

# Analysis of Robust Quasi-deviances for Generalized Linear Models

Eva Cantoni

Department of Econometrics  
University of Geneva  
CH - 1211 Geneva 4, Switzerland

August 2002  
revised May 2003

## Acknowledgements

We gratefully acknowledge the support of the Swiss National Science Foundation (Project Number 1214-66989).

## 1 Motivation

Generalized linear models (McCullagh and Nelder 1989) are a popular technique for modeling a large variety of continuous and discrete data. They assume that the response variables  $Y_i$ , for  $i = 1, \dots, n$ , come from a distribution belonging to the exponential family, such that  $E[Y_i] = \mu_i$  and  $V[Y_i] = V(\mu_i)$ , and that

$$\eta_i = g(\mu_i) = \mathbf{x}_i^T \boldsymbol{\beta}, \quad (1)$$

where  $\boldsymbol{\beta} \in \mathbb{R}^p$  is the vector of parameters,  $\mathbf{x}_i \in \mathbb{R}^p$ , and  $g(\cdot)$  is the link function.

The non-robustness of the maximum likelihood and the maximum quasi-likelihood estimators has been studied extensively in the literature. For model selection, the classical analysis-of-deviance approach shares the same bad robustness properties. To cope with this, Cantoni and Ronchetti (2001) propose a robust approach based on robust quasi-deviance functions for estimation and variable selection. We refer to that paper for a deeper discussion and the review of the literature.

## 2 Some theory

Cantoni and Ronchetti (2001) suggest the following set of estimating equations for the robust estimation of  $\boldsymbol{\beta}$ :

$$\sum_{i=1}^n \boldsymbol{\psi}(y_i, \mu_i) = \mathbf{0}, \quad (2)$$

where  $\boldsymbol{\psi}(y, \mu) = \nu(y, \mu)w(\mathbf{x})\boldsymbol{\mu}' - a(\boldsymbol{\beta})$ ,  $a(\boldsymbol{\beta}) = \frac{1}{n} \sum_{i=1}^n \mathbb{E}[\nu(y_i, \mu_i)]w(\mathbf{x}_i)\boldsymbol{\mu}_i'$  with the expectation taken with respect to the conditional distribution of  $y|\mathbf{x}$ ,  $\nu(\cdot, \cdot)$ ,  $w(\mathbf{x})$  are robustness weight functions, and  $\mu_i = \mu_i(\boldsymbol{\beta}) = g^{-1}(\mathbf{x}_i^T \boldsymbol{\beta})$ .

The estimator defined by equations (2) is an M-estimator characterized by the score function  $\boldsymbol{\psi}(y_i, \mu_i) = \nu(y_i, \mu_i)w(\mathbf{x}_i)\boldsymbol{\mu}_i' - a(\boldsymbol{\beta})$ . It is asymptotically normally distributed with variance  $\Omega = M(\boldsymbol{\psi}, F)^{-1}Q(\boldsymbol{\psi}, F)M(\boldsymbol{\psi}, F)^{-1}$ , where  $M(\boldsymbol{\psi}, F) = -\mathbb{E}[\frac{\partial}{\partial \boldsymbol{\beta}} \boldsymbol{\psi}(y, \mu)]$  and  $Q(\boldsymbol{\psi}, F) = \mathbb{E}[\boldsymbol{\psi}(y, \mu)\boldsymbol{\psi}(y, \mu)^T]$ . Its influence function (Hampel 1974) is proportional to  $\boldsymbol{\psi}$  (see Hampel, Ronchetti, Rousseeuw, and Stahel 1986), which is bounded with respect to  $y$  if  $\nu(y, \mu)$  is bounded, and with respect to  $\mathbf{x}$  if  $w(\mathbf{x})$  is suitably chosen to down-weight leverage points.

Moreover, the estimating equations (2) can be obtained by differentiating the robust quasi-likelihood function

$$Q_M(\mathbf{y}, \boldsymbol{\mu}) = \sum_{i=1}^n Q_M(y_i, \mu_i), \quad (3)$$

with respect to  $\boldsymbol{\beta}$ , where the functions  $Q_M(y_i, \mu_i)$  can be written as

$$Q_M(y_i, \mu_i) = \int_{\tilde{s}}^{\mu_i} \nu(y_i, t)w(\mathbf{x}_i)dt - \frac{1}{n} \sum_{j=1}^n \int_{\tilde{t}}^{\mu_j} \mathbb{E}[\nu(y_j, t)w(\mathbf{x}_j)]dt, \quad (4)$$

with  $\tilde{s}$  such that  $\nu(y_i, \tilde{s}) = 0$ , and  $\tilde{t}$  such that  $\mathbb{E}[\nu(y_i, \tilde{t})] = 0$ .

