Electronic Journal of Applied Statistical Analysis
EJASA, Electron. J. App. Stat. Anal.
http://siba-ese.unisalento.it/index.php/ejasa/index
e-ISSN: 2070-5948
DOI: 10.1285/i20705948v13n2p390

Computationally efficient univariate filtering for massive data
By Tsagris, Alenazi, Fafalios

Published: 14 October 2020

Electronic Journal of Applied Statistical Analysis
Vol. 13, Issue 02, October 2020, 390-412
DOI: 10.1285/i20705948v13n2p390

Computationally efficient univariate filtering for massive data

Tsagris Michail*[a], Alenazi Abdulaziz[b], and Fafalios Stefanos[c]

[a] *University of Crete, Department of Economics, Gallos Campus, Rethymnon, Greece,*
[b] *Northern Border University, Department of Mathematics, Arar, Saudi Arabia*
[c] *University of Crete, Department of Computer Science, Voutes Campus, Herakleion, Greece,*

Published: 14 October 2020

The vast availability of massive (or large scale) and big data has increased the computational cost of data analysis. One such case is the computational cost of the univariate filtering that typically involves fitting many univariate regression models and is essential for numerous variable selection algorithms to reduce the number of predictor variables. The paper manifests how to dramatically reduce that computational cost by employing the score test or the simple Pearson correlation. Extensive Monte Carlo simulation studies will demonstrate their advantages and disadvantages compared to the likelihood ratio test and examples with real data will illustrate the performance of the score test and the log-likelihood ratio test under realistic scenarios. Depending on the regression model used, the score test is $30 - 6,000$ times faster than the log-likelihood ratio test and produces nearly the same results. Hence this paper strongly recommends to substitute the log-likelihood ratio test with the score test for the task of univariate filtering when coping with massive data, big data, or even data whose sample size is in the order of a few tens of thousands or higher.

keywords: Univariate filtering, computational efficiency, score test, likelihood ratio tests, high-dimensional data.

1 Introduction

Massive or large scale data, which require high computing power, have become a frequent phenomenon nowadays. Reducing the computational cost entailed by massive data, using

*Corresponding author: mtsagris@uoc.gr

ISSN: 2070-5948
http://siba-ese.unisalento.it/index.php/ejasa/index

computationally efficient algorithms, is beneficiary for research and industry related purposes. In bionformatics for instance, analysis of numerous gene expression data that contain 55,000 variables and in computer science, analysis of big data[1] (order of Terabytes and higher) are common tasks. Computationally efficient algorithms are also highly desirable and required by banks, large scale institutions and companies that handle big data because those algorithms not only reduce the waiting time but further have an economic impact since they can reduce electricity expenses.

A common task met in both research and industry is variable selection (VS), described as follows. When a response variable Y (for example a phenotype, disease status, survival time) is given along with a set $\mathbf{X}$ of d predictor (or independent) variables, both consisting of n observations, VS attempts to identify the minimal set of predictor variables whose predictive capability on the response is optimal. In bioinformatics for instance, the goal is to identify the genes whose expression levels allow for early diagnosis of some disease Tsamardinos and Aliferis (2003).

Over the years, there has been an accumulation of VS algorithms in many data science fields, such as bioinformatics, statistics, machine learning, and signal processing. Most algorithms tackle the VS problem from an agglomerative, forward selection perspective. They commence with an empty set of variables and move forward by adding one or more variables at each time. Statistically Equivalent Signatures (Lagani et al., 2017), Forward Backward with Early Dropping (Borboudakis and Tsamardinos, 2019), Orthogonal Matching Pursuit (Chen et al., 1989; Pati et al., 1993; Davis et al., 1994), Sure Independence Screening (Fan and Lv, 2008; Fan and Song, 2010), forward selection (Weisberg, 1980) and forward stepwise regression (Weisberg, 1980) are some examples of VS algorithms that begin with univariate filtering. At that filtering step the most statistically significant variable, or the variable mostly correlated with the response is detected, while significant variables or a fraction of the most significant variables are retained for further analysis.

Univariate filtering with continuous responses is fast enough because of the fast implementation of the correlation between $\mathbf{y}$ and each of the $\mathbf{x}_i s$. With non-continuous responses though (count data, nominal, ordinal, survival), d univariate regressions and hence d log-likelihood ratio tests must be performed. This can be computationally really heavy with tens of thousands of variables or even with large sample sizes (hundreds of thousands).

Statistical softwares, such as R, are not computationally efficient in fitting numerous regression models when built-in commands are applied, such as *glm* or any regression model offered by a package, inside a *for* loop. Self implementation of the regression models and employment of parallel computing can assist reduce the execution time in R. The same recipe can be applied with C++, resulting in higher computational gains[2]. This raises the question of whether univariate filtering can become more efficient or extremely efficient, and effectively reduce the computational cost of numerous VS algorithms. The

[1]The main difference between big data and massive data is that the first cannot fit to the hard drive of a conventional desktop.

[2]Numerous C++ regressions models can be found in the R package *Rfast* (Papadakis et al., 2019).

answer is yes, by employment of the score test or of the Pearson correlation coefficient that allows for extremely computationally efficient univariate filtering. Further, specifically for logistic regression, the Welch's t-test (Welch, 1951) is another possibility.

We must note though that R has numerous packages that perform fast VS, according to different algorithms, for instance *glmnet* (Friedman et al., 2010), *leaps* (Lumley, 2020) and *bestglm* (McLeod et al., 2020). The drawback of those packages is that they are devised for certain classes of regression models.

In this paper we demonstrate, via simulation studies and experiments with real data at which different types of regression models will be considered, the computational advantages of the score test when employed for univariate filtering. The score test, also known as Rao's test (Rao, 1948) or Lagrange Multiplier test (Greene, 2003), is robust in the sense that it does not depend on the functional relationship between the response and the predictor variable(s) and it depends on the null distribution of the response y only through the MLE of the distribution under the H_0 (Chen, 1983). It is asymptotically equivalent to the log-likelihood ratio test (Greene, 2003) and for logistic and Poisson regression its formula is similar to the Pearson correlation coefficient (Hosmer Jr et al., 2013). Both the score test and Pearson correlation coefficient are applicable to numerous regression models, such as logistic, Poisson, negative binomial, Beta, Gamma, Weibull, etc. Further, we illustrate the performance of the Welch's t-test when the response variable is binary.

As a final task we apply all aforementioned testing procedures to two real datasets with large sample sizes. We expose the high computational gains of the score test (and of the Welch's t-test where applicable) in comparison to the log-likelihood ratio test. The first dataset contains $6,000$ observations, whereas the second one contains nearly $40,000$ observations. The rationale is to show the differences between the testing procedures and highlight that the score test requires the sample size to be at the order of tens of thousands so as to produce p-values that are equal to the p-values produced by the log-likelihood ratio test.

The next section presents the log-likelihood ratio test that relies upon fitting regression models, the score test, the Pearson correlation coefficient and the Welch' t-test and we show how one can apply them for the task of univariate filtering. Section 3 illustrates, via Monte Carlo simulation studies the computational gains of the score test, the Pearson correlation and of the Welch's t-test compared to the log-likelihood ratio test. Various regression models are examined, including inter comparisons among the tests in terms of type I error, correlation of the p-values and percentage of agreement of rejection of the H_0 and proportion of times the score test selects the most significant predictor variable. Section 4 illustrates the log-likelihood ratio and the score test using real data and finally Section 5 concludes the paper.

