Robust Inference for Generalized Linear Models: Binary and Poisson Regression

THÈSE N^{O} 4386 (2009)

PRÉSENTÉE LE 17 AVRIL 2009

À LA FACULTÉ SCIENCES DE BASE

CHAIRE DE STATISTIQUE APPLIQUÉE

PROGRAMME DOCTORAL EN MATHÉMATIQUES

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Sahar HOSSEINIAN

acceptée sur proposition du jury:

Prof. F. Eisenbrand, président du jury
Prof. S. Morgenthaler, directeur de thèse
Dr E. Cantoni, rapporteur
Dr G. Haesbroeck, rapporteur
Prof. V. Panaretos, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2009

To women in Iran

Acknowledgments

This present work, similar to a hike in the mountains of Switzerland, has taken time to achieve. This long-term project leaves some memories of people to whom I would like to express my gratitude.

I would like to express my deepest gratitude to my supervisor, Professor Stephan Morgenthaler for his continuous encouragement, patience, guidance and amiability throughout this work, without which this thesis would not have been accomplished.

I am also deeply grateful to Dr. Eva Cantoni (Univ. Genève, Switzerland), Dr. Gentiane Haesbroeck (Univ. Liège, Belgium) and Professor Victor Panaretos (EPFL Lausanne, Switzerland) for their willingness to be members of my jury. Thanks also to Professor Friedrich Eisenbrand (IMA, EPFL) for presiding the jury.

I would like to thank Professor Alireza Nematollahi for his help during the short period that he stayed at EPFL.
I express my thanks to all present and former graduate students of the three statistics chairs. In particular, I would like to thank Andrei to

bring me to EPFL, my office mate Nicolas for his help and his humor and Mehdi for all black board discussions.

Further, I thank all of my friends who encouraged me all during the years I was working on my thesis. I specially thank Mahkameh and Natasha my best friends far away, whose friendship accompanied me throughout this work. Very special thanks go to Rana, Basira and again Rana who gave me unconditional support all along this work. A big 'Thanks' to Soazig and Katia. Without their support and friendship I would never have reached the present state. I thank Jamie for his English corrections and Hamed for his help with LateX. And thanks to Meritxell for the coffee breaks during the thesis writing period.

And of course I am grateful to Super-Jürgen, who contributed with all his enthusiasm, his humanity and his patience throughout these years of my work. Thanks for staying with me through the difficult moments and shares the beautiful moments at Lutry.
Finally, my gratitude to my family, my parents and my brothers and big sisters who are far away, who motivated me to take this way. And special thanks to my family Baheri to to have welcomed me at Neuchâtel when I arrived in Switzerland.

Abstract

Generalized Linear Models have become a commonly used tool of data analysis. Such models are used to fit regressions for univariate responses with normal, gamma, binomial or Poisson distribution. Maximum likelihood is generally applied as fitting method.
In the usual regression setting the least absolute-deviations estimator (L_1-norm) is a popular alternative to least squares (L_2-norm) because of its simplicity and its robustness properties. In the first part of this thesis we examine the question of how much of these robustness features carry over to the setting of generalized linear models. We study a robust procedure based on the minimum absolute deviation estimator of Morgenthaler (1992), the L_q quasi-likelihood when $q = 1$. In particular, we investigate the influence function of these estimates and we compare their sensitivity to that of the maximum likelihood estimate.
Furthermore we particulary explore the L_q quasi-likelihood estimates in binary regression. These estimates are difficult to compute. We derive a simpler estimator, which has a similar form as the L_q quasi-likelihood estimate. The resulting estimating equation consists in a simple modification of the familiar maximum likelihood equation with the weights $w_q(\mu)$. This presents an improvement compared to other

robust estimates discussed in the literature that typically have weights, which depend on the couple $(\mathbf{x}_i, y_i)$ rather than on $\mu_i = h(x_i^T \beta)$ alone. Finally, we generalize this estimator to Poisson regression. The resulting estimating equation is a weighted maximum likelihood with weights that depend on μ only.

Keywords: Robustness, Generalized linear models, Binary regression, Poisson regression, Maximum quasi-likelihood.

Version abrégée

Les modèles linéaires généralisés (GLM) ont été dévéloppés afin de permettre l'ajustement de modèles pour des données réponses univariées issues de distributions telles que les lois Normale, Poisson, Binomiale, Exponentielle et Gamma. En général, le principe du maximum de vraisemblance est appliqué comme méthode d'ajustement pour ces modèles.

Dans les modèles de régression, les estimateurs de régression LAD (norme L_1) offrent des approches alternatives aux estimateurs des moindres carrés (norme L_2), grâce à leur caractéristique principale de robustesse. Dans le première partie de cette thèse, nous cherchons à déterminer si cette propriété de robustesse se retrouve pour les modèles linéaires généralisés. Nous examinons une approche de quasi-vraisemblance basée sur la norme L_q, introduite par Morgenthaler (1992). Plus particulièrement, nous nous intéressons à la fonction d'influence de cet estimateur et la comparons avec celle des estimateurs de vraisemblance .

Par la suite, nous étudions des méthodes de L_q quasi-vraisemblance pour les modèles binaires. Le calcul de ces estimateurs étant difficile, nous introduisons un nouvel estimateur similaire à ceux de la L_q quasi-

vraisemblance mais présentant une plus grande facilité de calcul. Le résultat est une modification de l'estimateur du maximum de vraisemblance pondérée par $w_q(\mu)$. Ceci constitue une amélioration par rapport à d'autres méthodes robustes dont la fonction de poids dépend de $(\mathbf{x}_i, y_i)$, au lieu de $\mu_i = h(x_i^T \beta)$ seulement.
Finalement, nous généralisons cet estimateur aux modèles de régression Poissonien. Notre résultat est une méthode de maximum de vraisemblance pondérée, dont les poids ne dépendent que de μ.

Mots-clés : Robustesse, Modèles linéaires généralisés, Régression binaire, Régression de Poisson, Maximum quasi-vraisemblance.

Contents

CHAPTER 1

Introduction

Statistics is the mathematical science of interpretation and explanation of data. It consists of methods for data analysis in a wide variety of scientific domains such as medicine, finance, crime and so forth. Regression analysis has became one of the most important tools for application in sciences. The most popular regression estimator was introduced in the work of Gauss and Legendre (see Plackett (1972), Stigler (1981) and Stigler (1986) for some historical discussions). This is the well-known *least squares* (LS) method, which corresponds to optimizing the fit by minimizing the sum of the squares of residuals. LS has been adopted because of its ease of computation.

However, real data often contains outliers. Outlying observations, both in the dependent and the explanatory variables, can have a strong effect on the LS estimate. The outlier problem is well-known and simply removing them as a solution is as old as statistics. In the last century a large body of literature on outliers and their dangers to classical statistical procedures has appeared.

To solve this problem, new statistical techniques have been developed that are not so easily affected by outliers. These are the robust methods. Robust (or resistant) procedures, pioneered in the work of E.S. Pearson, G.E.P. Box and J.W. Tukey (1960), are statistical techniques whose results remain trustworthy even if there is substantial contamination in the data. More formal theories of robustness have been developed in 1970s. In these theories, robustness signifies insensitivity to small deviations from the model assumption. Robust methods do not hide the outliers nor do they remove them. They will yield a reliable result even in their presence.
Least absolute deviations (LAD) or the L_1-norm is one of the principle alternative methods to LS (see Bloomfield and Steiger (1983) for more details). This method corresponds to a mathematical optimization technique similar to the least squares technique, which obtains the estimate by minimizing the absolute value of residuals. These estimates are resistant to outliers. Surprisingly, LAD curve-fitting was introduced a half-century earlier than LS by R. J. Boscovich (1757). Thirty years later P. S. Laplace (1789) gave an algebraic explanation of how to find the estimate of the parameter. Least absolute deviations has not been widely adopted because of their complicated computation. Today, the improvement in computer technology has removed this difficulty.

In the classical regression models, described above, the normal distribution plays a central role. Inference procedures for linear regression models in fact assume that the response variable Y follows a normal distribution. There are many practical situations where this assumption is not going to be even approximately satisfied. For example, suppose the response variable is a discrete variable, which we often encounter in counted responses, such as number of injuries, patients with particular diseases, and even the occurrence of natural phenomena including earthquakes. There are also many situations where the response variable is continuous, but the assumption of normality is completely unrealistic, such as distribution of stress in mechanical components and failure times of electronics components or systems.

Nelder and Wedderburn (1972) introduced the term *generalized linear model* (GLM) (McCullagh and Nelder, 1983). They unified various statistical models under this heading, namely regression models for univariate response variables that are distributed according to an exponential family. The exponential family includes the normal, binomial, Poisson, geometric, negative binomial, exponential, gamma and normal-inverse distributions. A generalized linear model links the expected value of the response, $E(y_i) = \mu_i$, to a linear prediction $\eta_i = \mathbf{x}_i^T \boldsymbol{\beta}$ with a monotone function, called the *link*. The classical parameter estimate for such models is the value that maximizes the likelihood, which in many cases is equivalent to maximizing a quasi-likelihood function, Wedderburn (1974).

The non-robustness of the maximum likelihood estimator for $\boldsymbol{\beta}$ has been extensively discussed. Robust alternatives for generalized linear models are proposed by many authors: see for instance, the work of: Stefanski *et al.* (1986b); Copas (1988); Künsch *et al.* (1989); Morgenthaler (1992); Carroll and Pederson (1993); Bianco and Yohai (1996); Marazzi and Ruffieux (1996), Cantoni and Ronchetti (2001) and Croux and Haesbroeck (2003).

The particular motivation of this thesis is to study the robustness problem in the context of generalized linear models and in a specific case, the logistic regression. Since in the classical regression model, the replacement of the L_2-norm by least absolute-deviations, the L_1-norm, has been shown to lead to resistance against outliers, we are interested in studying the specific LAD in generalized linear models. Morgenthaler (1992) replaced the L_2-norm in the definition of quasi-likelihood by an arbitrarily chosen L_q-norm for $q \geq 1$, expecting that this might similarly yield robust estimates. In this thesis we study the robustness behavior of this L_q quasi-likelihood for the GLM. In particular, we consider the logistic case and simplify these estimators in a new estimator. This new estimator has a similar form as the L_q quasi-likelihood estimate, but is easier to compute and results in more resistant estimates.

Thesis' Outline

In Chapter 2 a detailed overview of the theory and the historical development of regression models and generalized linear models is presented. It focuses on the robust estimates for these models. The chapter is organized to reflect the historical development of robust models from 1980 to 2008 and from basic to general. In Chapter 3 the robust behavior of the L_q quasi-likelihood estimator is examined and the computational difficulties of these estimators are studied. In Chapter 4 we introduce a new weighted maximum likelihood estimator, which is similar to the L_q quasi-likelihood for logistic regression. The properties of these estimators are discussed. We additionally propose the new weight function. Using simulation and examples, we compare these new estimates with estimates proposed in the literature. In Chapter 5 we generalize this estimator to the Poisson model. Finally, in Chapter 6, we present the conclusions and final remarks.

CHAPTER 2

Regression and Generalized Linear Models

2.1 Regression models

2.1.1 Linear regression

In this chapter we begin the discussion on fitting equations and the estimation of the parameters of simple linear regression models. In simple regression, we are interested in studying the relation between the explanatory variable X and the response variable Y,

$$y_i = \beta_1 + \beta_2 x_i + \varepsilon_i \qquad \text{for} \quad i = 1, \cdots, n\,, \tag{1}$$

where n is the number of observations. The error is assumed to be independent and be normally distributed with mean zero and unknown standard deviation σ. The goal is to estimate the regression coefficients $\boldsymbol{\beta} = (\beta_1\,, \beta_2)$.
The most popular regression estimator (Gauss and Legendre 1794-95) is the method of *least squares* (LS), which consists of estimating $\boldsymbol{\beta}$ by

$(\hat{\beta}_1, \hat{\beta}_2)$ such that the residuals

$$r_i = y_i - \hat{y}_i = y_i - (\hat{\beta}_1 + \hat{\beta}_2 x_i)$$

satisfy

$$\sum_{i=1}^{n} r_i^2 = \min_{\boldsymbol{\beta}}! \tag{2}$$

The solution to this optimization problem is,

$$\hat{\beta}_2 = \frac{\sum y_i (x_i - \bar{x})}{\sum (x_i - \bar{x})^2} \qquad \hat{\beta}_1 = \bar{y} - \beta_2 \bar{x}\,, \tag{3}$$

where $\bar{x} := \frac{1}{n} \sum x_i$, $\bar{y} := \frac{1}{n} \sum y_i$.

Example 1. *Figure 1 is a scatter plot of five points. The red point is an outlier in the direction of x. The figure shows the LS fit computed without the outlier has the better representation, however, the LS result has been strongly affected by a single outlier.*

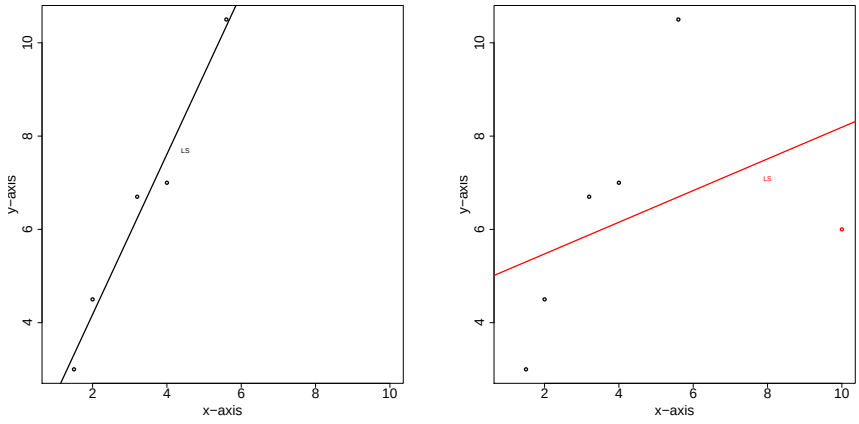

Figure 1: *Original data with five points and their LS line(left). Same data by adding an outlier in x direction and their LS line (right).*

Now we consider the more general case of a data set in multiple regression. Let $(\mathbf{x}_i \in \mathbb{R}^p, y_i \in \mathbb{R})$, for $i = 1, \cdots, n$ be pairs of explanatory and response variables. Where the vector $\mathbf{x}_i = (1, x_{i1}, \cdots, x_{i(p-1)})^T$. The data is assumed to follow the linear model

$$y_i = \mathbf{x}_i^T \boldsymbol{\beta} + \varepsilon_i \,, \tag{4}$$

where $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_p)$ are unknown parameters to be estimated and the error is a random variable with a normal distribution $\varepsilon_i \overset{i.i.d}{\sim} N(0, \sigma^2)$. The design matrix is $\mathbf{X} = (\mathbf{1}, \mathbf{x_1}, \cdots, \mathbf{x_{p-1}}) \in \mathbb{R}^{n \times p}$ with non-random (or fixed) variables. Let $\mathbf{y}$ and $\boldsymbol{\varepsilon}$ be the vectors with elements y_i and ε_i respectively. Then the linear model (4) may be written as

$$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}. \tag{5}$$

The fitted value $\widehat{\mathbf{y}}$ and residuals $\mathbf{r}$ correspond to

$$\widehat{\mathbf{y}} = X\boldsymbol{\beta} \qquad\qquad \mathbf{r} = \mathbf{y} - \widehat{\mathbf{y}}.$$

As in simple regression, the least squares technique results by minimizing the squares of residuals,

$$(\mathbf{y} - X\hat{\boldsymbol{\beta}})^T(\mathbf{y} - X\hat{\boldsymbol{\beta}}) = \min_{\boldsymbol{\beta}}!$$

This notation means the unknown parameters $\boldsymbol{\beta}$ is estimated by the value which minimizes the sum of the squared of residuals. Differentiating with respect to $\boldsymbol{\beta}$ yields $X^T(\mathbf{y} - X\hat{\boldsymbol{\beta}}) = 0$, then we can obtain

$$\hat{\boldsymbol{\beta}} = (X^T X)^{-1} X^T \mathbf{y}. \tag{6}$$

The least squares method has been adopted because of its ease of computation before the advent of electronic computer. However the LS method is still sensitive to the presence of outliers. In the next part of this chapter we study some of the robust methods that have been already proposed in the literature for regression models.

2.1.2 Robust regression methods

Certainly one can ask the question why we are interested in robust methods? Robust means remaining resistant against some unpredictable variations. In statistics, models are mere approximations of reality. The models that underlie many statistical procedures are overly optimistic and in real data gross errors occur with surprisingly large frequency. This kind of observation, which lies far from the mass of data is called outlier. The outlier can affect statistical models and results in an assumed model far from the true one. Robustness signifies insensitivity against some deviation from the true model. Robust procedures were pioneered in the works of E.S. Pearson, G.E.P. Box and J.W. Tukey (1960) and more formal theories of robustness have been developed in the 1970s.

In regression models, the aim of robust methods is to identify the outliers and highly influential data points, leverage points, and finally to describe the best fit for the data.

One of the first alternatives to LS, as a robust estimator, was proposed by Edgeworth (1887), who improved the Boscovich's (1757) proposal. This estimator is the *least absolute deviation* (LAD) or L_1 regression, which is determined by

$$\sum_{i=1}^{n} |r_i| = \min_{\boldsymbol{\beta}}! \tag{7}$$

The LAD estimator, like the median, is not uniquely defined[1]. Whereas the breakdown point (see 2.2.1) of the median is as high as 50%, the breakdown point of the LAD regression estimate is still 0% for outliers in the explanatory variable (high leverage points). This means that a single leverage point can affect the L_1 estimator.

The next innovation was Huber's M-estimator (Huber (1973) and Huber (1981)). Huber proposed to generalize the squares of residuals in LS to

$$\sum_{i=1}^{n} \rho(r_i/c\hat{\sigma}) = \min_{\boldsymbol{\beta}}! \tag{8}$$

[1]See e.g Harter (1977), Gentle et all. (1977) and Bloomfield and Steiger (1983).

Here ρ is a function, $\hat{\sigma}$ is an auxiliary scale estimate and c is a tuning constant. Differentiating this expression with respect to the regression coefficient $\boldsymbol{\beta}$, one can find the following system of equations:

$$\sum_{i=1}^{n} \psi(r_i/c\hat{\sigma})\mathbf{x}_i = \mathbf{0} \quad \in \mathbb{R}^p, \tag{9}$$

where $\mathbf{x}_i$ and $\mathbf{0}$ are the vectors $(1\,, x_{i1}\,, \cdots\,, x_{i(p-1)})$ and $(0\,, \cdots\,, 0)$, respectively. The LS and LAD estimators are M-estimators with ρ-function $\rho(r_i) = r_i^2$ and $\rho(r_i) = |r_i|$, respectively. After the general definition of M-estimator, different functions of $\psi(r_i)$ were proposed in the literature to achieve the best robust estimates.
Huber (1964) introduced an estimator, which is given by

$$\psi_c(x) = \begin{cases} x & \mid x \mid \leq c\,. \\ c\,\mathrm{sgn}(x) & \mid x \mid > c\,. \end{cases} \tag{10}$$

This M-estimator (9), like the L_1 estimator, is robust as long as no outliers in x-space are introduced. However, they are still affected by leverage points and their breakdown points are 0%. Therefore, in order to bound the influence of high leverage points, *generalized M-estimators* (GM-estimates) were introduced by considering some weight function of $\mathbf{x}_i$. Two particular forms of GM-estimates have been studied in the literature. The first, *Mallows-estimators*, were proposed by Mallows (1975) and replaces (9) by

$$\sum_{i=1}^{n} w(\mathbf{x}_i)\psi\left(\frac{r_i}{\hat{\sigma}}\right)\mathbf{x}_i = \mathbf{0}\,. \tag{11}$$

The second one, *Schweppe-estimators*, were first proposed by Schweppe (1975) and is also called *Hampel-Krasker-Welsch* estimator (Krasker and Welsch (1982)). These estimates are given by

$$\sum_{i=1}^{n} w(\mathbf{x}_i)\psi\big(\frac{r_i}{w(\mathbf{x}_i)\hat{\sigma}}\big)\mathbf{x}_i = \mathbf{0}\,. \tag{12}$$

The choice of w and ψ were considered to bound the influence of a single outlier. This influence can be measured by the influence function, which

was defined by Hampel (1974) (see 2.2.2). Therefore, the corresponding GM-estimator is now generally called the *bounded-influence-estimator*. Other estimators of the *GM-estimator* family were proposed in the literature, but none of them achieves a breakdown point greater than 30%.

Rousseeuw (1984) proposed the *least median squared* (LMS) method with 50% breakdown point, defined as

$$\text{med } r_i^2 = \min_{\boldsymbol{\beta}}! \tag{13}$$

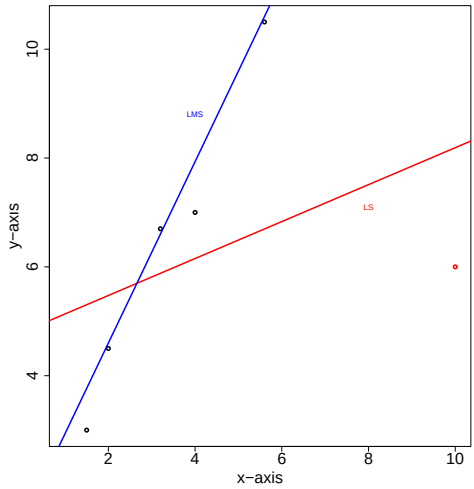

Figure 2: *Five original points with a leverage point and their LS and LMS fitting.*

In Figure 2 we consider Example 1 (page 20). The blue line represents the estimate fitted by the LMS method. The LMS estimate is resistant against all types of outliers, but it has an efficiency disadvantage. The LMS converges weakly with speed $n^{-1/3}$, rather than the usual[2] $n^{-1/2}$.

[2]See Rousseeuw and Leroy (1987) Section 4 of Chapter 4.

Hence Rousseeuw (1984) introduced the *least trimmed square* (LTS) estimate as

$$\sum_{i=1}^{h} (r_i^2)_{i:n} = \min_{\boldsymbol{\beta}}! \tag{14}$$

where $(r^2)_{1:n} \leq \cdots \leq (r^2)_{n:n}$ are the ordered squared residuals. The value of h defines how many of the largest squared residuals should not be considered for fitting the model. The best value of h is $n/2$ for which these estimates achieve their best breakdown points of 50%. Unlike the slow convergency rate of the LMS, the LTS converges at the rate $n^{-1/2}$.
In this section we reviewed several popular robust estimators for regression models. The next section describes two other aspects, which measure the robustness properties of an estimator, the breakdown point and the influence function.

2.2 Measuring robustness

In Section 2.1.2 we have seen that even a single outlier can change the result of some estimators. On the other hand, some estimators perform well with a certain percentage of data contamination. One can also be interested in measuring the effect of infinitesimal perturbations on the estimator. In the following we introduce two measures of the robustness of an estimator, the *breakdown point* and the *influence function.*

2.2.1 The Breakdown point

The breakdown point (BP) of an estimator $\hat{\boldsymbol{\beta}}$ of the true parameter $\boldsymbol{\beta}$ represents the largest percentage of data that can be contaminated such that $\hat{\boldsymbol{\beta}}$ still gives accurate information about $\boldsymbol{\beta}$.
The notation of BP was briefly described by F. Hample(1968) in his Ph.D. thesis. Hampel (1971) later gave a more general definition of BP. Here we first present a simple version, the finite-sample breakdown point, which was introduced by Donoho and Huber (1983). We will then describe the asymptotic BP (Hampel (1971)).

Let $Z_i = (\mathbf{x}_i \in \mathbb{R}^p, y_i \in \mathbb{R})$ for $i = 1, \cdots, n$ be a sample of size n. We consider some contamination of the original data set Z, resulting in the corrupted sample Z'. The fraction of corruption, ε, in the sample is given by

$$\varepsilon = \frac{n+m}{n} \quad \text{“adding” } m \text{ points to data set Z}$$

$$\varepsilon = \frac{m}{n} \quad \text{“replacing” } m \text{ points in data set Z.}$$

Let $\hat{\boldsymbol{\beta}}(Z)$ be a regression estimator. We first define the maximum bias that can be caused by such a contamination as

$$\text{bias}(\varepsilon, Z, \boldsymbol{\beta}) = \sup_{Z'} \mid \boldsymbol{\beta}(Z') - \boldsymbol{\beta}(Z) \mid \ . \tag{15}$$

If this bias is infinite, this means that the contamination has a large effect on the estimator. The BP (finite-sample) of the estimate $\boldsymbol{\beta}$ at sample Z is given by

$$\varepsilon^*(Z, \boldsymbol{\beta}) = \inf\{\varepsilon; \text{bias}(\varepsilon, Z, \boldsymbol{\beta}) = \infty\}\,, \tag{16}$$

which is the smallest fraction of corruption that causes an arbitrary large variation from the sample estimate. Note that this definition contains no distribution function. The following definition of breakdown point is introduced by Hampel (1971), namely the asymptotic contamination BP of an estimator.
Let $\hat{\boldsymbol{\beta}} \in \Theta$ be the asymptotic limit of our estimator at the underlying model distribution F. The BP of this estimator is denoted by $\varepsilon^*(\hat{\boldsymbol{\beta}}, F) \in (0, 1)$ such that for any $\varepsilon < \varepsilon^*$ and a probability function G, there exist a compact set $k \subset \Theta$ such that

$$\hat{\beta}((1-\varepsilon)F + \varepsilon G) \in k. \tag{17}$$

The breakdown point for different estimators has be studied separately and for reasonable estimates its value cannot exceed 50%.

2.2.2 Influence function

The influence function (IF) was introduced by Hampel (1974) under the name "influence curve" (IC). The IF is a tool to study the local robustness properties of an estimator when a small fraction of outliers appears in the sample.
The IF of an estimator $\boldsymbol{\beta}$ at the probability distribution $F_{(X,Y)}(x,y)$ is defined as

$$\mathrm{IF}(\mathbf{x},y;\boldsymbol{\beta},F)=\lim_{t\to 0}\frac{\boldsymbol{\beta}((1-t)F+t\triangle_{(\mathbf{x},y)})-\boldsymbol{\beta}(F)}{t}, \tag{18}$$

where $\triangle_{(\mathbf{x},y)}$ denotes the point mass 1 at $(\mathbf{x},y)$ and this limit exists. In the regression model, $F_{(X,Y)}(x,y)$ is the joint distributions of (X,Y), which is given by $H_X(x)G_{Y|X}$. H is the probability distribution of X and G is the probability distribution of the errors.
The IF measures the asymptotic bias caused by infinitesimal contamination in the data. From the influence function, one can compute the asymptotic variance of the estimator[3]

$$V(\boldsymbol{\beta},F)=\int \mathrm{IF}(\mathbf{x},y;\boldsymbol{\beta},F)\mathrm{IF}(\mathbf{x},y;\boldsymbol{\beta},F)^T dF(\mathbf{x},y). \tag{19}$$

Later, Hampel gave a general definition of the influence function for the M-estimator, which is the solution of the estimating equation

$$\sum \psi(\mathbf{x}_i,y_i;\boldsymbol{\beta})=0\in\mathbb{R}^p.$$

The IF of these estimates is given by the following expression (Hampel *et al.* (1986)):

$$\mathrm{IF}(\mathbf{x},y,\boldsymbol{\beta},F)=M(\psi,F)^{-1}\psi(\mathbf{x},y,\boldsymbol{\beta})\,, \tag{20}$$

where $M(\psi,F)=-E_F\big[\frac{\partial}{\partial\boldsymbol{\beta}}\psi(\mathbf{x},y,\boldsymbol{\beta})\big]\in\mathbb{R}^{p\times p}$.
Moreover, these estimators are asymptotically normal with asymptotic covariance matrix given by

$$V(T,F)=M(\psi,F)^{-1}Q(\psi,F)M(\psi,F)^{-1}\,,$$

[3]For more details see Hampel *et al.* (1986) p. 85.

where $Q(\psi, F) = E_F\left[\psi(\mathbf{x}, y, \boldsymbol{\beta})\psi(\mathbf{x}, y, \boldsymbol{\beta})^T\right]$.
The maximum likelihood estimator, for example, is an M-estimator corresponding to $\rho(\mathbf{x}, y, \boldsymbol{\beta}) = -\log g_{\boldsymbol{\beta}}(y, \boldsymbol{\beta}|\mathbf{x})$. Here $g_{\boldsymbol{\beta}}(.)$ is the conditional probability density function of y given $\mathbf{x}$. Then

$$\mathrm{IF}(\mathbf{x}, y, \boldsymbol{\beta}, F) = J(F)^{-1}\frac{\partial}{\partial \boldsymbol{\beta}}\log g_{\boldsymbol{\beta}}$$

and $V(\boldsymbol{\beta}, F) = J(F)^{-1}$ where

$$J(F) = \int (\frac{\partial}{\partial \boldsymbol{\beta}}\log g_{\boldsymbol{\beta}})^2 dF(\mathbf{x}, y),$$

where $F_{(X,Y)}(\mathbf{x}, y)$ is the joint distribution of (X, Y) as defined above.

2.2.3 Sensitivity curve

The definition of the influence function, described above, is asymptotic since we consider the estimator's asymptotic value. However, there exist a simple version called the *Sensitivity curve*(SC), introduced by Tukey (1970-71). Consider the sample $(z_1, \cdots, z_{n-1})$ of $n-1$ observations, then the SC of the estimator T in case of additional observation z is given by

$$SC_n(z) = n[T_n(z_1, \cdots, z_{n-1}, z) - T_{n-1}(z_1, \cdots, z_{n-1})]. \qquad (21)$$

From this definition one can show that the IF of an estimator is an asymptotic version of the SC. By considering the estimator $T_n(z_1, \cdots, z_n) = T(F_n)$ as a functional of empirical distribution F_n of $z_1, \cdots, z_n$, the SC can be transformed to

$$SC_n(z) = \frac{\left[T(1-\frac{1}{n})F_{n-1} + \frac{1}{n}\triangle_z) - T(F_{n-1})\right]}{1/n}$$

where F_{n-1} is the empirical distribution of $z_1, \cdots, z_{n-1}$. This expression is similar to the IF with contamination size $t = \frac{1}{n}$. This shows the fact that in many situations the $SC_n(z)$ converges to $\mathrm{IF}(z, T, F)$ when $n \to \infty$.

In the first section of this chapter we have discussed regression models. The basic assumption for the classical regression models is that the error follows a normal distribution and they are independent. In many cases, this assumption is not satisfied. Generalized linear models were introduced to unify various statistical models. The next section of this chapter will cover an overview of generalized linear models and their robust fitting.

2.3 Generalized linear models

The term *generalized linear models* (GLM) was coined by Nelder and Wedderburn (1972). They unified various statistical models for univariate response variables that follow a distribution from the exponential family[4].
Let $(\mathbf{x}_i \in \mathbb{R}^p, y_i \in \mathbb{R})$ for $i = 1, \cdots, n$ be the pair of explanatory and response variables. Generalized linear models relate the expected value of the response, $\mathbf{E}(y_i) = \mu_i$, to a linear prediction $\eta_i = \mathbf{x}_i^T \boldsymbol{\beta}$ as follows:

$$g[\mathbf{E}(y_i)] = g(\mu_i) = \mathbf{x}_i^T \boldsymbol{\beta}\,, \tag{22}$$

where $g(\mu)$ is some monotone function, called the *link*. The inverse link function is $h(\mathbf{x}_i^T \boldsymbol{\beta}) = \mu_i$. We assume that a random variable Y given X has a probability density function, depending on parameter $h(\mathbf{x}^T \boldsymbol{\beta})$, $f(y; h(\mathbf{x}^T \boldsymbol{\beta}))$. A set of distributions is said to belong to the exponential family, if it has a density of the form

$$f_Y(y, h(\mathbf{x}^T \boldsymbol{\beta}), \phi) = \exp\Big\{ \frac{y h(\mathbf{x}^T \boldsymbol{\beta}) - b(h(\mathbf{x}^T \boldsymbol{\beta}))}{a(\phi)} + c(y, \phi) \Big\}\,, \tag{23}$$

for some specific functions $a(.)\,, b(.)\,, c(.)$ and canonical parameter $h(\mathbf{x}^T \boldsymbol{\beta})$. Thus, many well known distributions are exponential families such as Poisson, normal, gamma and binomial distribution.

[4]For an elementary introduction to the subject see Dobson (2002) and for more details McCullagh and Nelder (1983).

Since the distribution of Y is an exponential family, we can directly derive its mean and variance. The log-likelihood of $f_Y(y; h(\mathbf{x}^T\boldsymbol{\beta}), \phi)$ is

$$l(h(\mathbf{x}^T\boldsymbol{\beta}), \phi, y) = \log f_Y(y; h(\mathbf{x}^T\boldsymbol{\beta}), \phi) = \frac{yh(\mathbf{x}^T\boldsymbol{\beta}) - b(h(\mathbf{x}^T\boldsymbol{\beta}))}{a(\phi)} + c(y, \phi)\,.$$

Then by considering the two relations $\mathbf{E}(\frac{\partial l}{\partial h(\mathbf{x}^T\boldsymbol{\beta})}) = 0$ and $\mathbf{E}(\frac{\partial^2 l}{\partial h(\mathbf{x}^T\boldsymbol{\beta})\partial h(\mathbf{x}^T\boldsymbol{\beta})}) + \mathbf{E}(\frac{\partial l}{\partial h(\mathbf{x}^T\boldsymbol{\beta})})^2 = 0$, one finds

$$\mathbf{E}(Y) = \mu = b'(h(\mathbf{x}^T\boldsymbol{\beta}))\,,$$

$$\mathrm{Var}(Y) = b''(h(\mathbf{x}^T\boldsymbol{\beta}))a(\phi)\,.$$

Thus, the variance of Y, $\mathrm{Var}(Y)$, is a product of two functions, the variance function, $V(\mu)$, which depends on canonical parameter $h(\mathbf{x}^T\boldsymbol{\beta})$, and $a(\phi)$ that does not depend on $h(\mathbf{x}^T\boldsymbol{\beta})$.
Different link functions, which relate the linear predictor to the expected value of Y are discussed in the literature. Note that each distribution as an exponential family, has a special link function called the *canonical link function*, which occurs when the canonical parameter defined in (23) is equal to linear prediction, $h(\mathbf{x}^T\boldsymbol{\beta}) = \mathbf{x}^T\boldsymbol{\beta}$. Table 2.3 summarizes the essential information for normal, binomial, exponential/gamma and Poisson distributions and their canonical link function.

Distribution	Notation	$V(\mu)$	C. link	Mean function, $h(\mathbf{x}^T\boldsymbol{\beta})$
Normal	$N(\mu, \sigma^2)$	1	identity	$\mathbf{x}^T\boldsymbol{\beta}$
Binomial	$B(n, \mu)$	$\mu(1-\mu)$	logit	$(1+\exp(-\mathbf{x}^T\boldsymbol{\beta}))^{-1}$
Exponential	$E(\mu)$	$1/\mu$	inverse	$(\mathbf{x}^T\boldsymbol{\beta})^{-1}$
Poisson	$P(\mu)$	μ	log	$\mathbf{x}^T\boldsymbol{\beta}$

In the generalized linear model, the unknown parameter $\boldsymbol{\beta}$ is typically estimated with the maximum likelihood method or the quasi-likelihood technique. In the next section we study these two methods.

2.3.1 Maximum likelihood estimates

Given the sample $(\mathbf{x}_i \in \mathbb{R}^p, y_i \in \mathbb{R})$ for $i = 1, \cdots, n$, where $y_i \sim f_Y(y_i, \mu_i)$. The *maximum likelihood estimator* (MLE) of parameter $\boldsymbol{\beta}$

in the model $\mu_i = h(\mathbf{x}_i\boldsymbol{\beta})$ is obtained by maximizing the log-likelihood, where the log-likelihood for independent observations $(y_1, \cdots, y_n)$ is given by $l(\mu) = \sum_i l_i(\boldsymbol{\beta})$ and

$$l_i(\mu) = \log f_Y(y_i; h(\mathbf{x}_i^T\boldsymbol{\beta}), \phi) = \frac{y_i h(\mathbf{x}_i^T\boldsymbol{\beta}) - b(h(\mathbf{x}_i^T\boldsymbol{\beta}))}{a(\phi)} + c(y_i, \phi).$$

Therefore, the log-likelihood as a function of $\boldsymbol{\beta}$ is

$$l(\boldsymbol{\beta}) = \sum_i l_i(\boldsymbol{\beta}),$$

then,

$$l(\boldsymbol{\beta}) = \max_{\boldsymbol{\beta}}! \tag{24}$$

This yields the p-dimensional system of equations, the score function, to estimate $\boldsymbol{\beta}$:

$$S(\hat{\boldsymbol{\beta}}) = \frac{\partial l}{\partial \hat{\boldsymbol{\beta}}} = \mathbf{0}. \tag{25}$$

2.3.2 Quasi-likelihood estimates

To derive a likelihood, we have to know the distribution function of the observation y, which is not always the case. Wedderburn (1974) introduced *maximum quasi-likelihood estimates* (MQLE) for fitting the generalized linear model. He showed that the quasi-likelihood is the same as log-likelihood function if and only if the family of response distribution is an exponential family. Suppose the response y_i has expectation μ_i and variance function, $V(\mu_i)$, where V is some known function. The parameter of interest, $\boldsymbol{\beta}$, relates to μ_i with inverse link function of $h(\mathbf{x}_i^T\boldsymbol{\beta}) = \mu_i$. Then for each observation the quasi-likelihood is given by

$$K(y_i, \mu_i) = \int_{y_i}^{\mu_i} \frac{y_i - t}{V(t)} dt\,, \tag{26}$$

or

$$\frac{\partial K(y_i, \mu_i)}{\partial \mu_i} = \frac{y_i - \mu_i}{V(\mu_i)}\,.$$

The function K has properties similar to those of log-likelihood[5].
To estimate $\boldsymbol{\beta}$, we maximize the quasi-likelihood function by differentiating with respect to $\boldsymbol{\beta}$, which yields

$$U(\boldsymbol{\beta}) = D^T V(\mu)^{-1}(\mathbf{y} - \boldsymbol{\mu}) = \mathbf{0}, \tag{27}$$

where D is the matrix of partial derivatives $D_{ij} = \partial\mu_i/\partial\beta_j$ and $V(\mu)$ is the diagonal matrix with elements $(V(\mu_1), \cdots, V(\mu_n))$. For $V(\mu) = 1$, the quasi likelihood reduces to the least square method.

