ℓ_1-Penalised Ordinal Polytomous Regression Estimators with Application to Gene Expression Studies

Stéphane Chrétien
National Physical Laboratory, Hampton Road, Teddington, United Kingdom
stephane.chretien@npl.co.uk

Christophe Guyeux
Computer Science Department, FEMTO-ST Institute, UMR 6174 CNRS, Université de Bourgogne Franche-Comté, 16 route de Gray, 25030 Besançon, France
christophe.guyeux@univ-fcomte.fr

Serge Moulin
Computer Science Department, FEMTO-ST Institute, UMR 6174 CNRS, Université de Bourgogne Franche-Comté, 16 route de Gray, 25030 Besançon, France
serge.moulin@univ-fcomte.fr

Abstract

Qualitative but ordered random variables, such as severity of a pathology, are of paramount importance in biostatistics and medicine. Understanding the conditional distribution of such qualitative variables as a function of other explanatory variables can be performed using a specific regression model known as ordinal polytomous regression. Variable selection in the ordinal polytomous regression model is a computationally difficult combinatorial optimisation problem which is however crucial when practitioners need to understand which covariates are physically related to the output and which covariates are not. One easy way to circumvent the computational hardness of variable selection is to introduce a penalised maximum likelihood estimator based on some well chosen non-smooth penalisation function such as, e.g., the ℓ_1-norm. In the case of the Gaussian linear model, the ℓ_1-penalised least-squares estimator, also known as LASSO estimator, has attracted a lot of attention in the last decade, both from the theoretical and algorithmic viewpoints. However, even in the Gaussian linear model, accurate calibration of the relaxation parameter, *i.e.*, the relative weight of the penalisation term in the estimation cost function is still considered a difficult problem that has to be addressed with caution. In the present paper, we apply ℓ_1-penalisation to the ordinal polytomous regression model and compare several hyper-parameter calibration strategies. Our main contributions are: (a) a useful and simple ℓ_1 penalised estimator for ordinal polytomous regression and a thorough description of how to apply Nesterov's accelerated gradient and the online Frank-Wolfe methods to the problem of computing this estimator, (b) a new hyper-parameter calibration method for the proposed model, based on the QUT idea of Giacobino et al. and (c) a code which can be freely used that implements the proposed estimation procedure.

2012 ACM Subject Classification Mathematics of computing → Regression analysis

Keywords and phrases LASSO, ordinal polytomous regression, Quantile Universal Threshold, Frank-Wolfe algorithm, Nesterov algorithm

Digital Object Identifier 10.4230/LIPIcs.WABI.2018.17

Funding Computations have been performed on the supercomputer facilities of the Mésocentre de calcul de Franche-Comté.

1 Introduction

Ordinal polytomous variables are of paramount importance in bioinformatics where practitioners may have to tackle qualitative but ordered data such as, e.g., the severity of a certain type of cancer [23], [24], [12], [13], [14], etc. Understanding how such variables can be explained by other variables such as, e.g., gene expressions, can help the research community investigate the influence of certain genes in the pathology under study. Oftentimes, only a small number of genes are relevant to the statistical modelling and variable selection needs to be performed in order to detect which of them should be ignored and which of them should not. The ordinal polytomous regression model is an adaptation of the classical regression model which is extremely well suited for this type of problem, and the goal of the present paper is to propose efficient approaches to the estimation and variable selection problems for this specific model.

1.1 When the number of covariates exceeds the number of observations: the blessing of sparsity

One important additional problem in standard gene expression studies is that the number of observations (e.g. patients) is often much smaller than the number of covariates (e.g. genes). In such cases, the problem cannot be expected to be solvable without some additional structure because the number of unknowns is larger than the number of observations. The main structural assumption which is usually made in such cases is that some sparsity property holds. In the example of gene expression analysis, it is usually considered natural to assume that only a small number of genes have significant influence on the output under study. Therefore, only a small number of regression coefficients should be nonzero in the estimator, although we cannot know before hand which are the ones which should be selected. Selecting the right variables in regression is often called "support recovery". Various approaches to variable selection have been proposed in the statistical literature. In practical applications, the most extensively used selection methods are the forward selection and the AIC/BIC information criteria based approaches [2], [22] [19]. Such methods however, can hardly be applied in situations where the number of covariates, e.g. genes, is large and one usually resorts to convex optimisation based strategies such as the LASSO [23] and its generalisations to nonlinear models [24], [25].