2 Log-likelihood ratio and Score tests for regression models and Pearson correlation coefficient

2.1 Computational efficiency

The issue of computational efficiency during the univariate filtering step has drawn sufficient research interest. Sikorska et al. (2013) proposed a computationally efficient approximation test when fitting thousands univariate logistic regressions, but it is not as computationally efficient as the score test. Redden et al. (2004) proposed a fast method, based on logistic regression, for obtaining the p-values of many median regressions. Obtaining the p-value of a logistic regression is much faster than obtaining the p-value of a median regression. When large sample sizes are available, adoption of the score test can make their method extremely computationally efficient compared to conducting numerous logistic regressions.

Computer nowadays have made parallel computations easier and more efficient. Tsamardinos et al. (2019) took advantage of the parallel computing and adopted the Forward Backward with Early Dropping algorithm (Borboudakis and Tsamardinos, 2019) to big (and massive) data. Parallel computing takes place not only across the predictor variables, but across the observations as well. The observations are split into folds and a logistic regression model is fitted in each fold. The results are then meta-analytically combined. This process produces accurate results with hundreds of thousands of observations and can lead to substantial improvements in terms of execution time, up to 10 times faster. The computational reduction during the univariate filtering though is not comparable to the one achieved by the score test.

On a different direction, Erdogdu et al. (2019) proved that, asymptotically, the regression coefficients of generalised linear models are proportional to the regression coefficients of a linear model. Our simulation studies (not shown here as they are outside the scope of this paper) provided evidence that this holds true for other regression models also, e.g. Weibull regression. Despite fitting a linear model is much cheaper than fitting a logistic regression model for instance, the computational gains are not as significant as one would think. Finding the proportionality factor, requires application of the Newton-Raphson or the golden-ratio algorithm that goes through the whole dataset at each step. Undoubtedly, this process is faster than simply fitting many (non-linear) regression models, yet, it is not as efficient as performing many score tests.

Another direction is to use sub-samples of the data instead of the whole dataset (Park et al., 2018) with accuracy being this strategy's trade-off. According to Park et al. (2018), their proposed method, that uses a portion of the data, can speed-up the maximum likelihood estimation of the model from 6 up to 629 times compared to using the full dataset while guaranteeing the same model predictions with 95% probability. The score test on the contrary, will be shown to produce the same results (almost equivalent p-values) as the log-likelihood ratio test with large sample sizes.

2.2 Univariate filtering

Assume a response variable $\mathbf{Y}$, a $n \times 1$ vector of observations $\mathbf{y}$ and a set of predictor variables, an $n \times d$ matrix $\mathbf{X}$, where n denotes the sample size and d denotes the number of variables are given. At first a regression model with only the intercept is fitted and its log-likelihood is computed. Then for each variable a regression model is fitted and the following hypotheses are tested:

$$H_0 : g(\mathbb{E}(y)) = a_j.$$
$$H_1 : g(\mathbb{E}(y)) = a_j + b_j x_j \quad (j = 1, \ldots, d).$$

Univariate filtering identifies the statistically significant predictor variables, or the $b_j s$ that are statistically significantly different from zero.

2.3 Log-likelihood ratio test

For each regression model in H_1 its associated log-likelihood is computed and hence the log-likelihood ratio test statistic is computed by

$$\Lambda = 2\left(\ell_1 - \ell_0\right), \tag{1}$$

where ℓ_1 and ℓ_0 are the log-likelihood values under the H_1 and H_0 respectively. Under H_0, the log-likelihood ratio test follows a χ^2 distribution with 1 degree of freedom, $\Lambda \dot{\sim} \chi^2_1$ (Young and Smith, 2005).

2.4 Score test

The score function is the derivative of the log-likelihood $U(\theta) = \frac{\partial \ell(\theta)}{\partial \theta}$, where ℓ and θ denote the log-likelihood and the value of the parameter of interest respectively. Application of the central limit theorem combined with Slutsky's lemma, states that under the H_0, $n^{-1/2} U(\theta_0) \xrightarrow{d} N\left(0, I^{-1}(\theta_0)\right)$, where θ_0 denotes the parameter value under H_0 and $I(\theta)$ is the Fisher information. Hence the score test asymptotically follows a χ^2 distribution with 1 degree of freedom

$$S^2 = \frac{U(\theta_0)^2}{Var(U(\theta_0))} \dot{\sim} \chi^2_1. \tag{2}$$

2.5 Pearson correlation coefficient

The sample Pearson correlation coefficient is computed by

$$r = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{\sqrt{n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i\right)^2} \sqrt{n \sum_{i=1}^n y_i^2 - \left(\sum_{i=1}^n y_i\right)^2}}. \tag{3}$$

Under H_0 (the two variables X and Y are linearly independent), the test statistic $Z = 0.5 \log \frac{1+r}{1-r} \sqrt{n-3} \dot{\sim} N(0,1)$ distribution, while for small n, $N(0,1)$ can be substituted by the t_{n-3} distribution, where t_ν denotes the t distribution with ν degree of freedom.

2.6 Large sample asymptotics of the score test

The asymptotic proximity of the two tests can be explained by the fact, that the log-likelihood ratio test and the score test differ by $O_p\left(n^{-1/2}\right)$ Young and Smith (2005), where the $O_p()$ notation indicates a random variable that is asymptotically bounded in probability[3]. In addition, both scores are parametrisation invariant[4]. For a comparison of the log-likelihood ratio test and score test in terms of the expected length of their confidence intervals the reader is referred to Mukerjee and Reid (2001).

2.7 Score test formula for some selected regression models

Below we present some formulas of the (square root of the) score test for some common regression models.

- With binary responses (0 or 1), logistic regression is usually employed. The log-likelihood of the logistic regression is given by

$$\ell_1 = \sum_{i=1}^{n} \left[y_i \log p_i + (1 - y_i) \log (1 - p_i) \right],$$

  where $p_i = \frac{1}{1+e^{-a-bx_i}}$. The score test takes the following form[5] (Hosmer et al., 2013)

$$S_{Bin} = \frac{\sum_{i=1}^{n} y_i x_i - \hat{p} \sum_{i=1}^{n} x_i}{\sqrt{\left[\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i \right)^2 / n \right] \left[\hat{p} \left(1 - \hat{p} \right) \right]}}, \tag{4}$$

  where $\hat{p} = \sum_{i=1}^{n} \frac{y_i}{n}$.

- With strictly positive response values, Gamma regression is an ordinarily selected model, whose log-likelihood is given by

$$\ell_1 = \sum_{i=1}^{n} \left[\alpha - \frac{y_i}{\mu_i} - \log (\mu_i) + \alpha \log (\alpha y_i) - \log (\alpha) \right],$$

  where $\mu_i = e^{a+bx_i}$. The score test for Gamma regression has the following formula

$$S_{Ga} = \frac{\sum_{i=1}^{n} x_i - \frac{\sum_{i=1}^{n} y_i x_i}{\hat{\alpha}/\hat{\beta}}}{\sqrt{\sum_{i=1}^{n} x_i^2 / \hat{\alpha}}}, \tag{5}$$

[3] Broadly speaking, the notation $Y_n = O_p(a_n)$ means that Y_n/a_n is bounded in probability as $n \to \infty$. That is, given $\epsilon > 0$, there exists $k > 0$ and n_0 such that, for all $n > n_0$, $P(|Y_n/a_n| < k) > 1 - \epsilon$.