Since in both the MLE and the MQLE, the system of equations to solve is nonlinear, one needs to use numerical method. The *Newton-Raphson procedure* is one of the best known and most widely used techniques, it was proposed by Isaac Newton and published in "Method of Fluxions" in 1736. Although the method was described by Joseph Raphson in "Analysis Aequationum" in 1690, the relevant sections of "Method of Fluxions" were written earlier, in 1671.
Consider the score equation $u(x) = 0$. The Newton-Raphson procedure requires the evaluation of a function $u(x)$ and its derivative $u'(x)$, at arbitrary points x. Geometrically, this method consists of extending the tangent at a current point x_i until it crosses zero, then setting the next approximation x_{i+1} to the value where the tangent intersect the abscissa. Algebraically, the method derives from Taylor series expansion of a function in the neighborhood of a point. At each iteration u is replaced by its Taylor expansion of order 1. Therefore, at iteration i, we have

$$u(x_i + \varepsilon) \approx u(x_i) + u'(x_i)\varepsilon + \frac{u''(x_i)}{2!}\varepsilon^2 + \cdots = 0\,,$$

then

$$x_{i+1} = x_i - \frac{u(x_i)}{u'(x_i)}.$$

The convergence of this method is not guaranteed but in nice circumstances it converges quadratically, which means the accuracy of the

[5]For more details see Wedderburn (1974).

approximation is doubled. One of the reasons that the method fails to converge happens when the denominator u' tends to zero in which case x_i becomes unreliable. For this reason we sometimes prefer other methods, such as Brent's method or the secant method, to achieve convergence. On the other hand, we can use the Newton-Raphson method with a suitable initial value to start the iteration.

The process of fitting in the classical linear models or in the GLM can be considered a way of replacing the observation $\mathbf{y}$ by a set of $\hat{\mathbf{y}}$ in regression or $\hat{\boldsymbol{\mu}}$ in the GLM. A question that might arise is how large the discrepancy of the fitted model is. One of the ways to measure this discrepancy is called residual, $r_i = y_i - \hat{y}_i$, for regression models and deviance for the GLM, which is defined as follows. The *deviance function* is twice the difference between the maximum achievable log-likelihood, which is attained at $\boldsymbol{\beta}^m$ (under model), and the maximum log-likelihood that is attained with fitted parameters $\hat{\boldsymbol{\beta}}$,

$$D(\mathbf{y}, \hat{\boldsymbol{\beta}}) = 2l(\mathbf{y}, \boldsymbol{\beta}^m) - 2l(\mathbf{y}, \hat{\boldsymbol{\beta}}). \tag{28}$$

The random variable $D(\mathbf{Y}, \hat{\boldsymbol{\beta}})$ is asymptotically distributed as χ^2_{n-p} where p is the number of fitted parameters and n is sample size. This fact is used to check the goodness of fit.
The deviance is the analogue of the residual sum of squares, which is familiar from the linear model. Thus one can estimate the unknown parameters by minimizing the deviance function, equivalent to maximizing the log-likelihood. In robust methods for generalized linear models, different functions of deviance have been suggested to obtain a robust estimator.

Example 2. *In this example we consider the mortality rates from a non-infectious disease for a large population. The number of deaths Y in a population, as independent events, can be modeled by a Poisson distribution.*

Figure 3 (left) is a scatter plot of death rate per 100,000 men (on a logarithmic scale) and their age group. This data corresponds to the

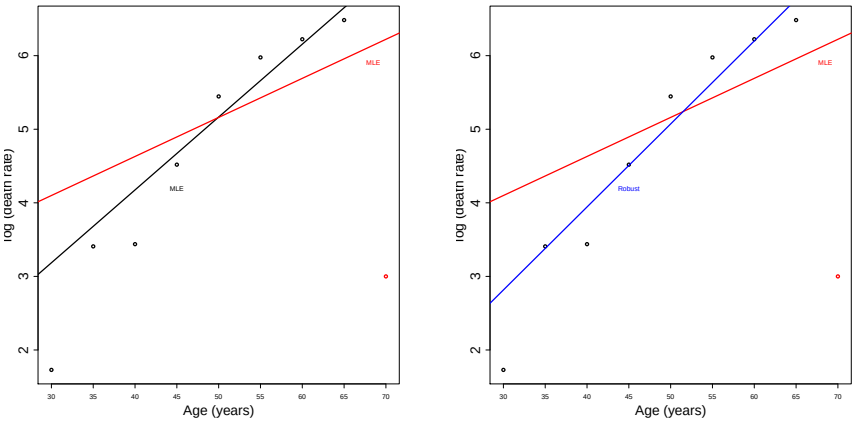

Figure 3: *Original data and an outlier with their MLE fitted lines. The black line shows the fitting without the outliers; the red line with the outliers. (left). Same data using a robust fitting method (right).*

Hunter region of New South Wales, Australia in 1991[6]. The red point represents a small death rate for a group of old men. This is considered as an added outlier. It shows that the MLE result has been affected strongly by this single outlier. In Figure 3 (right) the blue line represents the estimates computed using a robust method, which resists the effect of outliers.

As we have seen in this example, like the least squares method the maximum (quasi) likelihood methods are not robust. In the next section of this chapter we study robust methods for generalized linear models.

2.3.3 Robust generalized linear models

To robustify the MLE, Stefanski *et al.* (1986a) studied optimally bounded score functions for the GLM. They generalized results obtained by Krasker and Welsch (1982) for classical linear models. They suggested

[6]See Dobson (2002).

to find bounded-influence estimators, which minimize certain functions of the asymptotic covariance matrix. Consider the M-estimator of the form

$$\sum_{i=1}^{n} \psi(y_i, x_i, \boldsymbol{\beta}) = \mathbf{0}. \tag{29}$$

They defined a scalar measure of influence, the *self-standardized sensitivity* of the estimator $\hat{\boldsymbol{\beta}}$ as

$$S(\boldsymbol{\psi}) = \sup_{(\mathbf{x},y)} (\boldsymbol{\psi}^T W^{-1} \boldsymbol{\psi})^{1/2}, \tag{30}$$

where

$$W_{\boldsymbol{\psi}}(\boldsymbol{\beta}) = \mathop{\mathbf{E}}_{\boldsymbol{\beta}} \{\boldsymbol{\psi}(\boldsymbol{\beta}) \boldsymbol{\psi}(\boldsymbol{\beta})^T\}.$$

For maximum likelihood, $\boldsymbol{\psi} = l' = \partial l(\mathbf{y}, \mathbf{x}, \boldsymbol{\beta}) / \partial \boldsymbol{\beta}$ (25) and in general $S(l') = +\infty$. To obtain robustness, Stefanski *et al.* (1986a) chose the estimator $\hat{\boldsymbol{\beta}}_{\boldsymbol{\psi}}$ for which $S(\boldsymbol{\psi}) < b < \infty$ for a positive constant b. They determined the optimal score function, which turned out to be equal to

$$\boldsymbol{\psi}_{BI}(\mathbf{y}, \mathbf{x}, \boldsymbol{\beta}) = (l - C) min^{\frac{1}{2}} \big[1, \frac{b^2}{(l - C)^T B^{-1} (l - C)}\big], \tag{31}$$

where $l = l(\mathbf{y}, \mathbf{x}, \boldsymbol{\beta})$ is the log-likelihood and $C(\boldsymbol{\beta}) \in \mathbb{R}^P$ and $B(\boldsymbol{\beta}) \in \mathbb{R}^{P \times P}$ are defined to satisfy

$$\mathbf{E}\{\boldsymbol{\psi}_{BI}(\mathbf{y}, \mathbf{x}, \boldsymbol{\beta})\} = 0 \qquad B(\boldsymbol{\beta}) = \mathbf{E}\{\boldsymbol{\psi}_{BI} \boldsymbol{\psi}_{BI}^T\}.$$

The robust estimator of Stefanski *et al.* (1986a) is difficult to compute. Künsch *et al.* (1989) introduced another estimator, called the *conditionally unbiased bounded-influence estimate*. Künsch *et al.*'s estimator is in the subclass of M-estimators, whose score function is conditionally unbiased and satisfy

$$\mathop{\mathbf{E}}_{\boldsymbol{\beta}} (\boldsymbol{\psi}(y, \mathbf{x}, \boldsymbol{\beta} | \mathbf{x})) = \int \boldsymbol{\psi}(y, \mathbf{x}, \boldsymbol{\beta}) P_{\boldsymbol{\beta}}(dy | \mathbf{x}) = 0. \tag{32}$$

They introduced the following optimal score function,

$$\boldsymbol{\psi}_{cond}(\mathbf{y}, \mathbf{x}, \boldsymbol{\beta}, b, B) = W\big\{\mathbf{y} - \boldsymbol{\mu} - c(\mathbf{x}^T \boldsymbol{\beta}, \frac{b}{(\mathbf{x}^T B \mathbf{x})^{1/2}})\big\} \mathbf{x}, \tag{33}$$

where $b > 0$ is the bound which satisfy $S(\boldsymbol{\psi}) \leq b$, the function $c(.)$ is a bias correction, B is the dispersion matrix defined by

$$B(\boldsymbol{\beta}) = \mathbf{E}\{\boldsymbol{\psi}_{cond}\boldsymbol{\psi}_{cond}^T\}\,,$$

and $(\mathbf{x}^T B\mathbf{x})^{1/2}$ is a leverage measure. The weight function, W, down-weight atypical observations and is given by

$$W = W_b(\mathbf{y}, \mathbf{x}, \boldsymbol{\beta}, b, B) = W_b(r(\mathbf{y}, \mathbf{x}, \boldsymbol{\beta}, b, B)(\mathbf{x}^T B\mathbf{x})^{1/2})\,,$$

with $r(\mathbf{y}, \mathbf{x}, \boldsymbol{\beta}, b, B) = \mathbf{y} - \boldsymbol{\mu} - c(\mathbf{x}^T\boldsymbol{\beta}, \frac{b}{(\mathbf{x}^T B\mathbf{x})^{1/2}})$, $W_b(a) = H_b(a)/a$ and $H_b(a)$ is the Huber function (10). For a more detailed description of how to implement these estimators, in particular how to estimate the matrix B, see Künsch *et al.* (1989).

The development of robust models for the GLM continued with the work of Morgenthaler (1992). He introduced the L_q quasi-likelihood estimates, which replace the L_2-norm in the definition of quasi-likelihood by an arbitrary L_q-norm for $q \geq 1$. This was intended as a robust estimator similar to the LAD estimator in classical regression. We will study this estimator in detail in Chapter 3.

More recently, Cantoni and Ronchetti (2001) defined a general class of Mallows quasi-likelihood estimator. This is in general an M-estimator in which the influence of outliers in x and y direction is bounded separately. They considered the M-estimator of the form

$$\sum_{i=1}^{n}\left[\psi_c(r_i)W(\mathbf{x}_i)h'(\eta_i)V(\mu_i)^{-1/2} - a(\boldsymbol{\beta})\right] = \mathbf{0}\,, \tag{34}$$

where $r_i = (y_i - \mu_i)V(\mu_i)^{-1/2}$ are the Pearson residuals and

$$a(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}\mathbf{E}[\psi_c(r_i)]W(\mathbf{x}_i)h'(\eta_i)V(\mu_i)^{-1/2}$$

ensures the Fisher consistency of the estimator. $\psi_c(.)$ and $W(.)$ are suggested as simple functions to obtain a robust estimator. Cantoni and Ronchetti proposed $\psi_c(.)$ as a Huber function, (10), and $W(\mathbf{x}_i) = \sqrt{1-h_i}$ where h_i is the i^{th} diagonal element of hat matrix $H = X(X^TX)^{-1}X^T$.

Another choice for $W(\mathbf{x}_i)$ is the inverse of Mahalanobis distance. Note that if $W(\mathbf{x}_i) = 1$ for all i, we obtain the classical quasi-likelihood model. In particular, these estimates are applied in binomial and Poisson models.
In this thesis we focused on robust methods for logistic regression. We therefore consider the model where the response variables have a binomial distribution with link function logit. In the next section of this chapter, we review the robust estimates suggested in literature for logistic models.

2.4 Logistic regression and robust logistic regression

2.4.1 Binary regression

Suppose that the response variables y_i have binomial distributions $y_i \sim Bin(n_i, \mu_i)$ and the nonrandom explanatory variables $\mathbf{x}_i \in \mathbb{R}^P$. Then, the Bernoulli GLM is given by

$$\mathbf{E}(Y_i/n_i) = \mu_i = h(\eta_i)\,,$$

with $\eta_i = \mathbf{x}_i^T\boldsymbol{\beta}$. The link function $g(\mu)$ can be chosen logit, loglog, probit or cloglog. The link function, which is most commonly used is logit. It is given by

$$g(\mu_i) = \log(\mu_i/(1-\mu_i))\,.$$

In this research we focus on logit as link function and study the logistic regression in detail.
Consider the simple case where $n_i = 1$, then $y_1\,, \cdots\,, y_n$ are variables which take the value 1 with probability μ_i or the value 0 with probability $1-\mu_i$. The log-likelihood function is given by

$$l(\boldsymbol{\beta}) = \log L(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left[y_i \log \mu_i + (1-y_i)\log(1-\mu_i)\right].$$

Differentiation yields the maximum likelihood equations:

$$\sum_{i=1}^{n} \frac{y_i - \mu_i}{V(\mu_i)} h'(\eta_i)\mathbf{x}_i = \mathbf{0}\,, \tag{35}$$

where $V(\mu_i) = \mu_i(1-\mu_i)$. In the case of logit link function $h'(\eta_i) = \frac{\partial h(\eta_i)}{\partial \eta_i} = \mu_i(1-\mu_i) = V(\mu_i)$, therefore (35) simplifies to

$$\sum_{i=1}^{n} (y_i - \mu_i)\mathbf{x}_i = \mathbf{0}\,. \tag{36}$$

In the next section we give an overview of robust estimators of the logistic regression parameters $\boldsymbol{\beta}$.

2.4.2 Robust estimates for logistic regression

Development of robust models for logistic regression started with the work of Pregibon (1981), who introduced the diagnostic measures to help the analyst detect outliers and leverage points and quantifying their effect on various aspects of the maximum likelihood fit. Pregibon (1982) also suggested a robustifying modification to the MLE. Instead of minimizing the deviance (28), he proposed to use a function of the deviance to minimize, which is less sensitive to outliers:

$$\rho(D(\mathbf{y}, \hat{\boldsymbol{\beta}})) = \min_{\boldsymbol{\beta}}! \tag{37}$$

where ρ is a monotone and non-decreasing function. Differentiation with respect to $\boldsymbol{\beta}$ results in a weighted version of the maximum likelihood score equations,

$$\sum_{i=1}^{n} w_i(y_i - \mu_i)\mathbf{x}_i = \mathbf{0}\,, \tag{38}$$

where $w_i = \frac{\partial}{\partial D_i}\rho(D_i)$. Using Huber's loss function,

$$\rho(D) = \begin{cases} D & D \le k \\ 2(Dk)^{1/2} - k & D > k\,, \end{cases}$$

with a constant k, he obtained

$$w_i = \begin{cases} 1 & D_i \leq k \\ (k/D_i)^{1/2} & D_i > k\,. \end{cases}$$

Since this weight function only downweights the outliers in the sense of observations with large deviance and is still sensitive to leverage points, he also included a weight function $\pi(\mathbf{x}_i)$ in equation (38).
The deviance tapering method of Pregibon can be implemented for any generalized linear model. Pregibon's estimators are asymptotically biasd for logistic models.
Copas (1988) studied the estimate of Pregibon and showed that they exhibit comparatively strong bias for small samples.
Carroll and Pederson (1993) proposed another version of Pregibon's estimator, which for small samples is less biased than the estimator of Pregibon (1982) or Copas (1988). They suggested a simple weighted MLE with bounded influence by downweighting leverage observations. They defined the measure of leverage of an observation $\mathbf{x}$ by

$$u_n(\mathbf{x}) = \left(\big(X - \widehat{\boldsymbol{\mu}}_n\big)^T \widehat{\boldsymbol{\Sigma}}_n \big(X - \widehat{\boldsymbol{\mu}}_n\big) \right)^{\frac{1}{2}}. \tag{39}$$

They considered robust location vectors $\widehat{\boldsymbol{\mu}}_n$ and a robust dispersion matrix $\widehat{\boldsymbol{\Sigma}}_n$ of the design matrix $X = (\mathbf{x}_1, \cdots, \mathbf{x}_n)$ instead of the mean and the dispersion matrix of X and proposed the M-estimator equation

$$\sum_{i=1}^{n} W_i (y_i - \mu_i)\mathbf{x}_i = \mathbf{0}\,, \tag{40}$$

where $W_i = W(u_n(\mathbf{x}_i))$ with non-decreasing function such as $W(a) = (1 - \frac{a^2}{c^2})^3 I(\mid a \mid \leq c)$.
Studying the non-consistency of Pregibon's estimator continued with the work of Bianco and Yohai (1996). They proposed a Fisher-consistent version of Pregibon's estimator. These estimators are defined by

$$\rho(D(\mathbf{y}, \hat{\boldsymbol{\beta}})) + G(\boldsymbol{\mu}) + G(1 - \boldsymbol{\mu}) = \min_{\boldsymbol{\beta}}! \tag{41}$$

where ρ is a non-decreasing and bounded function, $\mu = h(\eta_i)$ and

$$G(u) = \int_0^u \rho'(-\log t)dt\,. \tag{42}$$

In particular, they studied a family of ρ functions given by

$$\rho_c(t) = \begin{cases} t - t^2/2c & t \le c \\ c/2 & t > c\,, \end{cases} \tag{43}$$

with tuning parameter c. This function smoothly truncates the identity function (corresponding to the MLE) for values larger than a constant c. The correction term $t^2/2c$ makes the function smooth.
Any other bounded ρ function, which makes the estimate qualitatively robust in Hampel's sense, can be used. We will denote these estimates with the notation BY.
Croux and Haesbroeck (2003) found by numerical experiments that, when they used the ρ function (43), it frequently occurred that the BY estimator "explodes" to infinity and does not exist (Croux *et al.* (2002)). Therefore they found a sufficient condition on ρ to guarantee a finite minimum of (41). Assuming that there is overlap in the sample, they proposed the function $\psi(t) = \rho'(t)$ as,

$$\boldsymbol{\psi}_d^{CH}(t) = \exp(-\sqrt{\max(t,d)}). \tag{44}$$

The constant d can be chosen to achieve both robustness and efficiency together.
Let $G(\boldsymbol{\mu}) + G(1-\boldsymbol{\mu}) = q(\boldsymbol{\mu})$, then

$$q'(\boldsymbol{\mu}) = \big(\psi^{CH}(-\log \boldsymbol{\mu}) - \psi^{CH}(-\log(1-\boldsymbol{\mu}))\big)\boldsymbol{\mu}(1-\boldsymbol{\mu}). \tag{45}$$

Differentiating (41) with respect to $\boldsymbol{\beta}$ yields the M-estimator

$$\sum_{i=1}^{n} \psi^{CH}(D(\mathbf{y}, \hat{\boldsymbol{\beta}}))(y_i - \mu_i)\mathbf{x}_i - q'(\mu_i) = \mathbf{0}. \tag{46}$$

This equation can also be written as

$$\sum_{i=1}^{n} \psi^{CH} = \sum_{i=1}^{n} \left[\psi^{CH}(D(\mathbf{y}, \hat{\boldsymbol{\beta}}))(y_i - \mu_i) - \mathbf{E}(\psi^{CH}(D(\mathbf{y}, \hat{\boldsymbol{\beta}}))(y_i - \mu_i))\right]\mathbf{x}_i = \mathbf{0}. \tag{47}$$

From now on we will use the notation $\mathrm{BY_{CH}}$ for the Bianco and Yohai estimator with the ψ^{CH} function defined by Croux and Haesbroeck. Croux and Haesbroeck showed that bad leverage points have a very small influence on these estimates. However, the observation in x space in the neighborhood of the hyperplane orthogonal to parameters, $\mathbf{x}_i^T \boldsymbol{\beta} = 0$, which separate the observation $y = 1$ and $y = 0$, can still have a large influence on the $\mathrm{BY_{CH}}$ estimates. Therefore, they suggested the weighted BY_{CH} estimator with

$$\sum_{i=1}^{n} w_i \boldsymbol{\psi}^{CH} = \mathbf{0}\,, \tag{48}$$

where $\boldsymbol{\psi}^{CH_{\mathrm{BY}}}$ is the same as (47) and $w_i = W(\mathrm{rmd}_i)$ for rmd the robust Mahalanobis distance and the function

$$W(t) = \begin{cases} 1 & t^2 \leq \chi^2_{p,0.975} \\ 0 & \text{else}\,. \end{cases} \tag{49}$$

As we have seen so far, the robust approaches for logistic models aim at minimizing some function of deviance. Markatou *et al.* (1997) proposed another approach based on the idea of replacing the maximum likelihood score function with a weighted score equations to achieve efficient estimates with good breakdown properties. The weights are a function of residuals and are selected to downweight an observation that is not consistent with the assumed model. However, this method normally does not downweight a cluster of observations.
Suppose $y_i \sim Bin(n_i, \mu_i)$ for $i = 1, \cdots, n$ with repeated covariate $\mathbf{x}_i$.

Let $d_i = \frac{y_i}{n_i}$, $\mu_i = (1 + e^{-\mathbf{x}_i^T \boldsymbol{\beta}})^{-1}$ and $\hat{\mu}$ an estimate of μ. They defined a new residual as

$$r_1(\mathbf{x}_i) = \frac{d_i}{\hat{\mu}_i} - 1 \qquad \text{for} \quad y = 1\,,$$

$$r_0(\mathbf{x}_i) = \frac{1 - d_i}{1 - \hat{\mu}_i} - 1 \qquad \text{for} \quad y = 0.$$

Markatou *et al.* (1997) proposed the estimate obtained by minimizing

$$\sum_{i=1}^{n} n_i[\mu_i G(r_1(\mathbf{x}_i)) + (1 - \mu_i) G(r_0(\mathbf{x}_i))] = \min_{\boldsymbol{\beta}}! \tag{50}$$

for a given convex function G. Let $A(t) = G'(t)$, then the solution of (50) satisfies

$$\sum_{i=1}^{n} n_i[A(r_1(\mathbf{x}_i)) - A(r_0(\mathbf{x}_i))]\mu_i(1 - \mu_i) = \mathbf{0}. \tag{51}$$

Or

$$\sum_{i=1}^{n} n_i[w_1(\mathbf{x}_i) y_i(1 - \mu_i) - w_0(\mathbf{x}_i)(n_i - \mu_i)\mu_i]\mathbf{x}_i = \mathbf{0}\,, \tag{52}$$

where $w_1(\mathbf{x}_i) = \frac{A(r_1)+1}{r_1+1}$ and $w_0(\mathbf{x}_i) = \frac{A(r_0)+1}{r_0+1}$.
Markatou *et al.* (1997) used the residual adjustment function RAF (Lindsay (1994)), which is given by

$$A(d) = 2[(d+1)^{\frac{1}{2}} - 1]\,.$$

They also used the negative exponential RAF (Lindsay (1994)) as

$$A_{\mathrm{NE}}(d) = 2 - (2 + d)e^{-d}.$$

These models of Markatou *et al.* are not applicable to binary data, $y_i \sim Bin(1, \mu_i)$.
Ruckstuhl and Welsh (2001) continued in the same direction as Markatou, Basu and Lindsay. They proposed a new class of estimators called the E-estimator. These estimates depend on tuning constants

$0 \leq c_1 \leq 1$ and $1 \leq c_2 \leq \infty$. Suppose $f_\mu(k) = \mathrm{P}(y = k)$ is the probability function of $y_i \sim Bin(n_i, \mu_i)$ and $p_n(k) = \frac{1}{n}\sum_{i=1}^{n} I(Y_i = k)$, $k = 0, \cdots, m$, the proportion of observations equal to k in the sample size n. The E-estimator $\hat{\boldsymbol{\mu}}$ of μ is obtained by minimizing

$$\sum_{k=0}^{m} \rho\big(\frac{p_n(k)}{f_\mu(k)}\big) f_\mu(k) = \min_{\boldsymbol{\beta}}, \tag{53}$$

with

$$\rho(x) = \begin{cases} (\log(c_1)+1)x - c_1 & x < c_1 \\ x\log(x) & c_1 < x \leq c_2 \\ (\log(c_2)+1)x - c_2 & x > c_2\,. \end{cases} \tag{54}$$

Equation (53) can be written as the solution of

$$\sum_{k=0}^{m} w\big(\frac{p_n(k)}{f_\mu(k)}\big) S_\mu(k) p_n(k) = \mathbf{0}\,, \tag{55}$$

where $S_\mu(k) = \frac{k - n_i\mu_i}{\mu_i(1-\mu_i)}$ and

$$w(x) = \rho'(x) - \rho(x)/x = I(c_1 \leq x \leq c_2) + \frac{c_1}{x} I(x < c_1) + \frac{c_2}{x} I(x > c_2).$$

This equation could be used for a large class of estimators with a suitable weight function. Note that Markatou *et al.* used w as a function of $(\frac{p_n(k)}{f_\mu(k)} - 1)$ instead of $(\frac{p_n(k)}{f_\mu(k)})$.
Using the negative exponential RAF in Markatou *et al.* is similar to the E-estimator. However, the E-estimator is less smooth and decreases rapidly for large x.

Since our research is based on the L_q quasi-likelihood estimator proposed by Morgenthaler (1992), we study these estimators in detail. The next chapter covers the L_q quasi-likelihood estimators and their robustness behavior.

CHAPTER 3

L_q Quasi-Likelihood Estimators

3.1 Definition

Let $(\mathbf{x}_i \in \mathbb{R}^p, y_i \in \mathbb{R})$, for $i = 1, \cdots, n$, be a pair of explanatory and response variables. A generalized linear model relates the expected value of the response to the linear prediction such as $g[E(y_i)] = g(\mu_i) = \mathbf{x}_i^T \boldsymbol{\beta}$. The response variable Y given X has the probability distribution $G(y, \boldsymbol{\beta}|\mathbf{x})$. Since we assume that the components of Y are independent, the covariance matrix, $V(\mu)$ must be diagonal, namely that $V(\mu) = diag(V(\mu_1), \cdots V(\mu_n))$. Quasi-likelihoods are used in fitting the model when we have only the information for the expectation of response variables, μ, and the variance function, $V(\mu)$, as a function of μ. Consider the quasi-likelihood of each observation as

$$\frac{\partial K(y_i, \mu_i)}{\partial \mu_i} = \frac{y_i - \mu_i}{V(\mu_i)} \,. \tag{56}$$

The quasi-likelihood estimating equations for the unknown parameters $\boldsymbol{\beta}$ are obtained by differentiating $K(y_i, \mu_i)$ with respect to $\boldsymbol{\beta}$, which yields

$$U(\boldsymbol{\beta}) = D^T V(\mu)^{-1}(\mathbf{y} - \boldsymbol{\mu}) = \mathbf{0}, \tag{57}$$

where D is the matrix of partial derivatives $D_{ij} = \partial\mu_i/\partial\beta_j = h'(\eta_i)x_{ij}$ with $h'(\eta_i) = \partial\mu_i/\partial\eta_i$. In the classical regression model, the replacement of the L_2-norm by least absolute-deviations, the L_1-norm, is resistant against outliers. Morgenthaler (1992) replaced the L_2-norm in the definition of quasi-likelihood (56) by the L_q-norm for $q \geq 1$ to obtain similarly robust estimators. He proposed the L_q quasi-likelihood estimators, which are determined by

$$\sum_{i=1}^{n} \left| \frac{y_i - \mu_i}{V(\mu_i)^{1/2}} \right|^q = \min_{\boldsymbol{\beta}}! \tag{58}$$

The gradient of (58) is proportional to

$$V(\mu)^{-q/2}\{|mathbfy - \boldsymbol{\mu}|^{q-1}\mathrm{sgn}(\mathbf{y} - \boldsymbol{\mu})\}\,, \tag{59}$$

where sgn(.) denotes the sign-function that takes values ± 1. With $q = 2$, this formula yields the usual quasi-likelihood. The estimating equation for $\boldsymbol{\beta}$ is

$$U_q(\boldsymbol{\beta}) = D^T V(\mu)^{-q/2}\{|y - \mu|^{q-1}\mathrm{sgn}(y - \mu) - c(\mu)\} = \mathbf{0}\,, \tag{60}$$

where

$$c(\mu_i) = \mathop{\mathbf{E}}_{Y|X}\{|Y_i - \mu_i|^{q-1}\mathrm{sgn}(Y_i - \mu_i)\} \tag{61}$$

is the bias correction to provide conditionally Fisher consistency and D and $V(\mu)$ are defined as above.

The computation of $c(\mu)$ is not always easy and it is not possible to give a closed form expression. Therefore, often one needs to compute $c(\mu)$ numerically or by Monte Carlo simulation. Suppose the observations y_i has the distribution $f_{\mu_i}(y)$ $(i = 1, \cdots, n)$. For $(j = 1, \cdots, m)$, we generate the sample z_{ij} from the distribution $f_{\mu_i}(y)$ and estimate $\widehat{c}(\mu_i)$

is given by

$$\widehat{c}(\mu_i) = \frac{1}{m}\sum_{j=1}^{m} |z_{ij} - \mu_i|^{q-1}\mathrm{sgn}(z_{ij} - \mu_i). \tag{62}$$

Let consider the response variables have the exponential distribution, one can find,

$$c(\mu) = \begin{cases} 0 & q = 2 \\ 1 - 2\mathrm{P}(y \le \mu) & q = 1 \\ a\sqrt{\mu} & q = 1.5 \\ b\mu^2 & q = 3\,, \end{cases} \tag{63}$$

for the constant a and b. Then we proposed a general form $c(\mu) \simeq A\mu^{q-1}$ when the response variables are from exponential distribution. To define the covariate A, we can estimate $\widehat{c}(\mu)$ from (62) and fit a regression model between $\hat{c}_i$ and μ_i^{q-1}.
For the binomial model, $y_i \sim Bin(n_i, \mu_i)$, $c(\mu)$ is equal to

$$\begin{aligned} c(\mu_i) = -\sum_{y_i=0}^{\lfloor n_i\mu_i \rfloor} (n_i\mu_i - y_i)^{q-1}\mathrm{P}(Y_i = y_i) \\ + \sum_{y_i=\lceil n_i\mu_i \rceil}^{n_i} (y_i - n_i\mu_i)^{q-1}\mathrm{P}(Y_i = y_i)\,. \end{aligned} \tag{64}$$

Therefore, there are considerable difficulties for computing $c(\mu)$ in L_q quasi-likelihood estimators.

3.1.1 Computation and an example

Computation of M-estimators is usually done iteratively. We have developed an algorithm to compute the L_q quasi-likelihood estimator, which is illustrated by an example in this section.

Morgenthaler (1992) considered the quasi-likelihood estimating equation in the form of

$$U_q(\hat{\boldsymbol{\beta}}) = \sum_{i=1}^{n} \frac{h'(\eta_i)}{V(\mu_i)^{q/2}} \big\{ |y_i - \mu_i|^{q-1} \mathrm{sgn}(y_i - \mu_i) - c(\mu_i) \big\} \mathbf{x}_i = \mathbf{0} \,. \quad (65)$$

Or

$$U_q(\hat{\boldsymbol{\beta}}) = \sum_{i=1}^{n} w_i \mathbf{x}_i (y_i - \hat{\mu}_i) = \mathbf{0} \,, \quad (66)$$

where $w_i = W(y_i, \hat{\mu}_i, q)$ is the weight function,

$$W(y_i, \mu_i, q) = \frac{h'(\eta_i) \big\{ |y_i - \mu_i|^{q-1} \mathrm{sgn}(y_i - \mu_i) - c(\mu_i) \big\}}{V(\mu_i)^{q/2} (y_i - \mu_i)} \,. \quad (67)$$

The Newton-Raphson method leads to the updated $\boldsymbol{\beta}$

$$\hat{\boldsymbol{\beta}}_{new} = \hat{\boldsymbol{\beta}}_{old} - (U'(\hat{\boldsymbol{\beta}}_{old}))^{-1} U(\hat{\boldsymbol{\beta}}_{old}) \,, \quad (68)$$

where U' is a matrix with elements $U_{ij} = \partial U_i / \partial \beta_j$ as

$$\frac{\partial U_i}{\partial \beta_j} = \sum_{i=1}^{n} w_i' h'(\eta_i) \mathbf{x}_i \mathbf{x}_j (y_i - \mu_i) - w_i h'(\eta_i) \mathbf{x}_i \mathbf{x}_j \,,$$

where $w_i' = \partial w_i / \partial \mu_i$. This leads to

$$\begin{aligned} \hat{\boldsymbol{\beta}}_{new} &= \hat{\boldsymbol{\beta}}_{old} + (X^T N X)^{-1} U(\hat{\boldsymbol{\beta}}_{old}) \\ &= (X^T N X)^{-1} (X^T N) \{ X \hat{\boldsymbol{\beta}}_{old} + N^{-1} W (y - \mu) \} \,, \end{aligned} \quad (69)$$

where W and N are diagonal matrices. The former has weights w_i on the diagonal, the later with diagonal elements

$$N_{ii} = h'(\eta_i) \{ w_i - w_i'(\mu_i)(y_i - \mu_i) \} \qquad (i = 1, \cdots, n) \,.$$

Example 3. *McCullagh and Nelder (1983) discuss an example involving clotting times of plasma induced by two different lots of an agent. The response was measured for nine different dilutions x. A gamma regression model with the inverse link was fitted. The estimators are computed by using maximum likelihood, L_1 quasi-likelihood and L_2 quasi-likelihood. Table 3 contains a summary of the results.*

	MLE and $q = 2$
Lot1	$\mu_1^{-1} = -0.0166(0.003) + 0.0153(0.0018)\log(x)$
Lot2	$\mu_2^{-1} = -0.0239(0.003) + 0.0236(0.0018)\log(x)$
	$q = 1$
Lot1	$\mu_1^{-1} = -0.0176(0.00137) + 0.0157(0.0065)\log(x)$
Lot2	$\mu_2^{-1} = -0.0240(0.023) + 0.0235(0.01)\log(x)$

Table 1: *Estimated regression parameters for L_1 and maximum likelihood which is equal to L_2. The values in parenthesis are the asymptotic standard deviations corresponding to the parameters. The link used is the inverse clotting time and the two lines correspond to two different agents. There is little difference between the estimates in this example.*

3.1.2 Asymptotic covariance matrix

The asymptotic covariance matrix of the L_q quasi-likelihood estimator, $\hat{\boldsymbol{\beta}}$, is considerably more complicated than the analog formula for the MLE, namely

$$\mathrm{Cov}(\hat{\boldsymbol{\beta}}) = (D^T V^{-q/2} Q D)^{-1} (D^T V^{-q/2} R V^{-q/2} D)(D^T V^{-q/2} Q D)^{-1}. \quad (70)$$

Here, Q denotes the diagonal matrix with diagonal elements

$$Q_{ii} = (q-1)\mathbf{E}(|Y_i - \mu_i|)^{q-2} - \mathbf{E}\{|Y_i - \mu_i|^{q-1} \partial \mathrm{sgn}(Y_i - \mu_i)/\partial \mu_i\} + c'(\mu_i).$$

The second term of this formula is zero except for $q = 1$, in which case it is $-2f_Y(\mu_i)$[1]. R also is diagonal with elements

$$R_{ii} = \mathbf{E}\big(|Y_i - \mu_i|^{2q-2}\big) - c(\mu_i)^2. \quad (71)$$

[1]For more details see 79.

3.2 Influence function of L_q quasi-likelihood estimators

We are interested in investigating the robustness properties of L_q quasi-likelihood estimators. Therefore, we measure the asymptotic bias caused by infinitesimal contamination, the influence function. This measure has been explained in details in section (2.2.2).

Proposition 3.2.1. *Let T_q be the root of equation (60), that is, the L_q quasi-likelihood estimator of a generalized linear model. Its influence function at the model F is equal to*

$$IF_q(\mathbf{x}_*, y_*) = M(\psi, F)^{-1}\psi(\mathbf{x}_*, y_*, \mu_*) \in \mathbb{R}^p\,,$$

where

$$\psi(\mathbf{x}_*, y_*, \mu_*) = \frac{h'(\eta_*)}{V(\mu_*)^{q/2}}\Big\{|y_* - \mu_*|^{q-1} sgn(y_* - \mu_*) - c(\mu_*)\Big\}\,\mathbf{x}_*\,, \quad (72)$$

where $M(\psi, F) = -E_F\big[\frac{\partial}{\partial\boldsymbol{\beta}}\psi(x, y, \mu)\big]$, y_* is an added element to the response variable and $\mathbf{x}_*$ is an added vector $(1, x_{*1}, ..., x_{*p})$ to the design matrix.

