KU LEUVEN

G0B16
ROBUST STATISTICS

Project 2. Mallow's type M-estimator of (Cantoni and Ronchetti, 2001)

Dzmitry Sei (r0913246)

Supervised by:
Prof. Mia Hubert
Prof. Stefan Van Aelst

June 6, 2023

FACULTY OF SCIENCE

1 Introduction

This report presents the development of a robust classifier capable of accurately predicting class labels. To achieve this, we employ a logistic regression model with Mallow's Type M-Estimator, as introduced by Cantoni and Ronchetti in 2001. By leveraging this classifier, we address the challenges posed by outliers and data discrepancies, making it suitable for datasets containing noisy or inconsistent samples.

2 Limitations of the Maximum Likelihood Approach

The influence function of logistic regression with the standard Maximum Likelihood Estimator (MLE) for the coefficients β is determined by the following equation:

$$IF(y,x,\hat{\beta}) = J(\hat{\beta})^{-1}(y-\mu(\hat{\beta}^t x))x^t$$

where $J(\hat{\beta})$ is Fisher information matrix $E[(y-\mu)^2 x^t x]$. The influence function is unbounded in the design space (x) and bounded in the response space (y). While extreme values in the design space can have a significant impact on the estimator, the effect of errors in the response variable (misclassification errors) is not as clear. Studies conducted by Copas (1988) and Pregibon (1982) have demonstrated that misclassification errors can introduce bias into the MLE. Consequently, it is crucial to employ robust estimators that are less affected by deviations in both the predictor variables (x) and the responses (y)[4].

3 Mallow's estimator GLM approach

3.1 Mallow estimator

Logistic regression is a GLM for binary classification. The estimation of β in a generalized linear model (GLM) with non-ordinary variance is achieved through the utilization of the quasi-likelihood method.

$$\sum_{i=1}^{n} (\frac{y_i - \mu_i}{\phi \upsilon_{\mu_i}}) \mu_i^{'} = 0, \tag{1}$$

where $var(y_i) = \phi \upsilon_{\mu_i}$. **In the context of Bernoulli logistic distribution (in our case) $\phi = 1$.** Non-robustness of this approach for logistic regression was shown in the previous section.

The Mallow's estimator is a robust version of 1, Mallow's estimator modifies the quasi-likelihood method by down-weighting the influence of outliers or influential observations, making it more resistant to deviations from the assumed model structure.

Given the Pearson residuals $r_i = \frac{y_i - \mu_i}{\sqrt{\phi v_{\mu_i}}}$, the M-estimating equations for β of a GLM from exponential family is given by the with the following formula

$$\sum_{i=1}^{n} \left[\psi(r_i;\beta,\phi,c)\omega(x_i) r_i \frac{1}{\sqrt{\phi v_{\mu_i}}} \mu_i^{'} - \alpha(\beta) \right] = 0, \tag{2}$$

where $\mu_i^{'} = \frac{\partial \mu_i}{\partial \beta}$ and $\alpha(\beta)$ is a correction term to ensure Fisher consistency. It is computed explicitly in [2].

The function $\psi(r_i;\beta,\phi,c)$ and the weights $\omega(x_i)$ introduce new ingredients in comparison to the classical GLM estimators obtained through maximum quasi-likelihood, where the estimating equations 1 is derived using $\psi(r_i;\beta,\phi,c) = r_i$ and $\omega(x_i) = 1$ for all i. $\psi(r_i;\beta,\phi,c)$ is added to control deviations in the y-space and the weights $\omega(x_i)$ to down weight leverage points.

2 can be rewritten as

$$\sum_{i=1}^{n} \left[\tilde{\omega}(r_i;\beta,\phi,c)\omega(x_i) r_i \frac{1}{\sqrt{\phi v_{\mu_i}}} \mu_i^{'} - \alpha(\beta) \right], \tag{3}$$

where $\tilde{\omega}(r_i;\beta,\phi,c) = \psi(r;\beta,\psi,c)/r$. Weights $\tilde{\omega}$ and ω can be used for diagnostics of outliers and leverage points.

