rho_Opt` computes values of the optimal rho function for specified value of the function argument. The functions `rho_Opt` and psi_Opt for 85% normal distribution efficiency are used to produce the results in Figure 2. At this efficiency the flat spot of the psi function on the small interval around zero is clearly visible, and the rho function is equal to zero on this interval.

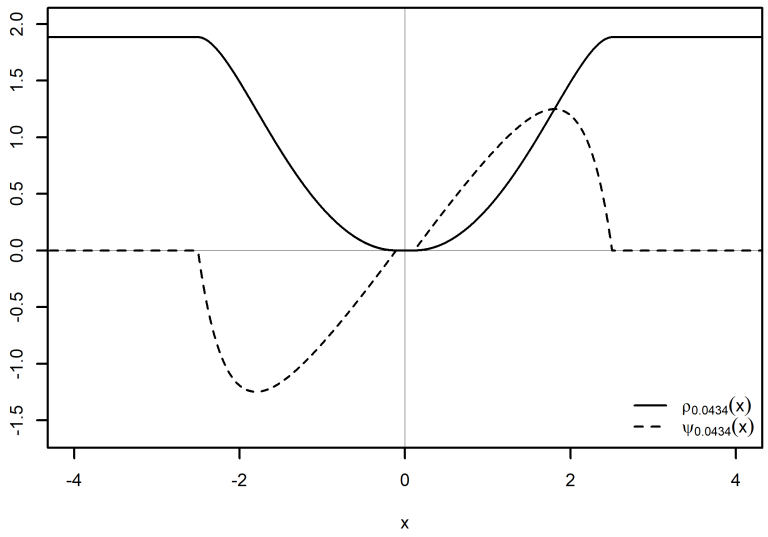

Figure 2: Optimal Rho and Psi Functions for 85% Normal Distribution Efficiency

Figure 3 shows the rho function shapes for normal distribution efficiencies of 85%, 90%, 95%, 99%.

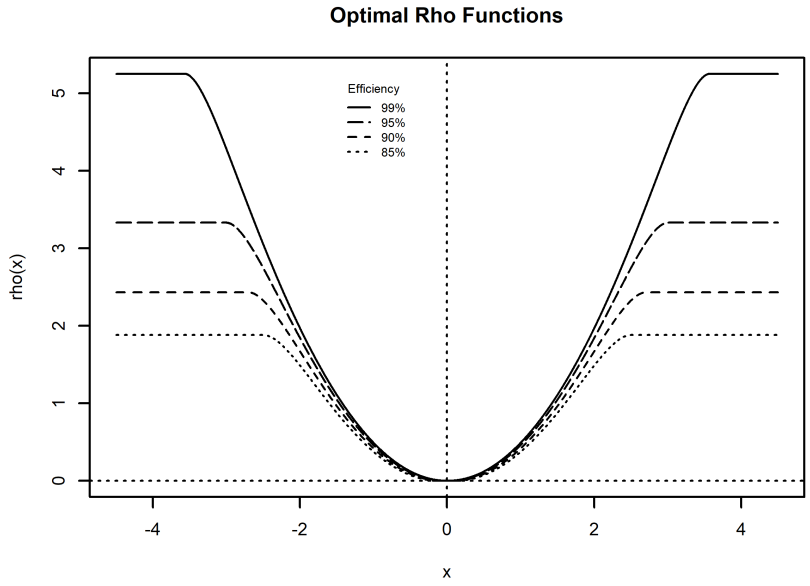

Figure 3: Optimal Rho Functions for Various Normal Distribution Efficiencies

2.3 The Optimal Weight Function

The weight function corresponding to the optimal psi function is:

$$w_a(x) = \frac{\psi_a(x)}{x} \tag{6}$$

The function `wgt_Opt` computes the value of the weight function $w_a(x)$ for a specified value of x, and this function is used to plot the shapes of the weight functions in Figure 4 for values of a corresponding to efficiency values 85%, 90%, 95%, 99%.