Proof. The influence function is a direct consequence of the general formula of IF for the M-estimator (20) (Hampel *et al.* (1986)) by noting that $\eta_* = \mathbf{x}_*^T\boldsymbol{\beta}$, $\mu_* = h(\eta_*)$, $h'(\eta_*) = \partial\mu_*/\partial\eta_*$. □

Moreover, these estimators are asymptotically normal with asymptotic covariance matrix given by (19)

$$V(T, F) = M(\psi, F)^{-1}Q(\psi, F)M(\psi, F)^{-1}\,,$$

where $Q(\psi, F) = E_F\big[\psi(x, y, \mu)\psi(x, y, \mu)^T\big]$.

3.2.1 Influence function when $q = 2$

Assuming $q = 2$, one finds that $c(\mu) = \mathbf{E}\{|y - \mu|\mathrm{sgn}(y - \mu)\} = \mathbf{E}(y - \mu) = 0$ and the M-estimator is calculated by the equation

$$\psi(\mathbf{x}_*, y_*, \mu_*) = \frac{h'(\eta_*)}{V(\mu_*)}(y_* - \mu_*)\,\mathbf{x}_*\,. \quad (73)$$

And

$$\frac{\partial \psi_*}{\partial \beta_j} = V(\mu_*)^{-1}\{h''(\eta_*)(y_* - \mu_*)\mathbf{x}_* - h'(\eta_*)\mathbf{x}_*\mathbf{x}_j h'(\eta_i)\}\,,$$

where $h''(\eta_*) = \partial^2\mu_*/\partial\eta_*^2$. Therefore, the matrix $M(\psi, F)$ can be easily computed as

$$M(\psi, F) = \mathbf{E}_F(-\frac{\partial \psi}{\partial \boldsymbol{\beta}}) = \mathbf{E}_F\big(\frac{h'(\eta)^2}{V(\mu)}\ XX^T\big)\,, \tag{74}$$

which can be estimated by

$$\widehat{M}(\psi, F) = D^T V(\mu)^{-1} D\,, \tag{75}$$

where D and $V(\mu)$ are defined as below. It follows that

$$\mathrm{IF}_2(\mathbf{x}_*, y_*) = (D^T V(\mu)^{-1} D)^{-1}\Big\{\frac{h'(\eta_*)}{V(\mu_*)}\ (y_* - \mu_*)\Big\}\,\mathbf{x}_*\,, \tag{76}$$

for $\eta_* = \mathbf{x}_*^T\boldsymbol{\beta}$ and $\mu_* = h(\eta_*)$.

3.2.2 Influence function when $q = 1$

When $q = 1$, the correction for consistency is defined by $c(\mu) = \mathbf{E}_F\big(\mathrm{sgn}(Y - \mu)\big) = 1 - 2\mathrm{P}(y \le \mu)$ and the M-estimator in (72) is calculated by the equation

$$\psi(\mathbf{x}_*, y_*, \mu_*) = \frac{h'(\eta_*)}{\sqrt{V(\mu_*)}}\Big(\mathrm{sgn}(y_* - \mu_*) - c(\mu_*)\Big)\,\mathbf{x}_*\,. \tag{77}$$

The negative derivative of the estimating equation at the true parameter value has expectation[2]

$$\begin{aligned} M(\psi, F) = -\mathbf{E}_F(\frac{\partial \psi_*}{\partial \beta_j}) = \\ - E_F\bigg(h'(\eta)^2/V(\mu)^{1/2}\ \mathrm{d}\big(\mathrm{sgn}(y - \mu)\big)/\mathrm{d}\mu\ XX^T\bigg)\,. \end{aligned} \tag{78}$$

[2]For more details see McCullagh and Nelder (1983).

Let $f(y|\mathbf{x}_i)$ denote the conditional density of the response y given $\mathbf{x}_i$ and let the interval with endpoints a and b be the support of f and $g(x,y) = \mathrm{sgn}(y-\mu)$. By using integration by parts we can obtain

$$\begin{aligned}\int_a^b g'(x,y)f(y|x)dy &= g(x,y)f(y|x)\Big|_a^b - \int_a^b g(x,y)f'(y|x)dy \\ &= 2f_Y(\mu)\end{aligned}$$

Since $\mathrm{sgn}(y-\mu) = -\mathrm{sgn}(\mu-y)$ or

$$\lim_{\varepsilon\to\infty}\frac{\mathrm{sgn}(y-\mu+\varepsilon)-\mathrm{sgn}(y-\mu)}{\varepsilon} = -\lim_{\varepsilon\to\infty}\frac{\mathrm{sgn}(\mu-y-\varepsilon)-\mathrm{sgn}(\mu-y)}{\varepsilon}\,. \tag{79}$$

Then

$$\mathrm{d}(\mathrm{sgn}(y-\mu))/\mathrm{d}\mu = -\mathrm{d}(\mathrm{sgn}(y-\mu))/\mathrm{d}y. \tag{80}$$

From (80) and (79), one can then obtain

$$\begin{aligned}M(\psi,F) &= E_X\left(\frac{h'(\eta)}{\sqrt{V(\mu)}}XX^T\int_a^b\big(\mathrm{d}(\mathrm{sgn}(y-\mu))/\mathrm{d}y\big)g(y|\mu)\mathrm{d}y\right) \\ &= 2f_Y(\mu)E_X\left(\frac{h'(\eta)}{\sqrt{V(\mu)}}XX^T\right).\end{aligned}$$

This matrix can be estimated as

$$\widehat{M}(\psi,F) = D^TV^{-1/2}SD\,, \tag{81}$$

where D and V are defined in (60), and S is a diagonal matrix with diagonal elements $s_i = 2f_Y(\mu_i)$ for $i = 1,\cdots,n$.

Note that in the case of a discrete distribution, one can use summation by parts[3] to find S. For example, let $y_i|\mathbf{x}_i$ have a binomial distribution $Bin(n_i,\mu_i)$, we find that $s_i = 2\mathrm{P}(y = [n_i\mu_i]+1)$.

Furthermore, one obtains the following expression for the influence function of L_1 quasi-likelihood estimators

$$\mathrm{IF}_1(\mathbf{x}_*,y_*) = (D^TV^{-1/2}SD)^{-1}\Big\{\frac{h'(\eta_*)}{\sqrt{V(\mu_*)}}(\mathrm{sgn}(y_*-\mu_*)-c(\mu_*))\Big\}\,\mathbf{x}_*\,. \tag{82}$$

[3]See appendix.

3.2.3 Comparing the influence function for $q = 1$ and $q = 2$

Similar to the LAD method as a robust alternative in classical models, we are interested in verifying the behavior of the L_1 quasi-likelihood and comparing it to the MLE. As we can see in (76) the influence function is not bounded when $q = 2$, which means that the MLE is sensitive to outliers and leverage points. This is similar to the least-square estimators in the usual regression model. Equation (82) shows that the L_1 quasi-likelihood estimators are not influenced by outliers which is similar to the least absolute deviation (LAD) regression case. What is different between the LAD linear regression and the L_1 quasi-likelihood is the behavioral at leverage points. Depending on which link function we choose, the influence function of the L_1 quasi-likelihood can be bounded even if leverage points appear.

Theorem 3.2.1. *Suppose the variance function $V(\mu)$ is the diagonal matrix of the response variable and $h(\eta)$ is the inverse link function, where $\eta_i = \mathbf{x}_i^T\boldsymbol{\beta}$ and $h'(\eta_i) = \partial\mu_i/\partial\eta_i$. If the link function satisfies $\frac{|h'(\eta_i)|}{\sqrt{V(\mu_i)}}||\mathbf{x}_i|| < b$, then it follows that the influence function of L_q estimator with $q = 1$ remains bounded even if $||\mathbf{x}_i|| \to \infty$ for a bound $b > 0$.*

Proof. The proof is discussed in Theorem 3.2.2. □

In the case of an exponential distribution, both the inverse link function $g(\mu) = 1/\mu$ and the identity $g(\mu) = \mu$ satisfy the condition of Theorem (3.2.1). For the binomial model the links “logit", “probit", “cloglog" and “loglog" are all covered by this theorem.
In order to verify the behavior of these influence functions and in particular to compare $q = 1$ to $q = 2$, we analyze them for the exponential and the binomial model with their respective canonical links.

Example 4. *We consider a case with $p = 2$, a slope β and an intercept α. Figure (4) presents the influence function of β for an exponential model. One can see that the influence function is unbounded when $q = 2$ whereas it is bounded when $q = 1$ even for leverage points.*

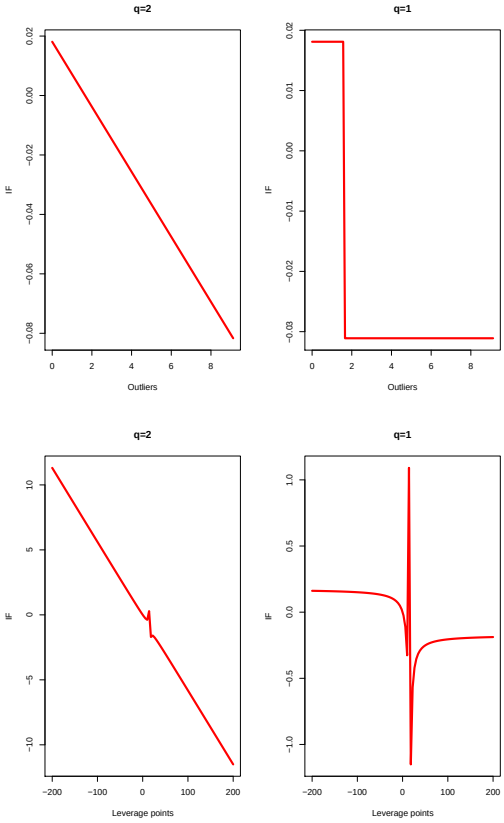

Figure 4: *The influence function of β when $q = 2$ and $q = 1$ for the exponential model.*

We also examined the binomial model. For binomial responses the outliers themselves cannot completely distort the estimators. The real danger is posed by bad leverage points. In Figure (5) the red points represent these kind of outliers. Due to the fact that the binomial model is complicated to investigate, in this thesis we focus on the logistic regression model and study the behavior of the L_q quasi-likelihood

estimator. We propose a new estimator, which is similar to the L_q quasi-likelihood estimator and results in a more resistant estimator.

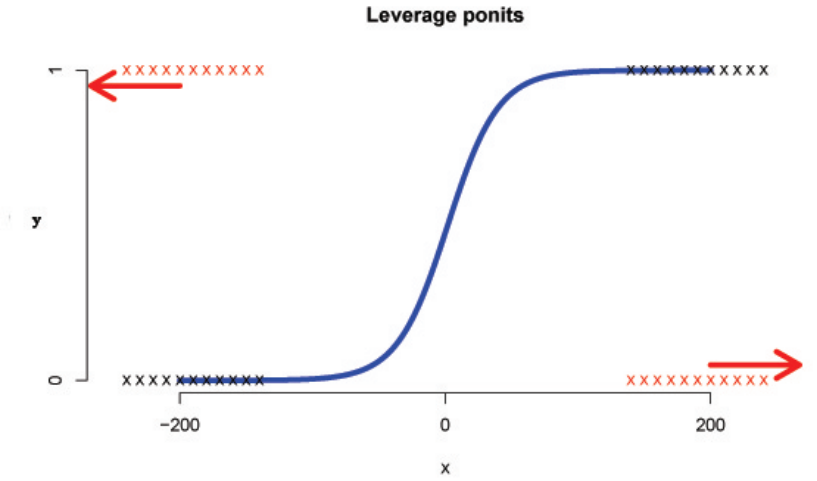

Figure 5: *Demonstration of the bad and good leverage points for the binomial model with $\beta > 0$.*

Example 5. *We consider the binomial models with $p = 2$, a slope β and an intercept α. In Figure (6) we compare the L_2 and the L_1 estimators under the influence of an additional observation $\mathbf{x}_i$. It shows that the bad leverage points can greatly influence β when $q = 2$. However the influence of these observations on the L_1 estimator converges to zero. We can say that these estimators are not sensitive to outliers when $q = 1$, and depending on the link function they can be insensitive to leverage points.*

In (2.4.2), we have seen that Croux and Haesbroeck (2003) proposed a weighted version of the BY estimator (Bianco and Yohai (1996)) to

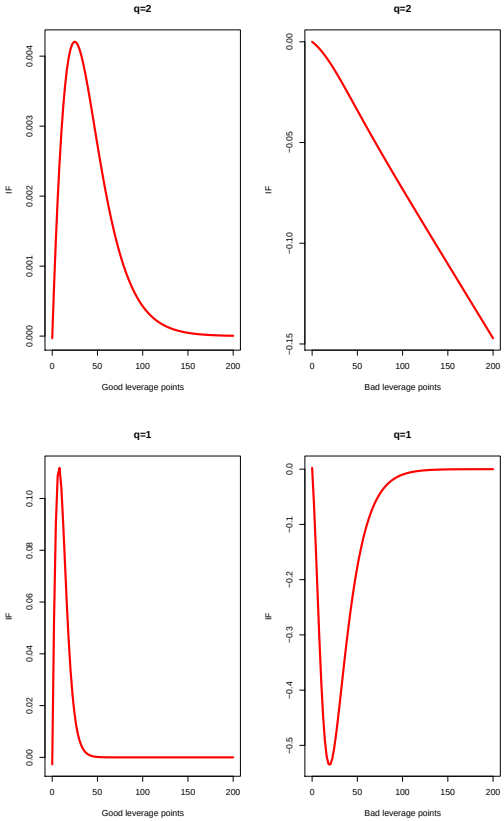

Figure 6: *The influence function of $\beta > 0$ when $q = 2$ and $q = 1$ for the binomial model.*

achieve the bounded influence function for multi-dimension explanatory variables. We study L_q quasi-likelihood estimators for the multi-dimensional explanatory variables as well. In Figure (7) we examine the influence function of L_1 quasi-likelihood for the logistic regression with two independent normally distributed covariates with β_1 and β_2 positive, $\eta_i = \beta_0 + \beta_1 x_i 1 + \beta_2 x_i 2$, and fixing $y = 1$. One can see that when the covariates both tend to ∞ and even $-\infty$ (bad leverage points)

the influence becomes zero. There is a very small influence when one of covariates tends to ∞ and the other to $-\infty$. In some cases like the BY estimator, this small influence could tend to infinity. The reason is explained in the following theorem.

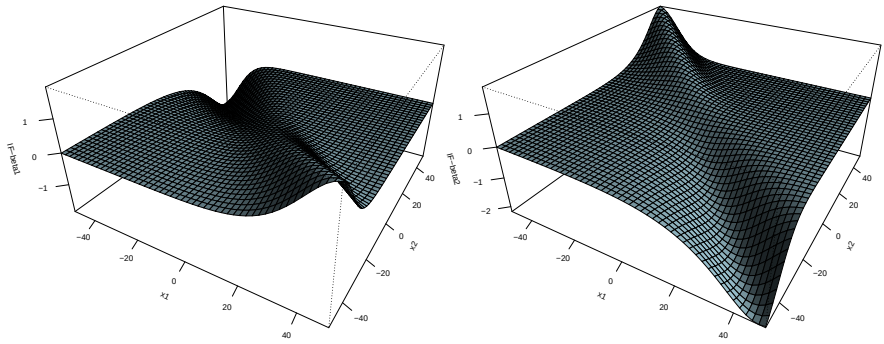

Figure 7: *The influence function of $\beta_1 > 0$ and $\beta_2 > 0$ when $q = 1$.*

Theorem 3.2.2. *The influence function of the L_q estimator with $q = 1$ is bounded for all values of $(\mathbf{x}_*, y_*)$ with the exception of the limits of the influence function along sequences for which the Euclidean norm $\|\mathbf{x}_*\| \to \infty$ and $\beta_2\, x_2^* + \cdots + \beta_p x_p^* = 0$.*

Proof. The boundedness of the influence function follows from the fact that as $\|\mathbf{x}_*\|$ tends to infinity, the corresponding $\eta_* = {\mathbf{x}_*}^T\boldsymbol{\beta}$ tends with very few exceptions to $\pm\infty$. The exceptional set is $\{\mathbf{x}_* : \mathbf{x}_*^T\boldsymbol{\beta} \to$ constant, as $\|\mathbf{x}_*\| \to \infty\}$. Since $\mathbf{x}_*^T\boldsymbol{\beta} = \beta_1 + \beta_2\, x_2^* + \cdots + \beta_p x_p^*$, the condition can only be satisfied if the constant in question is β_1 and the vector $[x_2^*, \ldots, x_p^*]$ is orthogonal to $[\beta_2, \ldots, \beta_p]$. In this case, $\frac{|h'(\eta_i)|}{\sqrt{V(\mu_i)}}$ is a constant.

□

3.3 Bias and Breakdown point

We next examine the breakdown point of L_q quasi-likelihood estimators. We illustrate the asymptotic bias of L_q quasi-likelihood caused by the contamination. Suppose the design matrix and binomial response are

$$X = \begin{pmatrix} 1 & x \\ 1 & 0 \\ 1 & 1 \end{pmatrix} \quad \begin{pmatrix} y_i & (n_i - y_i) \\ 1 & 0 \\ 10 & 10 \\ 19 & 1 \end{pmatrix}$$

We then change the top left element of matrix X, x, from -1 to 10 to create a bad leverage point. We contaminate $3\%\,, 15\%\,, 20\%\,, 33\%$ and 50% of the data and calculate the bias of $\hat{\boldsymbol{\beta}}$ estimated by L_q quasi-likelihood for $q = 2$ and $q = 1$. One hopes that for the more contaminated data the L_1 quasi-likelihood estimator remains more stable than the MLE. But as the Figure 8 shows, the L_1 quasi-likelihood estimator is biased even by 3% contamination.

Croux *et al.* (2002) proved that the maximum likelihood estimator (L_2 quasi-likelihood estimators) for the logistic regression model does not explode to infinity but rather implodes to zero when adding some outliers to the data set.
We have seen that the influence function of the L_1 quasi-likelihood estimators is bounded when $\mathbf{x}_i^T \tilde{\boldsymbol{\beta}} \neq 0$. Therefore, it would be expected that the L_1 quasi-likelihood estimator remains stable when adding some outliers in the data set. Instead, we find that the L_1 quasi-likelihood could be worse than L_2. As Figure 9 shows, L_2 tends slowly to zero but L_1 could drop very suddenly to zero. In this example we consider the same data as above, except the binomial response is modified. We study how the sensitivity curves (2.2.3) behaves if we change even a

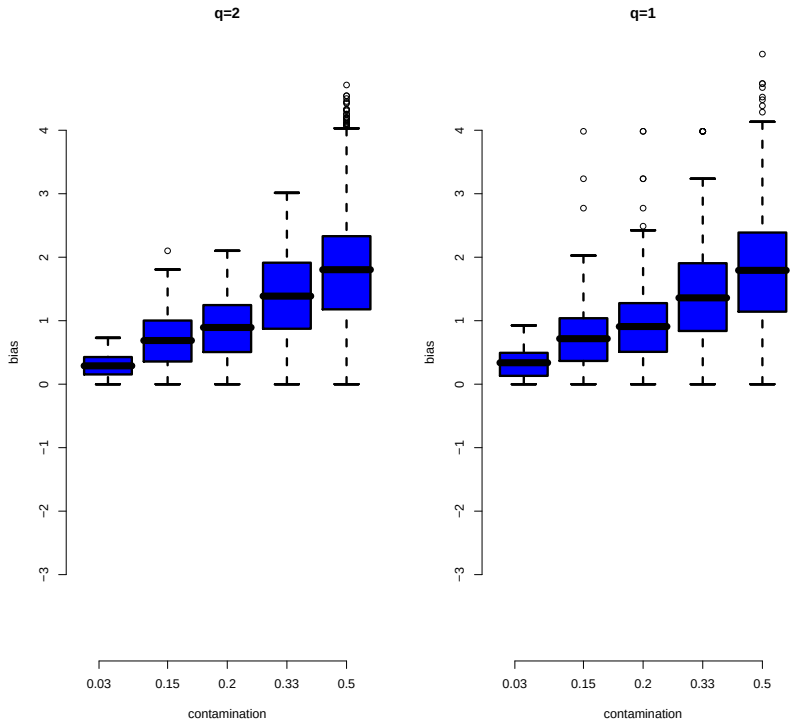

Figure 8: *Bias distribution of L_q quasi-likelihood estimators with bad leverage contaminations when $q = 1$ and $q = 2$.*

single observation among $n_i = 4000$ points.

$$\begin{pmatrix} y_i & (n_i - y_i) \\ 1 & 0 \\ 2000 & 2000 \\ 3800 & 200 \end{pmatrix}. \tag{83}$$

For binomial models, the sensitivity curve shows that the L_1 estimator could be worse than L_2 after contaminating even a bad leverage point.

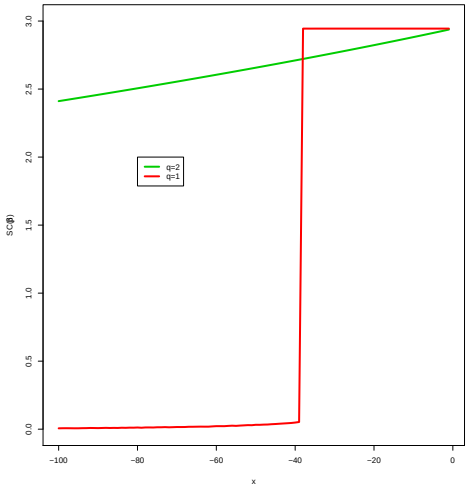

Figure 9: *The sensitivity curve of L_q quasi-likelihood with $q = 1$ and $q = 2$.*

Therefore we are interested in studying mathematically the general formula of these estimators for $q = 1$ and $q = 2$.

Consider the n binomial response variables $y_i \sim Bin(n_i, \mu_i)$ and the design matrix X with the link function logit $g(\mu_i) = \eta_i$.
Suppose for $i \geq 2$ the vector of explanatory variables, $\mathbf{x}_i$, are the cluster of observations close to each other with $\mu^{(2)}$. We then consider the vector $\mathbf{x}_1$ which has $y = 1$ far from the mass of data as a contaminated point with $\mu^{(1)}$. We want to investigate how the L_q quasi estimator reacts when $\|\mathbf{x}_1\| \to \infty$.
Therefore, one can simplify $\mathbf{x}_2 \simeq \mathbf{x}_3 \simeq \cdots \simeq \mathbf{x}_n = \mathbf{z}$ with $\mu_2 \simeq \cdots \simeq \mu_n \simeq \mu^{(2)} = (1 + e^{-\mathbf{z}^T\boldsymbol{\beta}})^{-1}$ and $\mu^{(1)} = (1 + e^{-\mathbf{x}_1^T\boldsymbol{\beta}})^{-1}$ for the contaminated observation.
Under this assumption, when $q = 2$, the system of equation (73) can

be written as

$$\sum_{i=2}^{n}(y_i - n_i\mu^{(2)})\mathbf{z} = (1-\mu^{(1)})\mathbf{x}_1. \tag{84}$$

Therefore,

$$(1-\mu^{(1)}) = \frac{B}{\|\mathbf{x}_1\|} = C_1(\mathbf{x}_1), \tag{85}$$

where $B = \sum_{i=2}^{n}(y_i - n_i\mu^{(2)})\mathbf{z}$. Consequently

$$\hat{\boldsymbol{\beta}} \simeq \frac{1}{\|\mathbf{x}_1\|}\big(\log(\frac{1-C_1(\mathbf{x}_1)}{C_1(\mathbf{x}_1)})\big). \tag{86}$$

When $\|\mathbf{x}_1\| \to \infty$, then $C_1(\mathbf{x}_1) \to 0$ and the value of $\log(\frac{1-C_1(\mathbf{x}_1)}{C_1(\mathbf{x}_1)})$ diverges to infinity. However, the divergence rate of $\|\mathbf{x}_1\|$ is higher. Therefore, this can explain why $\hat{\boldsymbol{\beta}}$ implodes to zero when $\|\mathbf{x}_1\| \to \infty$.

For $q = 1$, we consider a binomial response variable with m success events out of n, which are close to each other. Let $\mathbf{z}$ be the explanatory variable. Therefore, $\mu^{(2)}$ for this group is estimated as $\frac{m}{n}$. Consider a contaminated point far from this group with $y = 1$ and $\mathbf{x}$. The system of equation (77) is simplified to:

$$\sqrt{V(\mu^{(2)})}\sum_{i=2}^{n}\big\{\mathrm{sgn}(y_i - n_i\mu^{(2)}) - c(\mu^{(2)})\big\}\mathbf{z} = \sqrt{V(\mu^{(1)})}(1-c(\mu^{(1)}))\mathbf{x}. \tag{87}$$

Or

$$\sqrt{\frac{m(n-m)}{n}}\big(2m-n-nc(\frac{m}{n})\big)\mathbf{z} = \sqrt{\mu^{(1)}(1-\mu^{(1)})}(1-c(\mu^{(1)}))\mathbf{x}. \tag{88}$$

Since $(1-c(\mu^{(1)}) \simeq 1$, one gets

$$\sqrt{\mu^{(1)}(1-\mu^{(1)})} = \frac{D}{\|\mathbf{x}\|} = C_2(\mathbf{x}_1), \tag{89}$$

where $D = \sqrt{\frac{m(n-m)}{n}}\big(2m-n-nc(\frac{m}{n})\big)\mathbf{z}$. Consequently

$$\mu^{(1)} = \frac{1 \pm \sqrt{1-C_2(\mathbf{x})}}{2}.$$

Since $C_2(\mathbf{x}) \to 0$ when $\|\mathbf{x}\| \to \infty$, therefore, $\mu^{(1)} \to 0$ or $\mu^{(1)} \to 1$. This shows the instability of the computation of the L_1 estimator. For both results, similar to L_2, $\boldsymbol{\beta}$ implodes to zero.

3.3.1 Computation and optimization problem

We should note that the computation of the L_q estimator can exhibit numerical instabilities. This is due to the fact that the denominator of $W(y_i, \mu_i, q)$, $(y_i - \hat{\mu}_i)$ may be close to zero. This often happens when the estimated parameters $\hat{\mu}_i$ are close to the response variables y_i. Then, $W \to \infty$ and the algorithm does not converge.
As a solution we propose a method that numerically approximates the estimator using an iteratively refining search algorithm. The algorithm works as follows.
Let $\boldsymbol{\beta}_{old}$ be the last good estimate, obtained using (69). We use $\boldsymbol{\beta}_{old}$ as the starting point of the algorithm and search for better estimates in its vicinity.

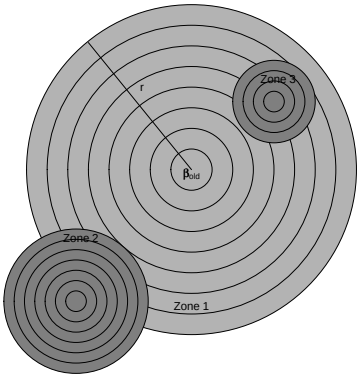

Figure 10: *The optimization of algorithm to find the L_q quasi-likelihood estimator.*

In a first step, the algorithm searches on concentric surfaces (zone 1) around $\boldsymbol{\beta}_{old}$, up to a maximum radius $r = 0.1\hat{S}(\boldsymbol{\beta}_{old})$, as shown in Figure 10. $\hat{S}(\boldsymbol{\beta}_{old})$ is the estimated standard error of $\boldsymbol{\beta}_{old}$ (70). On each concentric surface, we evaluate a fixed number of points. The algorithm then chooses as best estimate $\boldsymbol{\beta}$ the point with the smallest gradient among all evaluated points.

There are three possible outcomes of this algorithm. First, the best estimate has a gradient smaller than the target precision. In this case, the algorithm stops.

Second, the best estimate may lie on the border of the initial search zone. In this case, we take the best estimate as a new starting point and repeat the algorithm, since a better estimate may be found further outside of the initial zone (search in zone 2).

As a third possibility, the best estimate may lie inside the initial zone. In this case, we also take the best estimate as the new starting point and repeat the algorithm, but with a smaller radius to refine the estimate (search in zone 3).

The algorithm yields a final estimate $\boldsymbol{\beta}$ with a gradient smaller than an configurable tolerance.

This method has two drawbacks. First, it may not converge to an estimate which satisfies the targeted tolerance. Second, the method may be too slow, compared to simple iterative gradient-based searches such as the program glm in R.

These numerical problem and difficulties computing L_q quasi-likelihood estimators leads us to the idea of simplifying the L_q quasi-likelihood estimators. The next chapter proposes a new estimator for the binary regression, namely the weighted maximum likelihood estimator. We propose a new weight, which depends only on μ and tutoring q. These estimators are similar to L_q quasi-likelihood and results in more resistant estimator.

CHAPTER 4

Robust Binary Regression

In this chapter, we introduce a new estimator in binary regression. This estimator is based on the minimum absolute deviation estimator of Morgenthaler (1992). The resulting estimating equation in obtained through a simple modification of the familiar maximum likelihood equation. This estimator can be considered as a weighted maximum likelihood with weights that depend only on μ and q.

Let $(\mathbf{x}_i \in \mathbb{R}^p, y_i \in \mathbb{R})$, for $i = 1, \ldots, n$, be a pair of explanatory and response variables. The responses have a Bernoulli distribution $y_i \sim \text{Bernoulli}(\mu_i)$. In other words, y_i is either 0 or 1 and the probability $P(y_i = 1) = \mu_i$. The expectation and variance of Y_i are $E(Y_i) = \mu_i$ and $\text{Var}(Y_i) = \mu_i(1 - \mu_i) = V(\mu_i)$. Let g be an increasing link function, which maps the interval of probabilities $(0, 1)$ to the real numbers $\mathbb{R}$. Examples of such transformations are the logistic $g(\mu) = \ln(\mu/(1 - \mu))$, the probit $g(\mu) = \Phi^{-1}(\mu)$, and the complementary loglog $g(\mu) = \ln(-\ln(1 - \mu))$. The regression parameters

$\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$ enter into the model through

$$g(\mu_i) = \mathbf{x}_i^T \boldsymbol{\beta}\,, \tag{90}$$

where $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$ are the values of the explanatory variables for the i-th individual.
An excellent introduction to data analysis procedures based on the binary regression model is the log-likelihood, which is given by McCullagh and Nelder (1983)[1],

$$l(\beta) = \sum_{i=1}^{n} \left[y_i \ln\left(\frac{\mu_i}{1-\mu_i}\right) + \ln(1-\mu_i) \right] \tag{91}$$

and the likelihood estimating equations are

$$\sum_{i=1}^{n} \frac{h'(\mathbf{x}_i^T \beta)}{V(\mu_i)} (y_i - \mu_i)\mathbf{x}_i = 0\,, \tag{92}$$

where $h(\cdot)$ denotes the inverse link $h(\cdot) = g^{-1}(\cdot)$. The likelihood equations are particularly simple for the logistic link, because for $\eta = \text{logistic}(\mu)$, $h'(\eta) = \mu(1-\mu)$.
The likelihood equation (92) is sensitive to large values of $|y_i - \hat{\mu}_i|$, particularly if these are combined with $\mathbf{x}_i$ values near the edge of the design space. For example, a dose response, in which $\hat{P}(y_i = 1)$ increases with the dose can be inverted by adding a single observation with $y = 0$ for a sufficiently high dose. As in ordinary regression, we call such observations "bad leverage points", see Section (5).

Most of the robust estimators proposed in the literature (see 2.4.2) are based on modifications of the log-likelihood. To limit the influence of bad leverage points, one can put a weight on the contribution of an observation to the likelihood equation. For $\mathbf{x}_i$-values at the edge of the design space a lower weight is proposed. This type of estimating equations is considered in the general formula of the *weighted maximum*

[1] Chapter 4 p. 114.

likelihood estimate (WMLE) Maronna *et al.* (2006). The WMLE in the binary regression case is

$$\sum_{i=1}^{n} w_i \frac{h'(\eta_i)}{V(\mu_i)}(y_i - \mu_i)\mathbf{x}_i = \mathbf{0}\,, \tag{93}$$

where $h'(\eta_i) = \partial h_i/\partial \mu_i$ and w_i is a weight function.

Morgenthaler (1992) considered the quasi-log-likelihoods obtained by the q-th power of $\frac{Y_i-\mu_i}{V(\mu)}$, the L_q quasi-likelihood estimator (65). In binary regression all values of q except $q = 2$ lead to an asymptotically biased estimating equation, because

$$E\left(|Y_i - \mu_i|^{q-1}\mathrm{sgn}(Y_i - \mu_i)\right) = (1-\mu_i)^{q-1}\,P(Y_i = 1) - \mu_i^{q-1}\,P(Y_i = 0)$$

is nonzero. One can show that

$$\left\{|y_i - \mu_i|^{q-1}\mathrm{sgn}(y_i - \mu_i)\right\} = y_i(1-\mu_i)^{q-1} + (y_i - 1)\mu_i^{q-1}\,.$$

Therefore, the bias-corrected contribution of the i-th observation is

$$|y_i - \mu_i|^{q-1}\mathrm{sgn}(y_i - \mu_i) - \left((1-\mu_i)^{q-1}\,\mu_i - \mu_i^{q-1}\,(1-\mu_i)\right) = \left((1-\mu_i)^{q-1} + \mu_i^{q-1}\right)(y_i - \mu_i)\,.$$

This equality can again be easily checked for $y_i = 1$ and $y_i = 0$. Thus, the estimating equation for all these quasi-log-likelihoods is the same as the likelihood equation apart from the extra weight

$$w_q(\mu_i) = V(\mu_i)^{(2-q)/2}\,\left((1-\mu_i)^{q-1} + \mu_i^{q-1}\right)\,. \tag{94}$$

Therefore, the estimating equation is

$$\sum_{i=1}^{n}\,\psi(\mathbf{x}_i, y_i) = \sum_{i=1}^{n}\,\frac{h'(\eta_i)}{V(\mu_i)}\,w_q(\mu_i)\,(y_i - \mu_i)\mathbf{x}_i = \mathbf{0}\,. \tag{95}$$

However with this simple weight similar robust estimates discussed in the literature typically have weights that depend on the couple $(\mathbf{x}_i, y_i)$

rather than on $\mu_i = h(\eta_i)$ alone. Examples are weights that depend on a combination of the contribution to the deviance of the i-th observations and the Mahalanobis distance of $\mathbf{x}_i$ from the center of the design.
When the data is grouped and we observe n_i times at the same $\mathbf{x}_i$, the response y_i can be modeled as a binomial random variable with n_i trials and with probability of success μ_i, $y_i \sim Bin(n_i, \mu_i)$. In the grouped case, the estimating equation becomes

$$\sum_{i=1}^{n} \frac{h'(\eta_i)}{V(\mu_i)} w_q(\mu_i) (y_i - n_i \mu_i)\mathbf{x}_i = \mathbf{0}\,. \tag{96}$$

Interestingly, the equation (96) is not identical to the one of the L_q quasi-likelihood in the grouped case

$$\sum_{i=1}^{n} \frac{h'(\eta_i)}{V(\mu_i))^{q/2}} \left(|y_i - \mu_i|^{q-1}\mathrm{sgn}(y_i - \mu_i) - c(\mu_i)\right) \mathbf{x}_i = \mathbf{0}\,,$$

because $c(\mu_i) = E\{|y_i - \mu_i|^{q-1}\mathrm{sgn}(y_i - n_i\mu_i)\}$ does not have a simple form.

From now on we will use the notation BL_q for this new estimator (95), which is a simplified version of L_q quasi-likelihood estimator for binary regression models.
The weight function (94) is identically equal to 1 for $q = 2$, which shows the well-known fact that the L_2 quasi-likelihood is equal to the maximum likelihood procedure of Wedderburn (1974). When $0 < q < 2$, however, the weight tends to zero as μ tends to 0 or 1. Observations with a success probability close to 0 or 1 are thus given less weight. The full weight of 1 is given to those observations with success probability of 0.5. Note also that $w_{1-u}(\mu) = w_{1+u}(\mu)$ for all μ and for $0 < u < 1$. Thus values of q less than 1 are of no interest. Figure 11 shows the function $w_q(\mu)$ of the logistic model for various value of $1 \le q \le 2$.
Figure 12 presents $w_q(\mu)\frac{h'(\eta_i)}{V(\mu_i)}$, where the different value of q correspond to logit and loglog (clogclog) link function. One can see that for $q > 2$ this weight function does not downweight the leverage observations, whereas it downweights for $q < 2$.