These functions can be used to construct a robust measure of discrepancy between two nested models  $M_p$  (with  $p$  parameters) and  $M_{p-q}$  (with  $p-q$  parameters) by means of the test statistic

$$\Lambda_{QM} = 2 \left[ \sum_{i=1}^n Q_M(y_i, \hat{\mu}_i) - \sum_{i=1}^n Q_M(y_i, \dot{\mu}_i) \right], \quad (5)$$

where  $\hat{\mu}_i = \mu_i(\hat{\boldsymbol{\beta}})$  and  $\dot{\mu}_i = \mu_i(\dot{\boldsymbol{\beta}})$  are the estimates under model  $M_p$  and  $M_{p-q}$  respectively. Note that  $\Lambda_{QM}$ , which is independent of  $\tilde{s}$  and  $\tilde{t}$ , can be used to perform variable selection in an analysis-of-deviance type of approach.

Cantoni and Ronchetti (2001) proved that, under some regularity conditions, the test statistic  $\Lambda_{QM}$  in (5) is asymptotically equivalent to a quadratic form in normal variables and is therefore distributed according to a linear combination of  $\chi_1^2$  variables, more precisely:

$$\Lambda_{QM} \stackrel{n \rightarrow \infty}{\sim} \sum_{i=1}^q d_i N_i^2, \quad (6)$$

where  $N_1, \dots, N_q$  are independent standard normal variables,  $d_1, \dots, d_q$  are the  $q$  positive eigenvalues of the matrix  $Q(\boldsymbol{\psi}, F)(M^{-1}(\boldsymbol{\psi}, F) - \tilde{M}^+(\boldsymbol{\psi}, F))$ , and  $\tilde{M}^+(\boldsymbol{\psi}, F)$  is such that  $\tilde{M}^+(\boldsymbol{\psi}, F)_{11} = M(\boldsymbol{\psi}, F)_{11}^{-1}$  and  $\tilde{M}^+(\boldsymbol{\psi}, F)_{12} = 0$ ,  $\tilde{M}^+(\boldsymbol{\psi}, F)_{21} = 0$ ,  $\tilde{M}^+(\boldsymbol{\psi}, F)_{22} = 0$ .

The particular case when

$$\nu(y_i, \mu_i) = \psi_c(r_i) \frac{1}{V^{1/2}(\mu_i)} \quad (7)$$

with  $r_i = \frac{y_i - \mu_i}{V^{1/2}(\mu_i)}$  being the Pearson residuals and  $\psi_c$  the Huber function defined by

$$\psi_c(r) = \begin{cases} r & |r| \leq c \\ c \operatorname{sign}(r) & |r| > c \end{cases} \quad (8)$$

is studied in details in Cantoni and Ronchetti (2001), and the S-PLUS code implementing it is provided here.

Note also that  $\nu(y_i, \mu_i) = (y_i - \mu_i)/V(\mu_i)$  and  $w(\mathbf{x}) \equiv 1$  reproduce the classical approach.

Cantoni and Ronchetti (2001) also computed the asymptotic level and power under contamination, and showed that they are both bounded if an M-estimator with bounded influence function is used for  $\boldsymbol{\beta}$  (see Proposition 2 and Corollary 1 therein). A nice consequence of these results is the possibility to define a procedure to choose the tuning constant appearing in  $\nu$  (see for example (7) and (8)) subject to a maximal bias on the level of the test statistic in a neighborhood of the model. This will depend on the maximal bias allowed on the test statistics, the proportion of contamination and the distribution of the test statistic.

### 3 The code

The S-PLUS code implementing the estimating equations (2) and the test statistic (5) with  $\nu(y_i, \mu_i)$  as defined by (7) and (8) is provided here for logistic (Bernoulli),

binomial and Poisson models as an archived file: `robGLM.tar`. It has been developed under S-PLUS Version 3.4 Release 1 for Sun SPARC, SunOS 5.3 : 1996 and tested under S-PLUS Version 6.0 Release 1 for Sun SPARC, SunOS 5.6 : 2000. The code also runs in S-PLUS 6.1 Release 1 under Windows2000.