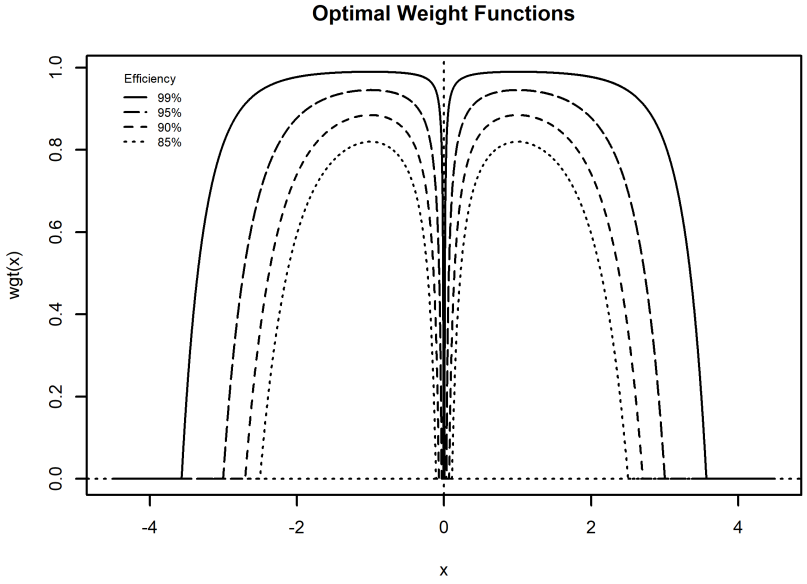

Figure 4: Optimal Weight Function for Various Normal Distribution Efficiencies

We refer to the behavior of the weight functions near the origin, where they have value zero on a more or less small interval, as a "sink hole". [3]

3 A Modified Optimal Estimator

The optimal estimators have the curious feature of intervals centered at the origin for which the the optimal psi and rho functions have value zero, and the optimal weight functions have a sink-hole. Consequently, the weight functions do not satisfy a condition of being non-increasing that is needed to insure convergence of an iterative re-weighting algorithm for the initial S-estimator and the final M-estimator in the MM-estimator algorithm (see Sections 5.7.1 and 4.5.2 of Maronna et al. (2019)). It is therefore desirable to have a smooth approximation modification of the optimal psi function that is linear with slope one at the origin, and a resulting weight function that is non-increasing. We derive such an approximation and refer to the resulting regression estimator as the modOpt estimator. The corresponding R psi, rho and weight functions are `psi_modOpt`, `rho_modOpt` and `wgt_modOpt`.

Substituting the expression (1) of the optimal psi function into the definition (6) of the optimal weight function, gives the following expression of the optimal weight function:

[3] In a geology context, a sink hole is a cavity in the ground caused by water erosion, and providing a route for surface water to disappear underground.

$$\mathrm{w}_a(x) = \begin{cases} 1 - \dfrac{a}{|x|\,\phi(x)} & |x| \in (\text{lower},\ \text{upper}) \\ \\ 0 & \text{otherwise}\,. \end{cases} \tag{7}$$

The derivative of the optimal weight function is

$$\mathrm{w}'_a(x) = \begin{cases} \dfrac{a\,(1 - x^2)}{x^2\,\phi(x)} & |x| \in (\text{lower},\ \text{upper}) \\ \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

which shows that the optimal weight function is increasing for $0 < x < 1$ and decreasing for $1 < x <$ upper, and at $x = 1$ takes on its maximum value of

$$1 - \frac{a}{\phi(1)}. \tag{9}$$

The idea is to modify the weight function by joining those maximum values at $x = \pm 1$ with a horizontal straight line, which yields the initial modified psi function candidate:

$$\psi_a^{(0)}(x) = \begin{cases} \left(1 - \dfrac{a}{\phi(1)}\right) x & -1 \le x \le 1 \\ \\ x - \operatorname{sign}(x)\dfrac{a}{\phi(x)} & 1 < |x| < \text{upper} \\ \\ 0 & \text{otherwise}\,. \end{cases} \tag{10}$$

Note that $\psi_a^{(0)}(x)$ has a continuous derivative, and the slope of the linear piece is

$$1 - \frac{a}{\phi(1)}. \tag{11}$$

Since by convention the slope of a psi function at the origin is one, we define our modified optimal psi function as:

$$\psi_a^{(\mathrm{mod})}(x) = \left(\frac{\phi(1)}{\phi(1) - a}\right) \psi_a^{(0)}(x) = \begin{cases} x & -1 \le x \le 1 \\ \\ \left(\dfrac{\phi(1)}{\phi(1) - a}\right)\left(x - \operatorname{sign}(x)\dfrac{a}{\phi(x)}\right) & 1 < |x| < \text{upper} \\ \\ 0 & \text{otherwise}\,. \end{cases} \tag{12}$$

3.1 Effiency and Tuning Constant *a* for the Modified Optimal Psi and Rho

Just as in the case of the optimal psi, there is function to compute the normal distribuiton efficiency for a specified constant a, and a function to compute the constant a for a specified normal distribution effiency. For example:

```
computeConstFromEfficiency_modOpt(0.95)
```

```
## [1] 0.01316352
```

```
computeEfficiencyFromConst_modOpt(0.01316352)
```

```
## [1] 0.95
```