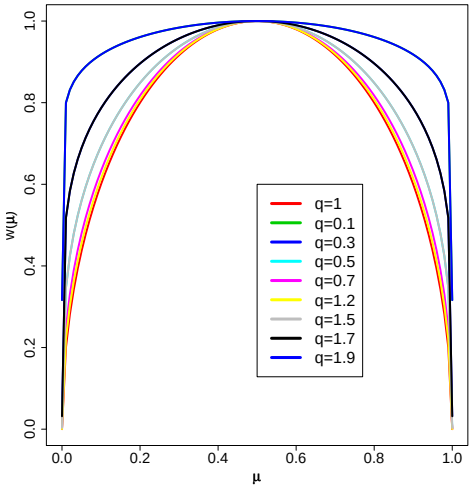

Figure 11: *The weight function $w_q(\mu)$ associated with the L_q quasi-likelihood estimator for the logistic function $1 \leq q < 2$.*

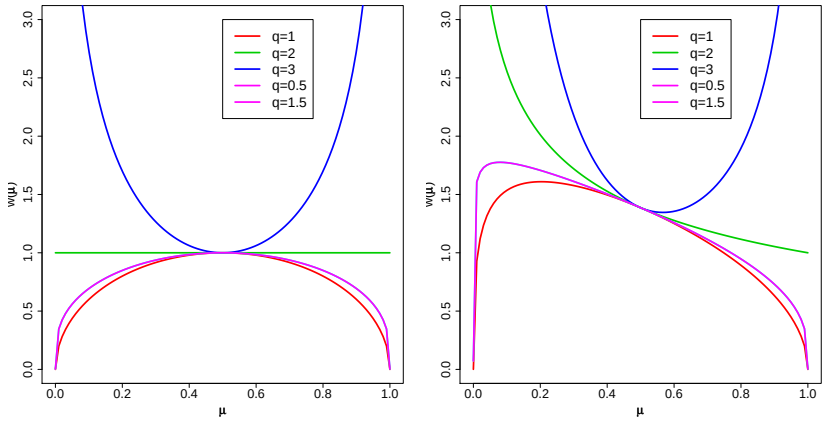

Figure 12: *The function $w_q(\mu)\frac{h'(\eta_i)}{V(\mu_i)}$ for link functions logit, loglog (cloglog).*

4.1 Asymptotic covariance matrix

Consider the linear expression (68), we can show that the approximate covariance matrix of $\boldsymbol{\beta}$ is equal to

$$\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\beta}}) = B_n^{-1}\mathrm{Cov}(U(\boldsymbol{\beta}))B_n^{-1}\,, \tag{97}$$

where $U(\boldsymbol{\beta})$ is defined by the system of equations (95),

$$U(\boldsymbol{\beta}) = \sum_{i=1}^{n} \frac{h'(\eta_i)}{V(\mu_i)} w_q(\mu_i)(y_i - \mu_i)\mathbf{x}_i = \mathbf{0} \qquad \in \mathbb{R}^p,$$

and the matrix B_n is the expectation of the negative derivative of the estimating equation at the true parameter

$$B_n = -\mathbf{E}(\partial U(\boldsymbol{\beta})/\partial\boldsymbol{\beta}) \qquad \in \mathbb{R}^{p\times p}.$$

Therefore, the asymptotic covariance matrix of $\hat{\boldsymbol{\beta}}$ is estimated by[2]

$$\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\beta}}) = (D^T V^{-q/2} Q D)^{-1}(D^T V^{-q/2} R V^{-q/2} D)(D^T V^{-q/2} Q D)^{-1}\,. \tag{98}$$

Here Q denotes the diagonal matrix with diagonal elements

$$Q_i = \mu_i^{q-1} + (1-\mu_i)^{q-1}. \tag{99}$$

R is also a diagonal matrix with diagonal elements

$$R_i = \mu_i(1-\mu_i)Q_i^2.$$

The L_q quasi-likelihood estimate for grouped data has an asymptotic variance covariance matrix, which is considerably more complicated than (98), see (70).

[2]For more details see McCullagh and Nelder (1983).

In order to illustrate our formula for the asymptotic covariance of the BL_q estimator, we conduct a simulation experiment. The estimators of interest are BL_2, BL_1, $\mathrm{BL}_{1.2}$, $\mathrm{BL}_{1.7}$. We compare the asymptotic variance of these estimates with the corresponding mean-squared error (MSE) computed by Monte Carlo simulations for the model. The simulations are carried out for $p = 3$ dimensions and a sample size of $n = 100$ or $n = 1000$. The explanatory variables $\mathbf{x}_i$ are distributed according to a normal distribution $\mathbb{N}(0, 1)$. The true parameter values, $\boldsymbol{\beta}^m = (\beta_1\,, \beta_2\,, \beta_3) \in \mathbb{R}^p$, are set to $(1, 2, 2)$. We use the link function logit $g(\mu) = \log(\mu/(1+\mu)) = \eta = \mathbf{x}^T\boldsymbol{\beta}$. $N = 1000$ samples $(y_1, \ldots, y_n)$ are generated according to $Bernoulli(\mu_i)$ with $\mu_i = h(\eta_i)$.
Let $\hat{\beta}_k$ be the estimated parameter for the k-th sample in each generation for $k = 1, \ldots, N$. The mean-squared error and the variance of these estimates are given by

$$\mathrm{MSE1} = \frac{n}{N}\sum_{k=1}^{N}\sum_{j=1}^{p}(\hat{\beta}_{kj} - \bar{\hat{\beta}}_{kj})^2 \qquad \mathrm{MSE2} = \frac{n}{N}\sum_{k=1}^{N}\sum_{j=1}^{p}(\hat{\beta}_{kj} - \beta^m)^2$$

$$\mathrm{Var1} = n \times \frac{1}{N}\sum_{k=1}^{N}\sum_{l=1}^{p}\widehat{\mathrm{Var}_l}(\hat{\beta}_k) \qquad \mathrm{Var2} = n \times \sum_{l=1}^{p}\widehat{\mathrm{Var}_l}(\beta^m).$$

Here Var1 computes the covariance matrix (98) of the estimated parameter for the k-th sample. Var2 imputes the true parameter $\boldsymbol{\beta}^m$ in (98). These two values should converge to the same value for large n.
MSE1 gives the mean-squared error of the estimated parameter for the k-th sample in each generation and the mean of these estimated parameters. MSE2 computes the mean-squared error of the estimated parameters and the true parameters. MSE1 and MSE2 should converge to the same value when n is large enough.
In general for large n, the mean-squared error converges to the variance of the parameters.
Table 2 illustrates how for large n, all four values, MSE1, MSE2, Var1 and Var2, converge.
Figure 13 presents the covariance matrix of BL_q for values $1 \leq q < 2$. We can see that it has a high efficiency for different values of q. The

n=100				
	MSE1	VAR1	MSE2	VAR2
BL_2	80.18	67.53	85.94	54.87
BL_1	100.59	80.92	109.40	61.26
$BL_{1.2}$	97.63	79.37	105.96	56.31
$BL_{1.7}$	83.50	69.96	89.48	60.68
n=1000				
	MSE1	VAR1	MSE2	VAR2
BL_2	53.23	53.40	53.67	52.44
BL_1	61.21	60.37	61.83	58.97
$BL_{1.2}$	60.45	59.70	61.04	53.85
$BL_{1.7}$	55.01	54.88	55.46	58.35

Table 2: *Comparing the asymptotic covariance and the mean squared error of the BL_q estimator for the various value of q.*

efficiency is computed as the ratio of the lowest feasible variance and the actual variance. For $q = 1$ it is equal to 0.90.

An alternative formula for the covariance

To compute the asymptotic covariance of the regression parameters, we need to have a model for the generation of the explanatory variable. In the following, we assume that $\mathbf{x}_i$ are independently drawn from a distribution H.
Let $D_{ij} = \partial\mu_i/\partial\beta_j$. One can show that

$$D^T V^{-q/2} Q D = D^T A D \tag{100}$$

$$D^T V^{-q/2} R V^{-q/2} D = D^T B D\,, \tag{101}$$

where A is the diagonal matrix with diagonal elements $A(\mathbf{x}_i) = V(\mu_i)^{-q/2} Q_i$ and B is also a diagonal matrix with diagonal elements $B(\mathbf{x}_i) = V(\mu_i)^{1-q} S_i$ with $S_i = Q_i^2$. Q_i is defined in (99). Therefore, the covariance matrix

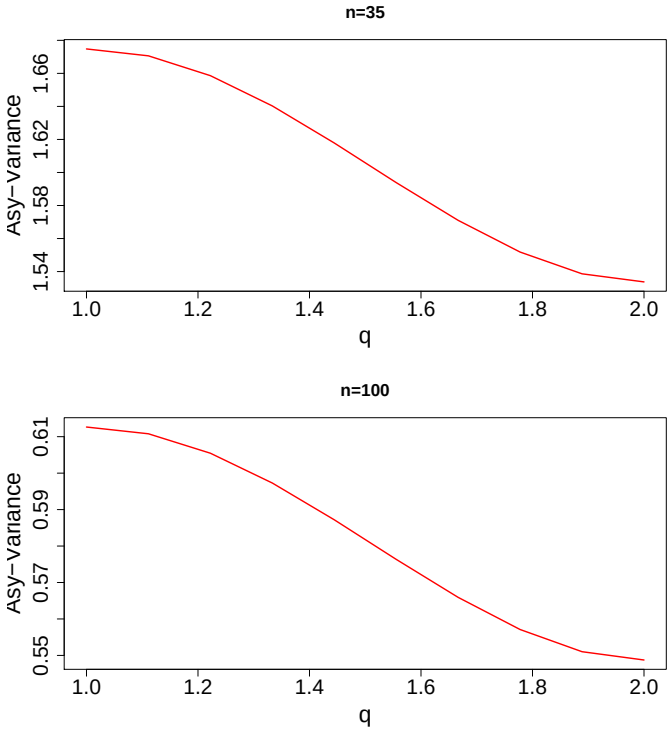

Figure 13: *The asymptotic covariance matrix of BL_q for various value of $1 \leq q < 2$.*

(98) becomes

$$\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\beta}}) = (D^T AD)^{-1}(D^T BD)(D^T AD)^{-1}\,. \tag{102}$$

This matrix depends only on the explanatory variable X. We can show that equations (100) and (101) are given by the following expressions

$$\frac{1}{n}D^T AD = \frac{1}{n}\sum_{i=1}^{n} A(\mathbf{x}_i)h'(\eta_i)\mathbf{x}_i h'(\eta_i)\mathbf{x}_i^T \tag{103}$$

$$\frac{1}{n}D^T BD = \frac{1}{n}\sum_{i=1}^{n} B(\mathbf{x}_i)h'(\eta_i)\mathbf{x}_i h'(\eta_i)\mathbf{x}_i^T. \tag{104}$$

Let $\mathbf{x}_1, \ldots, \mathbf{x_n}$ are i.i.d. realizations of a random variable $\mathbf{X}$ with a known distribution function H. Their expectation and variance are given by $\mathbf{E}(\mathbf{X}) = \boldsymbol{\mu}_x$ and $\Sigma = \mathbf{E}\big((\mathbf{X} - \boldsymbol{\mu}_x)(\mathbf{X} - \boldsymbol{\mu}_x)^T\big)$.
Since $\lim_{n\to\infty} \sum_{i=1}^{n} \mathbf{X}^T\mathbf{X} \overset{a.s}{\to} \mathbf{E}(\mathbf{X}^T\mathbf{X})$, then for large n we can obtain the following asymptotic value of (103) and (104):

$$\lim_{n\to\infty} \frac{1}{n}(D^T AD) = \int A(\mathbf{x})(h'(\eta))^2 \mathbf{x}\mathbf{x}^T dH(\mathbf{x}), \tag{105}$$

$$\lim_{n\to\infty} \frac{1}{n}(D^T BD) \to \int B(\mathbf{x})(h'(\eta))^2 \mathbf{x}\mathbf{x}^T dH(\mathbf{x}). \tag{106}$$

Therefore, the asymptotic covariance matrix (102) becomes

$$\mathrm{Cov}(\hat{\boldsymbol{\beta}})_{\mathrm{asy}} = \left(\int A(\mathbf{x})(h'(\eta))^2 \mathbf{x}\mathbf{x}^T dH(\mathbf{x})\right)^{-1}$$
$$\left(\int B(\mathbf{x})(h'(\eta))^2 \mathbf{x}\mathbf{x}^T dH(\mathbf{x})\right)\left(\int A(\mathbf{x})(h'(\eta))^2 \mathbf{x}\mathbf{x}^T dH(\mathbf{x})\right)^{-1}. \tag{107}$$

To illustrate this formula and compute the efficiency of the BL_q estimator for different value of q with respect to MLE, we conduct a simulations study. We use a basic Monte Carlo integration method to estimate (107). The simulation are carried out for $p = 4$ dimensions with $\boldsymbol{\beta} = (1, 2, 2, 2, 2)$. We generate $N = 1000$ explanatory variables $\mathbf{x}$ from the two following distribution of $H(\mathbf{x})$,

$$H_1(\mathbf{x}) = \big(\mathbf{1}, \mathbb{N}_{p-1}(\mathbf{0}, \Sigma)\big) \tag{108}$$

$$H_2(\mathbf{x}) = \big(\mathbf{1}, (1-\varepsilon)\mathbb{N}_{p-1}(\mathbf{0}, \Sigma) + \varepsilon\mathbb{N}_{p-1}(\mathbf{3}, \Sigma)\big), \tag{109}$$

with $\varepsilon = 30\%$. We choose three different Σ,

$$\Sigma_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \Sigma_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \Sigma_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/4 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Table 3 compares the efficiency of BL_q estimator for $q = 1$ and $q = 1.5$ with MLE. In this table two distributions are considered for the $\mathbf{x}$ with three different Σ.

$H_1(\mathbf{x})$		
	$q = 1$	$q = 1.5$
Σ_1	0.88	0.93
Σ_1	0.90	0.95
Σ_1	0.93	0.96
$H_2(\mathbf{x})$		
	$q = 1$	$q = 1.5$
Σ_1	0.89	0.94
Σ_2	0.89	0.94
Σ_3	0.92	0.96

Table 3: *Comparing the efficiency of the BL_q estimator for $q = 1$ and $q = 1.5$. Two distributions are considered for X with three different Σ.*

4.2 Computation

The computation of BL_q estimator as an M-estimator is done iteratively. We developed an algorithm similar to the L_q quasi-likelihood estimate to compute these estimates. We use the Newton-Raphson method to iteratively compute $\boldsymbol{\beta}$ as

$$\begin{aligned}\hat{\boldsymbol{\beta}}_{new} &= \hat{\boldsymbol{\beta}}_{old} + (X^T N X)^{-1} U(\hat{\boldsymbol{\beta}}_{old}) \\ &= (X^T N X)^{-1}(X^T N)\{X\hat{\boldsymbol{\beta}}_{old} + N^{-1}W(y-\mu)\}\,,\end{aligned}$$

where W and N are diagonal matrices. The former has weights $w_q(\mu_i)$ as given in (94) on the diagonal. The latter has diagonal elements

$$N_{ii} = h'(\eta_i)\{w_q(\mu_i) - w_q(\mu_i)'(\mu_i)(y_i - \mu_i)\} \qquad (i = 1, \ldots, n)\,.$$

For most values of p, this algorithm to estimate BL_q is stable compared to the algorithm for L_q. However, for $p > 3$ we have observed in our simulations that the algorithm occasionally, but rarely, fails to converge.

An alternative to this procedure can be based on the glm algorithm

with case weights to iteratively compute $\boldsymbol{\beta}$, but with an updated weight function at each iteration.
This is possible because our weight function, $w_q(\mu)$, only depends on $\mathbf{x}$ and not y. Furthermore, we propose an iteratively refining search algorithm for the weights. The algorithm works as follows.
We first identify the outliers in the space of the explanatory variables. The classical approach is to compute for each observation the Mahalanobis distance based on arithmetic mean and covariance matrix, which are not robust. We use the minimum covariance determinant (MCD) estimator (Rousseeuw (1985)) to replace the mean and covariance with their robust estimates. After identifying the potential leverage points, we estimate the initial parameters $\boldsymbol{\beta}_{in}$ from the reduced data obtained by deleting the potential leverage points. We use the weight $w_0 = w_q(h(\mathbf{x}^T\boldsymbol{\beta}_{in}))$ as the starting point. We update the weights w_i in iteration i using R's glm function

$$\boldsymbol{\beta}_i = coefficient(glm(y \sim X, \text{weight} = w_i, \text{family=binomial}))$$

and

$$w_i = w_q(h(\mathbf{x}^T\boldsymbol{\beta}_i)).$$

The algorithm stops when the difference between two parameters is smaller than an configurable tolerance. It yields a final estimate $\hat{\boldsymbol{\beta}}$.
In a simulation study we compared this algorithm to the algorithm based on Newton-Raphson. Both converge to the same results. In the case of $p > 3$, where the algorithm based on Newton-Raphson does not converge, the glm-based algorithm fails to converge too.
However, the glm-based algorithm may exhibit the problem that it tends to converge to the MLE, when we have the leverage points extremely far from the mass of data. Therefore, in this case, we can say that the algorithm based on Newton-Raphson is stable more and converges to the estimate.

4.2.1 The case of multiple roots

The BL_1 estimator is an M-estimator with a non-convex ρ function. This means that there may be more than one root for the system of equations $\sum_{i=1}^{n} \psi(\mathbf{x}, y, \boldsymbol{\beta}) = \mathbf{0}$.
We therefore modify the algorithm proposed above such that it can search for multiple roots of this system of equations. Among the multiple solutions, we can then use a secondary criterion to determine the best fit, or we may report several solutions to the end user.
In this algorithm we thus determine the best estimator with the smallest residual. The algorithm is organized as follows.
The idea is to start with different initial parameters to determine the different roots of the system of equations. For $p = 2$, let $\boldsymbol{\beta}_{in} = (\beta_1, \beta_2)$ be the initial parameters. They can be estimated using the maximum likelihood method. We have seen in binary regression that if β_2 changes to $-\beta_2$, then one can say that the parameters are influenced by the existence of outliers. Therefore, we suggest to consider as initial parameters of the algorithm all combinations of different signs of $\boldsymbol{\beta}_{in}$. For example one can consider $(-\beta_1, \beta_2), (\beta_1, \beta_2), (\beta_1, -\beta_2), (-\beta_1, -\beta_2)$. Hence there are $2^p = 4$ combinations as initial parameters.
We illustrate this algorithm in 4 examples.

Example 6. *Let the explanatory variables x_1 and x_2 be distributed according to a normal distribution $\mathbb{N}(0, 1)$ for $n = 35$ observations. The true parameter values are set to $(1, 2, 2)$ for $p = 3$. The logit link function $g(\mu) = log(\frac{\mu}{1-\mu}) = \eta = X^T\beta$ is used. The dependent variable y_i is generated according to $Bin(1, \mu_i)$, where $\mu_i = h(\eta_i)$. Six bad leverage points are added to the data, i.e., 6 instances of $y = 1$ and 6 vectors of $[1, -10, -10]$ are added to the design matrix. The results are summarized in Figure 14.*
Each of the lines in Figure 14 shows the discriminating hyperplane, where we have $x_2 = -\frac{\hat{\beta}_1}{\hat{\beta}_3} - \frac{\hat{\beta}_2}{\hat{\beta}_3} x_1 = 0$ with the different estimated parameters. The red line shows the maximum likelihood estimated parameters for Example 6. The blue and green lines are the two roots of the

robust estimates BL_1. These are computed with the algorithm based on multiple roots.

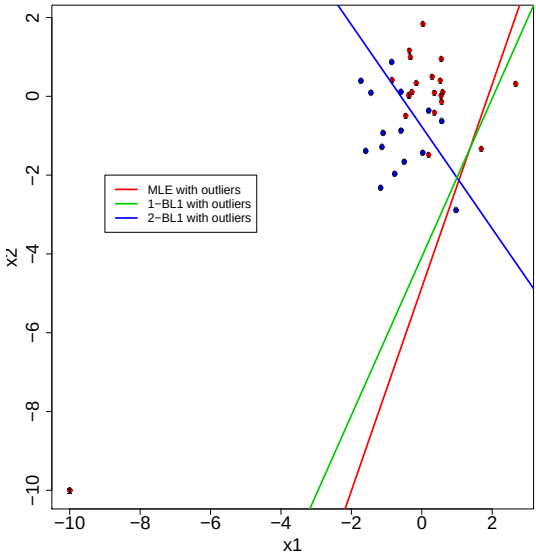

Figure 14: *The results computed for Example 6.*

Example 7. *In this example we consider the same model as in Example 6. We contaminate the data as follows:*

$$X[i,j] = (1-\varepsilon)N(\mu_1,\sigma_1) + \varepsilon N(\mu_2,\sigma_2)$$

$$y[i] = (1-\varepsilon)Bin(1,\mu[i]) + \varepsilon Bin(1,0.5).$$

Here ε indicates the percentage of contamination. We choose arbitrary but different $(\mu_1,\mu_2,\sigma_1,\sigma_2)$ with different percentages of contamination. The results are summarized in Figure 15.
The lines shows the discriminating hyperplane similar to Example 6.

The red and lines present the maximum likelihood and the BL_1 estimated parameters, which are computed for the data without outliers. The two blue lines are the results of the multiple root algorithm for $q = 1$.
In the case of 40% contaminations, the algorithm results only in one solution
Among the multiple solutions, we can then use the criterion of the smallest residual to determine the best fit or both results may be reported as the several solutions.

4.3 Influence function of the BL_q estimator

Similar to L_q quasi-likelihood, the influence function of the weighted maximum likelihood estimate BL_q can be derived directly from the general formula of IF for the M-estimator (20) (Hampel *et al.* (1986)),

$$\mathrm{IF}_q(\mathbf{x}_*, y_*) = M(\psi, F)^{-1}\psi(\mathbf{x}_*, y_*, \mu_*) \in \mathbb{R}^p\,,$$

with

$$\psi(\mathbf{x}_*, y_*, \mu_*) = \frac{h'(\eta_*)}{V(\mu_*)} V(\mu_i)^{(2-q)/2} \left((1-\mu_i)^{q-1} + \mu_i^{q-1}\right)(y_* - \mu)\,\mathbf{x}_*\,, \tag{110}$$

and $M(\psi, F) = -E_F\big[\frac{\partial}{\partial \boldsymbol{\beta}}\psi(x, Y, \mu)\big]$. Here, y_* is an added element to the response variable, $\mathbf{x}_*$ is an added vector $(1, x_{*1}, ..., x_{*p})$ to the design matrix and $\eta_* = \mathbf{x}_*^T\boldsymbol{\beta}$, $\mu_* = h(\eta_*)$, $h'(\eta_*) = \partial\mu_*/\partial\eta_*$. The matrix M can be estimated by

$$\hat{M}_q(\psi, F) = D^T V^{-q/2} Q D\,,$$

and Q is a diagonal matrix with elements

$$Q_i = \mu_i^{q-1} + (1-\mu_i)^{q-1}. \tag{111}$$

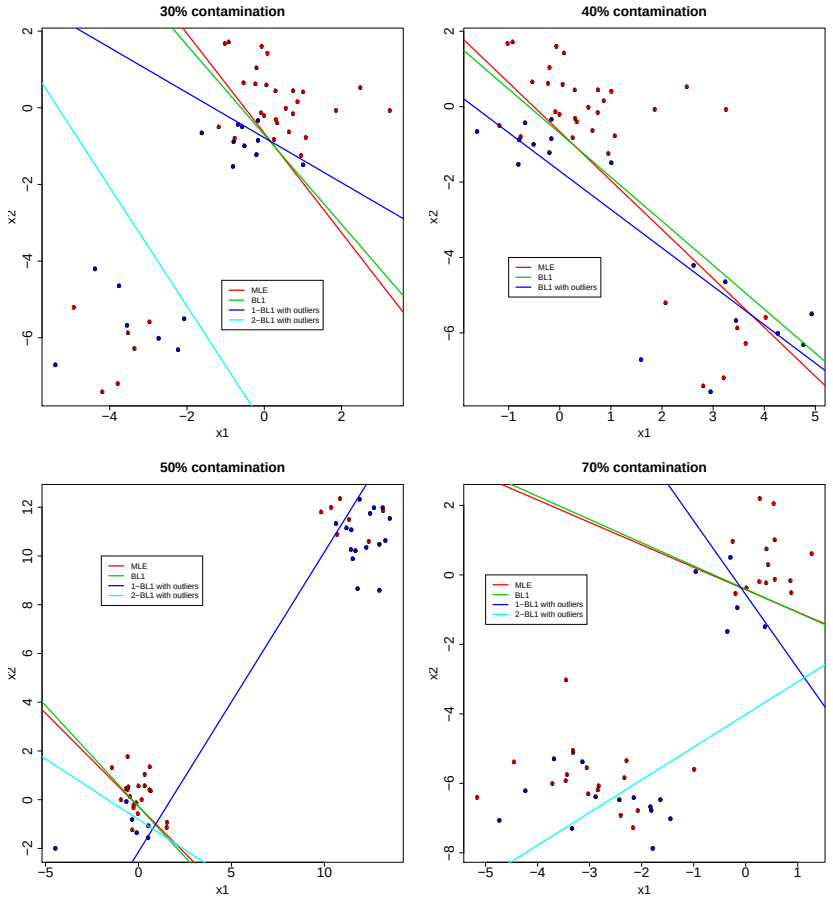

Figure 15: *The different computed results for Example 7 for different percentages of contamination.*

Theorem 3.2.1 and Theorem 3.2.2 are applicable to the BL_q estimator.

Example 8. *We consider Example 5 and study the IF of the* BL_q *estimator. Figure 16 shows that the bad leverage points could influence the estimates when* $q = 2$ *, whereas it converges to zero when* $q = 1$*.*

Figure 17 presents the influence function of the BL_q estimator for $q = 1.2$ and $q = 1.5$. This shows that for q approaching 2, the estimator is more sensitive to contaminated observations.

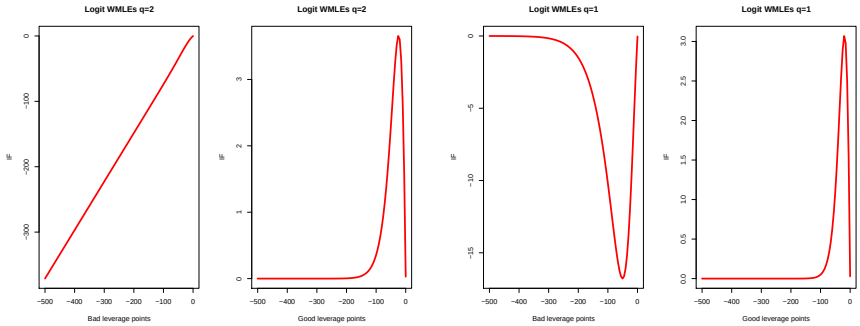

Figure 16: *The influence function of the BL_q estimator when $q = 2$ and $q = 1$ for the logistic regression.*

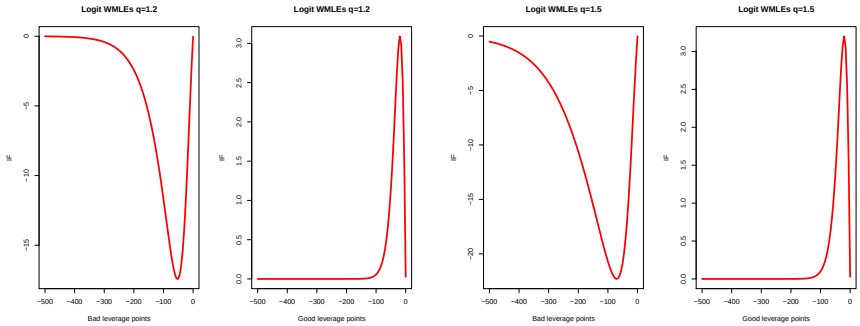

Figure 17: *The influence function of BL_q estimator when $q = 1.2$ and $q = 1.5$ for the logistic regression.*

For multi-dimensional explanatory variables, we have the same results as L_q quasi-likelihood estimate that when the covariates both tend to

∞ or $-\infty$ the IF becomes 0 (Figure 18). Moreover, there always exists the risk that was discussed in Theorem 3.2.2.

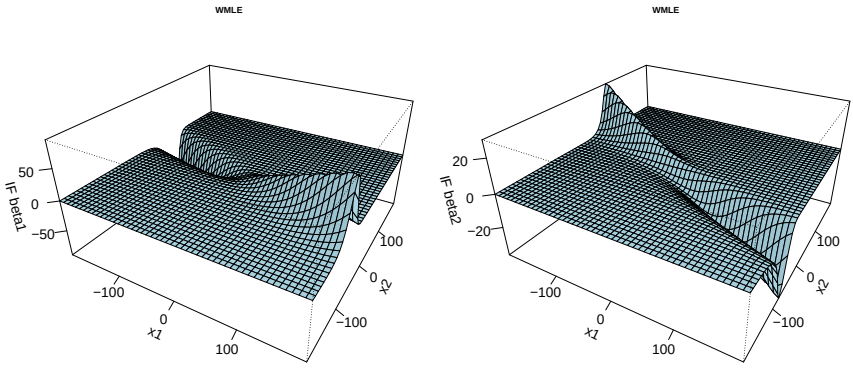

Figure 18: *The influence function of $\beta_1 > 0$ and $\beta_2 > 0$ when $q = 1$.*

4.3.1 Bias and sensitivity curve

We have seen (Figure 9) that the L_1 quasi-likelihood estimate implodes to zero (Croux *et al.* (2002)) with worse behavior than the maximum likelihood estimate. We are therefore interested in studying the sensitivity curve of the BL_q to investigate how this estimator behaves compared to the MLE. We consider a similar example as (3.3). Let

$$X = \begin{pmatrix} 1 & x \\ 1 & 2 \\ 1 & 3 \end{pmatrix} \quad \begin{pmatrix} y_i & (n_i - y_i) \\ 1 & 0 \\ 200 & 200 \\ 380 & 20 \end{pmatrix}.$$

The sensitivity curves (2.2.3) of BL_q estimates are represented in Figure 19. We move the top right element x of X from -1 to -100 and estimate the BL_1 and BL_2 for each value. We can see that the BL_1 estimate remains stable. In this example we consider a higher degree

of contamination, 1/400, compared to (3.3). This result illustrates that the BL_1 is more resistant than L_1 quasi-likelihood.

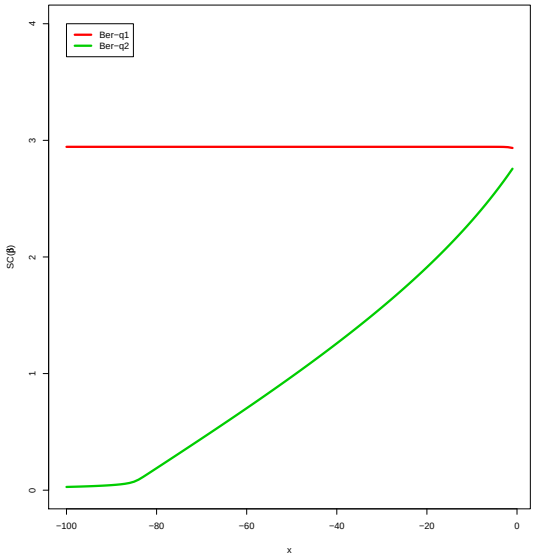

Figure 19: *The sensitivity curve of BL_q quasi-likelihood with $q = 1$ and $q = 2$.*

We next examine the breakdown point of BL_q quasi-likelihood estimates. We illustrate the asymptotic bias of BL_q quasi-likelihood caused by the contamination. Suppose the explanatory variables, $X \in \mathbb{R}^3$, are distributed according to a normal distribution $\mathbb{N}(0, 1)$. The true parameter values are set to $\boldsymbol{\beta} = (1, 2, 2)$. The logit link function $g(\mu) = \log(\frac{\mu}{(1-\mu)}) = \eta = \mathbf{x}^T\boldsymbol{\beta}$ is used and $N = 1000$ samples of y_i for $i = 1, \ldots, n$ are generated according to $Bin(1, \mu_i)$, where $\mu_i = h(\eta_i)$. We add m leverage points to the $n = 500$ original observations. These m leverage points are added as follows. In each generation we add m instances of $y = 1$ to the sample $(y_1, \ldots, y_n)$ and m vector of $[1, -5, -5]$ are added to the design matrix X. The value of m represents the per-

centages of contamination.
Let $\hat{\boldsymbol{\beta}}_k$ be the estimated parameter for the k-th sample in each generation for $k = 1, \ldots, N$. We then compute the bias of the estimator over N runs.

$$\text{Bias1} = ||\frac{1}{N}\sum_{k=1}^{N}\hat{\boldsymbol{\beta}}_k - \boldsymbol{\beta}||.$$

We contaminate $2\%\,, 3\%\,, 4\%\,, 5\%\,, 10\%\,, 20\%\,, 34\%\,, 50\%$ of the data and calculate the bias of BL_q estimates for $q = 2$ and $q = 1$.
As we expected for the more contaminated data, the BL_1 quasi-likelihood estimate remains more stable than the MLE. But as Figure 20 shows, the BL_1 estimate is heavily biased at 34% contamination, which shows that its breakdown point is smaller that 34%.
We have studied the bias in many other simulation studies and all them confirm the breakdown point greater than 20% for the BL_1 estimate. We have not been able to confirm this result theoretically.

4.4 Some alternative weight functions for BL_q

Any weight function, which depends only on μ will lead to an asymptotically unbiased estimating equation. In this section some alternatives are explored.
Let $\mathbf{x}_i \in \mathbb{R}^p$ be the explanatory variables and Y_i be the independent Bernoulli variables with success probabilities $\mu_i = \mathbf{P}(Y_i = 1 \mid \mathbf{x}_i)$ through the relation $\mu_i(\boldsymbol{\beta}) = \mu_i = 1/(1 + exp(-\mathbf{x}_i^T\boldsymbol{\beta}))$.
We again consider a Mallows estimator with a weight function that only depends on μ and q. We introduce three new weight functions to improve the robustness of these weighted maximum likelihood estimates. The first weight function has three parts and depends on $\mu\,$, q and two constants c and ε.

$$W_q^I(\mu) = \begin{cases} \mu^{q-1} + (1-\mu)^{1-q}(\mu(1-\mu))^{1-\frac{q}{2}} & |\ \mu - \frac{1}{2}\ | < c_2 \\ \frac{W_q(c+\varepsilon)}{-\varepsilon}\text{sgn}(\mu - \frac{1}{2})(\mu - u) & c_2 < |\ \mu - \frac{1}{2}\ | < c_1 \\ 0 & |\ \mu - \frac{1}{2}\ | > c_1\,, \end{cases} \tag{112}$$

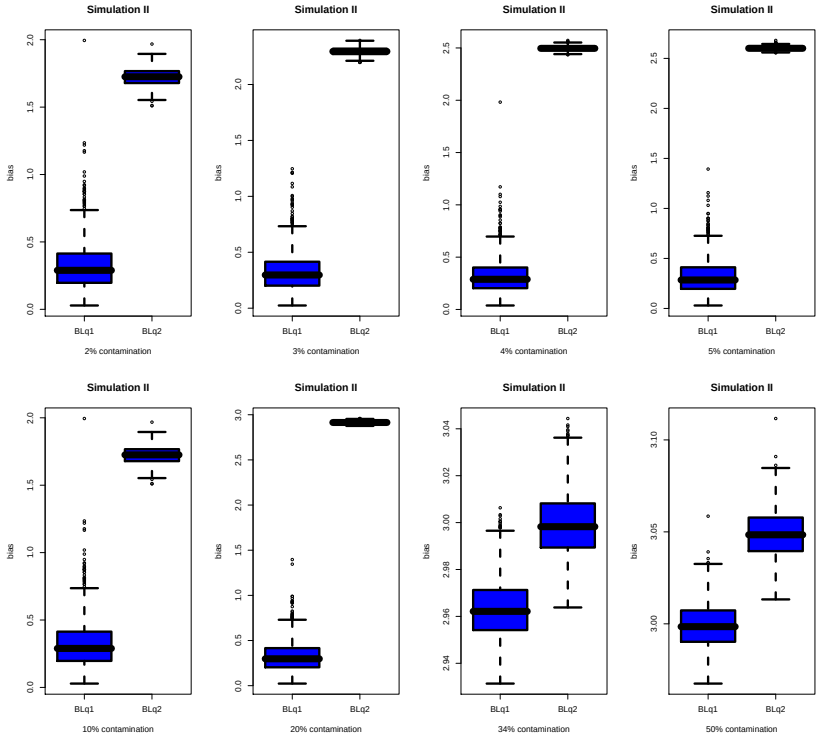

Figure 20: *Bias distribution of BL_q quasi-likelihood estimates with bad leverage contaminations when $q = 1$ and $q = 2$.*

where $c_1 = \frac{1}{2} - c$, $c_2 = \frac{1}{2} - c - \varepsilon$, sgn() denotes the sign-function that takes the values ± 1 and $u = (1-2c)I_{(\mu > \frac{1}{2})} + c$ with indicator function I.

Figure 21 presents this weight function for different values of q. Since for $q > 2$ this weight function does not downweight the leverage observations, we only consider the cases $1 \leq q \leq 2$.
We denote the estimator with this first weight function by W1-BL$_q$. This estimator yields stable estimates where the "bad leverage point"

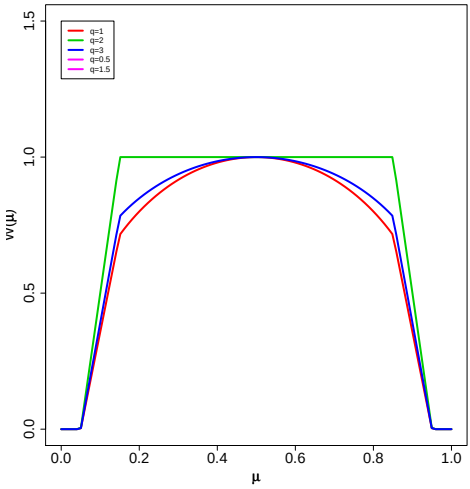

Figure 21: *The first new weight function for* $\varepsilon = 10\%\,, c = 5\%$.

is very far from the mass of data. This is when it attributes a weight of zero to data close to the edge of the design space. In the following example we illustrate this case.

Example 9. *We consider an example with* y *and* x*, which is presented in Figure 22.*

We add a bad leverage point and move this point far from the data. The results are shown in Figure 23.
The blue point is considered as a point extremely far from the design space. The W1-BL_1 *estimator approaches the estimate that is computed from the model without the bad leverage point. However, the* BL_1 *estimator is moderately influenced by this point (the blue lines in Figure 23). The other leverage points give the same result as the blue one in both models.*

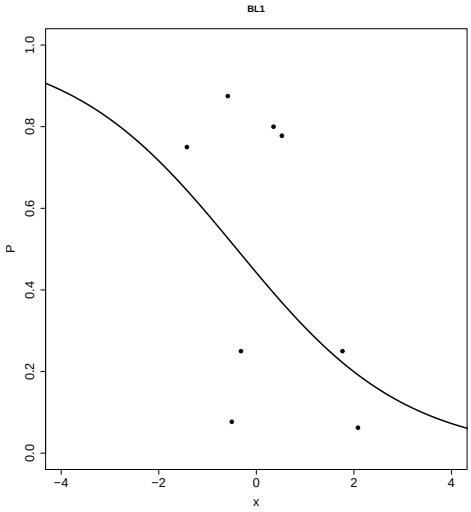

Figure 22: *The data from Example 9. The model is fitted with the W1-BL_1 estimator.*

This difference is because of a weight zero for data close to the edge of design space. These estimates have the disadvantage of numerical instability. In simulation studies, we have observed that the Newton-Raphson algorithm cannot converge when we have far a way contaminated data. This is due to the zero values in the matrix of the weight function.