3.2 Choice of ψ and $\omega(x)$

3.2.1 Choice ψ

The function ψ plays a crucial role in controlling the impact of large residuals in the estimation process. It is essential for ψ to be bounded to effectively manage these residuals. Common choices for ψ include functions that level off, such as the Huber function, or functions that exhibit a redescending. The selection of ψ is often accompanied by tuning it with a constant c. This constant is typically chosen to ensure a desired level of asymptotic efficiency in the estimation procedure.

Determining the value of c for a desired efficiency level in GLM models is more complex than in linear regression due to the influence of design and lack of general results. The estimated efficiency can be inspected afterward, allowing for model refitting with different c values if needed.

In this project Huber ψ function is used. **The value of c equal to** 1.8 **was empirically obtained as the optimal choice.**

3.2.2 Choice of $\omega(x_i)$

Robust estimators in linear models suggest a specific approach for determining the weights $\omega(x_i)$. One simple method is to set $\omega(x_i) = \sqrt{(1-h_{ii})}$, where h_{ii} represents the leverage of observation x_i. In the current implementation of the *robustbase* package, besides the default equal weights $\omega(x_i) = 1$ for all i), an alternative option is available. This option lets users choose weights based on the Mahalanobis distances d_i [3].

$$\omega(x_i) = \frac{1}{\sqrt{1 + 8max(0,(d_i^2 - q))/\sqrt{2q}}}$$

On the test data, the h_{ii} based approach shows better performance. That is why we use it during the model-building process.

3.3 β distribution and significance test

According to [2] $\hat{\beta} \sim N(\beta, \widehat{Var})$, where $\widehat{Var})$ is estimated using $\hat{\beta}$.This allows testing the significance of a coefficient estimation, using z-statistics, namely

$$z - statistic = \hat{\beta}_j / \sqrt{\widehat{Var}}. \tag{4}$$

Using 4 p-value for a coefficient can be calculated.

4 Robust Estimation of the Full Model

In this section, we construct a robust classifier that utilizes all the covariates. Next, we compare the performance of this estimator with logistic regression. Then, we employ the weights of residuals and data points obtained during the training of the robust classifier to diagnose robust outliers and leverage points. Only the train dataset is used for model building.

4.1 Model building

On the 1 the result of training is shown. As we can see models disagree about the significance of *MCoreJ* and *AreaJ*.