Note that the following computations for the optimal psi give virtually identical results as for the modified optimal psi:

```
computeConstFromEfficiency_Opt(0.95)
```

```
## [1] 0.01317965
```

```
computeEfficiencyFromConst_Opt(0.01317965)
```

```
## [1] 0.95
```

The function `psi_modOpt` is used to compute values of the modified optimal function $\psi_a^{(\text{mod})}(x)$.[4] Figure xx shows the attractive shape of the resulting 85% normal distribution efficiency $\psi_a^{(\text{mod})}(x)$, with its unity slope linearity around the origing, as compared with the flat spot of the corresponding optimal $\psi_a(x)$ based on the same a and scaled by $\phi(1)/\left(\phi(1) - a\right)$.

[4]The function `psi_modOpt` makes use of the utility function `computeTuningPsi_modOpt`, just as the function `psi_Opt` makes use of the utility function `computeTuningPsi_modOpt` for the value of a and the values of the support interval end points.

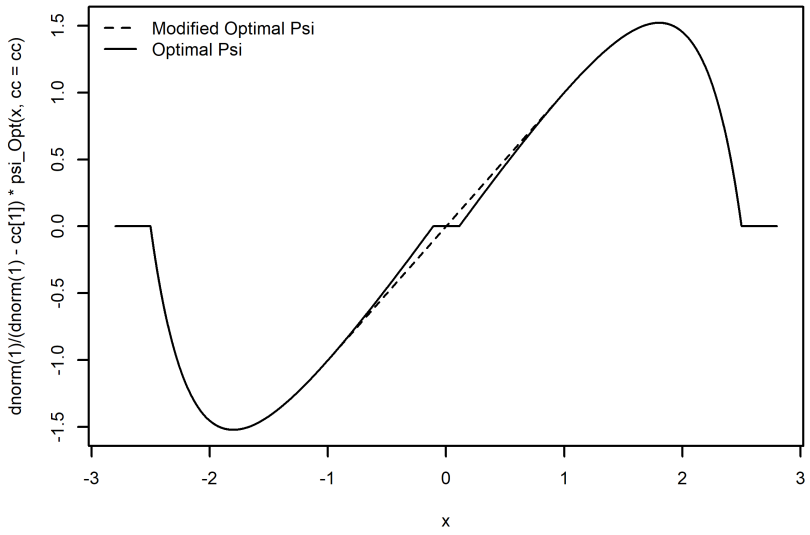

Figure 5: Optimal and Modified Optimal Psi Functions for 85% Normal Distribution Efficiency

3.2 Modified Optimal Weight Functions

Aside from the small differences in a region relatively near the origin, the modified optimal psi function shape concides with that of the optimal psi function, and there is little need for making plots of the modified optimal psi functions for the efficiencies in Figure 1. However, Figure 6 shows that the values of the weight functions obtained from the modified optimal psi function differ quite substantially near the origin from those the optimal psi based weight functions shown in Figure 4. In particular, the "sink hole" around the origin for the optimal psi function no longer exists in the modified optimal weight function, which is non-increasing for $x \geq 0$. Thus the modified weight function satisfies the sufficient condition that insures convergence of the iterative re-weighting algorithm for the initial S-estimator and the final M-estimator in the MM-estimator algorithm mentioned earlier.

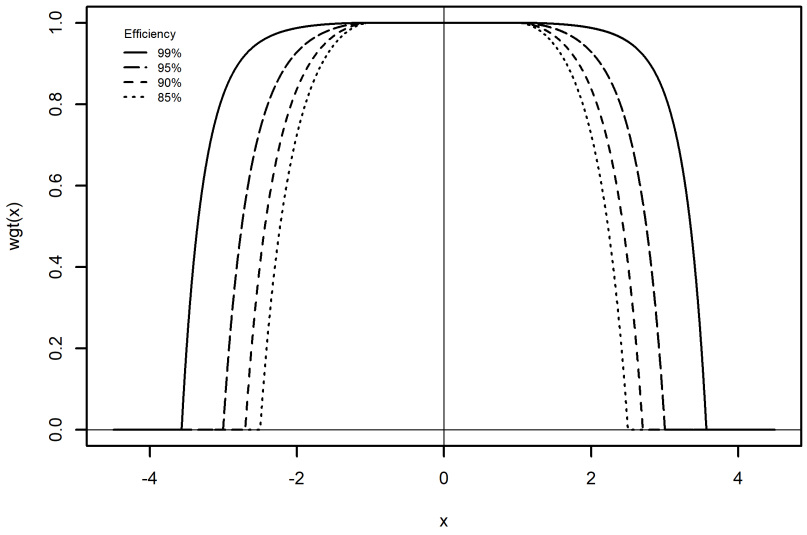

Figure 6: Modified Optimal Weight Function for Various Normal Distribution Efficiencies

4 MM-Estimator Implementation

An MM-estimator requires an initial S-estimator based on an M-scale with breakdown point one-half in order to insure that the final high-efficiency M-estimator has breakdown point one-half. For any bounded loss function, such an initial estimator may be obtained by a re-scaling of the loss function for the final estimator. For details see Section 5.5 of Maronna et al. (2019).