1.2 Previous work on variable selection via ℓ_1-norm penalisation

Convex optimisation based variable selection approaches are often based on penalised log-likelihood estimation, where the penalisation term is the ℓ_1-norm. In the linear model, it was discovered in [7] that under certain specific properties of the design matrix, known as the Restricted Isometry Property, the ℓ_1-norm penalised least ℓ_∞ estimator, aka the Dantzig estimator, would recover the location of the non-zero components exactly. This type of result, was then proven for the ℓ_1-penalised least ℓ_2 estimator, aka the LASSO estimator under weaker assumptions, including incoherence of the design matrix in [8]. The work [6] provided interesting alternative views on the statistical properties of the LASSO and Dantzig estimators which are still extensively used in the current literature on this topic.

Even when neither the Restricted Isometry Property nor the incoherence assumptions are satisfied, the mere computational tractability of ℓ_1-penalisation based estimators makes them the method of choice when the problem size is large.

1.3 The problem of hyper-parameter calibration

The main advantage of ℓ_1-based penalisation is to reduce the estimation problem to a convex optimisation one if the hyper-parameter, *i.e.* the relative weight associated with the ℓ_1-penalisation term, is calibrated to an appropriate value. In practice however, finding the right value for this hyper-parameter is often a complicated issue.

Most theoretical works come up with a formula for the hyper-parameter, see e.g. [8]. Such types of results are very important because they prove existence of a value of the hyper-parameter that will allow exact support recovery of the sparse regression vector under appropriate, e.g. incoherence assumptions of the design matrix. The theoretical value often gives the right order of dependencies with respect to the dimension of the problem, the standard deviation of the noise, and other important structural parameters, and is therefore a good indicator of how well conditioned the problem is, at least in theory.

In practice, however, the noise level is not known beforehand and therefore, hyper-parameter calibration cannot be performed without joint variance estimation. Reference [10] presents efficient methods for solving this joint estimation/calibration problem and present preliminary computational experiments showing practical relevance of the overall approach. The square-root LASSO [5] is another interesting alternative but is sometimes reported to have slightly worse performance in practice.

The usually preferred practical approach to hyper-parameter calibration is Cross Validation [3]. The Cross-Validation approach is very intuitive and had nice theoretical properties when the number of covariates is smaller than the number of observations. A drawback of Cross-Validation is the computational burden of re-sampling and computing the LASSO estimator a large number of times. Moreover, Cross-Validation is oriented towards prediction performance rather than accurate support selection. An alternative approach devised in [11], based on the Hedge algorithm of [16] and the stochastic Frank-Wolfe algorithm, was shown to outperform Cross Validation in terms of computational time for the linear model as well.

Recently, [17] devised a very efficient method called Quantile Universal Thresholding for hyper-parameter calibration in the linear model with a view towards efficient variable selection. Extensive numerical experiments provided in [17] show that Quantile Universal Thresholding outperforms Cross-Validation, although Cross-Validation has to be performed when the noise variance is unknown. Fortunately enough, recent work on fast variance estimation, as described e.g. in [18] or based on [11], should however allow to overcome the burden of using Cross-Validation as a subroutine in the Quantile Universal Thresholding procedure of [17].

1.4 Contributions of the paper

The main contributions of the present paper are threefold. The first is to present a ℓ_1-penalised maximum likelihood estimator for the ordered polytomous model and present efficient methods for computing this estimator. The second contribution is an efficient hyper-parameter calibration procedure based on recent work [17]. The last contribution is a freely available software implementation which can be downloaded online [1].