[4] Parametrisation invariance requires that the conclusions of a statistical analysis be unchanged for any reasonably smooth one-to-one function of θ (Young and Smith, 2005).

[5] The formula in (4) is equivalent to the square of the Cochran-Armitage test statistic for testing trends in a single $2 \times J$ contingency table (Chen, 1983). It is also worthwhile noticing that the formula for the logistic regression (4) and for the Poisson regression (see Appendix) are very similar to the Pearson correlation coefficient (3). This is a cornerstone feature of the score test for these two regression models that will reduce the computational burden significantly. R's command *cor* is pretty fast and the score test for the Poisson and the logistic regression rely on this command to achieve high speed.

where $\hat{\alpha}$ and $\hat{\beta}$ are the MLE estimates of the parameters of the Gamma regression under H_0.

- Beta regression is appropriate for responses that lie within $(0, 1)$ with the log-likelihood being

$$\ell_1 = \sum_{i=1}^{n} \{\log\Gamma(\phi) - \log\Gamma(\mu_i\phi) - \log\Gamma[(1-\mu_i)\phi] + (\mu_i\phi - 1)\log(y_i) + \\ + [(1-\mu_i)\phi - 1]\log(1-y_i)\},$$

where $\mu_i = \frac{1}{1+e^{-a-bx_i}}$. The relevant score test is given by

$$S_{Be} = \frac{\sum_{i=1}^{n} x_i \log\frac{y_i}{1-y_i} - \sum_{i=1}^{n} x_i \left[\psi(\hat{\alpha}) - \psi(\hat{\beta})\right]}{\sqrt{\sum_{i=1}^{n} x_i^2 \left[\psi'(\hat{\alpha}) + \psi'(\hat{\beta})\right]}}, \tag{6}$$

where $\hat{\alpha}$ and $\hat{\beta}$ are the MLE estimates of the parameters of the Beta regression under H_0, $\psi(.)$ and $\psi'(.)$ are the digamma and trigamma functions respectively. Note, that we consider regression for the location parameter only and not for the dispersion parameter.

2.8 Welch's t-test when the response is binary

When the response is binary, the Welch's t-test Welch (1951) can also be used and its test statistic is given by

$$T_w = \frac{|\bar{x}_1 - \bar{x}_0|}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_0^2}{n_0}}}, \tag{7}$$

where, in our case, $\bar{x}_1$ and $\bar{x}_0$ & s_1^2 and s_0^2 denote the two sample means and variances of the predictor variable x in the two subgroups $y = 1$ and $y = 0$ respectively. Under H_0, $T_w \sim t_\nu$, where ν is given by (Satterthwaite, 1946; Welch, 1951)

$$\nu \simeq \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{s_1^4}{n_1^2(n_1-1)} + \frac{s_2^4}{n_2^2(n_2-1)}}. \tag{8}$$

According to (Boulesteix, 2007) this is one of the standard approaches for such cases. To the best of our knowledge this test is not frequently employed by variable selection algorithms and has gone unnoticed. One possible reason could be that no one has performed simulation studies or empirical evaluation studies to show its, undermined, value. The non parametric alternative, Wilcoxon-Mann-Whitney test is not suggested because it tends to inflate the type I error (Tsagris et al., 2020a).

3 Monte Carlo simulations

Three regression models will be examined: logistic regression, Gamma regression and Beta regression. For all regression models, response values were generated from the relevant distributions and predictor variables were generated from the standard normal distribution. Since the score test for the logistic regression is very similar to the Pearson correlation, the latter will be excluded from this regression. In all cases, the five axes of comparison or five metrics are: a) Computational cost, b) Type I error, c) Correlation of the p-values (of the score test, of the log-likelihood ratio test and of the Welch's t-test & Pearson correlation when performed), d) Agreement in the decision (reject/not reject H_0) and e) Proportion of times the test detects the most significant predictor variable. The motivation behind this last metric can be seen in VS algorithms such as the (generalised) Orthogonal Matching Pursuit (Tsagris et al., 2018, 2020b) that selects the most significant variable in the first step.

3.1 Example 1: Logistic regression

Binary response values were generated from a Bernoulli distribution ($Ber(p)$) with various probabilities of success $p = (0.1, 0.2, 0.3, 0.4, 0.5)$ while for each case, $d = 500$ random predictor variables were generated from a standard normal distribution. The sample size varied from $10,000$ up to $1,000,000$. For each combination of probability of success and sample size the aforementioned four metrics (a)-(d) were computed. This process was repeated 10 times and the average performance metrics are reported.

Table 1 shows computational cost (in seconds) of each test for the 500 predictor variables for different sample sizes. The computational cost of both tests increases with the sample size, with the log-likelihood ratio test requiring up to 6 minutes with large sample sizes, while the score test never exceeds 6 seconds. Figure 1 presents the speed-up factor[6] across the various probabilities of success as a function of the sample size. The log-likelihood ratio test is between 30 to 70 times slower than the score test.

Table 2 contains the estimated type I error for both tests. These are in close agreement and when the sample size is $20,000$ or higher the estimated errors have the same value up to the 3rd digit. The correlation of the p-values of the two tests is perfect when the sample size is $20,000$ or larger (see Table 3). The percentage of agreement in the decision of rejection of the H_0 is also perfect (see Table 3) for the same sample sizes.

The Welch's t-test produces similar results to the score test and hence are not presented. The speed-up factors ranged from 34 up to 59 and the estimated type I errors were almost identical. The correlation of the log-likelihood ratio test p-values with the Welch's t-test p-values was always 1 and the percentage of agreement in rejecting the null hypothesis was either 0.998, 0.999 or 1.

The fifth axis of comparison is the proportion of times the test detects the most significant predictor variable. For this purpose we randomly selected a predictor variable X' and associated it with the response variable via the following formula $Y \sim Ber(\frac{1}{1+e^{-\beta X'}})$, where $\beta = (0.1, 0.2, 0.3, 0.4, 0.5)$ and the sample sizes n were the same. We counted the

[6]The number of times the log-likelihood ratio test is slower than the score test.

Table 1: **Logistic regression**: Computational cost (in seconds) of the log-likelihood ratio test (Λ) and the score test (S^2) for different sample sizes and probabilities of success. The fastest method is highlighted in bold.