4.1 MM-Estimator Based on Optimal Rho

In the case of $\rho_a(x)$, the scaled rho function $\tilde{\rho}_a(x)$ with breakdown point one-half is obtained as follows for the case of a 95% efficient final M-estimator.

First one computes the constants for the 95% efficient final M-estimator with:

```
(cc = computeTuningPsi_Opt(0.95))
```

```
##              a          lower          upper              c Psi_Opt(lower)
##   0.0131796499   0.0330545358   3.0032809091   1.0000000000  -0.0005459033
##       rho(Inf)
##   3.3313697906
```

Here the constant c is the default scaling constant for the psi and rho functions.

Then one computes the much smaller value of c needed to obtain a breakdown point one-half initial estimator with:

```
(ccHbp = computeTuningChi_Opt(cc))


##              a          lower          upper              c Psi_Opt(lower)
##   0.0131796499   0.0330545358   3.0032809091   0.3799299223  -0.0005459033
##       rho(Inf)
##   3.3313697906
```

We compute the efficiency of the high breakdown point initial M-scale estimator with:

```
computeEfficiencyFromConst_Opt(ccHbp[1], ccHbp[4])


## [1] 0.2021623
```

4.2 MM-Estimator Based on the Modified Optimal Rho

In the case of the modified optimal psi and rho, the calculations are similar to those above, except that one uses the functions for the modified optimal psi:

```
(cc = computeTuningPsi_modOpt(0.95))


##          a  normConst      upper          c Psi_Opt(1)   rho(Inf)
## 0.01316352 1.05753107 3.00373939 1.00000000 0.46057111 3.53690811
```

```
(ccHbp = computeTuningChi_modOpt(cc))


##          a  normConst      upper          c Psi_Opt(1)   rho(Inf)
## 0.01316352 1.05753107 3.00373939 0.38124404 0.46057111 3.53690811
```

```
computeEfficiencyFromConst_modOpt(ccHbp[1], ccHbp[4])


## [1] 0.2430879
```

4.3 Implementation in the RobStatTM Package

The code in the R package `optimalRhoPsi` (see Appendix) used for this document is all R code. However, C code versions (needed by the underlying `robustbase` package code) of the optimal estimator for the 50% breakdown point initial estimator, and a default 95% final estimator are included in the RobStatTM package. Those functions are as follows.

The optimal psi function, its derivative, and its rho function can be evalutated using the Mspi function; the optimal weight function can be evaluated using the `Mwgt` function. These functions are extended implementations (to include the optimal, and modified optimal psi families) of the functions with the same names found in the robustbase package. Note that these functions are not exported by RobStatTM. For example

```
RobStatTM:::Mpsi(x, cc, "opt", deriv = 0)
```

evaluates the optimal psi function with tuning vector `cc` at `x`. When `deriv = 1` the deriative of the optimal psi function is evaluated, and when `deriv = -1` the optimal rho function is evaluated. The optimal weight function can be evaulated using the function

```
RobStatTM:::Mwgt(x, cc, "opt")
```

The default tuning vectors for the initial estimator result in a breakdown point 50% but low normal distribution efficiency of about 28%, and the default tuning paramters for final estimator result in a breakdown point 50% and a high normal distribution efficiency of 95%.

are accesible through the `lmrobdet.control` function. For example:

```
library(RobStatTM)
lmrobdet.control(psi = "optimal")[c("tuning.chi", "tuning.psi")]
```

An MM estimator can then be fit using the `lmrobdetMM` function in RobStatTM:

```
lmrobdetMM(stack.loss ~ ., data = stackloss, control = lmrob.control(efficiency = 0.95
    family = "mopt"))
```

NOTE: Those values of efficiency and the rho/psi/weights family are the defaults, so the `control =` argument would not need to be used for those choices, but but for other choices of efficiency and/or the choice of the optimal rho/psi/weights family by using `family = "opt"`, the `control =` argument needs to be used.

References

Borchers, H. W. (2017). *pracma: Practical Numerical Math Functions*. R package version 2.0.7. URL: https://CRAN.R-project.org/package=pracma.