2 Methods

2.1 The model and the penalised estimator

2.1.1 The standard polytomous regression model

In the ordinal polytomous regression model, the independent qualitative output variables Y_i, $i = 1, \ldots, n$ with Q modalities $m_1, \ldots, m_Q$, are assumed to result from the quantification of a latent continuous variable $Y_i^* = X_i^t \beta^0 + \epsilon_i$, $i = 1, \ldots, n$, where X_i is a p-dimensional vector of covariates and where the residual ϵ_i has logistic cumulative distribution function $\Phi(y) = \frac{\exp(y)}{1+\exp(y)}$. More precisely, setting $-\infty = \gamma_0^0 < \cdots < \gamma_{Q-1}^0 < \gamma_Q^0 = +\infty$, we have $Y_i = m_q$ if and only if $Y_i^* \in]\gamma_{q-1}, \gamma_q]$. For $q = 1, \ldots, Q$, let us denote I_q the subset of {1...n} such that $i \in I_q$ if and only if $Y_i = m_q$. Let us denote by γ the vector $\gamma = (\gamma_1, \ldots, \gamma_{q-1})$. The conditional likelihood given $X_1, \ldots, X_n$ for this model is:

$$L_{Y|X}(\beta, \gamma) = \prod_{q=1}^{Q} \prod_{i \in I_q} \left(\Phi\left(X_i^t \beta - \gamma_{q-1}\right) - \Phi\left(X_i^t \beta - \gamma_q\right) \right). \tag{1}$$

where X is the $n \times p$ matrix such that X_i is its i^{th} row for all i in 1, ..., n.

The conditional log-likelihood is given by

$$l_{Y|X}(\beta, \gamma) = \sum_{i=1}^{n} \sum_{q=1}^{Q} 1_{\{Y_i = m_q\}} \log\left(\Phi\left(X_i^t \beta - \gamma_{q-1}\right) - \Phi\left(X_i^t \beta - \gamma_q\right) \right),$$

The parameters of this model are usually estimated using the maximum likelihood principle, *i.e.*, by finding the vector $(\hat{\beta}, \hat{\gamma})$ that maximizes $l_{Y|X}$. Maximization of the log-likelihood is made easy by the well known fact that the conditional log-likelihood function is concave.

The problem with this approach is that it cannot work when p is larger than n because, in this case, the Hessian matrix is easily shown to be singular. The situation where p is larger than n is however frequent in gene expression analysis as in many other problems, and one needs an estimator which can perform variable selection in such settings with low computational complexity. The next section introduces such an estimator based on ℓ_1 penalisation.

2.1.2 The penalised maximum likelihood estimator

One estimator of choice for the type of problem we just described (*i.e.* ordinal polytomous regression) is the ℓ_1-penalised maximum likelihood estimator given by

$$(\hat{\beta}, \hat{\gamma}) \quad \in \quad \operatorname{argmax}_{(b,c) \in \mathbb{R}^p \times \mathbb{R}^{Q-1}} \; l_{Y|X}(b, c) - \lambda \, \|b\|_1, \tag{2}$$

where λ is a relaxation parameter. This estimator corresponds exactly to the LASSO in the case where the log-likelihood is the one of the linear model. The main motivation for introducing this estimator is Theorem 1.2 in [8] about the LASSO. This theorem states that for a sufficiently sparse β in the linear model $Y = X\beta + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$, the risk of the LASSO estimator is near optimal, *i.e.* is comparable to the risk obtained with an oracle estimator which would know the support of β ahead of time. Moreover, support recovery is proved to hold with large probability for a vast majority of possible supports.

The assumptions in this theorem are the following:

1. X has low coherence, *i.e.* the maximum scalar product of two columns of X is less than $A_0 / \log(p)$;

2. the support and sign pattern of β have uniform distribution;
3. the nonzero components of β have magnitude above the noise level times a log factor.

It is therefore natural to expect that an appropriate translation of this result to the case of (ordinal or not) polytomous regression model will hold as well. In the sequel, we will present simulation based results on the penalised conditional likelihood estimator from the view point of variable selection.