	Probability of success									
	$p = 0.1$		$p = 0.2$		$p = 0.3$		$p = 0.4$		$p = 0.5$	
Sample size	Λ	S^2	Λ	S^2	Λ	S^2	Λ	S^2	Λ	S^2
1x10^4	1.72	**0.04**	1.52	**0.03**	1.3	**0.04**	1.83	**0.04**	1.78	**0.04**
2x10^4	4.11	**0.12**	3.09	**0.07**	3.08	**0.07**	3.88	**0.06**	3.15	**0.05**
5x10^4	13.5	**0.34**	9.26	**0.2**	7.79	**0.17**	11.18	**0.16**	8.38	**0.15**
1x10^5	25.99	**0.63**	18.27	**0.34**	14.99	**0.3**	20.2	**0.29**	16.55	**0.31**
2x10^5	58.86	**1.23**	38.16	**0.68**	32.34	**0.66**	43.31	**0.69**	33.85	**0.61**
3x10^5	87.67	**1.96**	58.00	**1.07**	48.85	**0.98**	62.26	**0.91**	50.67	**0.92**
5x10^5	107.04	**2.24**	104.1	**2.18**	81.51	**1.64**	105.2	**1.62**	86.32	**1.58**
7x10^5	183.1	**3.94**	132.19	**2.51**	113.28	**2.33**	146.48	**2.25**	123.37	**2.15**
1x10^6	254.99	**5.62**	178.27	**3.26**	156.82	**3.24**	207.02	**3.26**	178.18	**3.10**

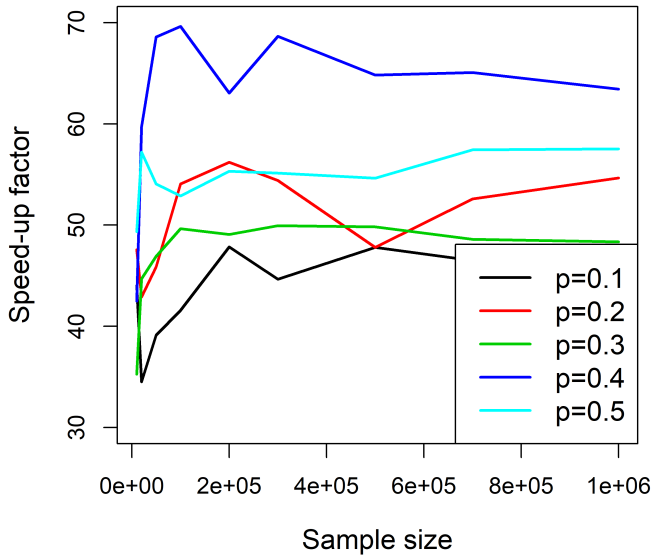


Figure 1: **Logistic regression**: Speed-up factor of the Λ test against the S^2 test. This is an estimate of how many times the Λ test is slower than the S^2 test.

Table 2: **Logistic regression**: Estimated type I error of the log-likelihood ratio test (Λ) and the score test (S^2) for different sample sizes and probabilities of success.

	Probability of success									
Sample	$p = 0.1$		$p = 0.2$		$p = 0.3$		$p = 0.4$		$p = 0.5$	
size	Λ	S^2	Λ	S^2	Λ	S^2	Λ	S^2	Λ	S^2
1x10^4	0.056	0.056	0.053	0.053	0.053	0.052	0.052	0.051	0.049	0.049
2x10^4	0.051	0.051	0.055	0.055	0.049	0.049	0.048	0.048	0.048	0.048
5x10^4	0.053	0.053	0.050	0.050	0.052	0.052	0.053	0.053	0.045	0.045
1x10^5	0.051	0.051	0.052	0.052	0.050	0.050	0.047	0.047	0.048	0.048
2x10^5	0.048	0.048	0.053	0.053	0.052	0.052	0.050	0.050	0.050	0.050
3x10^5	0.047	0.047	0.048	0.048	0.051	0.051	0.049	0.049	0.057	0.057
5x10^5	0.052	0.052	0.052	0.052	0.049	0.049	0.048	0.048	0.046	0.046
7x10^5	0.050	0.050	0.048	0.048	0.054	0.054	0.052	0.052	0.047	0.047
1x10^6	0.050	0.050	0.052	0.052	0.045	0.045	0.048	0.048	0.051	0.051

Table 3: **Logistic regression**: Correlation of the Λ and S^2 test p-values and percentage of agreement in rejecting H_0 for different sample sizes and probabilities of success.

	Correlation of the p-values					**Percentage of agreement**				
	Probability of success (p)					Probability of success (p)				
Sample size	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5
1x10^4	1	1	1	1	1	1	1	0.999	0.999	1
$\geq$ 2x10^4	1	1	1	1	1	1	1	1	1	1

number of times the score test, the log-likelihood ratio test and the Welch's t-test selected that variable as the one with the lowest p-value. All tests were always, unanimously, detecting the randomly selected variable except for $n = 10,000$ and $\beta = 0.1$, in which case all tests detected this variables in 94% of the times.

3.2 Example 2: Gamma regression

Response values were generated from Gamma distribution with shape and scale parameters equal to 1 and 5 (Ga(1,5)) and equal to 5 and 5 (Ga(5,5)). Accordingly, $d = 500$ random predictor variables were independently generated from the standard normal distribution. The sample sizes varied from $10,000$ up to $1,000,000$ and for each Gamma

distribution and sample size the four performance metrics were computed and averaged over 10 repetitions. The results are presented in Tables 4, 5 and 6.

Table 4 summarizes the computational cost of the log-likelihood ratio test and of the score test. The computational cost of the log-likelihood ratio test for large sample sizes is as high as 6 minutes, whereas for the score test it never exceeds the 4 seconds. The speed-up factor varies from 52 up to 93. For both Gamma distributions considered the computational gain (speed-up factor) of the score test is large and then decreases until it reaches a plateau at about 55, for sample sizes equal to hundreds of thousands.

The estimated type I errors of both tests are in close agreement as can be seen in Table 5, even for sample sizes equal to $10,000$. Their estimated type I errors become equal when the sample sizes are $50,000$ or more for both Gamma distributions.

The correlation of the p-values of both tests (see Table 6) reaches 1 for sample sizes equal to or greater than $100,000$. The percentage of agreement of the tests in rejecting the H_0 or not (see Table 6) also reaches 1 for the sample sizes equal to or greater than $100,000$. Nonetheless, the correlation is satisfactorily high for smaller sample sizes and never drops below 0.999.

Table 4: **Gamma regression**: Computational cost (in seconds) of the log-likelihood ratio test (Λ) and the score test (S^2) for different sample sizes and parameter values with $d = 500$ predictor variables. Fastest method is highlighted in bold. The speed-up factor columns depict the number of times Λ is slower than S^2.

	Gamma parameters					
	$\alpha = 1$, $\beta = 5$			$\alpha = 5$, $\beta = 5$		
Sample size	Λ	S^2	Speed-up factor	Λ	S^2	Speed-up factor
1x10^4	1.43	**0.02**	71.50	1.87	**0.02**	93.50
2x10^4	2.83	**0.04**	70.75	3.61	**0.04**	90.25
5x10^4	7.24	**0.10**	72.40	8.68	**0.11**	78.91
1x10^5	16.21	**0.23**	70.48	18.85	**0.30**	62.83
2x10^5	28.74	**0.56**	51.32	34.28	**0.60**	57.13
3x10^5	47.08	**0.87**	54.11	53.21	**0.96**	55.43
5x10^5	84.17	**1.47**	57.23	82.26	**1.58**	52.06
7x10^5	132.08	**2.43**	54.36	103.29	**1.97**	52.43
1x10^6	188.98	**3.40**	55.58	143.54	**2.54**	56.51

For the proportion of times the test detects the most significant predictor variable we randomly selected a predictor variable X' and associated it with the response variable via the following formula $Y \sim Ga(e^{\beta X'}, 5)$ and $Y \sim Ga(e^{\beta X'}, 10)$, where $\beta = (0.1, 0.2, 0.3, 0.4, 0.5)$ and the sample sizes n were the same as before. We counted the number of times the score test and the log-likelihood ratio test selected that variable as the one with the lowest p-value. All tests were always and in both cases, unanimously,

Table 5: **Gamma regression**: Estimated type I error of the log-likelihood ratio test (Λ) and the score test (S^2) for different sample sizes and parameter values with $d = 100$ predictor variables.