Maronna, R. A. et al. (2019). *Robust Statistics: Theory and Methods*. 2nd Edition. John Wiley & Sons, Ltd.

Svarc, M., Yohai, V. J. and Zamar, R. H. (2002). "Optimal Bias Robust M-Estimates of Regression". In: *Statistical Data Analysis Based on the L1-Norm and Related Methods*. Springer, pp. 191–200.

Yohai, V. J. (June 1987). "High Breakdown Point and High Efficiency Robust Estimates for Regression". In: *The Annals of Statistics* 15.2, pp. 642–656. DOI: 10.1214/aos/1176350366. URL: http://dx.doi.org/10.1214/aos/1176350366.

Yohai, V. J. and Zamar, R. H. (1997). "Optimal locally robust M-estimates of regression". In: *Journal of Statistical Planning and Inference* 64.2, pp. 309–323. ISSN: 0378-3758. DOI: 10.1016/S0378-3758(97)00040-2. URL: http://www.sciencedirect.com/science/article/pii/S0378375897000402.

Zhang, S. and Jin, J. (1996). *Computation of Special Functions*. New York: John Wiley & Sons, Inc. ISBN: 0-471-11963-6.

Appendix

A1 The optimalRhoPsi R Package

The `optimalRhoPsi` R package is available at `https://github.com/kjellpk/optimalRhoPsi/`. You install and load the package with the first two code lines below, and the third code line prints the names of all the functions in the package:

```
devtools::install_github("kjellpk/optimalRhoPsi")
library(optimalRhoPsi)
ls("package:optimalRhoPsi")
```

A1.1 Functions for Optimal Psi

The function `computeTuningPsi_Opt` (contained in the function `computeEfficiency_Opt`)

Computes the tuning parameter vector for the optimal psi for a specified normal distribution efficiency. For example, for 80% efficiency the vector of tuning parameters is:

```
(cc80 <- computeTuningPsi_Opt(0.8))
```

```
##              a          lower          upper              c Psi_Opt(lower)
##     0.05988905     0.15186061     2.34452872     1.00000000    -0.01135436
##       rho(Inf)
##     1.48673738
```

where `a` is the optimal psi parameter a, `lower` and `upper` are the endpoints of the positive support interval, `c` is the rescaling constant for the initial estimator, (*is this because the initial estimator is 80% efficient?*) `Psi_Opt(lower)` is Ψ_a(lower), and `rho(Inf)` is the supremum of the optimal rho function. Note that only `a` and `c` are strictly (*really?*) required. The other 4 components are included only to avoid unnecessary recalculation.

Functions for evaluating the optimal psi, rho and weights values:

- `psi_Opt`: optimal psi function
- `Psi_Opt`: indefinite integral of psi used by `rho_Opt`
- `rho_Opt`: optimal rho function
- `psip_Opt`: derivative of the optimal psi function
- `wgt_Opt`: optimal weight function

Each of the above five functions requires 2 arguments. The first, `x`, is a vector of values where the function should be evaluated. The second, `cc`, is the vector of tuning parameters, e.g., the vector `cc80` computed by `computeTuningPsi_Opt`.

A1.2 Functions for Modified Optimal Psi

The function `computeTuningPsi_modOpt` (that is contained in the function `computeEfficiency_modOpt`):

Computes the tuning parameter vector for the Modified Optimal psi for a specified Gaussian efficiency. For example, for 80% efficiency the vector of tuning parameters is:

```
(cm80 <- computeTuningPsi_modOpt(0.8))
```

```
##          a  normConst      upper          c Psi_Opt(1)   rho(Inf)
## 0.06046518 1.33313135 2.33952935 1.00000000 0.31888775 2.02550885
```

where `a` is the modified optimal psi parameter a, `normConst` is $\phi(1)/(\phi(1) - a)$, `upper` is the upper endpoint of the positive support interval for the optimal psi (with parameter a), `c` is the rescaling constant for the initial estimator, `Psi_Opt(1)` is $\Psi_a(1)$, and `rho(Inf)` is the supremum of the modified optimal rho function. Note that only `a` and `c` are strictly required. The other 4 components are included only to avoid unnecessary recalculation.

Functions for evaluating the modified optimal psi, rho and weights values:

- `psi_modOpt`: modified optimal psi function
- `rho_modOpt`: modified optimal rho function
- `psip_modOpt`: derivative of the modified optimal psi function
- `wgt_modOpt`: modified optimal weight function

Each of the above five functions requires 2 arguments. The first, `x`, is a vector of values where the function should be evaluated. The second, `cc`, is the vector of tuning parameters, e.g., the vector `cm80` computed by `computeTuningPsi_modOpt`.