2.2 Algorithms

2.2.1 Nesterov's algorithm

In [21, 20, 4], Nesterov introduced a new approach to convex minimization with possibly non-differentiable functions. Nesterov's method consists of smoothing the non-differentiable function and then applying a refined first order scheme to the problem. The main interest of this approach is that at iteration k, a bound of $O(1/k^2)$ on the error is guaranteed, whereas standard gradient methods only guarantee $O(1/k)$. Let us now describe a simple version of this method.

The first step is to smooth the ℓ_1-norm function. Notice that, for the vector β, $\|\beta\|_1$ can be written

$$\|\beta\|_1 = \max_{\|u\|_\infty \leq 1} u^t \beta, \tag{3}$$

and the maximizer in this expression is simply $\text{sign}(\beta)$, where $\text{sign}(\beta)$ is the vector with the component-wise signs of β. A possible simple smoothing of the ℓ_1-norm is given by

$$\ell_{1,\mu}(\beta) = \max_{\|u\|_\infty \leq 1} u^t \beta - \frac{\mu}{2}\|u\|_2^2. \tag{4}$$

Notice that the maximizer u^*_β in (4) exists due to continuity and coercivity, and is unique due to the strict convexity of $\|\cdot\|_2^2$. The main interesting feature of this smoothing is the following proposition.

▶ **Proposition 2.1.** *The function $\ell_{1,\mu}$ is differentiable with Lipschitz gradient. Moreover, the gradient is given by*

$$\nabla \ell_{1,\mu} = u^*_\beta \tag{5}$$

where u^*_β is the unique maximizer in (4) and the Lipschitz constant of the gradient is $L_1 = 1/\mu$.

Proof. See [20, Theorem 1]. ◀

With this result in hand, we can present Nesterov's accelerated gradient algorithm for smooth optimisation in Algorithm 1 below. In order to implement the algorithm, one needs to know the Lipschitz constant of the gradient of minus the log-likelihood, which is unknown, and the Lipschitz constant of the smoothed ℓ_1-norm penalty, which is $1/\mu$. In practice, the Lipschitz constant of the gradient of minus the log-likelihood can be estimated by random sampling and computing ratio between the norm of the difference between gradients at sampled points and the norm of the difference of these sample points.

Algorithm 1 Nesterov's algorithm for penalised log-likelihood estimation.

Input An initial point $\theta^{(0)} = (\beta^{(0)}, \gamma^{(0)})$, e.g. $\theta^{(0)} = 0$, the relaxation coefficient λ, the Lipschitz constants L_0 (resp. L_1) of the gradient of - the log-likelihood (resp. of $\ell_{1,\mu}$) and the maximum number of iterations $N \in \mathbb{N}_*$

for $k = 0...N-1$ **do**

Compute $g^{(k)} = \nabla\left(-l(\theta^{(k)}) + \lambda \ell_{1,\mu}(\beta^{(k)})\right)$

Compute $\theta^{(k,1)}$:

$$\theta^{(k,1)} = \operatorname{argmin}_{\tau \in \mathbb{R}^{p+Q-1}} \langle g^{(k)}, \tau - \theta^{(k)} \rangle + \frac{L_0 + L_1}{2} \|\tau - \theta^{(k)}\|_2^2.$$

Compute $\theta^{(k,2)}$:

$$\theta^{(k,2)} = \operatorname{argmin}_{\tau \in \mathbb{R}^{p+Q-1}} \Big(\sum_{0 \le k' \le k} \frac{1}{2(k'+1)} g^{(k')}, \tau - \theta^{(k')} \rangle \Big) + \frac{L_0 + L_1}{2} \|\tau - \theta^{(0)}\|_2^2.$$

Update $\theta^{(k+1)}$:

$$\theta^{(k+1)} = \frac{k+1}{k+3}\, \theta^{(k,1)} + \frac{2}{k+3}\, \theta^{(k,2)}.$$

end for

Output $\widehat{\theta}^{(N)}$.

2.2.2 The Frank-Wolfe algorithm

The Frank–Wolfe (FW) algorithm, proposed by Marguerite Frank and Philip Wolfe in 1956 [15], is another convex optimisation algorithm. The difference between the FW algorithm and the Nesterov one is that FW applies to constrained optimisation.