	Gamma parameters			
	$\alpha = 1$, $\beta = 5$		$\alpha = 5$, $\beta = 5$	
Sample size	Λ	S^2	Λ	S^2
1x10^4	0.048	0.049	0.046	0.045
2x10^4	0.048	0.048	0.050	0.050
5x10^4	0.052	0.052	0.049	0.049
1x10^5	0.056	0.055	0.055	0.055
2x10^5	0.048	0.047	0.047	0.047
3x10^5	0.049	0.050	0.050	0.050
5x10^5	0.046	0.046	0.052	0.052
7x10^5	0.055	0.055	0.046	0.046
1x10^6	0.050	0.050	0.049	0.049

Table 6: **Gamma regression**: Correlation of the Λ and S^2 test p-values and percentage of agreement in rejecting H_0 for different sample sizes and probabilities of success.

	Correlation of p-values		**Percentage of agreement**	
	Gamma parameters			
Sample size	$\alpha = 1$, $\beta = 5$	$\alpha = 5$, $\beta = 5$	$\alpha = 1$, $\beta = 5$	$\alpha = 1$, $\beta = 5$
1x10^4	0.999	0.999	0.999	0.999
2x10^4	0.999	1	0.999	0.999
5x10^4	0.999	1	0.999	1
$\geq$ 1x10^5	1	1	1	1

detecting the randomly selected variable regardless of the sample size and the magnitude of the regression coefficient β.

3.3 Example 3: Beta regression

The response values this time were generated from a Beta distribution Be(α, β), where the shape parameters were (α, β)=(5, 10), (0.5, 0.5) and (10, 5). In this scenario $d = 100$ and $d = 500$ random predictor variables were generated from standard normal

distribution, while the sample sizes varied from 100 up to 20,000. The reason we chose small sample sizes is that the score test has been shown to be size correct even for small sample sizes (Cribari-Neto and Queiroz, 2014). Beta regression, implemented in the R package *betareg* (Cribari-Neto and Zeileis, 2010), is not implemented in C++ but utilizes the R built-in function *optim* and hence the computational cost increases considerably with sample size.

The average duration (in seconds) of computing the p-values of the log-likelihood ratio test after 100 univariate Beta regressions and the p-values of 100 score tests appears in Table 7. The speed-up factors are more than 4,000, meaning that the application of 100 Beta regressions can be more than 4,000 times slower than the application of 100 score tests. The estimated type I errors (see Table 8) are nearly the same. Note that this time the sample size was only as large as 20,000, as the time required for the Beta regressions increases with the sample size. A similar picture is seen by examining the computational cost in Table 9 and the estimated type I error in Table 10 for the case of 500 predictor variables. The computational cost is dramatically smaller than performing 500 Beta regressions in R. The speed-up factor ranges from 154 up to 5,809, indicating that performing many Beta regressions can be thousands of times slower than performing many score tests.

Surprisingly enough, computation of many score tests is faster than computation of many Pearson correlation coefficients (see Table 9). However, Table 10 shows that the estimated type I error of the score test and of the Pearson correlation coefficient do not fully agree even for sample sizes equal to 1,000,000.

In order to see whether this disagreement was significant and to see what are the possible implications, the probability of identifying the most significant variable was computed for the score test and for the Pearson correlation coefficient. If the score test and the Pearson correlation coefficient agree in the most significant variable, then their type I error differences can be deemed negligible. In this case, one predictor variable (X') was randomly chosen from the 500 predictor variables. The response values were then generated from $Be\left(\frac{1}{1+e^{-X'}}, 0.5\right)$, $Be\left(\frac{1}{1+e^{X'}}, 5\right)$ and $Be\left(\frac{1}{1+e^{X'}}, 10\right)$. The results, presented in Table 11, show that when the sample sizes exceed 500 there is perfect agreement, in detecting the most statistically significant variable, between the score test and the Pearson correlation coefficient.

4 Examples with real data

Monte Carlo studies are based on simulating the predictor variables and the response variable from parametric models followed by parametric regression models. Hence, the data generating mechanism is expected to be recovered with large sample sizes. This is the ideal scenario where we expect the testing procedures to perform well. On the contrary, with real data will we cannot know beforehand the true generating model of the real data. Practitioners working with real data are more interested to know the performance of a testing procedure (and of an algorithm in general) with real data, under realistic situations. We compared the performance of the score test with the

Table 7: **Beta regression**: Computational cost (in seconds) of the log-likelihood ratio test (Λ) and the score test (S^2) for different sample sizes and parameter values with $p = 100$ predictor variables. Fastest method is highlighted in bold. The speed-up factor columns depict the number of times Λ is slower than S^2.

	Beta parameters								
	$\alpha = 5$, $\beta = 10$			$\alpha = 0.5$, $\beta = 0.5$			$\alpha = 10$, $\beta = 5$		
Sample size	Λ	S^2	Speed-up factor	Λ	S^2	Speed-up factor	Λ	S^2	Speed-up factor
100	1.54	**0.01**	154	1.64	**0.01**	164	1.73	**0.01**	173
500	2.73	**0.01**	273	2.63	**0.01**	263	3.19	**0.01**	319
1,000	4.37	**0.01**	437	4.11	**0.01**	411	5.71	**0.01**	571
5,000	21.18	**0.01**	2118	16.61	**0.01**	1661	24.11	**0.01**	2411
10,000	54.99	**0.01**	5499	35.46	**0.01**	1773	58.09	**0.01**	5809
20,000	88.68	**0.02**	4434	66.03	**0.02**	3315	91.62	**0.02**	4581

Table 8: **Beta regression**: Estimated type I error of the log-likelihood ratio test (Λ) and the score test (S^2) for different sample sizes and parameter values with $p = 100$ predictor variables.

	Beta parameters					
	$\alpha = 5$, $\beta = 10$		$\alpha = 0.5$, $\beta = 0.5$		$\alpha = 10$, $\beta = 5$	
Sample size	Λ	S^2	Λ	S^2	Λ	S^2
100	0.052	0.048	0.064	0.061	0.054	0.052
500	0.051	0.048	0.054	0.053	0.037	0.036
1,000	0.050	0.051	0.048	0.047	0.059	0.057
5,000	0.067	0.066	0.044	0.044	0.046	0.046
10,000	0.047	0.047	0.041	0.041	0.048	0.048
20,000	0.052	0.052	0.054	0.054	0.047	0.047

log-likelihood ratio test, where we assume that the parametric regression fits the data adequately. The computational cost of the log-likelihood and of the score test, the correlation of their corresponding p-values and the percentage of agreement in rejecting/not rejecting the H_0 were assessed.

Two datasets were downloaded from the https://archive.ics.uci.edu/ml/index.phpUC Irvine Machine Learning Repository, namely the *Gisette* dataset and the *Online News*

Table 9: **Beta regression**: Computational cost (in seconds) of the score test (S^2) and the Pearson correlation coefficient test (Z) for different sample sizes and parameter values with $d = 500$ predictor variables. Fastest method is highlighted in bold.