Because of numerical instability we introduce a second weight function (113), which has a simpler form with a smooth function. This weight function depends on μ and q.

$$W_q^I(\mu) = (1 - (2\mu - 1)^2)^q. \tag{113}$$

Figure 24 presents this weight function for different values of q. To improve the efficiency and robustness of the weighted maximum likelihood with the weight function $W_q^{II}(\mu)$, the best value of q can be

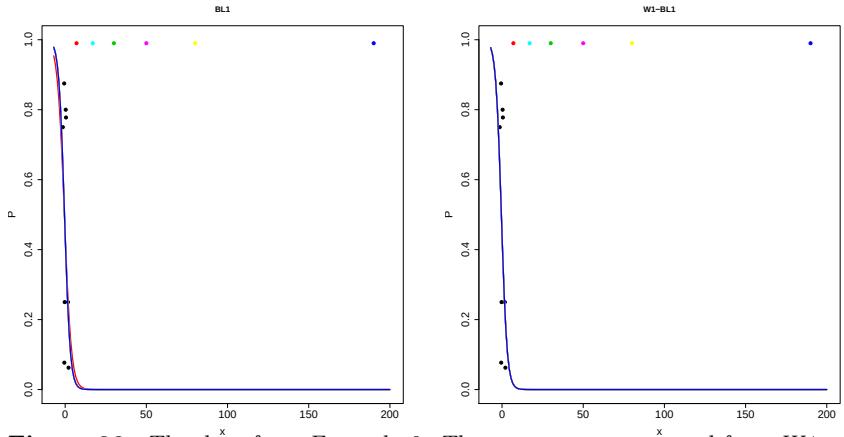

Figure 23: *The data from Example 9. The parameters computed from W1-BL_1 and BL_1 are compared.*

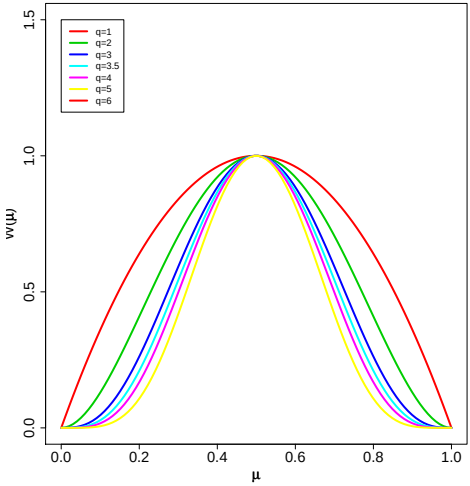

Figure 24: *The second new weight function for different value of q.*

$q = 2$ or $q = 3$. Unfortunately, similar to the first weight function, this weight function also exhibits numerical instability.
Thus we introduce the third weight function, which combines the $W_q^I(\mu)$ and $W_q^{II}(\mu)$. This weight function depends on μ, q and a constant c.

$$W_q^{III}(\mu) = \begin{cases} \mu^{q-1} + (1-\mu)^{1-q}(\mu(1-\mu))^{1-\frac{q}{2}} & \mid \mu - \frac{1}{2} \mid < c \\ (1-(2\mu-1)^2)^q & \mid \mu - \frac{1}{2} \mid > c\,. \end{cases} \tag{114}$$

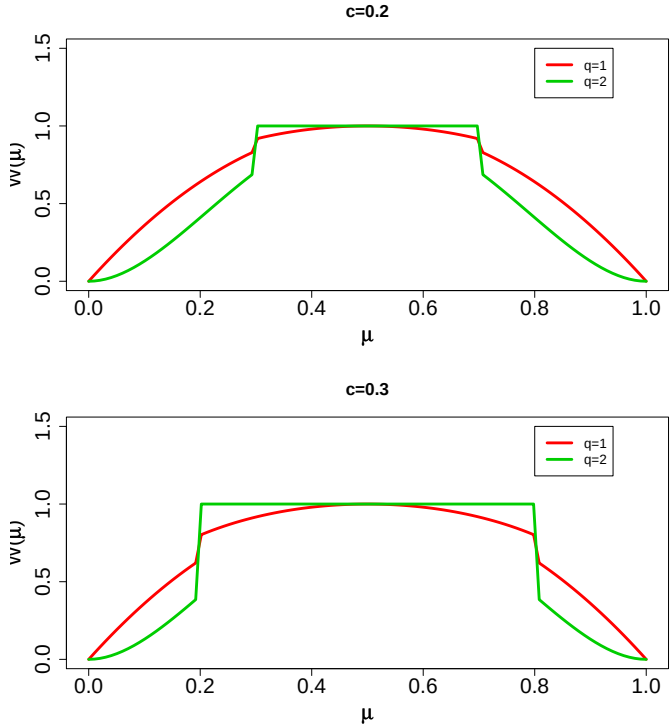

Figure 25: *The third new weight function for different values of q, $c = 0.2$ and $c = 0.3$.*

Figure 25 shows this weight function, for different values of q, with $c = 0.2$ and $c = 0.3$. For $c \geq 0.7$ this weight function is equal to the

weight function of the BL_q. In order to improve the robustness and the efficiency of this estimator, we choose the value of $c = 0.2$ or $c = 0.3$ and $1 \leq q \leq 2$.
The numerical stability of the weighted function maximum likelihood with the weight function $W_q^{III}(\mu)$ is better that the WMLE with the first and second weight function. However the BL_q estimate with the weight function (94) is still superior.

4.5 $\sqrt{n}$-consistency of the BL_q estimator

The asymptotic behavior of M-estimator has been extensively discussed especially following the important paper by Huber (Huber (1967) and also Huber (1981)). We are interested in studying under which conditions there exists a $\sqrt{n}$-consistent estimator solution for (95).
Let the design matrix be $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p) \in \mathbb{R}^{n \times p}$, where $\mathbf{x}_1$ is the vector $(1, \ldots, 1)$. Let Y_i be the independent Bernoulli variables with success probabilities $\mu_i = \mathbf{P}(Y_i = y_i \mid \mathbf{x}_i)$, where μ_i satisfies $g(\mu_i) = \mathbf{x}_i^T \boldsymbol{\beta}$. The BL_q estimator is the solution of the estimating equations

$$\sum_{i=1}^{n} \boldsymbol{\psi}(\mathbf{x}_i, y_i, \boldsymbol{\beta}) = \sum_{i=1}^{n} \frac{h'(\eta_i)}{V(\mu_i)} w_q(\mu_i)(y_i - \mu_i)\mathbf{x}_i = \mathbf{0} \qquad \in \mathbb{R}^p\,, \quad (115)$$

where $w_q(\mu_i) = V(\mu_i)^{\frac{2-q}{2}} \left(\mu_i^{q-1} + (1-\mu_i)^{q-1}\right)$.
This solution is defined as the point of global minimum of

$$\sum_{i=1}^{n} \boldsymbol{\rho}(\mathbf{x}_i, y_i, \boldsymbol{\beta}) = \min_{\boldsymbol{\beta}}\,, \quad (116)$$

where $\boldsymbol{\rho}(\mathbf{x}, y, \boldsymbol{\beta})$ is a function that satisfies $\frac{\partial}{\partial \boldsymbol{\beta}} \boldsymbol{\rho}(\mathbf{x}, y, \boldsymbol{\beta}) = \boldsymbol{\psi}(\mathbf{x}, y, \boldsymbol{\beta})$.
In order to study the asymptotic properties of the BL_q estimator we need the first- and second-order derivatives of $\boldsymbol{\psi}$ with respect to the parameter $\boldsymbol{\beta}$.
Denote the components of the vector-valued function $\boldsymbol{\psi}$ by ψ_j for $j =$

$1, \ldots, p$, $\mathbf{x}$ and $\boldsymbol{\beta}$,

$$\psi_j = \psi_j(\mathbf{x}, y, \boldsymbol{\beta}) = \frac{h'(\eta)}{V(\mu)} w_q(\mu)(y - \mu)x_j. \tag{117}$$

Here x_j is the corresponding component of $\mathbf{x}$ and $V(\mu)$ is the variance function applied to $h(\eta)$.
The first-order derivative of $\boldsymbol{\psi}$ is a matrix $\in \mathbb{R}^{p\times p}$ $\dot{\boldsymbol{\psi}} = \left[\dot{\psi}_{jr}\right]_{jr}$ for r and $j = 1, 2, \ldots, p$, where

$$\begin{aligned} \dot{\psi}_{jr} &= \frac{\partial \psi_j}{\partial \beta_r} \\ &= -\frac{(h'(\eta))^2}{V(\mu)} w_q(\mu) x_j x_r + \frac{\partial}{\partial \beta_r}\Big[\frac{h'(\eta)}{V(\mu)} w_q(\mu)\Big](y - \mu)x_j. \end{aligned} \tag{118}$$

The second term is zero when taking the expected value at the model.
The second-order derivative of $\boldsymbol{\psi}$ is a tensor $\in \mathbb{R}^{p\times p\times p}$, $\ddot{\boldsymbol{\psi}} = \left[\ddot{\psi}_{jrk}\right]_{jrk}$ for $j, r, k = 1, 2, \ldots, p$, where $\ddot{\psi}_{jrk} \in \mathbb{R}$ is equal to

$$\frac{\partial^2 \psi_j}{\partial \beta_r \partial \beta_k} = \frac{\partial}{\partial \beta_k}[\dot{\psi}_{jr}]. \tag{119}$$

To simplify the computation we ignore the terms involving the second and third derivatives of the inverse link function, $h''(\eta)$ and $h'''(\eta)$, in the elements $\ddot{\psi}_{jrk}$. We then obtain[3]

$$\ddot{\psi}_{jrk} = \frac{(h'(\eta))^3}{V(\mu)}\big[w_q''(\mu)(y - \mu) - 2w_q'(\mu)\big]x_j x_r x_k. \tag{120}$$

Huber and the authors following him distinguished two cases. In *case A.* the M-estimator $\boldsymbol{\beta}^n$ is the solution of $\sum_{i=1}^n \boldsymbol{\rho}(\mathbf{x}_i, y_i, \boldsymbol{\beta}) = \min_{\boldsymbol{\beta}}!$. In *case B.* it is determined as the root of the system of equations $\sum_{i=1}^n \boldsymbol{\psi}(\mathbf{x}_i, y_i, \boldsymbol{\beta}) = \mathbf{0} \in \mathbb{R}^p$. The asymptotic behavior of the M-estimator for the two cases are described in Huber (1981) (p.130).
If the function $\boldsymbol{\rho}$ is not convex, then there may exist several roots for

[3]For more details see McCullagh and Nelder (1983).

the system of equations (115). In this case, Jurecková and Sen (1996) imposed certain conditions on $\boldsymbol{\rho}$ and the distribution function $F(\mathbf{x}, y|\boldsymbol{\beta})$ of $(\mathbf{X}, Y)$, in order to obtain sufficient conditions for the existence of a $\sqrt{n}$-consistent estimator. In the following section we review the theorem of Jurecková and Sen (1996) (p. 184-191).

4.5.1 Applying the Theorem of Jureckovà and Sen to the BL_q estimator

Considering the explanatory variables, we assume that $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are i.i.d. realizations of a random variable $\mathbf{X}$ with a known distribution function H. The joint distribution of $(\mathbf{X}, Y)$ is then equal to

$$F_{(X;Y)}(\mathbf{x}, y|\boldsymbol{\beta}) = H(\mathbf{x})G(y|\mathbf{x}, \boldsymbol{\beta}),$$

where $G(y|\mathbf{x}, \boldsymbol{\beta})$ is a Bernoulli distribution with the parameter $h(\mathbf{x}^T\boldsymbol{\beta})$, where $\boldsymbol{\beta} \in \mathbb{R}^p$. The true parameter of $\boldsymbol{\beta}$ is denoted by $\boldsymbol{\beta}^m$, that is, $Y_1, \ldots, Y_n$ are independent Bernoulli random variables with distribution $G(y|\mathbf{x}_i, \boldsymbol{\beta}^m)$.
We assume that $\mathbf{E}(\mathbf{X}\mathbf{X}^T)$ has full rank p and the fourth-order moments of $\mathbf{x}_i$ exists and is finite. We also assume that the function

$$\mathbf{h}(\mathbf{b}) = \mathbf{E}\boldsymbol{\rho}(\mathbf{X}, Y, \mathbf{b}) \tag{121}$$

exists for all $\mathbf{b}$ and has a unique minimum at $\boldsymbol{\beta}^m$. This implies that $\boldsymbol{\lambda}(\mathbf{b}) = \mathbf{E}\boldsymbol{\psi}(\mathbf{X}, Y, \mathbf{b})$ has a root at $\mathbf{b} = \boldsymbol{\beta}^m$.

Theorem 4.5.1. *(Jurecková and Sen (1996) Theorem 5.2.1)*
*Let the data be as described above and let $\boldsymbol{\rho}(\mathbf{x}, y, \boldsymbol{b})$ be absolutely continuous in the components of $\boldsymbol{b}$ and such that the function $\boldsymbol{h}(\boldsymbol{b})$ of (121) has a unique minimum at $\boldsymbol{b} = \boldsymbol{\beta}^m$. Then under conditions **A1.** and **A2.**, there exists a sequence $\boldsymbol{\beta}^n$ of solutions of the system of equations (115) such that*

$$\sqrt{n}||\boldsymbol{\beta}^n - \boldsymbol{\beta}^m|| = O_p(1) \quad as \quad n \to \infty. \tag{122}$$

Furthermore,

$$\boldsymbol{\beta}^n = \boldsymbol{\beta}^m + n^{-1}(-\Gamma(\boldsymbol{\beta}^m))^{-1} \sum_{i=1}^{n} \boldsymbol{\psi}(\mathbf{x}_i, y_i, \boldsymbol{\beta}^m) + O_p(n^{-1})\,. \qquad (123)$$

Under conditions of the Theorem 4.5.1, $\sqrt{n}(\boldsymbol{\beta}^n - \boldsymbol{\beta}^m)$ has asymptotically a p-dimensional normal distribution $N_p(0, A(\boldsymbol{\beta}^m))$ with

$$A(\boldsymbol{\beta}^m) = (\Gamma(\boldsymbol{\beta}^m))^{-1}\mathbf{E}(\boldsymbol{\psi}(\mathbf{X}, Y, \boldsymbol{\beta}^m)\boldsymbol{\psi}(\mathbf{X}, Y, \boldsymbol{\beta}^m)^T)\Gamma(\boldsymbol{\beta}^m))^{-1}\,,$$

where the matrix $\Gamma(\boldsymbol{\beta}^m)$ has the elements

$$\gamma_{jr} = -\mathbf{E}(\frac{(h'(\eta))^2}{V(\mu)} w_q(\mu)\mathbf{X}_j\mathbf{X}_r)\,. \qquad (124)$$

This follows from

$$\sqrt{n}(\boldsymbol{\beta}^n - \boldsymbol{\beta}^m) \stackrel{asypt.}{=} \sqrt{n}(-\Gamma)^{-1}(\bar{\boldsymbol{\psi}}_n)\,, \qquad (125)$$

where $\bar{\boldsymbol{\psi}}_n = \sum_{i=1}^{n} \psi(\mathbf{x}_i, y_i, \boldsymbol{\beta}^m)/n$.

In the following, the two conditions **A1.** and **A2.** are defined and applied to the BL_q estimator.

A1. *First-order derivatives.*
The ψ_j are absolutely continuous in β_k with derivative $\dot{\psi}_{jk} = \partial\psi_j/\partial\beta_k$ such that $\mathbf{E}(\dot{\psi}_{jk}^2) < \infty$ for $k = 1, \ldots, p$. The matrices $\Gamma(\boldsymbol{\beta}^m) \in \mathbb{R}^{p\times p}$ with elements $\gamma_{jk}(\boldsymbol{\beta}^m)$ and $\mathbf{B}(\boldsymbol{\beta}^m) \in \mathbb{R}^{p\times p}$ with elements $b_{jk}(\boldsymbol{\beta}^m)$ are positive definite, where

$$\gamma_{jk}(\boldsymbol{\beta}) = \mathbf{E}\dot{\psi}_{jk}(\boldsymbol{\beta})$$

and

$$b_{jk}(\boldsymbol{\beta}) = Cov(\psi_j, \psi_k).$$

For the BL_q estimator, $\boldsymbol{\psi}(\mathbf{x}, y, \boldsymbol{\beta})$ is in fact continuous in the components $\boldsymbol{\beta}_k$ and boundedness of

$$\mathbf{E}(\dot{\psi}^2_{jk}(\mathbf{X}, Y, \boldsymbol{\beta})) \tag{126}$$

follows from boundedness of $\mathbf{E}(\mathbf{X}_j^2\mathbf{X}_k^2)$.
Furthermore, $\Gamma(\boldsymbol{\beta}^m) = [\gamma_{jr}(\beta)]_{jr}$ is positive definite under the condition we imposed on H that $\mathbf{E}(\mathbf{X}\mathbf{X}^T)$ has full rank p. Finally, the matrix $B(\boldsymbol{\beta}^m)$ is a covariance and thus positive definite. We have

$$\begin{aligned} b_{jk}(\beta) &= \mathrm{Cov}(\psi_j(\mathbf{X}, Y, \boldsymbol{\beta}), \psi_k(\mathbf{X}, Y, \boldsymbol{\beta})) \\ &= \mathrm{Cov}(A(\mathbf{X})(Y-\mu)\mathbf{X}_j, A(\mathbf{X})(Y-\mu)\mathbf{X}_k) \\ &= \mathbf{E}(A(\mathbf{X})^2\mathbf{X}_j\mathbf{X}_k)\,, \end{aligned} \tag{127}$$

where $A(\mathbf{x}) = \frac{h'(\eta)}{\sqrt{V(\mu)}} w_q(\mu)$. With the assumption on H already mentioned the positive definiteness of $\mathbf{B}(\boldsymbol{\beta}^m) = [b_{jk}(\beta)]_{jk}$ follows.

A2. *Second and third derivatives.*
The functions $\dot{\psi}_{jk}(\mathbf{x}_i, y_i, \boldsymbol{\beta}^m)$ *are absolutely continuous in the component* $\boldsymbol{\beta} \in \Theta$. *There exist random variables* $M_{jrk}(\mathbf{X}, Y, \boldsymbol{\beta}^m)$ *such that* $m_{jrk} = \mathbf{E}M_{jrk}(\mathbf{X}, Y, \boldsymbol{\beta}^m) < \infty$ *and*

$$|\ddot{\psi}_{jrk}(\mathbf{x}, y, \boldsymbol{\beta}^m + \boldsymbol{b})| < M_{jrk}(\mathbf{X}, Y, \boldsymbol{\beta}^m) \text{ for all } ||\boldsymbol{b}|| \leq \delta\,, \delta > 0\,,$$

In the case of the BL_q estimator, $\dot{\boldsymbol{\psi}}_{jk}(\mathbf{x}, y, \boldsymbol{\beta})$ is continuous in the components of $\boldsymbol{\beta}$. Considering the equation (120), note that $h'(\eta)$ and $V(\mu)$ are positive and that $|y-\mu| < 1$. Using the *triangle inequality* it follows that

$$|\ \ddot{\psi}_{jrk}\ | \leq \frac{(h'(\eta))^3}{V(\mu)} \big[\ |\ w_q''(\mu)\ | + 2\ |\ w_q'(\mu)\ |\ \big]\ |\ x_j x_r x_k\ |\ . \tag{128}$$

For any $\delta > 0$ and $||\mathbf{b}|| \leq \delta$ we need to find random variables M_{jrk} that bound the right hand side when evaluated at $\boldsymbol{\beta}^m + \mathbf{b}$. Let

$$f(\boldsymbol{\beta}^m) = \frac{(h'(\eta))^3}{V(\mu)} \big[\ |\ w_q''(\mu)\ | + 2\ |\ w_q'(\mu)\ |\ \big]$$

and consider the Taylor series with remainder

$$f(\boldsymbol{\beta}^m + \mathbf{b}) = f(\boldsymbol{\beta}^m) + \mathbf{b}^T f'(\boldsymbol{\xi}).$$

This can be bounded in absolute value by

$$|f(\boldsymbol{\beta}^m)| + \delta D(f, \boldsymbol{\beta}^m, \delta)\,,$$

where $D(f, \boldsymbol{\beta}^m, \delta) = \max\left\{||f'(\boldsymbol{\beta} + \mathbf{b})|| : ||\mathbf{b}|| \leq \delta\right\}$. Finally we define

$$M_{jrk} = \left((|f(\boldsymbol{\beta}^m)| + \delta D(f, \boldsymbol{\beta}^m, \delta)) \mid \mathbf{X}_j \mathbf{X}_r \mathbf{X}_k \mid \right). \tag{129}$$

The value of m_{jrk} is of the form

$$\mathbf{E} M_{jrk} = \mathbf{E}\big(C(\mathbf{X}) \mid \mathbf{X}_j \mathbf{X}_r \mathbf{X}_k \mid \big)\,, \tag{130}$$

which is bounded because of our condition on the distribution H.

If ρ is convex in $\mathbf{b}$, then the solution of (116) is unique. However, when ρ is not convex (the BL_q estimator), there may exist more roots of the system (115), which satisfies (122). Let $\boldsymbol{\beta}^n$ and $\boldsymbol{\beta}^n_*$ be two such roots. By satisfying the conditions of Theorem 4.5.1 and from the *COROLLARY* 5.2.2 of Jurecková and Sen, we then have

$$\|\beta^n - \beta^n_*\| = O_p(n^{-1}) \qquad \text{as} \qquad n \to \infty\,,$$

which results directly from (123).
For this case, Jurecková and Sen (1996) considered three additional conditions and Theorem 5.2.2, which discusses the $\sqrt{n}$-consistency of this type of estimator.

Theorem 4.5.2. *(Jurecková and Sen (1996) Theorem 5.2.2)*

*Let the data be as described in Theorem (4.5.1) and let $\boldsymbol{\rho}(\mathbf{x}, y, \boldsymbol{b})$ be absolutely continuous in the components of $\boldsymbol{b}$ and such that the function $\boldsymbol{h}(\boldsymbol{b})$ of (121) has a unique minimum at $\boldsymbol{b} = \boldsymbol{\beta}^m$. Then under the conditions **B1.-B3.**, there exists a sequence $\boldsymbol{\beta}^n$ of roots of the equation (115) such that, as $n \to \infty$,*

$$\sqrt{n}(\boldsymbol{\beta}^n - \boldsymbol{\beta}^m) = O_p(1). \tag{131}$$

The conditions **B1.**, **B2.** and **B3.** are defined and applied to the BL_q estimator in the following.

B1. *Moments of derivatives.*
There exist $k > 0$ *and* $\delta > 0$ *such that for* $||\boldsymbol{b}|| < \delta$

$$\mathbf{E}(\psi_j^2(\mathbf{X}, Y, \boldsymbol{\beta}^m + \boldsymbol{b})) \leq k$$

$$\mathbf{E}(\dot{\psi}_{jr}^2(\mathbf{X}, Y, \boldsymbol{\beta}^m + \boldsymbol{b})) \leq k$$

$$\mathbf{E}(\ddot{\psi}_{jrk}^2(\mathbf{X}, Y, \boldsymbol{\beta}^m + \boldsymbol{b})) \leq k$$

For the case of BL_q estimator and the fact that $|y - \mu| < 1$ we have

$$\mathbf{E}(\psi_j^2(\mathbf{X}, Y, \boldsymbol{\beta})) < \mathbf{E}(L(\mathbf{X})\mathbf{X}_j^2)\,, \tag{132}$$

where $L(\mathbf{x}) = \frac{h'^2(\eta)}{V^2(\mu)} w_q^2(\mu)$.

$$\mathbf{E}(\dot{\psi}_{jr}^2(\mathbf{X}, Y, \boldsymbol{\beta})) < \mathbf{E}(M(\mathbf{X})\mathbf{X}_j^2\mathbf{X}_r^2)\,, \tag{133}$$

where $M(\mathbf{x}) = \frac{h'^4(\eta)}{V^2(\mu)} w_q^2(\mu) + \big(\frac{\partial}{\partial \beta_r}[\frac{h'(\eta)}{V(\mu)} w_q(\mu)]\big)^2$.

$$\mathbf{E}(\ddot{\psi}_{jrk}(\mathbf{X}, Y, \boldsymbol{\beta})) < \mathbf{E}(N(\mathbf{X})\mathbf{X}_j^2\mathbf{X}_r^2\mathbf{X}_k^2)\,, \tag{134}$$

where $N(\mathbf{x}) = \frac{h'^6(\eta)}{V^2(\mu)} \big(w_q''(\mu)^2 + 4w_q'(\mu)^2\big)$.

It is evident that there exist a

$$k(\boldsymbol{\beta}^m) = \mathbf{E}(M(\mathbf{X})\mathbf{X}_j^2\mathbf{X}_r^2) + \mathbf{E}(N(\mathbf{X})\mathbf{X}_j^2\mathbf{X}_r^2\mathbf{X}_k^2)\,,$$

which is positive and the equation (132), (133) and (134) are all smaller than $k(\beta^m)$.

Therefore, for any $\delta > 0$ and $||\mathbf{b}|| \leq \delta$ there exists a positive $k(\boldsymbol{\beta}^m + \mathbf{b})$ that bounds the right hand side of (132), (133) and (134) when they

are evaluated at $\boldsymbol{\beta}^m + \mathbf{b}$. To obtain such a universal bound, we can proceed as before. Thus, we have

$$k(\boldsymbol{\beta}^m + \mathbf{b}) = k(\boldsymbol{\beta}^m) + \mathbf{b}^T k'(\boldsymbol{\xi}).$$

This can be bounded by

$$K = k(\boldsymbol{\beta}^m) + \delta D(k, \boldsymbol{\beta}^m, \delta)\,,$$

where $D(f, \boldsymbol{\beta}^m, \delta) = \max\left\{||k'(\boldsymbol{\beta}+\mathbf{b})|| : ||\mathbf{b}|| \leq \delta\right\}$ and K is a function of $\boldsymbol{\beta}^m$ and δ. This apply condition **B1.** to the BL_q estimator.
B2. *Fisher consistency.*
$0 < \gamma(\boldsymbol{\beta}^m) < \infty$ *where* $\gamma(\boldsymbol{\beta}^m) = \mathbf{E}(-\dot{\psi}_{jr}(\mathbf{X}, Y, \boldsymbol{\beta}^m))$.

This is confirmed from (126) and (133).

B3. *Uniform continuity in the mean.*
There exist $\alpha > 0$ *and* $\delta > 0$ *and a random variables* $M_{jrk}(\mathbf{X}, Y, \boldsymbol{\beta}^m)$ *such that* $m_{jrk} = \mathbf{E}M_{jrk}(\mathbf{X}, Y, \boldsymbol{\beta}^m) < \infty$ *and*

$$|\ddot{\psi}_{jrk}(\mathbf{x}, y, \boldsymbol{\beta}^m + \boldsymbol{b}) - \ddot{\psi}_{jrk}(\mathbf{x}, y, \boldsymbol{\beta}^m)| \leq ||\boldsymbol{b}||^{\alpha} M_{jkr}(\mathbf{x}, y, \boldsymbol{\beta}^m) \quad \textit{for} \quad ||\boldsymbol{b}|| \leq \delta$$

We have

$$\begin{aligned}|\ddot{\psi}_{jrk}(\mathbf{x}, y, \boldsymbol{\beta}^m + \mathbf{b}) - \ddot{\psi}_{jrk}(\mathbf{x}, y, \boldsymbol{\beta}^m)| \leq \\ |\ddot{\psi}_{jrk}(\mathbf{x}, y, \boldsymbol{\beta}^m + \mathbf{b})| + |\ddot{\psi}_{jrk}(\mathbf{x}, y, \boldsymbol{\beta}^m)|.\end{aligned}$$

We have seen in (128) that

$$|\ddot{\psi}_{jrk}(\mathbf{x}, y, \boldsymbol{\beta}^m + \mathbf{b})| < f(\boldsymbol{\beta}^m)|x_j x_k x_r|\,,$$

and

$$|\ddot{\psi}_{jrk}(\mathbf{x}, y, \boldsymbol{\beta}^m + \mathbf{b})| + |\ddot{\psi}_{jrk}(\mathbf{x}, y, \boldsymbol{\beta}^m)| < M_{jrk}\,,$$

where M_{jrk} are defined in (136). We finally define a new M_{jrk} such that

$$M_{jrk} = \left((2|f(\boldsymbol{\beta}^m)| + \delta D(f, \boldsymbol{\beta}^m, \delta)) \mid \mathbf{X}_j \mathbf{X}_r \mathbf{X}_k \mid \right). \tag{135}$$

The value of m_{jrk} is of the form

$$\mathbf{E}M_{jrk} = \mathbf{E}\big(O(\mathbf{X}) \mid \mathbf{X}_j\mathbf{X}_r\mathbf{X}_k \mid \big)\,, \tag{136}$$

because of our condition on the distribution H. This verifies the condition **B2.** for $\alpha = 1$ and $\mathbf{b} = 1$.

The conditions **B1.-B3.** guarantee that for there exists at least one root of (115), $\boldsymbol{\beta}^n$ as a BL_q estimator, which is a $\sqrt{n}$-consistent estimator of $\boldsymbol{\beta}^m$.

4.6 Simulation

In order to test and compare the performance of Bernoulli estimates (WML), we conducted a simulation experiment. The estimators of interest are the MLE (BL_2), BL_1, $\mathrm{BL}_{1.2}$ and $\mathrm{BL}_{1.7}$, which are the maximum likelihood and the weighted maximum likelihood for $q = 1$, $q = 1.2$ and $q = 1.7$. Here the MLE is computed with the *glm* algorithm of R and the BL_2 is computed with our algorithm for the maximum likelihood estimator. We will compare the robustness and efficiency performance of the BL_q estimates with the estimator of Croux and Haesbroeck (2003) (BY_{CH}), the estimator of Cantoni and Ronchetti (2001) in which the weights are calculated with "cov.mcd" from R (*Cantoni*), the estimator of Künsch *et al.* (1989) (*CUBIF*) and the estimator of Mallows (*Mallows*).
The different estimators were computed for $N = 1000$ samples of dimension $p = 3$ or 7 and sample size $n = 35$ or $n = 100$. The explanatory variables are generated as independent unit normals and were kept fixed throughout the simulations.

- **Simulation I** The explanatory variables $\mathbf{x}_i \in \mathbb{R}^p$ are distributed according to a normal distribution $\mathbb{N}p(0, \mathbf{I})$. The true parameter values are set to $\boldsymbol{\beta} = (0.5, 1.0, 1.0)$ when $p = 3$ and $\boldsymbol{\beta} = (0.5\,, 0.35\,, \ldots\,, 0.35)$ when $p = 7$. The logit link function $g(\mu) = log(\frac{\mu}{1+\mu}) = \eta = X\beta$ was used and $N = 1000$ dependent variables

y_i were generated according to $Bin(1, \mu_i)$, with $\mu_i = h(\eta_i)$.

- **Simulation II** In each group of data set $n = 35$ or $n = 100$, one of the $\mathbf{x}_i$ vectors was replaced by the vector $[1, -10, \ldots, -10]$. This means that one of the observations that is outlying in the x-space was added. The corresponding response was set equal to 1, even though the simulated model would strongly favor $y = 0$. Thus, the outlying observation is a bad leverage point. The estimators are computed over $N = 1000$ generations.

For each simulated data set i, all the estimators were computed. Furthermore, we calculate the following summary statistics: (137)

$$\text{Bias} = \sum_{k=1}^{p} |\hat{\beta}_{ik} - \beta_k| \qquad \text{IQR} = \sum_{k=1}^{p} IQR(\hat{\beta}_{ik})\,, \tag{137}$$

where $\hat{\beta}_{ik}$ is the k-th component of the i-th simulation. The results are presented in Table 4.

When the observations with $y = 1$ and those with $y = 0$ have corresponding x-values that can be separated by a hyperplane, the logistic regression is not well-defined. Some of the simulations lead to such non-overlapping data sets. These were excluded.
This simulation study illustrates that when there is no contamination to the data set (Simulation I), the estimators behave similarly.

The result of Table 5 shows that under addition of the bad leverage point (Simulation II), the MLE is heavily biased, due to the fact that the MLE estimates of the regression parameters implode. Not surprisingly, its IQR is low, indicating a great stability of the estimate. In short, the MLE estimates very stably a wrong parameter value.

Simulation I						
	n=35		n=100			
	p=3		p=3		p=7	
	Bias	MSE	Bias	MSE	Bias	MSE
MLE	0.13	2.13	0.04	1.20	0.07	2.35
BL_2	0.13	2.13	0.04	1.20	0.07	2.35
$BL_{1.7}$	0.13	2.12	0.04	1.22	0.07	2.34
$BL_{1.2}$	0.15	2.19	0.04	1.24	0.08	2.40
BL_1	0.15	2.21	0.04	1.24	0.08	2.40
CUBIF	0.14	2.14	0.04	1.21	0.07	2.36
Cantoni	0.16	2.56	0.06	1.38	0.09	2.80
Mallows	0.14	2.14	0.04	1.21	0.07	2.36
BY_{CH}	0.14	2.18	0.04	1.24	0.08	2.40

Table 4: *Comparison of the asymptotic bias and the mean squared error of the robust estimators for Binary models in Simulation I.*

Among the robust estimators, BL1.7, CUBIF and Mallows could be called weakly robust. They have a behavior between the highly robust estimators and the MLE.
When the sample size grows to n=100, these intermediate estimators become more competitive, but retain a relatively larger bias.
The dimension $p = 7$ is clearly very challenging. All the robust estimators have quite a large IQR, but are roughly unbiased.
Overall, the estimators BL_1 and BY_{CH} are the winners of this comparison.

In Figure (26), the bias distribution of BL_q estimators for $q = 1$, $q = 1.2$ and $q = 1.7$ are compared with alternative robust estimators and the MLE. Simulation I is shown on the left, while Simulation II is on the right. The top row contains the results for $p = 3$, $n = 35$ and the bottom row those for $p = 7$, $n = 100$. We obtain similar results to β_1 (not included) for the bias distribution of $\beta_2\,, \beta_3 \cdots$ and β_7. The MLE

Simulation II						
	n=35		n=100			
	p=3		p=3		p=7	
	Bias	MSE	Bias	MSE	Bias	MSE
MLE	1.38	0.99	0.96	0.67	0.76	1.74
BL_2	1.38	0.99	0.96	0.67	0.76	1.74
$BL_{1.7}$	0.37	2.97	0.06	1.26	0.03	2.54
$BL_{1.2}$	0.13	2.09	0.06	1.20	0.11	2.50
BL_1	0.14	2.10	0.06	1.20	0.11	2.51
CUBIF	0.34	1.55	0.15	1.01	0.07	2.28
Cantoni	0.14	2.61	0.06	1.40	0.14	2.97
Mallows	0.34	1.55	0.15	1.01	0.07	2.28
BY_{CH}	0.05	2.25	0.01	1.20	0.07	2.49

Table 5: *Comparison of the asymptotic bias and the mean squared error of the robust estimators for Binary models in Simulation II .*

implodes to zero for the contaminated data set (Simulation II).
We also investigated the efficiency of these estimators for a different contamination rate and very high leverage points. We obtained similar results (not included) to those presented above. From this simulation experiment we can conclude that a simple method of weighted maximum likelihood with the weight function depending only on μ yields a robust estimator with a good overall behavior.