The main trick that is needed to implement the Frank-Wolfe algorithm is to reformulate the penalised problem

$$(\hat{\beta}, \hat{\gamma}) \in \operatorname{argmax}_{(b,c) \in \mathbb{R}^p \times \mathbb{R}^{Q-1}} \; l_{Y|X}(b,c) - \lambda \, \|b\|_1. \tag{6}$$

as a constrained optimisation problem

$$(\hat{\beta}, \hat{\gamma}) \in \operatorname{argmax}_{(b,c) \in \mathbb{R}^p \times \mathbb{R}^{Q-1}} \; l_{Y|X}(b,c) \text{ with } \|b\|_1 \le r \tag{7}$$

for an appropriate value of r. In this new formulation, the problem of choosing λ is translated into the problem of choosing r.

Generally speaking, each iteration of the FW algorithm consists of finding

$$s^k = \operatorname{argmin}_{\mathbf{s} \in \mathcal{D}} \mathbf{s}^{\mathrm{T}} \nabla \mathrm{f}(\mathbf{x}_\mathrm{k}),$$

then upgrade $\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{2}{k+2}(\mathbf{s}_k - \mathbf{x}_k)$, where f is the function to minimize, k is the current iteration, and $\mathcal{D}$ is the set on which we want to optimize f. In the case where $\mathcal{D}$ is the hypercube defined by $\|\beta\|_1 \le r$ (as in our case), determining s_k is simple, since it is the point:

- of coordinate r for the component such that $\nabla_\beta \, l_{\tilde{Y}|X}(\beta, \gamma)$ is minimal,
- and zero for all the other components.

However, the logistic regression is a special case in which constraints have to be put on β, but not on γ. Practically speaking, the choice has been to alternate iterations of the Frank-Wolfe algorithm (to optimize β with γ fixed) with a simple gradient descent (to optimize γ with β fixed).

Algorithm 2 Find a λ that cancels all β components following a dichotomy approach.

V: number of non-zero coefficients of β, as a function of λ.
δ: desired accuracy, set by the user (default value: $\delta = 0.01$).
$\lambda_{max} = 1$
while $\mathrm{V}(\lambda_{max}) \neq 0$ **do**
  $\lambda_{max} = \lambda_{max} \times 2$
end while
$\lambda_{min} = \frac{\lambda_{max}}{2}$
if $\lambda_{max} = 1$ **then**
  $\lambda_{min=0}$
end if
while $\lambda_{max} - \lambda_{min} \geq \delta$ **do**
  $\lambda_{mean} = \frac{\lambda_{max} - \lambda_{min}}{2}$
  if $V(\lambda_{mean}) = 0$ **then**
    $\lambda_{max} = \lambda_{mean}$
  else
    $\lambda_{min} = \lambda_{mean}$
  end if
end while
Output $\lambda_{\#} = \lambda_{mean}$.

2.3 Hyperparameter calibration

2.3.1 Selection of the parameter by AIC

The first implemented method to select the λ parameter is to use the Akaike information criterion (AIC) [2]. This AIC is a compromise between the likelihood of the model and the number of non-zero parameters. More precisely, $AIC = -2l_{Y|X}(\beta, \gamma) + 2\|\beta\|_0$, and the goal is to find a set of parameters that minimizes this value. The method of choosing lambda processes in three steps. In the first one, the objective to is determine a penalty $\lambda_{\#}$ that is large enough to cancel all β components. This objective is realized by using Algorithm 2.

One can then apply, e.g. Nesterov's or the stochastic Frank-Wolfe algorithm with different values of the hyperparameter. One possible set of values is $\lambda_0 = 0$, $\lambda_1 = \frac{\lambda_{\#}}{50}$, $\lambda_2 = \frac{2\times\lambda_{\#}}{50}$, ..., $\lambda_{49} = \frac{49\times\lambda_{\#}}{50}$. The AIC value is then computed for each obtained model and, at the end of the day, the model with smallest AIC is finally selected.