	Beta parameters					
	$\alpha = 5$, $\beta = 10$		$\alpha = 0.5$, $\beta = 0.5$		$\alpha = 10$, $\beta = 5$	
Sample size	S^2	Z	S^2	Z	S^2	Z
100	0.00	0.00	0.00	0.00	0.00	0.00
500	0.00	0.00	0.00	0.00	0.00	0.00
1,000	0.00	0.00	0.00	0.01	0.00	0.00
5,000	0.02	0.02	**0.01**	0.02	**0.01**	0.02
1x10^4	**0.03**	0.04	**0.02**	0.03	**0.02**	0.03
2x10^4	**0.05**	0.06	**0.05**	0.06	**0.05**	0.06
5x10^4	**0.12**	0.17	**0.12**	0.15	**0.12**	0.16
1x10^5	**0.29**	0.41	**0.23**	0.29	**0.25**	0.30
2x10^5	**0.56**	0.68	**0.47**	0.60	**0.47**	0.61
5x10^5	**1.36**	1.74	**1.19**	1.53	**1.34**	1.66
7x10^5	**1.94**	2.54	**1.65**	2.13	**2.02**	2.41
1x10^6	**2.48**	3.09	**2.72**	3.17	**2.88**	3.45

Popularity dataset. Both datasets have a binary response and are thus suitable for logistic regression. The first dataset is a handwritten digit recognition problem where the goal is to separate the highly confusible digits "4" and "9". This dataset is one of five datasets of the NIPS 2003 feature selection challenge (Guyon et al., 2005) and contains $5,999$ binary observations and $5,000$ predictor variables. The second dataset summarizes a heterogeneous set of features about articles published by Mashable in a period of two years (Fernandes et al., 2015) with the goal of predicting the popularity in social networks. The popularity of online news is often measured by considering the number of interactions in the Web and social networks (e.g., number of shares, likes and comments). The authors have binarised the popularity using a threshold of $1,400$ shares and thus have turned the regression problem into a classification problem. This dataset contains $39,644$ observations and 64 predictor variables.

For logistic regression in particular examination of the tests is straightforward since the response values are binary. To assess the score test in Gamma and Beta regressions some modifications must take place. All three cases are presented below.

- **Logistic regression and score test**. In order to obtain a better and more accurate picture of the computational cost and of the relevant performance metrics, we implemented 10 repetitions. Each time a bootstrap sample was generated

Table 10: **Beta regression**: Estimated type I error of the score test (S^2) and the Pearson correlation coefficient test (Z) for different sample sizes and parameter values with $d = 500$ predictor variables.

	Beta parameters					
	$\alpha = 5$, $\beta = 10$		$\alpha = 0.5$, $\beta = 0.5$		$\alpha = 10$, $\beta = 5$	
Sample size	S^2	Z	S^2	Z	S^2	Z
100	0.048	0.046	0.052	0.049	0.052	0.048
500	0.055	0.058	0.045	0.047	0.047	0.047
1,000	0.055	0.054	0.054	0.051	0.055	0.055
5,000	0.049	0.048	0.055	0.054	0.051	0.052
1x10^4	0.052	0.053	0.053	0.051	0.052	0.051
2x10^4	0.051	0.052	0.048	0.050	0.051	0.051
5x10^4	0.051	0.055	0.045	0.045	0.051	0.053
1x10^5	0.046	0.047	0.045	0.047	0.055	0.052
2x10^5	0.052	0.049	0.049	0.047	0.045	0.046
5x10^5	0.054	0.054	0.044	0.046	0.049	0.050
7x10^5	0.055	0.054	0.054	0.053	0.047	0.047
1x10^6	0.045	0.046	0.048	0.051	0.045	0.045

Table 11: **Beta regression**: Estimated probability of identifying the most significant predictor variable of the score test (S^2) and the Pearson correlation coefficient test (Z) for different sample sizes and parameter values with $p = 500$ predictor variables. The highest probability is highlighted in bold.

	Beta parameters					
	$\beta = 10$		$\beta = 0.5$		$\beta = 5$	
Sample size	S^2	Z	S^2	Z	S^2	Z
100	**0.92**	0.64	**0.90**	0.62	**0.88**	0.50
≥ 500	1.00	1.00	1.00	1.00	1.00	1.00

containing the response vector and the predictor variables matrix (the pairing was not distorted). For each bootstrap sample we computed the p-values from the score and the log-likelihood ratio tests and Welch's t-test.

- **Gamma regression and score test**. Since the response values are binary we created new response values. We generated non negative continuous random values

from a mixture of a Weibull and a folded normal distribution with the mixing proportion being equal to 50%. This process was repeated 10 times and each time we computed the p-values from the score and the log-likelihood ratio tests.

- **Beta regression and score test**. Similarly to the Gamma regression case, new response values were generated. We generated percentages from a mixture of a logistic normal distribution and a simplex distribution with the mixing proportion being equal to 50%. This process was repeated 5 times only for the first dataset (and 10 times for the second dataset), because fitting thousands of Beta regressions was shown to be highly computationally expensive. Each time we computed the p-values from the score and the log-likelihood ratio tests.

The performance metrics that were computed are a) the computational cost of the log-likelihood ratio test and of the score test, b) the correlation of their corresponding p-values and c) the percentage of agreement in rejecting/not rejecting the H_0. The average numbers of all metrics are reported in Table 12, corroborating the evidence of the simulation studies.

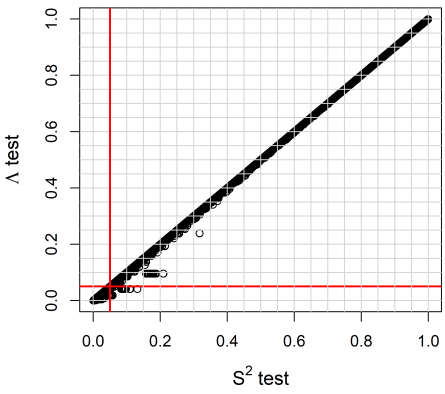
The first dataset (Gisette) contains $5,999$ observations and this explains why the correlation between the log-likelihood ratio p-values and score test p-values is 0.999. The second dataset (Online) contains $39,644$ observations and this is why the correlation of the p-values is 1. The same conclusions were drawn for Welch's t-test. Figure 2 visualizes the p-values obtained from the log-likelihood ratio test, the score test and the Welch's t-test. The results agree with the simulation studies also, for the Gamma and Beta regressions. The correlation of the p-values is only 0.997 even for the second dataset (Online). Table 6 reported that in order for the correlation of the p-values of the score test and the log-likelihood ratio test to be exactly 1 requires samples sizes of tens of thousands of observations. Finally the computational advantage of the score test (and of the Welch's t-test) over the log-likelihood ratio test is again evident for all three types of regressions.

5 Conclusions

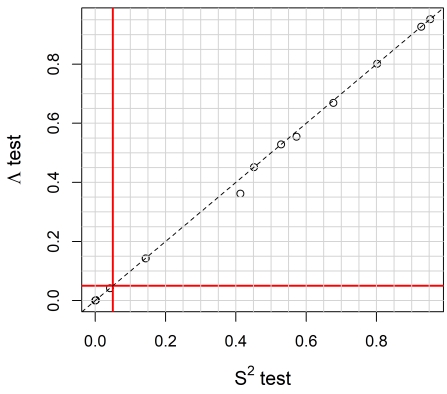
The score test was suggested as a faster alternative to log-likelihood ratio test that involves fitting many simple (with one predictor) regression models. Score test's only requirement, in order to be equivalent to the log-likelihood ratio test, is large sample size. This might sound like a disadvantage at first, but is actually an advantage. With massive or big data, computational cost becomes a serious problem and score test solves this problem effectively.