4.7 Examples

Example 10. *The data set* ***leukemia*** *of Cook and Weisberg (1982) (Chapter5, p. 193) contains* 33 *leukemia patients with the following variables. The response variable Y is one when the patient survives at least 52 weeks. The two covariates are* $\mathbf{X}_1$ *white blood cells (WBC) and* $\mathbf{X}_2$ *presence (1) or absence (0) of acute granuloma (AG) in the white*

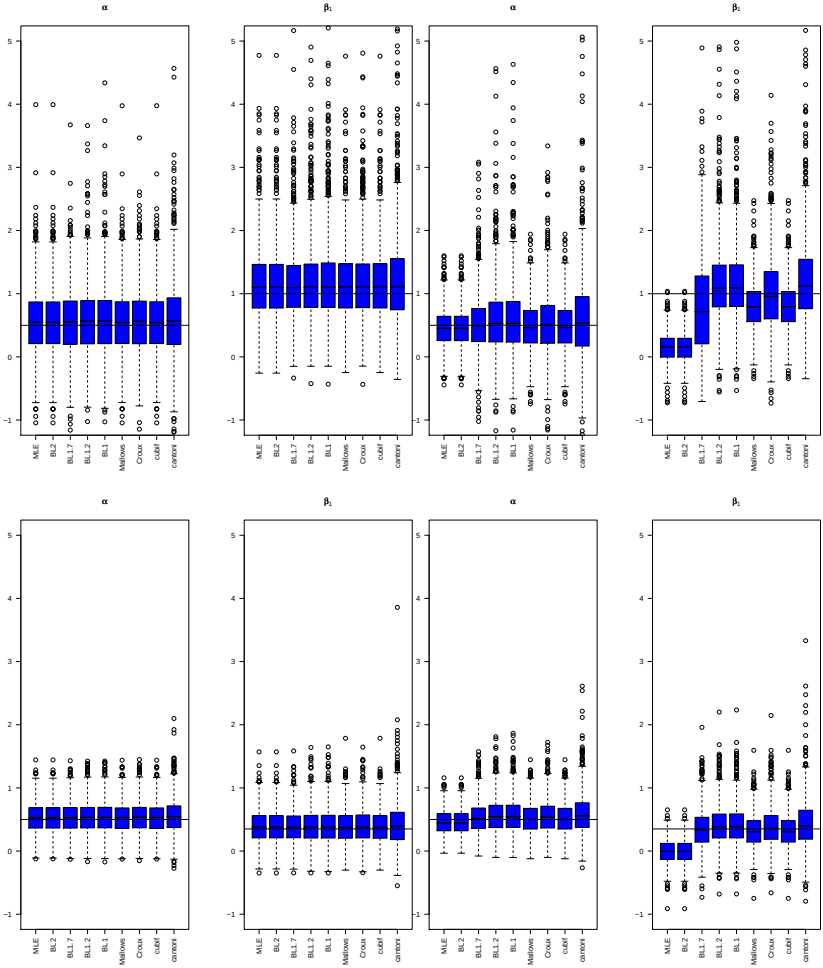

Figure 26: *The bias distribution of BL_q estimator is compared to alternative robust estimators in binary regression. Simulation I is shown on the left, while Simulation II is on the right. The top row contains the results for $p = 3$, $n = 35$ and the bottom row those for $p = 7$, $n = 100$.*

blood cells. Cook and Weisberg (1982) detected that the observation (15) corresponds to a patient with $WBC = 100000$ *who survived for a long period and this observation influenced the MLE.*
In Table 6 we report the estimated parameters and their asymptotic standard error within the parentheses corresponding to[4]

- *the MLE with the complete sample or* BL_2*, (MLE)*
- *the MLE without observation number 15 (*MLE_{15}*)*
- *the Pregibon's weighted MLE (*P_{WMLE}*)*
- *the estimator corresponding to Croux and Haesbroeck (2003) with* $c = 0.5$ *(*BY_{CH}*)*
- *the weighted M-estimate (*WBY_{CH}*)*
- *the* BL_1 *quasi-likelihood estimator (*BL_1*)*

Estimate	Intercept	WBC($\times 10^{-4}$)	AG
MLE	$-1.3(0.81)$	$-0.32(0.18)$	$2.26(0.95)$
MLE_{15}	$0.21(1.08)$	$-2.35(1.35)$	$2.56(1.23)$
P_{WMLE}	$0.17(1.08)$	$-2.25(1.32)$	$2.52(1.22)$
BY_{CH}	$0.16(1.66)$	$-1.77(2.33)$	$1.93(1.16)$
WBY_{CH}	$0.20(1.19)$	$-2.21(0.98)$	$2.40(1.30)$
BL_1	$0.14(0.76)$	$-2.05(0.93)$	$2.5(0.86)$

Table 6: *Comparison of the estimated parameters for the leukemia data set with their standard errors for the MLE,* P_{WMLE}*,* BY_{CH} *and* BL_1 *estimator.*

We can observe that coefficients fitted with BL_1 *are very similar to* MLE_{15} *,* P_{WMLE} *,* BY_{CH} *and* WBY_{CH} *with notable smaller standard error.*

[4]For more details see Maronna *et al.* (2006).

Example 11. *A well-known data set* ***skin*** *was introduced by Finney (1947) and studied by Pregibon (1982) and Croux and Haesbroeck (2003). The Binary response variable Y indicates presence or absence of vasoconstriction of the skin of the digits after air inspiration. The two covariates are $\mathbf{X}_1$ the volume of air inspired and $\mathbf{X}_2$ the inspiration rate.*
Figure 11 presents the skin data set with the estimated coefficients discussed above. The skin data set is difficult to examine. Observation 4 and 18 have been detected as being influential for the MLE. Deleting these two points yields a data set whose overlap depends only on one observation.

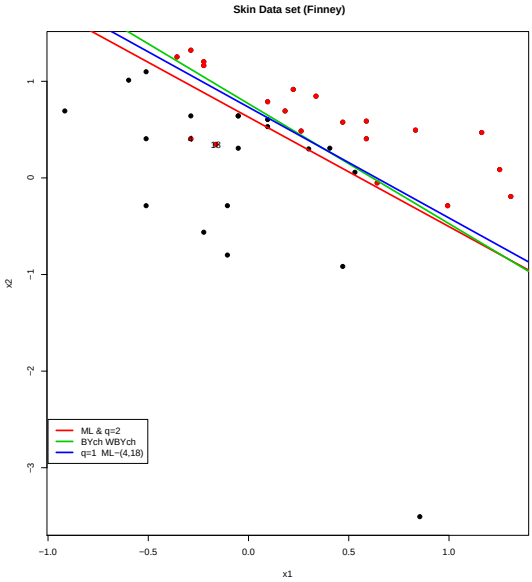

Figure 27: *Comparing the estimator for Skin data (Finney, 1947).*

Table 7 gives the estimated coefficients and their standard errors via the MLE, MLE_{15}, P_{WMLE}, CUBIF (Künsch et al. *(1989)), BY_{CH},*

WBY_{CH} and the BL_1 estimator.
Since the explanatory variables do not contain any outliers, BY_{CH} and WBY_{CH} are are identical. As we can see, the BL_1 estimate is similar to the MLE without observations 4 and 18. We therefore believe that the (BL_1) performs well with this data. Similar to the ML applied to the data set without the influential observations, the standard errors of the BL_1 are quite big.

Estimate	Intercept	*Log*VOL	*log*RATE
MLE	−2.875(1.32)	5.179(1.863)	4.562(1.837)
$MLE_{4,18}$	−24.58(14.02)	39.55(23.25)	31.94(17.76)
P_{WMLE}	−2.837(1.35)	5.139(1.882)	4.508(1.882)
CUBIF	−2.847(1.31)	5.141(1.85)	4.523(1.824)
BY_{CH}	−6.854(10.047)	10.738(15.307)	9.367(12.779)
WBY_{CH}	−6.824(9.85)	10.709(15.113)	9.335(12.540)
BL_1	−21.28(14.52)	34.41(23.64)	27.72(17.96)
$BL_{1.2}$	−21.19(13.04)	34.25(18.96)	27.61(14.49)
$BL_{1.5}$	−5.39(2.27)	8.54(2.44)	7.56(2.4)

Table 7: *Comparison of the estimated parameters for the Skin data set with their standard errors for the MLE, CUBIF, BY_{CH} and BL_q estimator with $q = 1\,, 1.2$ and 1.5.*

Example 12. *In this example we consider the data set from de Vijver M. J. et al. (2002). They used complementary DNA (cDNA) microarrays and the isolation of RNA to analyze breast cancer tissue. The* 25000*-genes arrays were studied to identify a gene-expression profile that is associated with prognosis in 295 patients with breast cancer.*
In this example we consider 11 genes that Nicolas Fournier, researcher at EPFL, had identified during his Master thesis for this data set. The Binary response variable Y indicates whether the selected patients had distant metastases at their survival time. We fit a logistic regression to introduce the relationship between diversity and these variables.

	BL_1 estimator		MLE	
	Coefficients	P-value	Coefficients	P-value
Intercept	−0.69 (0.24)	0.004	−0.67 (0.23)	0.0026
Contig40719_RC	−0.87 (0.47)	0.0624	−0.76 (0.44)	0.0822
NM_005133	0.27 (0.09)	0.0030	0.25 (0.09)	0.0042
Contig41652	0.51 (0.25)	0.0400	0.56 (0.24)	0.0201
NM_005375	−0.05 (0.04)	0.1294	−0.05 (0.04)	0.1214
NM_006622	−0.21 (0.06)	0.0004	−0.21 (0.06)	0.0003
NM_016109	0.13 (0.04)	0.0026	0.13 (0.04)	0.0022
Contig43806_RC	−0.07 (0.04)	0.1091	−0.07 (0.04)	0.1033
L27560	0.15 (0.04)	0.0001	0.14 (0.04)	0.0001
Contig43791_RC	−0.07 (0.04)	0.0408	−0.06 (0.03)	0.0601
NM_004524	0.16 (0.06)	0.0038	0.15 (0.06)	0.0055

Table 8: *Comparison of the estimated parameters for genes expression data set with their p-values. The coefficients are estimated by the MLE and BL_1 for the logistic regression model.*

In Table 8 we report the estimated parameters and their p-value via the BL_1 and the MLE. The values in parentheses are their standard errors. From Table 8 and based on the confidence interval for each of the explanatory variables, one can see that two of these variables (NM_005375 and Contig43806_RC) cannot be considered significant in the model. From the MLE the variables Contig40719_RC and Contig43791_RC are at the limit to be considered in the model. However, using the robust method the gene Contig43791_RC can stay in the model. To reduce the number of these variables we apply a backward stepwise procedure on the MLE and the BL_1. We keep the variable in the model if the p-value obtained from the robust method at each step of procedure is less than 5%.

The results of this analysis are presented in Table 9. As we can see the estimated parameters and their p-values, which are computed by the

maximum likelihood estimator and the robust models BL_1, are essentially similar.

	BL_1 estimator		MLE	
	Coefficients	P-value	Coefficients	P-value
NM_005133	0.24	$1.87e{-}03$	0.26	$3.70e{-}03$
Contig41652	0.54	$2.68e{-}02$	0.56	$1.87e{-}02$
NM_006622	-0.25	$1.12e{-}05$	-0.24	$1.21e{-}05$
NM_016109	0.18	$2.21e{-}06$	0.18	$1.07e{-}06$
L27560	0.19	$3.04e{-}07$	0.18	$3.05e{-}07$
Contig43791_RC	-0.07	$3.01e{-}02$	-0.06	$6.97e{-}02$
NM_004524	0.20	$2.69e{-}04$	0.19	$2.21e{-}04$

Table 9: *The coefficients estimation and corresponding P-value for final logistic regression model, estimated by the MLE and BL_1 for the genes expression data set.*

The approach to adapt the simple weighted maximum likelihood with the weight that depends only on μ in Poisson regression are studied in Chapter 5.

CHAPTER 5

Poisson Regression

In Chapter 4 we introduced the robust method for binary regression. In the present chapter we focus on a Poisson regression, where the response variables are in the set of non negative integers without any upper limit on the observed values.
Let y_i for $i = 1, \ldots, n$ be independent according to a Poisson distribution. The probability density is given by

$$\mathbf{P}(Y = y) = \frac{e^{-\mu}\mu^y}{y!} \qquad y = 0, 1, 2, \ldots , \tag{138}$$

where $\mu > 0$ is the expectation and the variance of y_i.
Consider a generalized linear model with response variables $y_i \sim \mathbb{P}(\mu_i)$ and the non-random explanatory variables $\mathbf{x}_i \in \mathbb{R}^p$. Let $g(.)$ be an monotone increasing function defined on the positive reals. The commonly used link function is "log". The regression parameters $\boldsymbol{\beta} = (\beta_1 , \ldots , \beta_p)^T$ enter into the model through

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

The Poisson log-likelihood function for a vector of independent observations y_i is

$$l(\boldsymbol{\beta}) = \log L(\boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i \log \mu_i - \mu_i). \tag{139}$$

Differentiation (139) with respect to $\boldsymbol{\beta}$ yields the maximum likelihood equations

$$\sum_{i=1}^{n} h'(\eta_i)(\frac{y_i - \mu_i}{\mu_i})\mathbf{x}_i = \mathbf{0}\,, \tag{140}$$

where $\mu_i = h(\eta_i)$ and $h'(\eta_i) = \partial\mu_i/\partial\eta_i$. For the canonical link function $\mu_i = e^{\mathbf{x}_i\boldsymbol{\beta}}$ the system of equations (140) become

$$\sum_{i=1}^{n} (y_i - e^{\mathbf{x}_i\boldsymbol{\beta}})\mathbf{x}_i = \mathbf{0}. \tag{141}$$

In the next section we give an overview of the robust estimates for Poisson regression parameters $\boldsymbol{\beta}$.

5.1 Robust Poisson regression

In the literature, there is no specific robust method for estimating Poisson regression parameters. However, the general robust methods for generalized linear models can be applied to Poisson models. The conditionally unbiased bounded influence estimator of Künsch *et al.* (1989) is one of the first robust models that can be used to estimate the parameters for these models.
More recently, a Mallows quasi-likelihood estimator of Cantoni and Ronchetti (2001) has been applied in particular to Poisson regression. These two estimators have been discussed in detail in Chapter 2.
In the present chapter we study the L_q quasi-likelihood estimator of Morgenthaler (1992) for Poisson regression models. Later we introduce two weighted maximum likelihood estimates for these models.

5.2 L_q quasi-likelihood for Poisson regression

Consider the system of equations (60) of $\boldsymbol{\beta}$:

$$U_q(\boldsymbol{\beta}) = \sum_{i=1}^{n} \frac{h'(\eta_i)}{V(\mu_i)^{q/2}} \{|y_i - \mu_i|^{q-1} \text{sgn}(y_i - \mu_i) - c(\mu_i)\} = \mathbf{0}\,. \quad (142)$$

For the Poisson model $y_i \sim \mathbb{P}(\mu_i)\,, c(\mu_i)$ is equal to

$$\begin{aligned} c(\mu_i) &= \underset{Y|X}{\mathbf{E}} \{|Y_i - \mu_i|^{q-1} \text{sgn}(Y_i - \mu_i)\} \\ &= -\sum_{y_i=0}^{\lfloor \mu_i \rfloor} (\mu_i - y_i)^{q-1} \mathbf{P}(Y_i = y_i) + \sum_{y_i=\lceil \mu_i \rceil}^{\infty} (y_i - \mu_i)^{q-1} \mathbf{P}(Y_i = y_i)\,. \end{aligned} \quad (143)$$

In the second part of this expression we need to compute an infinite sum. Given the exponential decay of the probabilities, this sum clearly converges. An approximate value can be obtained by computing $\sum_{j=\lceil \mu \rceil} (j - \mu)^{q-1} \mathbf{P}(Y_i = j)$ for a finite value of N .

In order to define the value of N, we performed a small simulation study. We compute these series for different value of $N = 10$, $N = 100$, $N = 1000$ and $N = 10000$ to find the N that bounds the maximum error by ε. This simulation study showed that we can fix the value of N to $4\bar{y}$ for sufficient accuracy.

5.2.1 L_1 quasi-likelihood for Poisson regression

Since in this research we are interested in studying the L_1 estimate for generalized linear models, we only consider the case of $q = 1$ of L_q quasi-likelihood estimates. For the canonical link function $\mu = e^{\eta}$, the system of equations (142) become

$$U_1(\boldsymbol{\beta}) = \sum_{i=1}^{n} V(\mu_i)^{1/2} \big\{ \text{sgn}(y_i - \mu_i) - 1 + 2\mathbf{P}(y_i \leq \mu_i) \big\} \mathbf{x}_i. \quad (144)$$

We can show that

$$U_1(\boldsymbol{\beta}) = \begin{cases} \sum_{i=1}^n V(\mu_i)^{1/2}\big(2\mathbf{P}(y_i \leq \mu_i)\big)\mathbf{x}_i & y_i > \mu_i \\ \sum_{i=1}^n V(\mu_i)^{1/2}\big(2\mathbf{P}(y_i \leq \mu_i) - 2\big)\mathbf{x}_i & y_i \leq \mu_i \,. \end{cases} \tag{145}$$

Or

$$U_1(\boldsymbol{\beta}) = \sum_{i=1}^n V(\mu_i)^{1/2}\big(2\mathbf{P}(y_i \leq \mu_i) - 1\big)W(y_i, \mu_i)\mathbf{x}_i \,, \tag{146}$$

where

$$W(y_i, \mu_i) = \begin{cases} 1 + \frac{1}{2\mathbf{P}(y_i \leq \mu_i) - 1} & y_i > \mu_i \\ 1 - \frac{1}{2\mathbf{P}(y_i \leq \mu_i) - 1} & y_i \leq \mu_i \,. \end{cases} \tag{147}$$

The weight function (147) in the estimating equation (146) is a function on μ and y.

One can consider an alternative approach to compute $\mathbf{P}(y_i \leq \mu_i)$. Let $Y(t)$ be a Poisson process with intensity λ, satisfying $Y(0) = 0$. It follows that $Y(t) \sim \mathbb{P}(t\lambda)$. Let T_i be the waiting times between the events counted by the Poisson process. These are independent and follow an exponential distribution with rate λ. From the definition of the Poisson process we find

$$\mathbf{P}(Y_i(t) > \mu) = \mathbf{P}(\sum_{j=1}^{\mu} T_j < t) = \mathbf{P}(G < t) \,,$$

where G is the random variable distributed according to a Gamma distribution $\mathbb{G}(\mu, \lambda)$.
To apply this formula to the computation of $c(\mu)$, we have to choose λ such that $t\lambda = \mu$, i.e., $\lambda = \mu/t$.

5.2.2 Computation of L_1 quasi-likelihood estimator

Let consider the system of equation (145) being of the form of

$$\sum_{i=1}^n w_i(y_i - \mu_i)\mathbf{x}_i \,, \tag{148}$$

where

$$w_i = V(\mu_i)^{1/2}\{\text{sgn}(y_i - \mu_i) - 1 + 2\mathbf{P}(y_i \leq \mu_i)\}/(y_i - \mu_i). \quad (149)$$

The computation of this weighted maximum likelihood is based on the Newton-Raphson iterative method, which uses the derivative

$$\frac{\partial U_k}{\partial \beta_r} = \sum_{i=1}^{n} h'(\eta_i) x_{ir} x_{ik} \{W_i'(y_i - \mu_i) - W_i\},$$

where $h'(\eta) = \partial \mu_i / \partial \eta_i$ and $w' = \partial w_i / \partial \mu_i$. This yields the Newton-Raphson update

$$\hat{\boldsymbol{\beta}}_{new} = \hat{\boldsymbol{\beta}}_{old} + (X^T N X)^{-1} U(\hat{\boldsymbol{\beta}}_{old}),$$

where N is a diagonal matrix with diagonal elements

$$N_{ii} = h'(\eta)\{w_i - w_i'(y_i - \mu_i)\}.$$

To simplify the computation we ignore the term involving the derivatives of the weight w_i' in the matrix N.

In robust binary regression, we have seen the advantage of a weighted maximum likelihood estimate with weight function that only depends on μ. In the binary case, this estimator is a simplified version of L_q quasi-likelihood estimates. In Poisson models, the simplification of the L_q estimate cannot yield a similar form of the estimator. Therefore, we are interested in finding a similar weight function for Poisson regression. In binary regression the response variables are limited to values of 0 or 1. However, there is no upper limit for a Poisson variable and to introduce a reasonable weight functions we need at least knowledge of the mean of the observed data. This fact yields the weight function that do not depend on μ_i alone. In the next section we introduce new weighted maximum likelihood estimates for Poisson models.

5.3 The WMLE for Poisson regression

Let the response variable have a Poisson distribution $y_i \sim \mathbb{P}(\mu_i)$ and let the explanatory variables be $\mathbf{x}_i \in \mathbb{R}^p$. We introduce two new WMLE

estimating equations. Several examples and simulations will then illustrate these two procedures.
Our first new weighted maximum likelihood system equation has the form

$$\sum_{i=1}^{n} \frac{h'(\eta_i)}{\mu_i} W_y(\mu_i)(y_i - \mu_i)\mathbf{x}_i = \mathbf{0}\,, \tag{150}$$

where $\mu_i = h(\eta_i)$, $h'(\eta_i) = \partial\mu_i/\partial\eta_i$ and the weight function $W_y(\mu)$

$$W_y(\mu) = \exp(-\frac{(\mu - m)^2}{2s^2})\,. \tag{151}$$

Here, m and s denote the arithmetic mean and standard deviation of the observations,

$$m = \frac{1}{n}\sum_{i=1}^{n} y_i \qquad s = \frac{1}{n}\sum_{i=1}^{n}(y_i - m)^2.$$

Alternatively, we compute a robust estimate for these two values from the observed data y_i. We use the minimum covariance determinant (MCD) estimator (Rousseeuw (1985)) to introduce a new group of observation. Furthers, we use this data set to compute the arithmetic mean and standard deviation m and s.
With this weight, the observations on the edge of the design space are given less weight. The full weight of 1 is given to those observations with μ equal to the sample mean.
Figure 28 shows the function $W_y(\mu)$ of the Poisson model for various values of the sample mean and variance. We will denote this estimator with WMLE_1.
Based on the definition of this weight function $W_y(\mu)$, one can expect that these weighted maximum likelihood estimates are biased because $\mathbf{E}\big(W(\mu_i)(Y_i - \mu_i)\big) \neq 0$.
However, this weight function, similar to the second weight function in binary regression, yields a numerically unstable estimate. The weight function $W_y(\mu)$ gives weight 1 only to the observations in a small interval around the center and quickly drops to zero for observation outside this interval. This abrupt change is a problem in real data with small

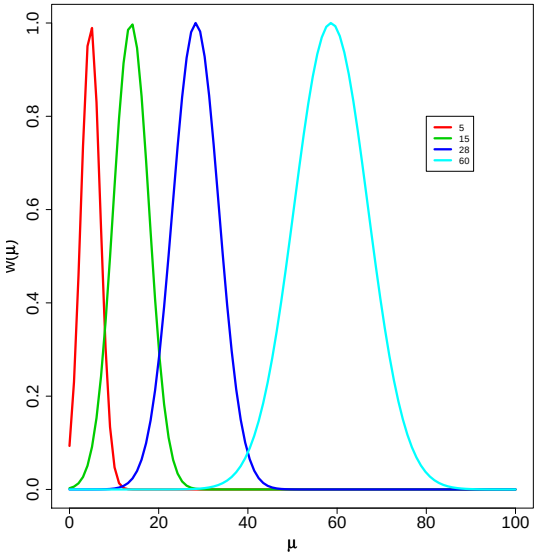

Figure 28: *The weight function $W_y(\mu)$ associated with four Poisson distributions with different parameters $\mu = 5\,, 15\,, 28$ and 60.*

sample size. In such cases, assigning a weight zero reduces the number of observations and the algorithm may converge to strange results. This problem is presented in Example 14.
Therefore, we introduce a second weight function for Poisson models, which only depends on μ and two constants c_1 and c_2. The system equation for this WMLE is given by

$$\sum_{i=1}^{n} \frac{h'(\eta_i)}{\mu_i} W^{\mathrm{MH}}(\mu_i)(y_i - \mu_i)\mathbf{x}_i = \mathbf{0}\,, \tag{152}$$

where $\mu_i = h(\eta_i)$, $h'(\eta_i) = \partial\mu_i/\partial\eta_i$ and weight function

$$W^{\mathrm{MH}}(\mu_i) = \begin{cases} 1 & \frac{v}{c_1} < \mu_i < c_1 v \\ \frac{c_1 \mu_i}{v} & \mu_i < \frac{v}{c_1} \\ \frac{c_2 v - \mu_i}{v} & c_1 v < \mu_i < c_2 v \\ 0 & \text{otherwise}\,, \end{cases} \tag{153}$$

where v is the median of μ. In this research we chose some what arbitrary the value of $c_1 = 2$ and $c_2 = 3$. This estimator is denoted by $\mathrm{WMLE}^{\mathrm{MH}}$. Figure 29 shows this weight function for three Poisson regression models, which is simulated from different values of mean and variance, respectively.

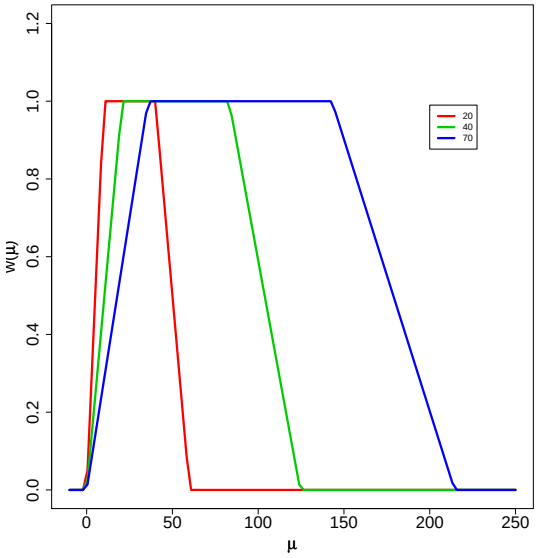

Figure 29: *The weight function* $W^{MH}(\mu)$ *for three different values of* $v = 20\,, 40$ *and* 70.

This weighted maximum likelihood estimator is illustrated in several examples later in this chapter.

5.3.1 Computation of WMLE

Consider the canonical link function, where the system of equations (152) become

$$U(\boldsymbol{\beta}) = \sum_{i=1}^{n} \frac{h'(\eta_i)}{\mu_i} W^{\mathrm{MH}}(\mu_i)(y_i - \mu_i)\mathbf{x}_i = \mathbf{0}\,. \tag{154}$$

The computation of these system of equations similar to (5.2.2) is based on the Newton-Raphson method. The update is

$$\hat{\boldsymbol{\beta}}_{new} = \hat{\boldsymbol{\beta}}_{old} + (X^T N X)^{-1} U(\hat{\boldsymbol{\beta}}_{old}),$$

where N is a diagonal matrix with diagonal elements

$$N_{ii} = h'(\eta)\{W^{\mathrm{MH}}(\mu_i) - W'^{\mathrm{MH}}(\mu_i)(y_i - \mu_i)\}.$$

To simplify the computation, we again ignore the term involving the derivatives of the weight $W'^{\mathrm{MH}}(\mu_i)$ in the matrix N.
Alternatively, $\boldsymbol{\beta}$ can be computed iteratively using the weighted glm algorithm with updated weight function.
For the WMLE_1 we can consider the same algorithm with the weight function $W_y(\mu)$.

5.4 Asymptotic covariance matrix

The asymptotic covariance matrix of the $WMLE^{\mathrm{MH}}$ is computed similar to the method of BL_q, where $U(\boldsymbol{\beta})$ is defined by the system of equations (152). This asymptotic covariance matrix is estimated by

$$\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\beta}}) = (D^T V^{-1} Q D)^{-1} (D^T V^{-1} R V^{-1} D)(D^T V^{-1} Q D)^{-1}\,. \tag{155}$$

Here Q denotes the diagonal matrix with diagonal elements

$$Q_i = W^{\mathrm{MH}}(\mu_i). \tag{156}$$

R is also a diagonal matrix with diagonal elements

$$R_i = Q_i^2 V(\mu_i).$$

V is a diagonal matrix of variance with diagonal elements $(V(\mu_1), cdots, V(\mu_n))$. In Chapter 4, we proposed an alternative formula for the asymptotic covariance matrix of the estimate (4.1). Here we generalize this formula for the WMLE$^{\mathrm{MH}}$. In the following, we assume that $\mathbf{x}_1, \ldots, \mathbf{x_n}$ are i.i.d. realizations of a random variable $\mathbf{X}$ with a known distribution function H.
Let $D_{ij} = \partial\mu_i/\partial\beta_j$. The asymptotic covariance matrix (155) becomes[1]

$$\mathrm{Cov}(\hat{\boldsymbol{\beta}})_{\mathrm{asy}} = \left(\int A(\mathbf{x})(h'(\eta))^2 \mathbf{x}\mathbf{x}^T dH(\mathbf{x}) \right)^{-1}$$
$$\left(\int B(\mathbf{x})(h'(\eta))^2 \mathbf{x}\mathbf{x}^T dH(\mathbf{x}) \right) \left(\int A(\mathbf{x})(h'(\eta))^2 \mathbf{x}\mathbf{x}^T dH(\mathbf{x}) \right)^{-1}. \quad (157)$$

Here $A(\mathbf{x})$ and $B(\mathbf{x})$ are given by

$$A(\mathbf{x}_i) = V(\mu_i)^{-1} Q_i\,, \quad (158)$$

and

$$B(\mathbf{x}_i) = V(\mu_i)^{-1} Q_i^2\,, \quad (159)$$

with Q_i defined in (156).

To illustrate this formula and compute the efficiency of the WMLE$^{\mathrm{MH}}$ estimator for different value of c_1 and c_2, we conducted a simulations study. We use a basic Monte Carlo integration method to estimate (157). The simulation are carried out for $p = 4$ dimensions with $\boldsymbol{\beta} = (1, 0.2, 0.2, 0.2)$. We generate $N = 1000$ explanatory variables $\mathbf{x}$ from the two following distribution of $H(\mathbf{x})$,

$$H_1(\mathbf{x}) = \left(\mathbf{1}, \mathbb{N}_{p-1}(\mathbf{0}, \Sigma)\right) \quad (160)$$
$$H_2(\mathbf{x}) = \left(\mathbf{1}, (1-\varepsilon)\mathbb{N}_{p-1}(\mathbf{0}, \Sigma) + \varepsilon \mathbb{N}_{p-1}(\mathbf{3}, \Sigma)\right), \quad (161)$$

[1] For more details see (4.1).

with $\varepsilon = 30\%$. We choose three different Σ,

$$\Sigma_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \Sigma_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \Sigma_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/4 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Table 10 compares the efficiency of the WMLE$^{\mathrm{MH}}$ related to the MLE for different values of (c_1, c_2) to the MLE. In this table two distributions are considered for the $\mathbf{x}$ with three different Σ.

$H_1(\mathbf{x})$		
(c_1, c_2)	(2,3)	(1,4)
Σ_1	0.79	0.72
Σ_2	0.81	0.70
Σ_3	0.81	0.70
$H_2(\mathbf{x})$		
(c_1, c_2)	(2,3)	(1,4)
Σ_1	0.82	0.73
Σ_2	0.81	0.70
Σ_3	0.80	0.70

Table 10: *Comparing the efficiency of the WMLE^{MH} for (c_1, c_2) equal to $(2, 3)$ or (1,4) to the MLE. Two distributions are considered for X with three different Σ.*

5.5 Influence function of the WMLE$^{\mathrm{MH}}$

The influence function of the weighted maximum likelihood estimate WMLE$^{\mathrm{MH}}$ can be derived directly from the general formula of IF for the general M-estimator (20) (Hampel *et al.* (1986)),

$$\mathrm{IF}(\mathbf{x}_*, y_*) = M(\psi_{\mathrm{MH}}, F)^{-1} \psi_{\mathrm{MH}}(\mathbf{x}_*, y_*, \mu_*) \in \mathbb{R}^p \,,$$

with

$$\psi_{\mathrm{MH}}(\mathbf{x}_*, y_*, \mu_*) = \frac{h'(\eta_*)}{V(\mu_*)} W^{\mathrm{MH}}(\mu_*)(y_* - \mu*)\mathbf{x}_* \,, \tag{162}$$

and $M(\psi_{\mathrm{MH}}, F) = -E_F\left[\frac{\partial}{\partial \boldsymbol{\beta}}\psi_{\mathrm{MH}}(x, y, \mu)\right]$. Here y_* is an added element to the response variable, $\mathbf{x}_*$ is an added vector $(1, x_{*1}, ..., x_{*p})$ to the design matrix and $\eta_* = \mathbf{x}_*^T\boldsymbol{\beta}$, $\mu_* = h(\eta_*)$, $h'(\eta_*) = \partial\mu_*/\partial\eta_*$. The matrix M can be estimated by

$$\hat{M}_q(\psi, F) = D^T V^{-1} Q D \,,$$

where Q is a diagonal matrix with elements

$$Q_i = W^{\mathrm{MH}}(\mu_i). \tag{163}$$

Example 13. *We consider the Poisson model with $p = 2$, a slope β and an intercept α. We define the model as follows:*

- *The explanatory variables $\mathbf{x}_i$ are distributed according to a normal distribution $\mathbb{N}(0, 1)$, for $i = (1, \ldots 100)$.*
- *The response variables y_i are distributed according to a Poisson distribution $\mathbb{P}(10)$.*
- *We add an outlier $y_* = 40$. We then add the vector $[1, x_*]$ to the original design matrix. Here x_* is varying from -200 to 300.*

Figure 30 compares the MLE and the WMLE^{MH} estimates under the influence of an additional observation of varying $\mathbf{x}_i$. It shows that the bad leverage points can significantly influence β in the MLE. However, the influence of these observations on the WMLE^{MH} converges to zero. We can say that the WMLE^{MH} is not sensitive to bad leverage points.

5.6 Simulation

In order to test and compare the performance of this estimator, we conducted a simulation experiment. The estimators of interest are WMLE_1

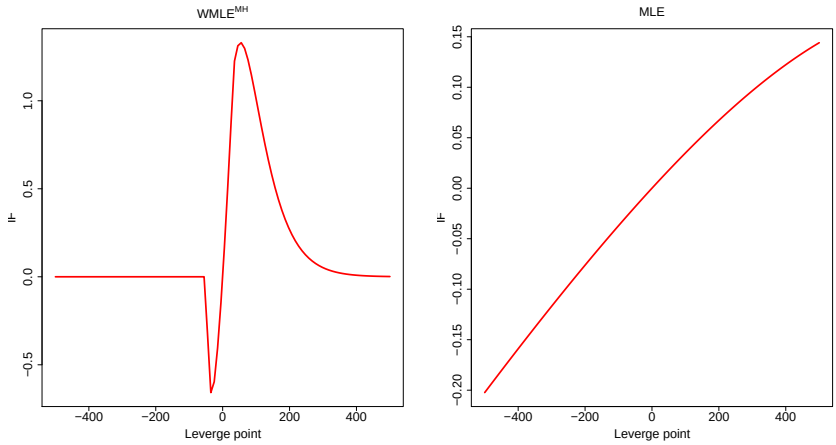

Figure 30: *Comparing the influence function of β, for MLE and the $WMLE^{MH}$.*

and WMLE^{MH}, computed using Newton-Raphson with weight functions $W_y(\mu)$ (151) and $W^{\text{MH}}(\mu)$ (153). We compare the robustness and efficiency performance of both estimates with MLE, Cantoni (weighted on X with "cov.mcd"), CUBIF and MALLOWS estimates[2].

The explanatory variables $\mathbf{x}_i \in \mathbb{R}^p$ are distributed according to a normal distribution $\mathbb{N}_p(0,1)$. The true parameter values are set to $\boldsymbol{\beta} = (1,2,2)$ when $p = 3$. The log link function $g(\mu) = \log(\mu) = \eta$ is used and $N = 1000$ dependent variables y_i were generated according to $\mathbb{P}(\mu_i)$, with $\mu_i = h(\eta_i)$.

We add $m = 1$ outliers (bad leverage points) to the $n = 100$ original observations. In each of the N generations, we add m times $y = 3\bar{y}$ to the dependent variable and m vectors $[1, -5, \ldots, -5]$ to the design matrix, respectively. Therefore, $y_{new} \in \mathbb{R}^{n+m}$ and $X \in \mathbb{R}^{(n+m)\times p}$. The estimators are computed over $N = 1000$ generations. Then, we compute the bias and mean-squared error (MSE) of estimates over N runs.

[2]These estimators are studied in Chapter 2.

Bias and MSE are computed as follows:

$$\text{Bias} = \| \frac{1}{N} \sum_{k=1}^{N} \hat{\boldsymbol{\beta}}_k - \boldsymbol{\beta} \| \qquad MSE = \frac{1}{N} \sum_{k=1}^{N} \| (\hat{\boldsymbol{\beta}}_k - \boldsymbol{\beta}) \|^2 \ . \quad (164)$$

The results are presented in Table 11. This simulation study illustrates that the weighted maximum likelihood estimator with the weight function $W^{\text{MH}}(\mu)$ is asymptotically unbiased similar to the estimator of Cantoni and Ronchetti (2001) and Künsch *et al.* (1989).
When there is no contamination to data set, the estimators behave similarly. After adding the outlier to the sample, the WMLE^{MH} estimator performs similar to its competitors Cantoni, CUBIF and Mallows. More generally we can say that a simple method of weighted maximum likelihood with the weight function depending only on μ yields a robust estimator with a good overall behavior.

	Original data		With outliers	
	Bias	MSE	Bias	MSE
MLE	0.00026	0.00028	2.87	8.26
WMLE_1	0.84	5.81	11.84	1022.68
WMLE^{MH}	0.0076	0.012	0.0066	0.012
Cantoni	0.00024	0.00036	0.000219	0.00036
Mallows	0.94	870	0.0019	0.0018
CUBIF	0.002	0.0018	0.002	0.0018

Table 11: *Comparison of the asymptotic bias and the mean squared error of the robust estimators for Poisson models (164).*

5.7 Examples

Example 14. *We consider the mortality rates from a non-infectious disease for a large population from Example 2. The estimated parameters are presented in Figure 32.*

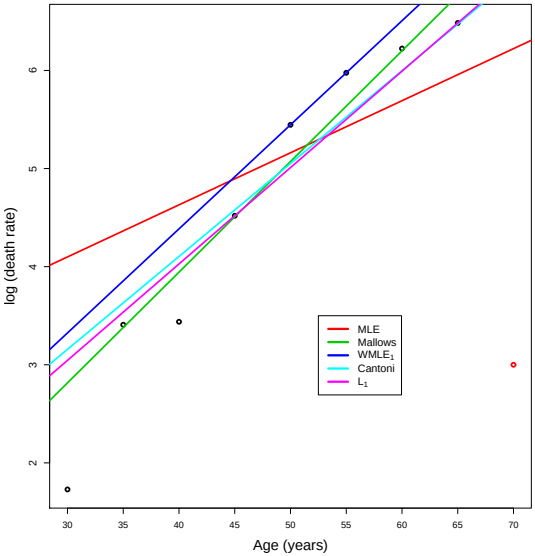

Figure 31: *Comparison of different estimators for Poisson regression. The WMLE is estimated by considering the first weight function $W_y(\mu)$.*

Example 15. *As an illustration, we consider the data arising from a clinical trial of 59 patients who suffer from epilepsy carried out by Leppik et al. (1985)*[3]*.*
The epileptic attacks can be simple or complex and the total number of attacks is modeled as a Poisson variable. They were randomized to receive either the anti-epileptic drug prog-abide or a placebo. The explanatory variables are the baseline seizure rate, recorded during an eight week period prior to randomization divided by 4, "Base". The age of the patients in years divided by 10, "Age", and the binary indicators "Trt" for the prog-abide group.
The interaction between treatment and baseline seizure rate is considered in the model. This interaction shows that the seizure rate for

[3]The information for this data set are summarized in Thall and Vail (1990).