The archived file `robGLM.tar` contains the following files:

- `robGLM/Ins3.4`
- `robGLM/Ins6.0`
- `robGLM/ROBGLMsfun.s`
- `robGLM/carrots.dump`
- `robGLM/possum.dump`
- `robGLM/help3.4/glm.rob.d`
- `robGLM/help3.4/quasi.rob.d`
- `robGLM/help6.0/glm.rob.sgml`
- `robGLM/help6.0/quasi.rob.sgml`
- `robGLM/helpHTML/glm.rob.html`
- `robGLM/helpHTML/quasi.rob.html`

The files `Ins3.4` and `Ins6.0` are scripts to install the code under S-PLUS version 3.4 and 6.0 for Unix respectively. The file `ROBGLMsfun.s` contains the whole set of S-PLUS functions. `carrots.dump` and `possum.dump` are dump files with the data used in Cantoni and Ronchetti (2001). The subdirectory `help3.4` (and `help6.0`) contains the help files for the respective versions of the software.

The main routines are `glm.rob` for estimation, and `quasi.rob` for model selection. See their help files and the example below for details.

## 3.1 Installation

### Unix

To restore the files from the archive use the command `tar xvf robGLM.tar`: a directory called `robGLM` is created. In order to use these functions, go to the directory `robGLM` and run the proper installation file depending on the software version you are using. You are then ready to open an S-PLUS session in this directory.

### Windows

Decompress the file `Mspline.tar` with, for example, WinZip. A folder `robGLM` is created. Move the HTML help files (the content of the folder `helpHTML`) to the folder `.Data\__Hhelp`. Source the code (`ROBGLMsfun.s`) within S-PLUS.

## 4 An example

As a tutorial example, we will reproduce the results of Section 5.1 in Cantoni and Ronchetti (2001), Tables 1 and 2. This application is on a well known dataset, called the damage carrots dataset. It is taken from Phelps (1982) and used in McCullagh and Nelder (1989) to illustrate diagnostic techniques because of the presence of a large outlier in the  $y$ -space.

We first restore the data from the dump file:

```
> restore("carrots.dump")
> carrots[1:4,]
  success total logdose block block1 block2
1      10     35   1.52     1      1      0
2      16     42   1.64     1      1      0
3       8     50   1.76     1      1      0
4       6     42   1.88     1      1      0
```

The data give the proportion of carrots showing insect damage in a soil experiment where a trial with three blocks and eight dose levels of insecticide were considered.

We model the data with a binomial model  $\mathcal{B}(m, \pi)$  with logit link, that is,

$$\log\left(\frac{\mu}{m - \mu}\right) = \beta_0 + \beta_1 \log(\text{dose}) + \beta_2 \text{block2} + \beta_3 \text{block1},$$

where `block1` and `block2` are dummy variables taking the value 1 if measures are taken in block 1 or 2 respectively.

The classical fit with the S-PLUS `glm` function can be obtained by

```
> carrots.class <- glm(cbind(success,total-success)~logdose+block2+block1,
  data=carrots,family=binomial)
```

with part of the summary being:

```
Call: glm(formula = cbind(success,total-success)~logdose+block2+block1,
  family=binomial, data=carrots)
```

Coefficients:

	Value	Std. Error	t value
(Intercept)	1.4802191	0.6554212	2.258424
logdose	-1.8173780	0.3433617	-5.292897
block2	0.8432622	0.2257122	3.736007
block1	0.5423816	0.2314707	2.343198

(Dispersion Parameter for Binomial family taken to be 1 )

Null Deviance: 83.34426 on 23 degrees of freedom

Residual Deviance: 39.97575 on 20 degrees of freedom

Note that this classical results can be reproduced with our software by letting  $c = \infty$ :

```
carrots.class2 <- glm.rob(as.matrix(carrots[,c(3,6,5)]),as.matrix(carrots[,1]),
  choice="binom",chuber=Inf,weights.on.x=F,ni=as.matrix(carrots[,2]))
```

Figure 1: Diagnostic fits for the carrots dataset.

The validation of the model with different techniques – plot of deviance residuals, plot of Pearson residuals, and Cook’s distance (see Figure 1, produced with `glm.diag` from the `bootlib` S-PLUS library that accompany Davison and Hinkley 1997) – highlights the presence of a single large outlier (observation 14). No leverage point appears from this analysis. We therefore decide to use the robust technique by letting  $c = 1.2$  in Equation (8). This value of  $c$  is obtained by allowing a maximal error of 0.02 on the asymptotic level and by fixing a percentage of contamination of 4% (see details in Section 5.1 in Cantoni and Ronchetti 2001). All the weights on the design were put to 1, because the  $h_i$  values (hat matrix) were small for all the observations. This gives the following command lines:

```
> carrots.rob <- glm.rob(as.matrix(carrots[,c(3,6,5)]),as.matrix(carrots[,1]),
  choice="binom",chuber=1.2,weights.on.x=F,ni=as.matrix(carrots[,2]))
```

From the object `carrrots.rob` we can for example extract the estimated coefficients and their standard deviation:

```
> carrots.rob$coeff
[1] 1.9386955 -2.0491545 0.6847383 0.4496158
> carrots.rob$sd.coef
[1] 0.6974640 0.3684791 0.2360487 0.2409132
```

which arises some doubts about the significance of the variable `block1`. Note that this fact is hidden in the classical analysis.