	Non-robust_Estimate	Non-robust_Std	Non-robust_z_value	Non-robust_p	Non-robust_Is_significant	Robust_Estimate	Robust_Std	Robust_z_value	Robust_p	Robust_Is_significant
(Intercept)	"-55.87147"	"6.63705"	"-8.41812"	"0"	"T"	"-68.36468"	"8.2599"	"-8.2767"	"0"	"T"
MAperF	"-0.00202"	"6e-04"	"-3.38985"	"7e-04"	"T"	"-0.00231"	"0.00066"	"-3.51509"	"0.00044"	"T"
MTotF	"-0.00095"	"0.00081"	"-1.16654"	"0.24339"	"F"	"-0.00099"	"0.00089"	"-1.10737"	"0.26814"	"F"
MCoreF	"0.00387"	"0.00084"	"4.63331"	"0"	"T"	"0.00462"	"0.00093"	"4.98971"	"0"	"T"
AreaF	"3e-05"	"1e-05"	"3.9214"	"9e-05"	"T"	"5e-05"	"1e-05"	"4.68839"	"0"	"T"
IR2F	"-0.00082"	"0.00055"	"-1.47984"	"0.13892"	"F"	"-0.00114"	"0.00061"	"-1.87961"	"0.06016"	"F"
csfF	"0.00108"	"0.00019"	"5.81414"	"0"	"T"	"0.00121"	"0.00021"	"5.76337"	"0"	"T"
EllipF	"0.00501"	"0.00115"	"4.34108"	"1e-05"	"T"	"0.00488"	"0.00127"	"3.84779"	"0.00012"	"T"
MAperJ	"-0.00028"	"0.00042"	"-0.67285"	"0.50104"	"F"	"-0.00036"	"0.00046"	"-0.80013"	"0.42363"	"F"
MTotJ	"-0.00047"	"0.00075"	"-0.62949"	"0.52903"	"F"	"-0.00028"	"0.00081"	"-0.34836"	"0.72757"	"F"
MCoreJ	"0.00139"	"0.00064"	"2.15855"	"0.03089"	"T"	"0.00118"	"0.00069"	"1.69918"	"0.08929"	"F"
AreaJ	"-1e-05"	"1e-05"	"-0.80245"	"0.42229"	"F"	"-2e-05"	"1e-05"	"-2.13316"	"0.03291"	"T"
IR2J	"0.00102"	"0.00065"	"1.57519"	"0.11521"	"F"	"0.00123"	"0.00072"	"1.70082"	"0.08898"	"F"
csfJ	"0.00039"	"0.00018"	"2.10226"	"0.03553"	"T"	"0.00051"	"2e-04"	"2.55792"	"0.01053"	"T"
EllipJ	"0.00013"	"0.00101"	"0.12507"	"0.90047"	"F"	"-4e-05"	"0.00109"	"-0.03855"	"0.96925"	"F"
MAperN	"0.00019"	"0.00037"	"0.5238"	"0.60042"	"F"	"4e-05"	"0.00039"	"0.1079"	"0.91408"	"F"
MTotN	"-0.00026"	"0.00066"	"-0.39712"	"0.69128"	"F"	"0.00022"	"0.00072"	"0.30455"	"0.76071"	"F"
MCoreN	"0.00089"	"0.00059"	"1.52033"	"0.12843"	"F"	"0.00086"	"0.00063"	"1.37421"	"0.16938"	"F"
AreaN	"0"	"1e-05"	"-0.01807"	"0.98558"	"F"	"2e-05"	"2e-05"	"1.03853"	"0.29902"	"F"
IR2N	"0.00055"	"5e-04"	"1.09743"	"0.27245"	"F"	"0.00052"	"0.00053"	"0.98371"	"0.32526"	"F"
csfN	"0.00025"	"0.00018"	"1.34416"	"0.1789"	"F"	"0.00025"	"2e-04"	"1.2666"	"0.2053"	"F"
EllipN	"-0.00017"	"0.00087"	"-0.20011"	"0.8414"	"F"	"-0.00055"	"0.00093"	"-0.58819"	"0.5564"	"F"

Figure 1: Trained models summary

Figure 2 depicts the fractional relationship between the non-robust and robust estimators. As we can non-robust method underestimates standard error. The observed phenomenon can be attributed to the inherent characteristics of the M-estimator and the non-robust estimator. The M-estimator, specifically designed to exhibit robustness against outliers, employs a strategy that reduces the influence of extreme observations. Consequently, this down weighting of extreme observations results in estimated coefficients with larger standard errors. The M-estimator achieves this robustness by being less reliant on individual observations and their potential impact on the estimated coefficients. In contrast, the non-robust estimator treats all observations equally, including outliers. This approach assigns equal weight to every data point, regardless of its magnitude or potential to distort the estimation. Consequently, if outliers exert a substantial impact on the estimated coefficients, the non-robust estimator can yield smaller standard errors due to the lack of down weighting.

The presence of outliers in the data significantly also impacts the observed effect of a *AreaN* coefficient, leading to pronounced disparities.

	Non-rob. Estimate/rob Estimate	Non-rob. Std. Error/rob. Std. Error
(Intercept)	0.81725631	0.8035271
MAperF	0.87191832	0.9041326
MTotF	0.95624617	0.9077376
MCoreF	0.83810500	0.9025718
AreaF	0.68475559	0.8186886
IR2F	0.71901639	0.9132517
csfF	0.89701895	0.8891861
EllipF	1.02566227	0.9091137
MAperJ	0.77960192	0.9270831
MTotJ	1.66042454	0.9188808
MCoreJ	1.17951369	0.9284968
AreaJ	0.34151095	0.9078398
IR2J	0.82815775	0.8942091
csfJ	0.75041473	0.9130624
EllipJ	-3.00395984	0.9258334
MAperN	4.55255634	0.9377782
MTotN	-1.20155574	0.9214555
MCoreN	1.03900002	0.9391429
AreaN	-0.01513909	0.8701363
IR2N	1.05598202	0.9465593
csfN	0.97328597	0.9171235
EllipN	0.31638574	0.9299707

Figure 2: Non-robust estimations/Robust estimation

4.2 Outlier detection

As mentioned before outlier detection is based on examining $\tilde{\omega}$ (w_r)and ω (w_x). But it should be noted that there is no single approach (especially considering that $\tilde{\omega}$ depends on tuning parameter c).