2.3.2 BIC Selection

λ is chosen in the same manner than for the AIC method, except for the fact that the value to optimize is, this time, $BIC = -2\ l_{Y|X}(\beta, \gamma) + \log(n)\|\beta\|_0$.

2.3.3 Adapting the Quantile Universal Threshold selection to ordinal polytomous regression

Quantile Universal Threshold (QUT) [17] is a simulation-based method. Its objective is to be sure that, if the vector Y to be predicted has no link with the matrix of predictive variables X, then the vector β of the regression coefficients will be the null vector with probability $1 - \alpha$, where α is set by the user (α is set to 5% in Section 3).

The working principle of QUT is as follows.

Algorithm 3 QUT : successive evaluations of gamma knowing lambda, and of lambda knowing gamma.

$\lambda = \sqrt{2} \times \log(2 \times \max(\mathrm{p}, 1)) \times \max(0.01, \mathrm{std}(\mathrm{Y}))$ (initialization of λ)
for i = 1 ... 3 **do**
  Choose γ based on the current λ with Nesterov.
  Choose λ based on the current γ with QUT.
end for
Output λ.

- Randomly pick a large number of vectors of the same size than Y. For instance, in the case study of Section 3, 100 vectors $\tilde{Y}_1 ... \tilde{Y}_{100}$ are picked as permutations of the original Y vector. That is to say, $\tilde{Y}_i$ has the same number of subjects in each category as the initial vector Y.
- For each random vector $\tilde{Y}_i$, find a λ_i large enough such that, when the ℓ_1-penalised maximum likelihood estimator described in Section 2.1.2 is optimized, β is the null vector.
- The obtained λ_i are sorted, and then we select the value such that a proportion $1 - \alpha$ of the λ_i is below this threshold.

To speed up the second step of this process, the following property is used: if $\lambda_\# = \|\nabla_\beta \, l_{\tilde{Y}|X}(\beta = \vec{0}, \gamma)\|_\infty$, then the optimisation of the ℓ_1-penalised maximum likelihood estimator with a penalty of $\lambda_\#$ returns $\beta = \vec{0}$. Please note that $\lambda_\#$ is not necessarily the smallest possible penalisation such as $\beta = \vec{0}$.
γ is required in order to compute $\lambda_\#$. However γ is not known, and λ is needed to calculate it. So, a loop has been implemented as in Algorithm 3.

Thanks to the shortcut $\lambda_\# = \|\nabla_\beta \, l_{\tilde{Y}|X}(\beta = \vec{0}, \gamma)\|_\infty$, the computation time to obtain λ is greatly reduced, leading to the fastest determination of λ (see Section 3), as it requires only a few the optimisation of the ℓ_1-penalised maximum likelihood estimator. Note that a version of the QUT whose second step is performed by dichotomy, as in Algorithm 2, has been implemented too, but it underperforms the other methods in terms of computation time.

2.3.4 Selection of the r parameter by Online Frank-Wolfe algorithm

The method follows the procedure described in [11] with small necessary adjustments in order to accommodate for the specific constraints associated with our estimator. We refer the reader to the associated longer report [9] for complete details.

3 Simulation results

3.1 Description of the experiments

We now assess the practical performance of the proposed methods. For this purpose, we performed various numerical experiments on simulated data. The simulation and testing procedure works as follows.

1. The number of subjects n, the number of variables p, the number of influential variables s, and the underlying threshold vector γ_0 are set (Section 3.2 contains the authors' choices).

2. The vector of underlying parameters β_0 is randomly picked. This vector is of size p such that $p - s$ of its components are null, while the other s components follow a Gaussian law $\mathcal{N}(0, 1)$
3. The matrix X of explanatory variables is then drawn. This is a matrix of size $n \times p$, in which each component follows a law $\mathcal{N}(0, 1)$.
4. The noise vector ϵ of Y^* is drawn. It is of size n, where each component follows a logistic(1,1) law.
5. $Y^* = X\beta_0 + \epsilon$ is computed, and then Y based on Y^* and γ_0.
6. Steps 2, 3, 4, and 5 above allows the construction of a database. They are repeated 50 times, leading to 50 different databases.
7. Each of these 50 databases is divided into a learning sample ($\frac{2}{3}$ of the subjects) and a testing one (the other third).
8. Each of the regression methods listed in Section 3.2 is finally applied to the 50 learning samples. The performances of the models are measured on the 50 corresponding test samples based on the criteria defined in Section 3.2.