We therefore perform a robust stepwise procedure to build the final model. We start by checking the pertinence to add the variable `logdose` to the null model and continue with the variable `block2` and then `block1`. We keep the same parameter setting as for the estimation procedure, that is,  $c = 1.2$  and all the weights on the design equal to 1. The S-PLUS command lines are:

```
step1 <- quasi.rob(as.matrix(carrots[,3]),as.matrix(carrots[,1]),out.col=1,
  choice="binom",chuber=1.2,weights.on.x=F,ni=as.matrix(carrots[,2]))
step2 <- quasi.rob(as.matrix(carrots[,c(3,6)]),as.matrix(carrots[,1]),
  out.col=2, choice="binom",chuber=1.2,weights.on.x=F,
  ni=as.matrix(carrots[,2]))
step3 <- quasi.rob(as.matrix(carrots[,c(3,6,5)]),as.matrix(carrots[,1]),
  out.col=3,choice="binom",chuber=1.2,weights.on.x=F,
  ni=as.matrix(carrots[,2]))
```

Note that the `out.col` argument is used to identify the column(s) of the explanatory variables that we are considering to add.

From each created object, one can extract the value of the residual robust quasi-deviance at the larger and at the submodel, the difference of quasi-deviances (statistic (5)) and its associated p-value, for example:

```
> step1$quasi.dev.hat
[1] 39.94188
> step1$quasi.dev.dot
[1] 60.47899
> step1$diff.quasi.dev
[1] 20.53711
> step1$pvalue
[1] 6.329916e-07
```

These results mean that it is worthwhile to add the variable `logdose` to the model at the 5% level. The p-values of the other two tests are `step2$pvalue = 0.017` and `step3$pvalue = 0.085`, which means that the significance of variable `block1` is not clearcut.

The classical results can be reproduced by putting `chuber=Inf` in the above command lines. This is equivalent to using the `anova` built-in S-PLUS function to compare nested models, as it is shown in the following.

If we want to test in a classical framework whether the variable `block1` plays an important role, then we would use the following commands with our software:

```

step3.class <- quasi.rob(as.matrix(carrots)[,c(3,6,5)],as.matrix(carrots)[,1],
  out.col=3,choice="binom",chuber=Inf,weights.on.x=F,ni=as.matrix(carrots)[,2])
> step3.class$quasi.dev
[1] 5.61448
> step3.class$pvalue
[1] 0.01781266

```

Using the S-PLUS anova function this will be:

```

> anova(carrots.glm,test="Chisq")
Analysis of Deviance Table

```

Binomial model

Response: cbind(carrots\$succes, carrots\$total - carrots\$succes)

Terms added sequentially (first to last)

	Df	Deviance	Resid.	Df	Resid.	Dev	Pr(Chi)
NULL				23		83.34426	
carrots\$logdose	1	28.61519		22		54.72906	0.00000009
carrots\$block2	1	9.13883		21		45.59023	0.00250241
carrots\$block1	1	5.61448		20		39.97575	0.01781266

where the last line matches our test in `step3.class`.

In the classical analysis no doubts arise on the significance of `block1` (at the 5% level). This is probably a masking effect due to the outlying observation.

## References

- Cantoni, E. and E. Ronchetti (2001). Robust inference for generalized linear models. *Journal of the American Statistical Association* 96, 1022–1030.
- Davison, A. C. and D. V. Hinkley (1997). *Bootstrap Methods and their Applications*. Cambridge: Cambridge University Press.
- Hampel, F. R. (1974). The influence curve and its role in robust estimation. *Journal of the American Statistical Association* 69, 383–393.
- Hampel, F. R., E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel (1986). *Robust Statistics: The Approach Based on Influence Functions*. New York: Wiley.
- McCullagh, P. and J. A. Nelder (1989). *Generalized Linear Models* (Second ed.). London: Chapman & Hall.

Phelps, K. (1982). Use of the complementary log-log function to describe dose-response relationships in insecticide evaluation field trials. In R. Gilchrist (Ed.), *Lecture Notes in Statistics, No. 14. GLIM.82: Proceedings of the International Conference on Generalized Linear Models*. Berlin/New York: Springer-Verlag.