For instance, in the context of *glmrob*, an observation may be detected as an outlier if $|w_r w_x| < 0.0001$. On the other hand [1] proposed an outlier detection method where an observation is flagged as an outlier if its weight, represented by w_r, falls below a specific threshold, such as 0.5.

The approach based on [1] finds 16 outliers (see the code for the exact outliers array) in the train data.3 highlights the outliers.All the dots under the 0.5 threshold are considered extreme.

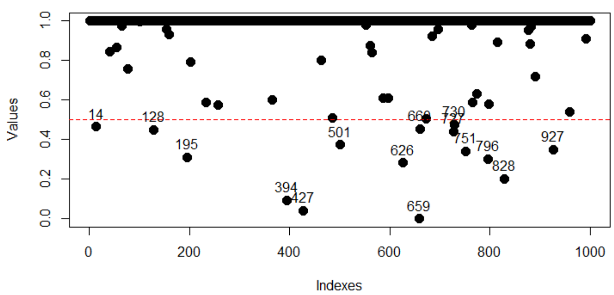

Figure 3: Full model outliers

5 Robust variable selection

According to [2], 2 can be seen as the derivative with respect to β of the robust quasi-likelihood function $\sum Q_M(y_i, \mu_i)$. This allows us to use the robust version of the deviance statistics.

$$\Lambda_{Q_M} = 2\left[\sum Q_M(y_i, \hat{\mu}_i) - \sum Q_M(y_i, \dot{\mu}_i)\right],$$

where $\hat{\mu}_i$ and $\dot{\mu}_i$ are the estimators under model M_p and nested model M_{p-q} respectively.

The quasi-deviance statistic can be used in an ANOVA test to compare two nested *glmrob* models.

Utilizing ANOVA we reduced the full model to the following:

$$logit(p_i) = \beta_0 + \beta_1 MCoreF + \beta_2 AreaF + \beta_3 IR2F + \beta_4 csfF + \beta_5 EllipF + \beta_6 AreaJ + \beta_7 IR2J + \beta_8 csfJ \tag{5}$$

5 (in robust and in non-robust context) is referred to as a truncated model In the text below.

6 Performance on Test Data

The algorithm, utilized for detecting the best-performing model and selecting the optimal cutoff point for separating stars from galaxies using predicted probabilities, is as follows:

1. Build a Receiver Operating Characteristic (ROC) curve for each available model.
2. Determine the cutoff point for each model based on the constructed ROC curve.
3. The best model is determined by selecting the one with the highest Area Under the Curve (AUC) value.

We obtained the following results.

Model	**AUC**	**Cutoff Point**
Full Robust	*0.9951*	0.354
Full Non-robust	0.9949	0.27
Truncated Robust	0.9948	0.284
Truncated Non-robust	0.9946	0.336

Based on the analysis, it can be observed that the Robust full model performs the best among the models, despite having similar AUC values. Furthermore, it is noteworthy that the Full robust model has the largest cutoff point compared to the other models.

The predicted probabilities for the test set are stored in the variable *rob_preds* in the provided code.

References

[1] Eva Cantoni and Elvezio Ronchetti. “A robust approach for skewed and heavy-tailed outcomes in the analysis of health care expenditures”. In: *Journal of Health Economics* 25.2 (2006), pp. 198–213.

[2] Eva Cantoni and Elvezio Ronchetti. “Robust inference for generalized linear models”. In: *Journal of the American Statistical Association* 96.455 (2001), pp. 1022–1030.

[3] Stephane Heritier et al. *Robust methods in biostatistics*. John Wiley & Sons, 2009.

[4] Maria-Pia Victoria-Feser. “Robust logistic regression for binomial responses”. In: *Available at SSRN 1763301* (2000).