3.2 Results

The methods we decided to compare are the following.

- λ parameter selection by AIC as in Section 2.3.1.
- λ parameter selection by BIC as in Section 2.3.2.
- λ parameter selection according to Quantile Universal Threshold, as presented in Section 2.3.3.
- The use of the Frank-Wolfe algorithm, to solve the constrained optimization with selection of the r parameter using Online Frank-Wolfe, as defined in Sections 2.2.2 and 2.3.4. This model is named "OFW" in Tables 1 and 2.
- The absence of variable selection. That is to say, the model obtained when the likelihood is maximized without penalty. This model is simply named "$\lambda = 0$" in Tables 1 and 2.
- The model that predicts, for each subject in the test sample, the largest category of the learning sample. It is named "null model" in Tables 1 and 2, as this is the best possibility if no explanatory variable is taken into account.

$\lambda = 0$ and the null model are only performed to check if the first four methods work well. Indeed, when dealing with the logistic regression, it is important to check if the predictive model is better than simply placing all patients in the majority category. Moreover, when working on variables selection, it can be useful to check if the obtained model is better than the one with no selection.

Two experiments have been performed. In the first one, $n > p$, there are 50 variables, the learning sample has been constituted by 200 subjects, while the test sample has 100 subjects (see Table 1). In the other experiment, $p > n$, the learning sample has 100 subjects, the test one has 50 subjects, and there are 200 variables (Table 2). In both cases, the number of significant variables was set to $s = 5$.

We considered $Q = 3$ categories for Y, and we set $\gamma_0 \in [0, 3]$, as unbalanced categories were wanted to complicate the regression problem. With this choice of γ_0, Y_i is in the first category for all $Y_i^* \leq 0$, *i.e.*, for half of the simulated subjects. The Nesterov algorithm runs for 200 iterations, while the Franck-Wolfe one iterates 200 times.

For each method, four performance criteria are studied.

Table 1 Monte Carlo simulations with $n_{\text{learning}} = 200, p = 50, n_{\text{test}} = 100$.

choice of λ (or r)	correctly ranked	average likelihood	prediction error	CRW	Time	λ (or r)	Nb of variables
BIC	66.7	0.48	0.35	59.4	2464.9	14.8	3.9
QUT $\|\|\|_\infty$	65.4	0.48	0.36	57.9	53.0	22.0	2.6
AIC	65.3	0.47	0.36	58.6	2458.9	10.2	7.1
OFW	63.3	0.46	0.39	55.2	199.0	7.2	39.6
$\lambda = 0$	60.0	0.44	0.43	55.3	20.1	0.0	50.0
null model	49.0	-	-	33.3	0	-	0

- The percentage of subjects in the test sample which are correctly ranked by the model fitted on the learning sample. This percentage is named "correctly ranked" in Tables 1 and 2.
- The average likelihood. That is, the geometric mean of the probabilities that the model fitted to the learning sample assigns the actual categories of subjects in the test sample. This is what we called "average likelihood" in Tables 1 and 2.
- The average prediction error. That is to say, the average gap between the predicted category and the actual category, named "prediction error" in Tables 1 and 2.
- The percentage of correctly ranked subjects, weighted by the size of the categories. More precisely, we calculate $100 \times \sum_{i=1}^{n} 1_{\text{prediction is right}} \frac{Q \times p}{\#I_{Y_i}}$, where $\#I_{Y_i}$ is the number of subjects in the same category than Y_i. This criterion attaches greater importance to the proper classification of subjects that are in a poorly represented category. It is referenced as "CRW" in Tables 1 and 2.