The score test and the Pearson correlation coefficient when used for univariate filtering were shown to be extremely computationally efficient when compared to the log-likelihood ratio test and produced exactly the same results with large sample sizes ($n > 10,000$) for logistic regression and Gamma regression. In addition, the Welch's t-test produced almost identical results to the score test. Hence, with massive or big data, the score test could substitute the log-likelihood ratio test, while for logistic regression

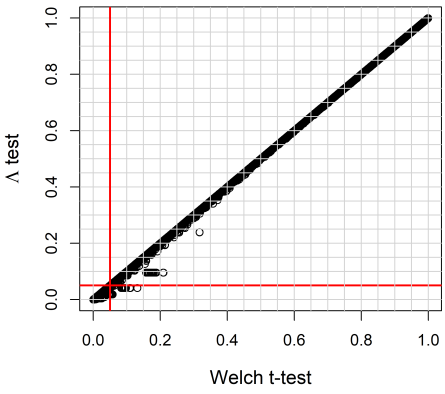
Gisette dataset **Online News Popularity dataset**



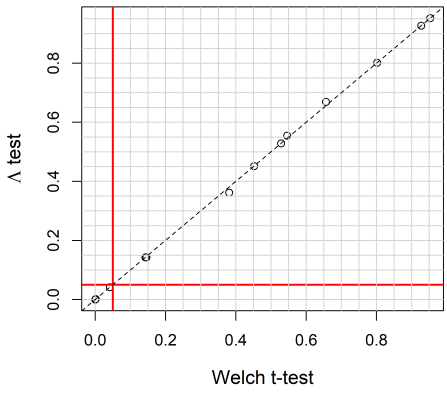
(a) S^2 **p-values versus** Λ **p-values**



(b) S^2 **p-values versus** Λ **p-values**



(c) Welch's t**-test p-values versus** Λ **p-values**



(d) Welch's t**-test p-values versus** Λ **p-values**

Figure 2: Scatter plot of the score test (S^2) p-values and of the Welch's t-test p-values versus the log-likelihood ratio test (Λ) p-values (using logistic regression). The dashed line refers to the 45° line that passes through the origin. The red lines delimit the rejection region at the 5% significance level for each test.

Table 12: **Real data examples**: Computational cost (in seconds) of Λ and S^2 tests, correlation of their p-values and percentage of agreement in rejecting H_0 for the two datasets and the three regression models.

		Computational cost			**Correlation of p-values**		**Percent of agreement**	
Regression	Dataset	Λ	S^2	Welch	S^2	Welch	S^2	Welch
Logistic	Gisette	19.875	**0.230**	**0.242**	0.999	0.999	0.991	0.991
	Online	1.438	**0.020**	**0.016**	1	0.999	1	1
Gamma	Gisette	13.913	**0.251**		0.995		0.973	
	Online	0.908	**0.008**		0.997		0.983	
Beta	Gisette	1208.310	**0.250**		1		0.998	
	Onine	70.988	**0.112**		1		1	

the Welch's t-test is another option. For Beta distributed response values, the Pearson correlation coefficient and the score test did not reach 100% agreement for small sample sizes. An interesting conclusion is that the score test is size correct even for small sample sizes corroborating the findings of Cribari-Neto and Queiroz (2014). This implies that the score test could replace the log-likelihood ratio test even for small sample sizes with Beta distributed response values.

In case of binary responses, computation of many score tests is between 30 to 70 times faster than the computation of a C++ implementation of the relevant, logistic regression based, log-likelihood ratio tests. With Beta regression, computation of numerous score tests is more than $6,000$ times faster than performing many log-likelihood ratio tests using Beta regression that has been implemented in R. Score test's computational efficiency is attributed to the fact that it fits a single regression model only, under the null hypothesis, unlike the log-likelihood ratio test that requires fitting many regression models under the alternative hypothesis as well.

Another conclusion this paper has reached to, is that despite R being rather "slow" (in comparison to Python or Matlab), with the proper computations it becomes extremely fast. The general advice "*It's your algorithm*" suits the results of this paper. Continuing with this, we would like to inform the reader that many score and log-likelihood ratio tests have been implemented in the R packages *Rfast* (Papadakis et al., 2020) and *Rfast2* (Papadakis et al., 2019). Furthermore, we are working towards improving the computational efficiency of the score test.

Due to the paper's space limitations not many regression cases could be covered. For instance, Poisson and negative binomial and Weibull regression for which the formulas of the score test are provided in the Appendix. The case of multinomial regression was not examined either, for which Welch's F-test for multiple samples (Welch, 1951) can be an alternative to the log-likelihood ratio test, with computational cost nearly equal

to that of the score test and results of similar accuracy.

In all cases and examples considered in this paper, only continuous predictor variables were used. Evidently, real data are not constrained to continuous predictors, but may include categorical variables as well. We could modify our R function to apply the score test for categorical predictors, but that would entail the employment of a *for* loop in R, thus increasing the computational burden. Tsagris et al. (2020b) treats categorical variables by applying Analysis of Variance.

Appendix

Asymptotic equivalence of the score test and log-likelihood ratio test

Below is a short proof of the asymptotic equivalence of the score test and log-likelihood ratio test when θ is scalar, as in the case this paper examines. By expanding the score function $U(\theta)$ using Taylor series about $\hat{\theta}$ one can obtain (Brazzale et al., 2007; Young and Smith, 2005)

$$\left(\hat{\theta}-\theta\right) I(\theta)^{1/2} = I(\theta)^{1/2} U(\theta)\left[1+o_p(1)\right].$$

A similar expression for the log-likelihood ratio test gives

$$\Lambda(\theta) = \left(\hat{\theta}-\theta\right)^2 I(\theta)\left[1+o_p(1)\right],$$

where $o_p(1)$ indicates a random variable that converges in probability to 0[7].

Score test formulas for some other regression models

Formulas of the score test for some other common regression models.

- With count data, the Poisson regression is the simplest model employed and its log-likelihood is given by

$$\ell_1 = \sum_{i=1}^{n}\left[y_i \log(\lambda_i) - \lambda_i - \log(y_i!)\right],$$

where $\lambda_i = e^{a+bx_i}$. The form of the score test in this case is

$$S_{Pois} = \frac{\sum_{i=1}^{n} y_i x_i - \bar{y}\sum_{i=1}^{n} x_i}{\sqrt{\left[\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2/n\right]\bar{y}}}, \tag{9}$$

where $\bar{y} = \sum_{i=1}^{n} \frac{y_i}{n}$.

[7]In general, the notation $Y_n = o_p(a_n)$ means that $Y_n/a_n \to 0$ as $n \to \infty$.

- With count data that exhibit overdispersion (variance is greater than the mean), the negative binomial regression is more suitable than the Poisson regression that assumes the dispersion parameter is 1 (mean is equal to the variance). The relevant log-likelihood is given by

$$\ell_1 = \sum_{i=1}^{n} \left[\log \Gamma (y_i + r) - \log (y_i!) - \log (r) + r \log \left(\frac{r}{r + \mu_i} \right) + y_i \log \left(\frac{\mu_i}{r + \mu_i} \right) \right],$$

where $\mu_i = e^{a+bx_i}$. The corresponding score test is given by

$$S_{NB} = \frac{\hat{p} \sum_{i=1}^{n} x_i y_i - (1 - \hat{p})\, \hat{r} \sum_{i=1}^{n} x_i}{\sqrt{\hat{p}^2 \left(\bar{y} + \bar{y}^2 / \hat{r} \right) \sum_{i=1}^{n} x_i^2}}, \tag{10}$$

where $\bar{y}$ is the sample mean, $\hat{p}$ and $\hat{r}$ are the MLE estimates of the parameters of the Negative Binomial regression under H_0.