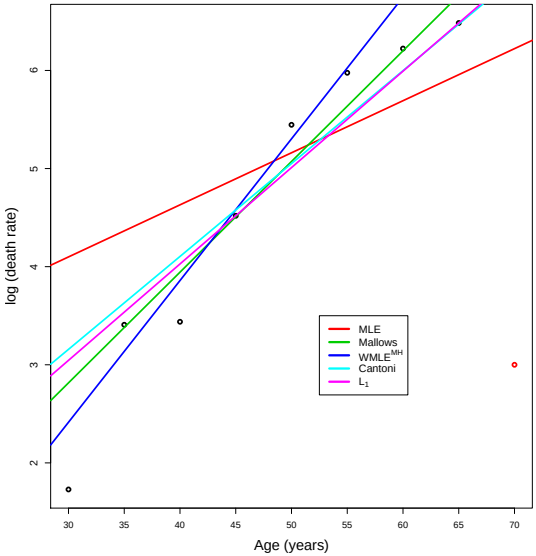

Figure 32: *Comparison of different estimators for Poisson regression. The second weight function $W^{MH}(\mu)$ is considered for the WMLE.*

the prog-abide group is either higher or lower than that for the placebo group. Which of two occurs depends on the baseline count. If the baseline count exceeds a critical threshold, the drug reduces the attacks, otherwise not.
Table 12 gives the parameter estimates, their standard errors, their t-values and their p-values. The Poisson regression parameters are estimated with three robust methods and they are compared with the MLE. This example shows that with the $WMLE^{MH}$ method we obtain the similar parameter estimates as using Cantoni's estimator Cantoni and Ronchetti (2001) for the epilepsy data set. Via the MLE method, the variable interaction is not significant to keep in the final model. However, with the $WMLE^{MH}$ we keep this variable in the model.

Example 16. *In this example, we consider the Possum data-set, which*

	Int.	Age	Base	Trt	Base:Trt
MLE	1.97	0.24	0.085	−0.26	0.008
st.err	0.13	0.04	0.004	0.077	0.004
t-value	14.48	5.90	23.31	−3.34	1.71
p-value	2e-16	3.7e-9	2e-16	0.0009	0.088
Cantoni	2.04	0.16	0.085	−0.32	0.012
st.err	0.15	0.047	0.004	0.087	0.005
t-value	13.2	3.4	20.4	−3.7	2.4
p-value	2e-16	0.0008	2e-16	0.0002	0.017
Mallows	1.84	0.129	0.13	−0.42	0.029
st.err	0.29	0.076	0.04	0.227	0.043
t-value	6.28	1.69	3.4	−1.86	0.66
p-value	3.4e-10	9.1e-02	6.7e-04	6.3e-02	5.1e-01
WMLE$^{\mathbf{MH}}$	2.13	0.044	0.128	−0.47	0.054
st.err	0.198	0.057	0.014	0.16	0.021
t-value	10.91	0.77	9.35	−3	2.58
p-value	1.1e-27	4.4e-01	9.2e-21	2.6e-03	1e-02

Table 12: *Comparison of the estimated parameters for the epilepsy data set with their standard errors , their t-values and their p-values.*

is studied in Lindsay and Nix (1990). This data was collected to analyze the diversity of arboreal marsupials in Montane ash forest in Australia. Arboreal marsupials were reported at dusk and night-time at 152 sites, each of 3 ha within with uniform vegetation in the Central Highlands of Victoria from July 1983 to June 1984 and March 1987 to February 1989.

For each site, the following measures were recorded[4]*: number of shrubs, number of cut stumps from past logging operations, number of stage (hollow-bearing trees), a bark index reflecting the quantity of decorticating bark, a habitat score indicating the suitability of nesting and foraging*

[4]Cantoni and Ronchetti (2001).

habitat for Linderbeater's possum, the basel area of acacia species, the species of eucalypt with greatest stand basel area (Eucalyptus regnans, Eucalyptus delegatension, Eucalyptus nitens) and the aspect of the site. We use the Poisson generalized linear models with log-link to introduce the relationship between diversity and these variables.

In Table 13, we report the estimated parameters via the WMLE^{MH},

Variable	WMLE$^{\mathrm{MH}}$	MLE	Cantoni's
Intercept	−1.01 (0.29)	−0.95 (0.27)	−0.88 (0.27)
Shrubs	0.01 (0.02)	0.01 (0.02)	0.01 (0.02)
Stumps	−0.25 (0.30)	−0.27 (0.29)	−0.22 (0.29)
Stags	0.04 (0.01)	0.04 (0.01)	0.04 (0.01)
Bark	0.05 (0.02)	0.04 (0.01)	0.04 (0.02)
Habitat	0.09 (0.04)	0.07 (0.04)	0.07 (0.04)
Acacia	0.05 (0.01)	0.02 (0.01)	0.02 (0.01)
E. deleg.	−0.00 (0.20)	−0.02 (0.19)	−0.02 (0.2)
E. nitens	0.16 (0.30)	0.12 (0.27)	0.13 (0.28)
E. regnans	–	–	–
NW.SE	0.10 (0.20)	0.07 (0.19)	0.07 (0.19)
SE.SW	0.10 (0.21)	0.12 (0.19)	0.08 (0.20)
SW.NW	−0.53 (0.27)	−0.49 (0.25)	−0.53 (0.26)
SW.NE	–	–	–

Table 13: *Comparison of the estimated parameters for the Possum data set with their standard errors for the WMLE^{MH}, the MLE and Cantoni's estimator.*

the MLE and the Cantoni's estimator. The values within the parentheses are their standard errors. From Table 13 and basing on the confidence interval for each of explanatory variables, one can see that many of these variables can not be considered significant in the model. To reduce the number of these variables we apply a forward stepwise procedure on the MLE and the WMLE^{MH}. We keep the variable in the model if the p-value obtained at each step of procedure is less than 5%*. The results of this analysis are presented in Table 14. As we can see*

the model chosen by the maximum likelihood estimator and the robust models WMLE^{MH} are essentially similar.
There is high correlation between variable habitat and acacia (0.54). Keeping the both variable in the model yield the un significant p-value for the variable habitat. Therefore, one of these variables should be considered in the models. Here, we keep the variable habitat in the final models.

MLE			
Variable	Coefficient.	Standard error	P-value
Intercept	−0.756	0.193	0.0000
Stags	0.036	0.01	0.0002
Bark	0.039	0.012	0.0025
Habitat	0.109	0.029	0.0001
SW.NE	−0.586	0.207	0.0072
WMLE$^{\mathrm{MH}}$			
Variable	Coefficient.	Standard error	P-value
Intercept	−0.848	0.225	0.00016
Stags	0.032	0.011	0.0032
Bark	0.046	0.015	0.0016
Habitat	0.123	0.031	0.0002
SW.NE	−0.611	0.227	0.0047

Table 14: *The coefficients estimation and corresponding standard errors for final Poisson model, estimated by the MLE and WMLE^{MH} of Possum data set.*

The WMLE$^{\mathrm{MH}}$ is resistant against outliers. The robust estimators for the Poisson regression proposed in the literature are based on modifications of the log-likelihood. They introduce a weight function, which limits the influence of bad leverage points and the outliers. For $\mathbf{x}_i$-values at the edge of the design space and for the high residuals a lower weight is proposed.
However with this simple weight, similar robust estimates discussed in

the literature typically have weights that depend on the couple (x_i, y_i) rather than on $\mu_i = h(\eta_i)$ alone. This new estimator WMLE^{MH} can be adopted more than the other robust estimator because of its simple weight function and ease of computation.

Conclusion and Outlook

In this thesis we have studied the robust inference in generalized linear models, particularly in logistic and Poisson regression.
Existing robust theory for the regression and generalized linear models have been presented in Chapter 2. We have seen that the least absolute-deviations estimator (L_1-norm) is an alternative to least squares in the regression models. In Chapter 3 we have studied the robust behaviors of the L_1-norm for generalized linear models. The general L_q quasi-likelihood estimator has been examined in detail. Their influence function shows that the L_q quasi-likelihood estimator is not sensitive to outliers and even leverage points with $q = 1$ for the canonical link functions. It has pointed out that these estimates have the computational difficulties in binary regression. In Chapter 4 we have introduced a new robust estimator in binary regression. This estimator is based on the minimum absolute deviation, which is similar to the L_q quasi-likelihood estimate for logistic regression. The resulting estimating equation is obtained through a simple modification of the familiar maximum likelihood equation. This estimator can be considered as a weighted maximum likelihood with weights that depend only on μ and q.

Similar robust estimates discussed in the literature typically have weights that depend on the couple $(\mathbf{x}, y)$. Our weight function depends on μ alone. The presented idea have been generalized in Chapter 5 to Poisson regression. The efficiency and the influence function of this new weighted maximum likelihood estimator have been examined. Furthermore, this simple robust estimator shows results comparable to other robust estimator discussed in the literature.
The approach to adaptive the simple weighted maximum likelihood with the weight that depends only on μ in the generalized linear model would be a interesting subject to follow.

Appendix

Theoretical background

Algebra

Definition 1. *Summation by part*

$$\sum_{k=m}^{n} f_k(g_{k+1} - g_k) = \left[f_{n+1}g_{n+1} - f_m g_m\right] - \sum_{k=m}^{n} g_{k+1}(f_{k+1} - f_k) \quad (165)$$

The O_p and o_p notations are used to describe the relation between the sequences. **?** defined these notations as follows.

Asymptotic statistics

Definition 2. *Let $X_1, X_2, \ldots$ be a sequence of real valued random variables. If for any $\varepsilon > 0$, there exists a constant M such that*

$$P(|X_n| \geq M) \leq \varepsilon \quad n = 1, 2, \cdots$$

we write $X_n = O_p(1)$. Such a sequence is said to be bounded in probability . If $X_n \xrightarrow{p} 0$ as $n \to \infty$, we write $X_n = o_p(1)$, which means that

X_n converges in probability to 0. If $X_n = o_p(1)$ then $X_n = O_p(1)$ but not the inverse.

Definition 3. *Let $W_1, W_2, \ldots, X_1, X_2, \ldots, Y_1, Y_2, \ldots,$ and $Z_1, Z_2, \ldots$ denote sequence real-valued random variables such that $Y_n > 0$ and $Z_n > 0$, $n = 1, 2, \ldots$.*

- *If $X_n = o_p(1)$ as $n \to \infty$, then $X_n = O_p(1)$ as $n \to \infty$.*
- *If $W_n = O_p(1)$ and $X_n = O_p(1)$ as $n \to \infty$, then $W_n + X_n = O_p(1)$ and $W_n X_n = O_p(1)$as $n \to \infty$; that is, $O_p(1) + O_p(1) = O_p(1)$ and $O_p(1)O_p(1) = O_p(1)$.*
- *If $W_n = O_p(1)$ and $X_n = o_p(1)$ as $n \to \infty$, then $W_n + X_n = O_p(1)$ and $W_n X_n = o_p(1)$as $n \to \infty$; that is, $O_p(1) + o_p(1) = O_p(1)$ and $O_p(1)o_p(1) = o_p(1)$.*
- *If $W_n = O_p(Z_n)$ and $X_n = O_p(Y_n)$ as $n \to \infty$, then*

  $$W_n X_n = O_p(Y_n Z_n) \text{ and } W_n + X_n = O_p(max(Z_n, Y_n)) \text{ as } n \to \infty;$$

  that is,

  $$O_p(Y_n) + O_p(Z_n) = O_p(max(Y_n, Z_n)) \text{ and } O_p(Y_n)O_p(Z_n) = O_p(Y_n Z_n).$$

This is summarized from **?** :

Definition 4. *Let X_n, X, Y_n and Y be random variables. Then*

- *$X_n \xrightarrow{a.s} X$ almost surely implies $X_n \xrightarrow{p} X$.*
- *$X_n \xrightarrow{p} X$ implies $X_n \xrightarrow{d} X$ in distribution.*
- *$X_n \xrightarrow{p} c$ for a constant c if and only if $X_n \xrightarrow{d} c$.*
- *if $X_n \xrightarrow{d} X$ and $||X_n - Y_n|| \xrightarrow{d} 0$, then $Y_n \xrightarrow{d} X$.*
- *$X_n \xrightarrow{p} X$ and $Y_n \xrightarrow{p} Y$ then $(X_n, Y_n) \xrightarrow{p} (X, Y)$ and $X_n + Y_n \xrightarrow{p} X + Y$.*

R Programs

The binary regression

Lq-quasi-likelihood estimator

```
#=============================================================
# File: Lq-helper-function.R
# The helper functions for Lq-bin-R
#
# Author: Sahar Hosseinian, EPFL (sahar.hosseinian@epfl.ch)
# Subject: Thesis "Robust Generalized Linear Models"
# Copyright 2008, 2009
#STAP/EPFL, Switzerland (http://stap.epfl.ch/)
#=============================================================

#-------------------------------------------------------------
# Function Vmu
#-------------------------------------------------------------
Vmu <- function(mu){return(mu*(1-mu))}

#-------------------------------------------------------------
# Function cb
# Calculating C the fisher consistency bias.
#-------------------------------------------------------------
cb <- function(ni,mu,ql)
{
        c.i<-c()
        m<-length(mu)
        for(i in 1:m)
        {
            densbin1 <- dbinom(ceiling(ni[i]*mu[i]):ni[i],
                               ni[i],mu[i])
            obsbin1 <- (ceiling(ni[i]*mu[i]):ni[i]-ni[i]*
                               mu[i])^(ql-1)
            densbin2 <- dbinom(0:(ceiling(ni[i]*mu[i])-1),
                               ni[i],mu[i])
            obsbin2 <- (ni[i]*mu[i]-0:(ceiling(ni[i]
```

```
                                        *mu[i])-1))^(ql-1)
            c.i[i]<-sum(densbin1*obsbin1)
                        - sum(densbin2*obsbin2)
        }
        round(c.i,4)
        if(ql==2)
        {
            c.i<-rep(0,length(mu))
        }
        return(c.i)
}
#-------------------------------------------------------------
# Function Wi
#-------------------------------------------------------------
Wi <- function (X,y,ql,ccc,mu,ni,eta,dinvlink)
{
        hprimVinverse <- (ni*dinvlink(eta))*((Vmu(mu)*ni))
                                ^(-ql/2)
        yminc <- ((abs(y-ni*mu))^(ql-1))*sign(y-ni*mu)-ccc
        res <- (hprimVinverse*yminc)/(y-ni*mu)
        res <- diag(res)
        return(res)
}
#-------------------------------------------------------------
# Function Ui
#-------------------------------------------------------------
Ui <- function (X,y,ni,mu,dinvlink,eta,ql,ccc)
{
        hprimVinverse <- (ni*dinvlink(eta))*((Vmu(mu)*ni))
                                ^(-ql/2)
        yminc <- ((abs(y-ni*mu))^(ql-1))*sign(y-ni*mu)-ccc
        Wsan <- diag((hprimVinverse*yminc)/(y-ni*mu))
        res<- t(X)%*%Wsan%*%matrix(y-ni*mu,ncol=1)
        return(res)
}
#-------------------------------------------------------------
# Function Mi
```

```
#-------------------------------------------------------------------
Mi <- function(ni,W,eta,dinvlink)
{
        res<-diag(dinvlink(eta))
        res<-res%*%W*ni
        return(res)
}

#-------------------------------------------------------------------
# File: Var-beta-Lq.R
# Find variance of beta of Lq-quasi-likelihood
#in binary regression.
#
# Author: Sahar Hosseinian, EPFL (sahar.hosseinian@epfl.ch)
# Subject: Thesis "Robust Generalized Linear Models"
# Copyright 2008, 2009
#STAP/EPFL, Switzerland (http://stap.epfl.ch/)
#-------------------------------------------------------------------

#-------------------------------------------------------------------
# Function Ri.i
#-------------------------------------------------------------------
Ri.i <- function(ni,mu,ql,ccc)
{
        R.i <- c()
        m <- length(mu)
        for (i in 1:m)
        {
            RES <-c()
            Comb <-c()
            for (j in 0:(ceiling(ni[i]*mu[i])-1))
            {
                Comb[j+1] <- factorial(ni[i])
                             /(factorial(ni[i]-j)
                             *factorial(j))
                RES[j+1] <- Comb[j+1]*(mu[i]^j)
                             *((1-mu[i])^(ni[i]-j))
                             *(ni[i]*mu[i]-j)^(2*ql-2)
```

```
            }
            for (k in (ceiling(ni[i]*mu[i])):ni[i])
            {
                Comb[k+1] <- factorial(ni[i])
                             /(factorial(ni[i]-k)
                             *factorial(k))
                RES[k+1] <- Comb[k+1]*(mu[i]^k)
                             *((1-mu[i])^(ni[i]-k))
                             *(k-ni[i]*mu[i])^(2*ql-2)
            }
            R.i[i]<- sum(RES)
        }
        return(R.i-ccc)
}

#-------------------------------------------------------------------
# Function Q4Var.i
#-------------------------------------------------------------------
Q4Var.i <- function(ni,mu,ql)
{
        if (ql==1)
        {
            Q.i <- -4* dbinom(round(ni*mu),ni,mu)
        }
        if (ql==2)
        {
            Q.i <- c(rep(1,length(mu)))
        }
        if(ql!=1 & ql!=2)
        {
            Q.i <- c()
            m <- length(mu)
            for (i in 1:m)
            {
                QRES <-c()
                QComb <-c()
                for (j in 0:(ceiling(ni[i]*mu[i])-1))
```

```
                {
                    QComb[j+1] <- factorial(ni[i])
                                  /(factorial(ni[i]-j)
                                  * factorial(j))
                    QRES[j+1] <- QComb[j+1]*(mu[i]^j)
                                  *((1-mu[i])^(ni[i]-j))
                                  *(ni[i]*mu[i]-j)^(ql-2)
                }
                for (k in (ceiling(ni[i]*mu[i])):ni[i])
                {
                    QComb[k+1] <- factorial(ni[i])
                                  /(factorial(ni[i]-k)
                                  *factorial(k))
                    QRES[k+1] <- QComb[k+1]*(mu[i]^k)
                                  *((1-mu[i])^(ni[i]-k))
                                  *(k-ni[i]*mu[i])^(ql-2)
                }
                Q.i[i]<- (ql-1)*sum(QRES)
            }
        }
        return (Q.i)
}

#-------------------------------------------------------------------
# Function Dnp.matrix
#-------------------------------------------------------------------
Dnp.matrix <- function(ni,X,eta,dinvlink)
{
        diag((ni*dinvlink(eta)))%*%X
}

#-------------------------------------------------------------------
# Function Var.of.beta
#-------------------------------------------------------------------
Var.of.beta <- function(Dnp, Rii, Q4Var, ni, Vmu, mu, ql)
{
        DVQD <- t(Dnp)%*%(diag((Vmu(mu)*ni)^(-ql/2)))
```

```
                %*%(diag(Q4Var))%*%Dnp
        DVRVD <- t(Dnp)%*%(diag((Vmu(mu)*ni)^(-ql/2)))
                %*%(diag(Rii))
                %*% (diag((Vmu(mu)*ni)^(-ql/2)))%*%Dnp

        return(solve(DVQD)%*%DVRVD%*% solve(DVQD))
}

#=============================================================
# File: Optimize-w-inf.R
# Optimize the estimated parameter beta if weight function
# becomes Inf
#
# Author: Sahar Hosseinian, EPFL (sahar.hosseinian@epfl.ch)
# Subject: Thesis "Robust Generalized Linear Models"
# Copyright 2008, 2009
# STAP/EPFL, Switzerland (http://stap.epfl.ch/)
#=============================================================

if (error.ind==1 &ql==1)
{
    warning("W is Inf")
    len.co.old = length(co.old)
    # IC of vector beta=std*0.1
    radius = 0.1*sqrt(diag(Varbetta))
    max.steps = 1000
    # number of element *steps=maxsteps
    steps = max.steps ^ (1/len.co.old)
    delta = radius / steps
    beta.grad = co.old
    eta.grad <- as.vector(X%*%beta.grad)
    mu.grad <-invlink(eta.grad)
    ccc.grad <- cb(ni,mu.grad,ql=ql)
    W.grad <- Wi(X,yy,ccc,ql=ql,mu.grad,ni
                ,eta.grad,dinvlink)
    U.grad <- Ui(X,yy,ni,mu.grad,dinvlink
                ,eta,ql,ccc)
    beta.max = co.old
```

```
best.grad = gradold
for (i in 1:len.co.old)
{
    beta.grad[i] = co.old[i] - radius[i]/2
    beta.max[i] = co.old[i] + radius[i]/2
}
while (beta.grad[len.co.old] <= beta.max[len.co.old])
{
    # Loop to iteratively increase all components
    # of the vector
    # When one component reaches the maximum, it is
    # reset and the next component is increased
    for (j in 1:len.co.old)
    {
        beta.grad[j] = beta.grad[j] + delta[j]
        if (beta.grad[j] <= beta.max[j])
        {
            break
        }
        else if (j<len.co.old)
        {
            beta.grad[j] = beta.max[j] - radius[j]
        }
    }
    eta.grad <- as.vector(X%*%beta.grad)
    mu.grad <-invlink(eta.grad)
    ccc <- cb(ni,mu.grad,ql=ql)
    W.grad <- Wi(X,yy,ccc,ql=ql,mu.grad,ni
                 ,eta.grad,dinvlink)
    U.grad <- Ui(X,yy,ni,mu.grad,dinvlink
                 ,eta,ql,ccc)
    grad.grad<-sqrt(sum(U.grad^2))
    if (grad.grad < best.grad)
    {
        co.new = beta.grad
        best.grad = grad.grad
    }
```

```
} #while 2
if (grad.grad >= 0.0001)
{
    if (max(abs(beta.grad - beta.max))>=
        max(radius*0.99))
    {
        cat("A better estimate was found on
        the border of the search region","\n")
        # cat("Repeat algorithm around co.new
        # with radius=", radius,"\n")

        # CI  beta +/- std*0.1
        radius = 0.1*diag(sqrt(Varbetta))
        max.steps = 1000
        # number of element *steps=maxsteps
        steps = max.steps ^ (1/len.co.old)
        delta = radius / steps
        co.old = co.new
        beta.grad = co.old
        beta.max = co.old
        best.grad = best.grad
        for (i in 1:len.co.old)
        {
            beta.grad[i] = co.old[i] - radius[i]/2
            beta.max[i] = co.old[i] + radius[i]/2
        }
        while (beta.grad[len.co.old] <= beta.max
               [len.co.old])
        {
            # Loop to iteratively increase all
            # components of the vector
            # When one component reaches the
            # maximum, it is reset and the
            #next component is increased
            for (j in 1:len.co.old)
            {
                beta.grad[j]←beta.grad[j]+delta[j]
```

```
                if (beta.grad[j] <= beta.max[j])
                {
                    break
                }
                else if (j<len.co.old)
                {
                    beta.grad[j]←beta.max[j]-radius[j]
                }
            }
            eta.grad ← as.vector(X%*%beta.grad)
            mu.grad ←invlink(eta.grad)
            ccc ← cb(ni,mu.grad,ql=ql)
            W.grad ← Wi(X,yy,ccc,ql=ql,mu.grad,ni
                        ,eta.grad,dinvlink)
            U.grad ← Ui(X,yy,ni,mu.grad,dinvlink
                        ,eta,ql,ccc)
            grad.grad←sqrt(sum(U.grad^2))
            if (grad.grad < best.grad)
            {
                co.new = beta.grad
                best.grad = grad.grad
            }
        } #while 3
    }
    else
    { #if it is not precise
        cat("Insufficient precision","\n")
        cat("Repeat algorithm around co.new
        with radius=", delta,"\n")
        radius = delta
        max.steps = 1000
        steps = max.steps ^ (1/len.co.old)
        delta = radius / steps
        co.old = co.new
        beta.grad = co.old
        beta.max = co.old
        best.grad = best.grad
```

```
for (i in 1:len.co.old)
{
    beta.grad[i]←co.old[i]-radius[i]/2
    beta.max[i]←co.old[i]+radius[i]/2
}
while (beta.grad[len.co.old]<=beta.max
    [len.co.old])
{
    # Loop to iteratively increase all
    # components of the vector
    # When one component reaches the maximum,
    # it is reset and the next component
    # is increased
    for (j in 1:len.co.old)
    {
        beta.grad[j]←beta.grad[j]+delta[j]
        if (beta.grad[j] <= beta.max[j])
        {
            break
        }
        else if (j<len.co.old)
        {
            beta.grad[j]←beta.max[j]-radius[j]
        }
    }
    eta.grad ← as.vector(X%*%beta.grad)
    mu.grad ←invlink(eta.grad)
    ccc ← cb(ni,mu.grad,ql=ql)
    W.grad ← Wi(X,yy,ccc,ql=ql,mu.grad,ni
        ,eta.grad,dinvlink)
    U.grad ← Ui(X,yy,ni,mu.grad,dinvlink
        ,eta,ql,ccc)
    grad.grad←sqrt(sum(U.grad^2))
    if (grad.grad < best.grad)
    {
        co.new = beta.grad
        best.grad = grad.grad
```

```
                }
            } # while 4
        }# if it is not precise
    }# if gradient is not zero
}

#=============================================================
#=============================================================
# File: Lq-bin.R
# Find the Lq quasi-likelihood estimated parameter beta
#in binary regression.
#
# Author: Sahar Hosseinian, EPFL (sahar.hosseinian@epfl.ch)
# Subject: Thesis "Robust Generalized Linear Models"
# Copyright 2008, 2009
#STAP/EPFL, Switzerland (http://stap.epfl.ch/)
#=============================================================

ajustbin <- function(X,y,linkfunction,ql,tol=10^(-6))
{
    n <- dim(X)[1]
    if(linkfunction=="logit")
    {
        link     <- function(mu) {return(log(mu/(1-mu)))}
        invlink  <- function(eta){return(1/(1+exp(-eta)))}
        dinvlink <- function(eta) {return(exp(-eta)
                    /(1+exp(-eta))^2)}
    }
    if(linkfunction=="probit")
    {
        link     <- function(mu) {return(qnorm(mu))}
        invlink  <- function(eta) {return(pnorm(eta))}
        dinvlink <- function(eta) {return(dnorm(eta))}
    }
    if(linkfunction=="cloglog")
    {
        link     <- function(mu){return(log(-log(1-mu)))}
        invlink  <- function(eta){return(1-exp(-exp(eta)))}
```

```
      dinvlink <- function(eta){return((exp(-exp(eta)))
                  *(exp(eta)))}
}
if(linkfunction=="loglog")
{
      link     <- function(mu){return(-log(-log(mu)))}
      invlink  <- function(eta){return(exp(-exp(-eta)))}
      dinvlink <- function(eta){return((exp(-eta))
                  *(exp(-exp(-eta))))}
}
if (sum(linkfunction==c("logit","probit","cloglog",
    "loglog"))!=1)
{
    warning ("Link is not available for binomial,
    available link are logit, probit,loglog and cloglog.
    Canonical link is considered.")
    link <- function(mu){return(log(mu/(1-mu)))}
    invlink <- function(eta){return(1/(1+exp(-eta)))}
    dinvlink <- function(eta){return(exp(-eta)
                /(1+exp(-eta))^2)}
}

source("Lq_helper_function.R") # Helper functions
source("Var_beta_Lq.R") # Compute variance of beta

if(NCOL(y)==1)
{
    y<-cbind(y,1-y)
}
X<-X[!apply(y,1,sum)==0,]
y<-y[!apply(y,1,sum)==0,]
ni <- y[,1]+y[,2]
yy <- y[,1]
probab<-(yy+1)/(ni+2)
weight<-(probab)*(1-probab)
        /(ni*(dinvlink(link(probab)))^2)
```

```
# The starting value
XWXinv <- solve((t(X)%*%diag(weight)%*%X), tol=1e-700)
startingpoint1<- XWXinv%*%(t(X)%*%diag(weight)
                  %*%link(probab))
startingpoint2 <- as.matrix(glm(y~X[,-1],
                   family=binomial)$coef)
error <- 10
error.ind <- 0
loop <- 0
co.old <- startingpoint1
while (error > tol)
{
    loop <- loop+1
    b <- matrix(co.old,ncol=1)
    eta <- as.vector(X%*%b)
    mu <-invlink(eta)
    probhat<-yy/ni
    ccc <- cb(ni,mu,ql=ql)
    W <- Wi(X,yy,ccc,ql=ql,mu,ni,eta,dinvlink)

    if (sum((yy-ni*mu)==0)==1)
    {
        error.ind <-1
        break
    }

    U <- Ui(X,yy,ni,mu,dinvlink,eta,ql,ccc)
    grad <- sqrt(sum(U^2))
    M <- Mi(ni,W,eta,dinvlink)
    XMXinv <- solve((t(X)%*% M %*%X), tol=1e-700)
    XtWyminmu <- t(X)%*%W%*%matrix(yy-ni*mu,ncol=1)
    co.new <- b + XMXinv %*% XtWyminmu

    if(any(is.na(co.new)))
    {
        warning("alogorithem did not converge")
        return(list(coefficients=co.new))
```

```
    }

    #To solve the problem when new beta is far from old
    #beta
    step<-co.new-co.old
    if (any(step>=10))
    {
        m<-min(0.7*sqrt(sum(co.old^2)/sum((step)^2)),1)
        co.new<-co.old+m*step
    }
    error <- sum(abs(co.new-co.old)/abs(co.new))
    co.old <- co.new
    Wold <- W
    Mold <- M
    etaold <- eta
    muold <- mu
    ymuold <- yy-ni*muold
    Uold <- U
    gradold <- sqrt(sum(U^2))
} #while

# Compute variance
Rii <- Ri.i(ni,mu,ql,ccc)
Q4Var <- Q4Var.i(ni,mu,ql)
Dnp <- Dnp.matrix(ni,X,eta,dinvlink)
Varbetta <- Var.of.beta(Dnp,Rii,Q4Var,ni,Vmu,mu,ql)

# Optimize estimate if weight-function is Inf
source("optimize-w-inf.R")

if(error.ind==1 & ql==1)
{
    gradian.print<-best.grad
}
else
{
    gradian.print <-gradold
```

```
    }

    number.of.loop <-loop
    eta.last <- as.vector(X%*%co.new)
    mu.last <-invlink(eta.last)
    ccc.last <- cb(ni,mu.last,ql=ql)
    W.last <- Wi(X,yy,ccc.last,ql=ql,mu.last,ni
                 ,eta.last,dinvlink)
    Rii.last <- Ri.i(ni,mu.last,ql,ccc.last)
    Q4Var.last <- Q4Var.i(ni,mu.last,ql)
    Dnp.last <- Dnp.matrix(ni,X,eta.last,dinvlink)
    Varbetta.last <- Var.of.beta(Dnp.last,Rii.last
                      ,Q4Var.last,ni,Vmu,mu.last,ql)
    Standard.err <- sqrt(diag(Varbetta.last))

    return(list(coefficients=co.new,
                gradient=gradian.print,
                iteration=number.of.loop,
                fitted.values=mu.last,
                Var.beta=Varbetta.last,
                St.err=Standard.err))
}
```

The BLq estimator

```
#=============================================================
# File: MCD-new-data.R
# This function uses mcd to find the robust mahalanobis
#distance and give the new datalist without outliers to
#have a better initial beta.
#
# Author: Sahar Hosseinian, EPFL (sahar.hosseinian@epfl.ch)
# Subject: Thesis "Robust Generalized Linear Models"
# Copyright 2008, 2009
#STAP/EPFL, Switzerland (http://stap.epfl.ch/)
#=============================================================

library(robustbase)
```

```
#-------------------------------------------------------------------
# Function mcd_mahalanobis
#-------------------------------------------------------------------
mcd_mahalanobis<- function(X)
{
    p <- dim(X)[2]
    X_mcd <- X[,2:p]
    mu_mcd <- covMcd(X_mcd)$center
    cov_mcd <- covMcd(X_mcd)$cov
    x.mu_mcd <- X_mcd-mu_mcd
    # Find just the diagonal elements
    dist_mcd <- rowSums((x.mu_mcd%*%solve(cov_mcd
                , tol=1e-700))*(x.mu_mcd))
    return(dist_mcd)
}

#-------------------------------------------------------------------
# Function New_data_mahalanobis
#-------------------------------------------------------------------
New_data_mahalanobis <- function(X,y)
{
    Xmah_mcd <- mcd_mahalanobis(X)
    data_mcd <- cbind(y,X)
    new_data_mcd <- data_mcd[(Xmah_mcd < quantile((Xmah_mcd)
                    ,prob=0.87)),]
    return(new_data_mcd)
}

#===================================================================
# File: Var-beta-BLq.R
# Find variance of BLq estimator in binary regression.
#
# Author: Sahar Hosseinian, EPFL (sahar.hosseinian@epfl.ch)
# Subject: Thesis "Robust Generalized Linear Models"
# Copyright 2008, 2009
#STAP/EPFL, Switzerland (http://stap.epfl.ch/)
#===================================================================
#-------------------------------------------------------------------
```

```
# Function Ri.i
#-------------------------------------------------------------------
Ri.i <- function(ni,mu,ql)
{
    R.i<- ni*mu*(1-mu)*(mu^(ql-1)+(1-mu)^(ql-1))^2
     return(R.i)
}

#-------------------------------------------------------------------
# Function Q4Var.i
#-------------------------------------------------------------------
Q4Var.i <- function(mu,ql)
{
    if (ql==2)
    {
        Q.i <- c(rep(1,length(mu)))
    }
    else
    {
        Q.i<-(mu^(ql-1)+(1-mu)^(ql-1))
    }
    return (Q.i)
}

#-------------------------------------------------------------------
# Function Dnp.matrix
#-------------------------------------------------------------------
Dnp.matrix <- function(ni,X,eta,dinvlink)
{
    res<-diag(ni*dinvlink(eta))
    res<-res%*%X
    return(res)
}

#-------------------------------------------------------------------
# Function Var.of.beta
#-------------------------------------------------------------------
```

```
Var.of.beta <- function(Dnp, Rii, Q4Var, ni, Vmu, mu, ql)
{
    DVQD <- t(Dnp)%*%(diag((Vmu(mu)*ni)^(-ql/2)))
            %*%(diag(Q4Var))%*%Dnp
    DVRVD <- t(Dnp)%*%(diag((Vmu(mu)*ni)^(-ql/2)))
             %*% (diag(Rii))%*%(diag((Vmu(mu)*ni)
             ^(-ql/2)))%*%Dnp
    return(solve(DVQD)%*%DVRVD%*% solve(DVQD))
}

#=============================================================
# File: BLq-bin.R
# Find the BLq estimated parameter beta in binary regression.
#
# Author: Sahar Hosseinian, EPFL (sahar.hosseinian@epfl.ch)
# Subject: Thesis "Robust Generalized Linear Models"
# Copyright 2008, 2009
#STAP/EPFL, Switzerland (http://stap.epfl.ch/)
#=============================================================

#-------------------------------------------------------------
# Function ajustbinY01q
#-------------------------------------------------------------
ajustbinY01q <- function(X,y,ql,tol=10^(-6))
{
    n <- dim(X)[1]
    #---------------------------------------------------------
    # Helper functions
    #---------------------------------------------------------
    normf    <- function(A) {sqrt(sum (A^2))}
    link     <- function(mu) {return(log(mu/(1-mu)))}
    invlink  <- function(eta){return(1/(1+exp(-eta)))}
    dinvlink <- function(eta) {return(exp(-eta)
                /(1+exp(-eta))^2)}

    Vmu <- function(mu){return(mu*(1-mu))}
    Wi <- function (mu,ql)
    {
```

```
    res1 <- (mu*(1-mu))^((2-ql)/2)
    res2 <- (mu^(ql-1)+(1-mu)^(ql-1))
    res<- diag(res1*res2)
    return(res)
}

Ui <- function (X,y,W,mu,ni)
{
    res<- (t(X)%*%W%*%matrix(y-ni*mu,ncol=1))
    return(res)
}

Mi <- function(W,eta,dinvlink,ni)
{
    res<-diag(ni*dinvlink(eta))
    res<-res%*%W
    return(res)
}
# Find new data list without outliers
source("MCD_new_data.R")
# Compute variance of beta
source("Var_beta_BLq.R")
#-------------------------------------------------------------------------

if(NCOL(y)==1)
{
    y<-cbind(y,1-y)
}
X<-X[!apply(y,1,sum)==0,]
y<-y[!apply(y,1,sum)==0,]
ni <- y[,1]+y[,2]
yy<-y[,1]
startingpoint2 <- as.matrix(glm(yy~X[,-1]
                    ,family=binomial)$coef)
newdata<- New_data_mahalanobis(X,yy)
yf <- newdata[,1]
p <- dim(X)[2]
```

```
xf ←newdata[,3:(p+1)]
startingpoint4 ←as.matrix(glm(yf~xf
                  ,family=binomial)$coef)
error ← 10
error.ind ← 0
loop ← 0
if(ql==2 |ql==1.9)
{
    co.old ← startingpoint2
}
else
{
    co.old ← startingpoint4
}

CC=0
while (error > tol)
{
    loop ← loop+1
    if (loop>500)
    {
        co.old ← startingpoint2
        loop ←1
    }
    b ← matrix(co.old ,ncol=1)
    eta ← as.vector(X%*%b)
    mu ←invlink(eta)
    W ← Wi(mu, ql)
    U ← Ui(X,yy,W,mu, ni)
    grad ← sqrt(sum(U^2))
    dinv ← dinvlink(eta)
    M ← Mi(W, eta , dinvlink , ni)
    XMXinv ← solve((t(X)%*% M %*%X), tol=1e-700)
    XtWyminmu ← t(X)%*%W%*%matrix(yy-ni*mu,ncol=1)
    co.new ← b + XMXinv %*% XtWyminmu

    if(any(is.na(co.new)))
```

```
    {
        warning("algorithm did not converge1")
        return(list(coefficients=co.new) )
    }
    #To solve the problem when new beta is far
    #from old beta
    stepp←co.new−co.old
    if (any(abs(stepp)>=10))
    {
        m←min(0.7*sqrt(sum(co.old^2)
            /sum((stepp)^2)),1)
        co.new←co.old+m*stepp
    }
    error ← sum(abs(co.new−co.old)/abs(co.new))


    if (normf(co.new)>normf(co.old))
    {
        CC=CC+1
    }
    if (normf(co.new)<normf(co.old))
    {
        CC=0
    }
    if (CC==200)
    {
        warning("algorithm did not converge2")
        return(list(coefficients=matrix(,,ncol=1
                        ,nrow=length(co.old)),
                        Var.beta=matrix(,,ncol=1
                        ,nrow=length(co.old))))
    }

    co.old ← co.new

}#while
```

```
    eta.last ← as.vector(X%*%co.new)
    mu.last ←invlink(eta.last)
    W.last ← Wi(mu.last,ql)
    U.last ← Ui(X,yy,W.last,mu.last,ni)

    #-------------------------------------------------------------
    # Compute variance of beta
    #-------------------------------------------------------------
    Rii.last ← Ri.i(ni,mu.last,ql)
    Q4Var.last ← Q4Var.i(mu.last,ql)
    Dnp.last ← Dnp.matrix(ni,X,eta.last,dinvlink)
    Varbetta.last ← Var.of.beta(Dnp.last,Rii.last
                    ,Q4Var.last,ni,Vmu,mu.last,ql)
    Standard.err ← sqrt(diag(Varbetta.last))

    return(list(coefficients=co.new,
                Var.beta=matrix(diag(Varbetta.last)
                        ,ncol=1,nrow=length(co.old)),
                gradient=grad.last,
                fitted.values=mu.last,
                Var.beta=Varbetta.last,
                St.err=Standard.err))
}
```

The case of multiple roots

```
#=============================================================
# File: BLq-beta-initial.R
# Find the BLq estimated parameter from the initial beta
#in binary regression.
#
# Author: Sahar Hosseinian, EPFL (sahar.hosseinian@epfl.ch)
# Subject: Thesis "Robust Generalized Linear Models"
# Copyright 2008, 2009
#STAP/EPFL, Switzerland (http://stap.epfl.ch/)
#=============================================================

F_beta_initial ← function(X,y,ql,beta_initial,tol=10^(-6))
```

```
{
    n ← dim(X)[1]

    #------------------------------------------------------------
    # Helper functions
    #------------------------------------------------------------
    normf    ← function(A) { sqrt(sum (A^2)) }
    link     ← function(mu) {return(log(mu/(1-mu)))}
    invlink  ← function(eta){return(1/(1+exp(-eta)))}
    dinvlink ← function(eta) {return(exp(-eta)
                /(1+exp(-eta))^2)}

    Vmu ← function(mu){return(mu*(1-mu))}

    Wi ← function (mu, ql)
    {
        res1 ← (mu*(1-mu))^((2-ql)/2)
        res2 ← (mu^(ql-1)+(1-mu)^(ql-1))
        res ←diag(res1*res2)
        return(res)
    }

    Ui ← function (X,y,W,mu, ni)
    {
        res← (t(X)%*%W%*%matrix(y-ni*mu, ncol=1))
        return(res)
    }

    Mi ← function(W, eta, dinvlink, ni)
    {
        res←diag(ni*dinvlink(eta))
        res←res%*%W
        return(res)
    }
    #------------------------------------------------------------

    if(NCOL(y)==1)
```

```
{
    y<-cbind(y,1-y)
}
X<-X[!apply(y,1,sum)==0,]
y<-y[!apply(y,1,sum)==0,]
ni <- y[,1]+y[,2]
yy<-y[,1]
error <- 10
loop <- 0
co.old <- beta_initial
CC=0

while (error > tol)
{
    loop <- loop+1
    if (loop>2000)
    {
        warning("algorithm did not converge-loop")
        return(list(coefficients=matrix(,,ncol=1
        ,nrow=length(co.old))))

    }
    b <- matrix(co.old,ncol=1)
    eta <- as.vector(X%*%b)
    mu <-invlink(eta)
    W <- Wi(mu, ql)
    U <- Ui(X,yy,W,mu,ni)
    grad <- sqrt(sum(U^2))
    dinv <- dinvlink(eta)
    M <- Mi(W,eta,dinvlink,ni)
    XMXinv <- solve((t(X)%*% M %*%X), tol=1e-700)
    XtWyminmu <- t(X)%*%W%*%matrix(yy-ni*mu,ncol=1)
    co.new <- b + XMXinv %*% XtWyminmu
    if(any(is.na(co.new)))
    {
        warning("algorithm did not converge")
        return(list(coefficients=co.new))
```

```
        }
        #To solve the problem when new beta is far
        #from old beta
        stepp<-co.new-co.old
        if (any(abs(stepp)>=10))
        {
            m<-min(0.7*sqrt(sum(co.old^2)/sum((stepp)^2)),1)
            co.new<-co.old+m*stepp
        }

        error <- sum(abs(co.new-co.old)/abs(co.new))

        if (normf(co.new)>normf(co.old))
        {
            CC=CC+1
        }
        if (normf(co.new)<normf(co.old))
        {
            CC=0
        }
        if (CC==200)
        {
            warning("algorithm did not converge")
            return(list(coefficients=matrix(,,ncol=1
            ,nrow=length(co.old))))
        }
        co.old <- co.new
    }#while
    return(list(coefficients=co.new))
}

#=============================================================
# File: BLq-multi-root.R
# Estimate multiple roots for the BLq quasi-likelihood
#function. Generate 2*2^p -1 initial beta, which are
#computed by glm from original data and MCD new dataset.
#The beta with the smallest residual is given as the
#best beta.
```

```
#
# Author: Sahar Hosseinian, EPFL (sahar.hosseinian@epfl.ch)
# Subject: Thesis "Robust Generalized Linear Models"
# Copyright 2008, 2009
#STAP/EPFL, Switzerland (http://stap.epfl.ch/)
#=============================================================
#
# Inputs:
#   beta_0: initial estimate
#   tol: minimum required distance for new beta from
#   old betas
#   X, y, ql and linkfunction for
#=============================================================

# Program Bernouli to find beta
source("BLq-beta-initial.R")

#-------------------------------------------------------------
# Helper functions for beta_initial_sign
#-------------------------------------------------------------
normf    <- function(A) {sqrt(sum (A^2))}
link     <- function(mu) {return(log(mu/(1-mu)))}
invlink  <- function(eta){return(1/(1+exp(-eta)))}
dinvlink <- function(eta) {return(exp(-eta)
            /(1+exp(-eta))^2)}

#-------------------------------------------------------------
# Function beta_initial_sign
#-------------------------------------------------------------
beta_initial_sign <- function(X , y, ql ,beta_0,beta_1
                      , tol=10^(-6))
{
        p=length(beta_0)
        rep_1 <- array(0, c(p, 2^p))

        for (i in 1:p)
        {
```

```
    rep_1[i,] <- rep(c(1,-1), times=2^(p-i)
    , each=2^(i-1))
}
beta_0_sign <- cbind((as.vector(beta_0)*rep_1),
                     (as.vector(beta_1)*rep_1))
new_estimate <- matrix(,,ncol=dim(beta_0_sign)[2],
                       nrow=dim(beta_0_sign)[1])

for(k in 1:dim(beta_0_sign)[2])
{
    new_estimate[,k] <- F_beta_initial(X,y,ql
                       ,beta_0_sign[,k],10^(-6))$coef
}
eta_estimate <- X%*%new_estimate
mu_estimate <- invlink(eta_estimate)
final_table <- cbind(y, mu_estimate)
norm_final <- c()

for( j in 1:(2*(2^p)))
{
    norm_final[j] <- normf(new_estimate[,j])
}
residual_final <- y-mu_estimate
V_mu_estimate <- mu_estimate*(1-mu_estimate)
Pearson_residual_final <- residual_final
                          / sqrt(V_mu_estimate)
norm_final_pearson <- c()
norm_final_residual <- c()

for( j in 1:(2*(2^p)))
{
    norm_final_pearson[j] <-
                normf(Pearson_residual_final[,j])
    norm_final_residual[j] <-
                normf(residual_final[,j])
}
min_norm_final_residual<-
```

```
                                apply(as.matrix(norm_final_residual)
                                ,2,min)
        all_estimate_min_res <- as.matrix(
                                new_estimate[,min_norm_final_residual
                                ==norm_final_residual])
        best_coef<- as.matrix(all_estimate_min_res[,1])
        return(list(coefficients=new_estimate,
                Mu_esimates=final_table,
                norm=norm_final,
                resf=residual_final,
                presf=Pearson_residual_final,
                norm.pearson=norm_final_pearson,
                norm.residual=norm_final_residual,
                bestcoef=best_coef))
}
```

The glm weight function algorithm

```
#=============================================================
# File: all-weight-binary.R
# Find four weight functions in the binary regression.
#
# Author: Sahar Hosseinian, EPFL (sahar.hosseinian@epfl.ch)
# Subject: Thesis "Robust Generalized Linear Models"
# Copyright 2008, 2009
#STAP/EPFL, Switzerland (http://stap.epfl.ch/)
#=============================================================



#=============================================================
# Function Wi
# Find the weight function for BLq estimator
# in binary regression, the simplified Lq.
#-------------------------------------------------------------
Wi <- function (mu, ql)
{
    res1 <- (mu*(1-mu))^((2-ql)/2)
```

```
    res2 <- (mu^(ql-1)+(1-mu)^(ql-1))
    res<- (res1*res2)
    return(res)
}
#=============================================
# Function N_Wmu
# Find the firts new weight function in binary regression.
#
#---------------------------------------------
# Function Imu
#---------------------------------------------
Imu <-function(mu)
{
    res<-c()
    res[mu>=0.5]=1
    res[mu<0.5]=0
    return(res)
}
#---------------------------------------------
# Function FU
#---------------------------------------------
FU <- function(mu,cc)
{
    res <- (1-2*cc)*Imu(mu)+cc
    return(res)
}
#---------------------------------------------
# Function Wmu_1
#---------------------------------------------
Wmu_1<- function(mu,ql)
{
    res <-((mu*(1-mu))^((2-ql)/2))*(mu^(ql-1)+ (1-mu)^(ql-1)
    return(res)
}
#---------------------------------------------
# Function Wmu_2
#---------------------------------------------
```

```
Wmu_2 <-function(mu, ql, cc, epsilon)
{
        u<-FU(mu, cc)
        W1<-Wmu_1((cc+epsilon), ql)
        res1 <- W1/(-epsilon)
        res2 <-  sign(mu-0.5)*(mu-u)
        return(res1*res2)
}
#-------------------------------------------------------------------
# Function Wmu_3
#-------------------------------------------------------------------
Wmu_3 <-0
#-------------------------------------------------------------------
# Function N_Wmu
#-------------------------------------------------------------------
N_Wmu <- function(mu, ql, cc, epsilon)
{
    W<-c()
    c2 <- (1-cc)-epsilon-0.5
    c1 <- 1-cc-0.5
    W[abs(mu-0.5)<c2]<- ((mu[abs(mu-0.5)<c2]
                         *(1-mu[abs(mu-0.5)<c2]))^((2-ql)/2))
                         *(mu[abs(mu-0.5)<c2]^(ql-1)
                         +(1-mu[abs(mu-0.5)<c2])^(ql-1))
    W[abs(mu-0.5)>c1]<-0
    if(any(abs(mu-0.5)>c2 & abs(mu-0.5)<c1))
    {
        W[abs(mu-0.5)>c2 & abs(mu-0.5)<c1]<-
                         Wmu_2(mu[abs(mu-0.5)>c2
                         & abs(mu-0.5)<c1], ql, cc, epsilon)
    }
    return(diag(W))
}
#-------------------------------------------------------------------
# Function S_Wmu
# Find the second new weight function in binary regression.
#-------------------------------------------------------------------
```

```
SN_Wmu <- function(mu, ql)
{
    res<-(1-(2*mu-1)^2)^ql
    return(res)
}
#=============================================================
# Function WT
# Find the third new weight function in binary regression.
#-------------------------------------------------------------
WT<- function(mu, ql, c2)
{
   Wmu_TNB <-c()
   n <- length(mu)
   for (i in 1:n)
   {
      if (abs(mu[i]-0.5)<c2)
      {
         Wmu_TNB[i] <- ((mu[i]*(1-mu[i]))^((2-ql)/2))
                       * (mu[i]^(ql-1)+ (1-mu[i])^(ql-1))
      }
      if (abs(mu[i]-0.5)>c2)
      {
         Wmu_TNB[i] <- (1-(2*mu[i]-1)^2)^ql
      }
   }
   return(Wmu_TNB)
}
#=============================================================

#=============================================================
# File: glm_weighted_BLq.R
# GLM weighted algorithm to find the WMLE for different
#weight functions in binary regression.
# wf is the weight function can be selected from
#wf==Ber, wf==N-Ber, wf=SN-Ber or wf==TN-Ber
#
# Author: Sahar Hosseinian, EPFL (sahar.hosseinian@epfl.ch)
# Subject: Thesis "Robust Generalized Linear Models"
```

```
# Copyright 2008, 2009
#STAP/EPFL, Switzerland (http://stap.epfl.ch/)
#-------------------------------------------------------------------

#-------------------------------------------------------------------
glm.ajust.weight<- function(X,y,ql,wf)
{
    n <- dim(X)[1]

    #-------------------------------------------------------------------
    # Helper functions
    #-------------------------------------------------------------------
    normf    <- function(A) { sqrt(sum (A^2)) }
    link     <- function(mu) {return(log(mu/(1-mu)))}
    invlink  <- function(eta){return(1/(1+exp(-eta)))}
    dinvlink <- function(eta) {return(exp(-eta)
                /(1+exp(-eta))^2)}

    # Find the weight functions
    source("all_weight_binary.R")
    # Find new data list without outliers
    source("MCD_new_data.R")
    #-------------------------------------------------------------------

    if(NCOL(y)==1)
    {
        y<-cbind(y,1-y)
    }
    X<-X[!apply(y,1,sum)==0,]
    y<-y[!apply(y,1,sum)==0,]
    ni <- y[,1]+y[,2]
    yy<-y[,1]
    newdata<- New_data_mahalanobis(X,yy)
    yf <- newdata[,1]
    p <- dim(X)[2]
    xf <-newdata[,3:(p+1)]
    startingbeta1 <-as.matrix(glm(yf~xf
```

```
                    ,family=binomial)$coef)
startingbeta2 <- as.matrix(glm(y~X[,-1]
                    ,family=binomial)$coef)

if(ql==2 |ql==1.9)
{
    co.old <- startingbeta2
}
else
{
    co.old <- startingbeta2
}

error <-10
loop <-0
CC=0

while(error >10^(-6))
{
    loop <- loop+1
    if (loop>1500)
    {
        co.old <- startingbeta2
        loop <-1
    }
    b <- matrix(co.old,ncol=1)
    eta <- as.vector(X%*%b)
    mu <-invlink(eta)
    if (wf=="Ber")  { ww <- (Wi(mu,ql))}
    if (wf=="N-Ber"){ ww <- diag(N_Wmu(mu,ql,0.1,0.1))}
    if (wf=="SN-Ber"){ ww <- SN_Wmu(mu,ql)}
    if (wf=="TN-Ber"){ ww <- WT(mu,ql,0.2)}

    if(all(ww==0))
    {
        warning("algorithm did not converge, ww =0")
        return(list(coefficients=matrix
```

```
            (,,ncol=1,nrow=length(co.old)),
            Var.beta=matrix(,,ncol=1,nrow=length(co.old))))

        }

        co.new <- matrix(glm(yy~X[,-1]
                    ,weights=ww,family=binomial)$coef,ncol=1)

        #To solve the problem when new beta is far
        #from old beta
        stepp<-co.new-co.old
        if ((any(abs(stepp)>=10)))
        {
            m<-min(0.7*sqrt(sum(co.old^2)/sum((stepp)^2)),1)
            co.new<-co.old+m*stepp
        }

        if (normf(co.new)>normf(co.old))
        {
            CC=CC+1
        }
        if (normf(co.new)<normf(co.old))
        {
            CC=0
        }

        if( (CC==50))
        {
            warning("algorithm did not converge2")
            return(list(coefficients=matrix(,,ncol=1
            ,nrow=length(co.old)),
            Var.beta=matrix(,,ncol=1,nrow=length(co.old))))
        }
        error <- sum(abs(co.new-co.old)/abs(co.new))
        co.old <- co.new
    }
    return(list(coefficients=co.new))
```

}

The Poisson regression

The L_1 quasi-likelihood estimator

```
#=============================================================
# File: L1-Poisson.R
# Find the L1 estimated parameter beta in Poisson regression.
#
# Author: Sahar Hosseinian, EPFL (sahar.hosseinian@epfl.ch)
# Subject: Thesis "Robust Generalized Linear Models"
# Copyright 2008, 2009
#STAP/EPFL, Switzerland (http://stap.epfl.ch/)
#=============================================================

#-------------------------------------------------------------
# Function ajust.poiss.Lq
#-------------------------------------------------------------
ajust.poiss.Lq <- function(X,y,tol=10^(-6))
{
    n <- dim(X)[1]
    #---------------------------------------------------------
    # Helper functions
    #---------------------------------------------------------
    link     <- function(mu) {return(log(mu))}
    invlink  <- function(eta){return(exp(eta))}
    dinvlink <- function(eta){return(exp(eta))}
    Vmu <- function(mu){return(mu)}
    Pmu <- function (mu)
    {
        P<-c()
        for (i in 1:n)
        {
            P[i]<-sum(dpois(0:(floor(mu[i])),mu[i]))
        }
        return(P)
    }
    Pmu_2<-function(mu,y)
```

```
{
P_mu<-Pmu(mu)
P<-c()
for (i in 1:n)
{
    if(y[i]<=mu[i]){P[i]=2*P_mu[i]}
    if(y[i]>mu[i]){P[i]=2*P_mu[i]-2}
}
return((P))
}
Wi2 <-function(mu,y,P)
{
    V<-(Vmu(mu))
    res<- (V^(1/2)%*%(P))/(y-mu)
    return(diag(res))
}
Wi <- function (mu,y,eta)
{
    V <- (Vmu(mu))^(1/2)
    ccc<-1-2*Pmu(mu)
    yminc <- sign(y-mu)-ccc
    res <- (V*yminc)/(y-mu)
    res <- diag(res)
    return(res)
}
Ui <- function (X,y,W,mu)
{
    res<- (t(X)%*%W%*%matrix(y-mu,ncol=1))
    return(res)
}
Mi <- function(mu,eta)
{
    res<-diag(dinvlink(eta))
    res<-res%*%W
    return(res)
}
#-----------------------------------------------------------------
```

```
    startingpoint2 ← as.matrix(glm(y~X[,-1]
                        ,family=poisson)$coef)
    error ← 10
    loop ← 0
    co.old ← startingpoint2
    while (error > tol)
    {
        loop ← loop+1
        b ← matrix(co.old ,ncol=1)
        eta ← as.vector(X%*%b)
        mu ←invlink(eta)
        P ← Pmu_2(mu,y)
        W ← Wi(mu,y,P)
        U ← Ui(X,y,W,mu)
        grad ← sqrt(sum(U^2))
        M ← Mi(mu,eta)
        XMXinv ← solve((t(X)%*% M %*%X), tol=1e-700)
        XtWyminmu ← t(X)%*%W%*%matrix(y-mu,ncol=1)
        co.new ← b + XMXinv %*% U
        error ← sum(abs(co.new-co.old)/abs(co.new))
        co.old ← co.new
    }#while
    return(list(coefficients=co.new))
}
```

The WMLE^{MH} estimator

```
#-------------------------------------------------------------------
# File: Var-beta-WMLE-MH.R
# Find variance of WMLE-MH in Poisson regression.
#
# Author: Sahar Hosseinian, EPFL (sahar.hosseinian@epfl.ch)
# Subject: Thesis "Robust Generalized Linear Models"
# Copyright 2008, 2009
#STAP/EPFL, Switzerland (http://stap.epfl.ch/)
#-------------------------------------------------------------------
#-------------------------------------------------------------------
# Function Ri.i
```

```
#-------------------------------------------------------------------------
Ri.i <- function(mu)
{
    m <-median(mu)
    W<- diag(Wi(mu,m))
    R.i<- W^2*mu
    return(R.i)
}

#-------------------------------------------------------------------------
# Function Q4Var.i
#-------------------------------------------------------------------------
Q4Var.i <- function(mu)
{
    m <-median(mu)
    Q.i<- diag(Wi(mu,m))
    return (Q.i)
}

#-------------------------------------------------------------------------
# Function Dnp.matrix
#-------------------------------------------------------------------------
Dnp.matrix <- function(X, eta, dinvlink)
{
    res<-diag(dinvlink(eta))
    res<-res%*%X
    return(res)
}

#-------------------------------------------------------------------------
# Function Var.of.beta
#-------------------------------------------------------------------------
Var.of.beta <- function(Dnp, Rii, Q4Var, Vmu, mu)
{
    DVQD <- t(Dnp)%*%(diag((Vmu(mu))^(-1)))%*%(diag(Q4Var))
            %*%Dnp
    DVRVD <- t(Dnp)%*%(diag((Vmu(mu))^(-1)))%*%(diag(Rii))
```

```
                %*%(diag((Vmu(mu))^(-1)))%*%Dnp
    return(solve(DVQD)%*%DVRVD%*% solve(DVQD))
}

#=============================================================
# File: WMLE-MH-Poisson.R
# Find the WMLE-MH estimated parameter beta
# in Poisson regression.
#
# Author: Sahar Hosseinian, EPFL (sahar.hosseinian@epfl.ch)
# Subject: Thesis "Robust Generalized Linear Models"
# Copyright 2008, 2009
# STAP/EPFL, Switzerland (http://stap.epfl.ch/)
#=============================================================

#-------------------------------------------------------------
# Function ajust.poisson.NR
#-------------------------------------------------------------
ajust.poisson.NR <- function(X,y,tol=10^(-6))
{
    n <- dim(X)[1]
    #---------------------------------------------------------
    # Helper functions
    #---------------------------------------------------------
    link     <- function(mu) {return(log(mu))}
    invlink  <- function(eta){return(exp(eta))}
    dinvlink <- function(eta){return(exp(eta))}
    Vmu <- function(mu){return(mu)}
    Wi <-function(mu,m)
    {
        a<-numeric(0)
        for(i in 1:length(mu))
        {
        res <-0
        if(mu[i]>2*m & mu[i]<3*m){res<- (3*m-mu[i])/m}
        if(mu[i]<2*m & mu[i]>(m/2)){res<-1}
        if(mu[i]<(m/2)&mu[i]>0){res<-2*mu[i]/m}
        if(mu[i]>3*m&mu[i]<0){res<-0}
```

```
      a<-c(a,res)
      }
 return(diag(a))
}
Ui <- function (X,y,W,mu)
{
     res<- (t(X)%*%W%*%matrix(y-mu,ncol=1))
     return(res)
}
Mi <- function(W,eta,dinvlink)
{
     res<-diag(dinvlink(eta))
     res<-res%*%W
     return(res)
}
# Find new data list without outliers
source("MCD_new_data.R")
# Compute variance of beta
source("Var_beta_WMLE_MH.R")
#-------------------------------------------------------------------
startingpoint2 <- as.matrix(glm(y~X[,-1]
                    ,family=poisson)$coef)
newdata<- New_data_mahalanobis(X,y)
yf <- newdata[,1]
 p <- dim(X)[2]
 xf <-newdata[,3:(p+1)]
startingpoint4 <- as.matrix(glm(yf~xf
                    ,family=poisson)$coef)
co.old <- startingpoint4
error <- 10
loop <- 0
while (error > tol)
{
     loop <- loop+1
     if (loop>500)
     {
          co.old <- startingpoint2
```

```
        loop <-1
    }
    b <- matrix(co.old,ncol=1)
    eta <- as.vector(X%*%b)
    mu <-invlink(eta)
    m<-median(mu)
    W <- Wi(mu,m)
    U <- Ui(X,y,W,mu)
    grad <- sqrt(sum(U^2))
    dinv <- dinvlink(eta)
    M <- Mi(W,eta,dinvlink)
    XMXinv <- solve((t(X)%*% M %*%X), tol=1e-700)
    XtWyminmu <- t(X)%*%W%*%matrix(y-mu,ncol=1)
    co.new <- b + XMXinv %*% XtWyminmu
    if(any(is.na(co.new)))
    {
        warning("algorithm did not converge")
        return(list(coefficients=co.new) )
    }
    #To solve the problem when new beta is far
    #from old beta
    stepp<-co.new-co.old
    if (any(abs(stepp)>=10))
    {
        mm<-min(0.7*sqrt(sum(co.old^2)
                /sum((stepp)^2)),1)
        co.new<-co.old+mm*stepp
    }
    error <- sum(abs(co.new-co.old)/abs(co.new))
    co.old <- co.new
    www<-diag(W)
}#while
eta.last <- as.vector(X%*%co.new)
mu.last <-invlink(eta.last)

#-----------------------------------------------------------
# Compute variance of beta
```

```
    #-------------------------------------------------------------------
    Rii.last <- Ri.i(mu.last)
    Q4Var.last <- Q4Var.i(mu.last)
    Dnp.last <- Dnp.matrix(X, eta.last, dinvlink)
    Varbetta.last <- Var.of.beta(Dnp.last, Rii.last
                        , Q4Var.last, Vmu, mu.last)
    Standard.err <- sqrt(diag(Varbetta.last))
    return(list(coefficients=co.new, St.err=Standard.err))
}
```

Bibliography

de Vijver M. J. et al., V. (2002) A gene expression signature as a predictor of survival in breast cancer. *Ann. Statist.* **347**(25), 1999–2009.

Bianco, A. M. and Yohai, V. J. (1996) Robust estimation in the logistic regression model. In *Robust statistics, data analysis, and computer intensive methods (Schloss Thurnau, 1994)*, volume 109 of *Lecture Notes in Statist.*, pp. 17–34. New York: Springer.

Bloomfield, P. and Steiger, W. L. (1983) *Least absolute deviations.* Volume 6 of *Progress in Probability and Statistics.* Boston, MA: Birkhäuser Boston Inc. ISBN 0-8176-3157-7. Theory, applications, and algorithms.

Cantoni, E. and Ronchetti, E. (2001) Robust inference for generalized linear models. *J. Amer. Statist. Assoc.* **96**(455), 1022–1030.

Carroll, R. J. and Pederson, S. (1993) On robustness in the logistic regression model. *J. Roy. Statist. Soc. Ser. B* **55**(3), 693–706.

Cook, R. D. and Weisberg, S. (1982) *Residuals and influence in regression*. Monographs on Statistics and Applied Probability. London: Chapman & Hall. ISBN 0-412-24280-X.

Copas, J. B. (1988) Binary regression models for contaminated data. *J. Roy. Statist. Soc. Ser. B* **50**(2), 225–265. With discussion.

Croux, C., Flandre, C. and Haesbroeck, G. (2002) The breakdown behavior of the maximum likelihood estimator in the logistic regression model. *Statist. Probab. Lett.* **60**(4), 377–386.

Croux, C. and Haesbroeck, G. (2003) Implementing the Bianco and Yohai estimator for logistic regression. *Comput. Statist. Data Anal.* **44**(1-2), 273–295. Special issue in honour of Stan Azen: a birthday celebration.

Dobson, A. J. (2002) *An introduction to generalized linear models*. Second edition. Chapman & Hall/CRC Texts in Statistical Science Series. Chapman & Hall/CRC, Boca Raton, FL. ISBN 1-58488-165-8.

Donoho, D. and Huber, P. J. (1983) The notion of breakdown point. In *A Festschrift for Erich L. Lehmann*, Wadsworth Statist./Probab. Ser., pp. 157–184. Belmont, CA: Wadsworth.

Finney, D. J. (1947) The estimation from individual records of relationship between dose and quantal response. *Biometrika* **34**(3/4), 320–334.

Hampel, F. R. (1971) A general qualitative definition of robustness. *Ann. Math. Statist.* **42**, 1887–1896.

Hampel, F. R. (1974) The influence curve and its role in robust estimation. *J. Amer. Statist. Assoc.* **69**, 383–393.

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J. and Stahel, W. A. (1986) *Robust statistics*. Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics. New York: John Wiley & Sons Inc. The approach based on influence functions.

Huber, P. J. (1964) Robust estimation of a location parameter. *Ann. Math. Statist.* **35**, 73–101.

Huber, P. J. (1967) The behavior of maximum likelihood estimates under nonstandard conditions. In *Proc. Fifth Berkeley Sympos. Math. Statist. and Probability (Berkeley, Calif., 1965/66), Vol. I: Statistics*, pp. 221–233. Berkeley, Calif.: Univ. California Press.

Huber, P. J. (1973) Robust regression: asymptotics, conjectures and Monte Carlo. *Ann. Statist.* **1**, 799–821.

Huber, P. J. (1981) *Robust statistics*. New York: John Wiley & Sons Inc. Wiley Series in Probability and Mathematical Statistics.

Jurecková, J. and Sen, P. K. (1996) Asymptotic representations and interrelations of robust estimators and their applications. In *Robust inference*, volume 15 of *Handbook of Statist.*, pp. 467–512. Amsterdam: North-Holland.

Krasker, W. S. and Welsch, R. E. (1982) Efficient bounded-influence regression estimation. *J. Amer. Statist. Assoc.* **77**(379), 595–604.

Künsch, H. R., Stefanski, L. A. and Carroll, R. J. (1989) Conditionally unbiased bounded-influence estimation in general regression models, with applications to generalized linear models. *J. Amer. Statist. Assoc.* **84**(406), 460–466.

Lindsay, B. G. (1994) Efficiency versus robustness: the case for minimum Hellinger distance and related methods. *Ann. Statist.* **22**(2), 1081–1114.

Lindsay, D. B.G., C. R. B. T. M. T. S. A. P. and Nix, H. A. (1990) The conservation of arboreal marsupials in the montane ash forests of the central highlands of victoria, south-east australia: I. factors influencing the occupancy of trees with hollows. *Biological Conservation* **54**, 111–131.

Mallows, C. L. (1975) On some topics in robustness,unpoblished memorandum, bell telephone laboratories murray hill, nj. .

Markatou, M., Basu, A. and Lindsay, B. (1997) Weighted likelihood estimating equations: the discrete case with applications to logistic regression. *J. Statist. Plann. Inference* **57**(2), 215–232. Robust statistics and data analysis, II.

Maronna, R. A., Martin, R. D. and Yohai, V. J. (2006) *Robust statistics*. Wiley Series in Probability and Statistics. Chichester: John Wiley & Sons Ltd. ISBN 978-0-470-01092-1; 0-470-01092-4. Theory and methods.

McCullagh, P. and Nelder, J. A. (1983) *Generalized linear models*. Monographs on Statistics and Applied Probability. London: Chapman & Hall. ISBN 0-412-23850-0.

Morgenthaler, S. (1992) Least-absolute-deviations fits for generalized linear models. *Biometrika* **79**(4), 747–754.

Myers, R. H., Montgomery, D. C. and Vining, G. G. (2002) *Generalized linear models*. Wiley Series in Probability and Statistics. New York: Wiley-Interscience [John Wiley & Sons]. ISBN 0-471-35573-9. With applications in engineering and the sciences.

Nelder, J. A. and Wedderburn, R. W. M. (1972) Generalized linear models. *JRSS* **135**(3), 370–384.

Pregibon, D. (1981) Logistic regression diagnostics. *Ann. Statist.* **9**(4), 705–724.

Pregibon, D. (1982) Resistant fits for some commonly used logistic models with medical applications. *Biometrics* **38**(2), 485–498.

Rousseeuw, P. (1985) Multivariate estimation with high breakdown point. In *Mathematical statistics and applications, Vol. B (Bad Tatzmannsdorf, 1983)*, pp. 283–297. Dordrecht: Reidel.

Rousseeuw, P. J. (1984) Least median of squares regression. *J. Amer. Statist. Assoc.* **79**(388), 871–880.

Rousseeuw, P. J. and Leroy, A. M. (1987) *Robust regression and outlier detection*. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. New York: John Wiley & Sons Inc. ISBN 0-471-85233-3.

Ruckstuhl, A. F. and Welsh, A. H. (2001) Robust fitting of the binomial model. *Ann. Statist.* **29**(4), 1117–1136.

Stefanski, L. A., Carroll, R. J. and Ruppert, D. (1986a) Optimally bounded score functions for generalized linear models with applications to logistic regression. *Biometrika* **73**(2), 413–424.

Stefanski, L. A., Carroll, R. J. and Ruppert, D. (1986b) Optimally bounded score functions for generalized linear models with applications to logistic regression. *Biometrika* **73**(2), 413–424.

Stigler, S. M. (1981) Gauss and the invention of least squares. *Ann. Statist.* **9**(3), 465–474.

Stigler, S. M. (1986) *The history of statistics*. Cambridge, MA: The Belknap Press of Harvard University Press. ISBN 0-674-40340-1. The measurement of uncertainty before 1900.

Thall, P. F. and Vail, S. C. (1990) Some covariance models for longitudinal count data with overdispersion. *Biometrics* **46**(3), 657–671.

Tukey, J. W. (1970-71) *Exploratory Data Analysis*. Addision-Wesley: Preliminary edition.

Wedderburn, R. W. M. (1974) Quasi-likelihood functions, generalized linear models, and the Gauss-Newton method. *Biometrika* **61**, 439–447.

Wedderburn, R. W. M. (1976) On the existence and uniqueness of the maximum likelihood estimates for certain generalized linear models. *Biometrika* **63**(1), 27–32.

SAHAR HOSSEINIAN

Chaire de Statistique Appliquée
Département de Mathématiques
Ecole Polytechnique Fédéral de Lausanne
CH-1015 Lausanne
Sahar.hosseinian@epfl.ch
Born on March 21, 1976 in Iran

My goal is to make use of my theoretical knowledge in statistics applied to biostatistics. I feel that my strong points are my ambition and my curiosity to learn.

EDUCATION

2004- Present **PhD student**, EPFL Thesis title: "Robust Statistics for generalized linear models", with Prof. S. Morgenthaler.

2001-2003 **Postgrad in Statistics,** University of Neuchâtel, Switzerland.

1994 -1999 **Bachelor's Degree in Statistics,** University of Shahid Beheshti, Tehran, Iran.

1990 -1994 **High School Diploma in Mathematics-Physics,** Aboureihan High-School, Tehran, Iran.

2003 -2004 **Certificate French Language and Civilization,** University of Neuchâtel, Switzerland (500 study hours).

PROFESSIONAL EXPERIENCE

3.2009- **Bio-Statistician**, Diagnoplex, Epalinges, Switzerland

7.2004-3.2009 **Teaching and Research Assistant**, Prof. S. Morgenthaler, Chair of Applied Statistics, EPFL.

- Specialization in robust generalized linear models and their applications.
- Implementation of statistical models in R.
- Analysis of data from healthcare studies, especially clinical trials.
- Analysis of genetics data and fitting of carcinogens models.
- Supervision of student projects on mathematical risk theory and insurance models.
- Responsible teaching assistant for the courses: "Statistics and probability for life science" and "Genetics data analysis". Development of exercises to allow students to understand statistical models by applying them to biological studies.

2003 -2004 **Teaching and Research Assistant (40%)**, Prof. Y. Tillé**,** Applied Statistics, Neuchâtel University

- Responsible for the course: "Statistics for psychology and social science".
- Survey of new methods in sampling theory.
- Implementation of statistical models in SPSS.

2001-2003 **Research Assistant (40%),** Prof. Y. Dodge**,** Statistics Group, Neuchâtel University

- Specialization in linear regression models and their applications.

- Responsible for the statistical project "Gulliver 2002" during Expo02: National study to analyze the views of Swiss citizens about Switzerland. Preparation of the survey, evaluation and comparison of the outcome with results of "Gulliver 1964".
- Participation in the organization of conferences and conventions.

7.2003- 9.2003 **Statistical analysis of customer satisfaction,** Orange Switzerland

2000-2001 **Executive assistant of the CEO of Kayson Constructions,** Tehran, Iran
- Preparation of agreements with international partners (France, Russia, Japan).
- Interface with the different departments of Kayson Constructions.

1999 – 2000 **Marketing Assistant,** Siemens Corporation, Tehran, Iran
- Communication with business partners, customer support, sales assistant at expositions.

CONTRIBUTED CONFERENCE PRESENTATIONS

S. Hosseinian and S. Morgenthaler, "Robustness in Generalized linear models", International Conference on Robust Statistics 2006, Lisbon, Portugal.
S. Hosseinian and S. Morgenthaler, "The L_1 Estimate in Generalized Linear Models", 39èmes Journées de Statistique 2007, Angers, France.
S. Hosseinian and S. Morgenthaler, "Weighted Maximum Likelihood Estimates in Logistic Regression", International Conference on Robust Statistics 2008, Antalya, Turkey.

COMPUTER SKILLS

R, S-plus, MINITAB, SPAD, SPSS (Statistical Software), Microsoft Office, LaTeX, Windows, Linux

LANGUAGES

Persian (Native Language) - French/ English (Fluent) - German/Arabic (Acquaintance)

OTHER INTERESTS

Water-polo, Salsa, Ski, Tennis, Travel, Reading

REFERENCES

Available upon request