- An alternative to Gamma regression is the Weibull regression, that is mainly used in biostatistics. Its log-likelihood is given by

$$\ell_1 = \sum_{i=1}^{n} \left[\log (\kappa) - \log (\lambda_i) + (\kappa - 1) \log \left(\frac{y_i}{\lambda_i} \right) - \left(\frac{y_i}{\lambda_i} \right)^{\kappa} \right],$$

where $\lambda_i = e^{a+bx_i}$. The relevant score test takes the following form

$$S_{Weib} = \frac{\frac{\sum_{i=1}^{n} x_i y_i^{\hat{\kappa}}}{\hat{\lambda}^{\hat{\kappa}}} - \sum_{i=1}^{n} x_i}{\sqrt{\sum_{i=1}^{n} x_i^2}}, \tag{11}$$

where $\hat{\kappa}$ and $\hat{\lambda}$ are the MLE estimates of the parameters of the Weibull regression under H_0.

References

Borboudakis, G. and Tsamardinos, I. (2019). Forward-backward selection with early dropping. *The Journal of Machine Learning Research*, 20(1):276–314.

Boulesteix, A.-L. (2007). WilcoxCV: an R package for fast variable selection in cross-validation. *Bioinformatics*, 23(13):1702–1704.

Brazzale, A. R., Davison, A. C., and Reid, N. (2007). *Applied Asymptotics: Case Studies in Small-Sample Statistics*. Cambridge University Press.

Chen, C.-F. (1983). Score tests for regression models. *Journal of the American Statistical Association*, 78(381):158–161.

Chen, S., Billings, S. A., and Luo, W. (1989). Orthogonal least squares methods and their application to non-linear system identification. *International Journal of Control*, 50(5):1873–1896.

Cribari-Neto, F. and Queiroz, M. P. (2014). On testing inference in Beta regressions. *Journal of Statistical Computation and Simulation*, 84(1):186–203.

Cribari-Neto, F. and Zeileis, A. (2010). Beta Regression in R. *Journal of Statistical Software*, 34(2).

Davis, G. M., Mallat, S. G., and Zhang, Z. (1994). Adaptive time-frequency decompositions. *Optical Engineering*, 33(7):2183–2192.

Erdogdu, M. A., Bayati, M., and Dicker, L. H. (2019). Scalable approximations for generalized linear problems. *The Journal of Machine Learning Research*, 20(1):231–275.

Fan, J. and Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5):849–911.

Fan, J. and Song, R. (2010). Sure independence screening in generalized linear models with NP-dimensionality. *The Annals of Statistics*, 38(6):3567–3604.

Fernandes, K., Vinagre, P., and Cortez, P. (2015). A proactive intelligent decision support system for predicting the popularity of online news. In *Portuguese Conference on Artificial Intelligence*, pages 535–546. Springer.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1.

Greene, W. H. (2003). *Econometric Analysis*. Pearson Education India.

Guyon, I., Gunn, S., Ben-Hur, A., and Dror, G. (2005). Result analysis of the NIPS 2003 feature selection challenge. In *Advances in Neural Information Processing Systems*, pages 545–552.

Hosmer Jr, D. W., Lemeshow, S., and Sturdivant, R. X. (2013). *Applied logistic regression*. John Wiley & Sons, 3rd Edition.

Lagani, V., Athineou, G., Farcomeni, A., Tsagris, M., and Tsamardinos, I. (2017). Feature selection with the R package MXM: Discovering statistically-equivalent feature subsets. *Journal of Statistical Software*, 80.

Lumley, T. (2020). *leaps: Regression Subset Selection*. R package version 3.1, https://CRAN.R-project.org/package=leaps.

McLeod, A., Xu, C., and Lai, Y. (2020). *bestglm: Best Subset GLM and Regression Utilities*. R package version 0.37.3, https://CRAN.R-project.org/package=bestglm.

Mukerjee, R. and Reid, N. (2001). Comparison of test statistics via expected lengths of associated confidence intervals. *Journal of Statistical Planning and Inference*, 97(1):141–151.

Papadakis, M., Tsagris, M., Dimitriadis, M., Fafalios, S., Tsamardinos, I., Fasiolo, M., Borboudakis, G., Burkardt, J., Zou, C., Lakiotaki, K., and Chatzipantsiou, C. (2020). *Rfast: A Collection of Efficient and Extremely Fast R Functions. R package version 1.9.9*. Available at the CRAN Repository: https://CRAN.R-project.org/package=Rfast.

Papadakis, M., Tsagris, M., Fafalios, S., and Dimitriadis, M. (2019). *Rfast2: A Collec-*

tion of Efficient and Extremely Fast R Functions II. R package version 0.0.5. Available at the CRAN Repository: https://CRAN.R-project.org/package=Rfast2.

Park, Y., Qing, J., Shen, X., and Mozafari, B. (2018). BlinkML: Efficient maximum likelihood estimation with probabilistic guarantees. *arXiv preprint arXiv:1812.10564*.

Pati, Y. C., Rezaiifar, R., and Krishnaprasad, P. S. (1993). Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, pages 40–44. IEEE.

Rao, C. R. (1948). Large sample tests of statistical hypotheses concerning several parameters with applications to problems of estimation. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 44, pages 50–57. Cambridge University Press.

Redden, D. T., Fernández, J. R., and Allison, D. B. (2004). A simple significance test for quantile regression. *Statistics in Medicine*, 23(16):2587–2597.

Satterthwaite, F. E. (1946). An approximate distribution of estimates of variance components. *Biometrics Bulletin*, 2(6):110–114.

Sikorska, K., Lesaffre, E., Groenen, P. F., and Eilers, P. H. (2013). GWAS on your notebook: fast semi-parallel linear and logistic regression for genome-wide association studies. *BMC Bioinformatics*, 14(1):166.

Tsagris, M., Alenazi, A., Verrou, K.-M., and Pandis, N. (2020a). Hypothesis testing for two population means: parametric or non-parametric test? *Journal of Statistical Computation and Simulation*, 90(2):252–270.

Tsagris, M., Papadovasilakis, Z., Lakiotaki, K., and Tsamardinos, I. (2018). Efficient feature selection on gene expression data: Which algorithm to use? *BioRxiv preprint https://www.biorxiv.org/content/10.1101/431734v1.abstract*.

Tsagris, M., Papadovasilakis, Z., Lakiotaki, K., and Tsamardinos, I. (2020b). A generalised OMP algorithm for feature selection with application to gene expression data. *arXiv preprint arXiv:2004.00281*.

Tsamardinos, I. and Aliferis, C. F. (2003). Towards principled feature selection: relevancy, filters and wrappers. In *AISTATS*.

Tsamardinos, I., Borboudakis, G., Katsogridakis, P., Pratikakis, P., and Christophides, V. (2019). A greedy feature selection algorithm for Big Data of high dimensionality. *Machine learning*, 108(2):149–202.

Weisberg, S. (1980). *Applied linear regression*. John Wiley & Sons.

Welch, B. L. (1951). On the comparison of several mean values: an alternative approach. *Biometrika*, 38(3-4):330–336.

Young, G. A. and Smith, R. (2005). *Essentials of Statistical Inference*. Cambridge University Press.