The "average likelihood" and "correctly-ranked weighted" criteria are relevant when classes are very unbalanced (like 98 %, 1 %, and 1 %), which can really occur in practice. In the case study, the "correctly ranked" criterion has been considered first, as this is probably the most natural criterion for not too unbalanced categories like the ones used during our simulations. Tables 1 and 2 are sorted according to this criterion.

Table 1 summarizes the results in the case where the number of subjects in the training sample is 200, the number of subjects in the testing one is 100, and the number of explanatory variables is 50. Table 2, summarizes the results in the case where the number of subjects in the training sample is 100, the number of subjects in the testing one is 50, and the number of explanatory variables is 200.

To finish describing Tables 1 and 2, let us note that "nb of variables" represents the number of variables that the model considers as influential.

Wilcoxon tests have also been performed in order to determine if the differences between the methods are statistically significant. Tables 3 and 4 show the results of these Wilcoxon tests. In the $n_{\text{learning}} = 200, p = 50$ case, the difference between correctly ranked subjects for BIC (66,7%) and QUT (65,4%) is significant with a p-value of $8,03 \times 10^{-3}$, even if this difference is only equal to 1,3%. Conversely, in the $n_{\text{learning}} = 100, p = 200$ case, the difference between QUT, BIC, OFW is not significant. This case may require more simulated data if we want to separate these methods correctly. Finally, in any cases, QUT, BIC, OFW, and AIC are significantly better than λ =0 and the null model.

Table 2 Monte Carlo simulations with $n_{\text{learning}} = 100, p = 200, n_{\text{test}} = 50$.

choice of λ (or r)	correctly ranked	average likelihood	prediction error	CRW	Time	λ (or r)	Nb of variables
QUT $\|\|\|_\infty$	61.0	0.43	0.42	52.4	33.5	16.9	1.3
BIC	60.7	0.44	0.42	54.0	982.9	11.9	3.5
OFW	59.8	0.43	0.42	50.2	72.4	6.4	54.1
AIC	55.4	0.42	0.48	50.1	995.8	8.9	9.2
null model	48.1	-	-	33.3	0	-	0
$\lambda = 0$	36.7	0.22	0.77	40.0	12.8	0.0	200.0

Table 3 Paired Wilcoxon tests associated to Monte Carlo simulations with $n_{\text{learning}} = 200, p = 50, n_{\text{test}} = 100$.

	QUT$\|\|\|_\infty$	AIC	OFW	$\lambda = 0$	null model
BIC	8.03×10^{-3}	3.69×10^{-3}	7.83×10^{-7}	1.71×10^{-9}	7.38×10^{-10}
QUT$\|\|\|_\infty$	-	7.03×10^{-1}	3.36×10^{-3}	9.54×10^{-8}	7.83×10^{-10}
AIC	-	-	8.99×10^{-4}	2.02×10^{-8}	7.32×10^{-10}
OFW	-	-	-	1.58×10^{-6}	7.44×10^{-10}
$\lambda = 0$	-	-	-	-	1.95×10^{-9}

Table 4 Paired Wilcoxon tests associated to Monte Carlo simulations with $n_{\text{learning}} = 100, p = 200, n_{\text{test}} = 50$.

	BIC	OFW	AIC	null model	$\lambda = 0$
QUT	6.74×10^{-1}	3.77×10^{-1}	2.93×10^{-4}	$8,66 \times 10^{-9}$	9.13×10^{-10}
BIC	-	4.86×10^{-1}	5.47×10^{-4}	$1,62 \times 10^{-8}$	1.32×10^{-9}
OFW	-	-	2.66×10^{-5}	$1,17 \times 10^{-8}$	1.31×10^{-9}
AIC	-	-	-	$1,87 \times 10^{-5}$	3.33×10^{-9}
null model	-	-	-	-	$6,95 \times 10^{-7}$

4 Discussion

First of all, the four variable selection methods work better than $\lambda = 0$ and the null model. This shows that the algorithms work correctly, and that variable selection is useful. The absence of variable selection is particularly harmful in the case where $p > n$, see Table 2. It makes sense because, in this case, the optimisation of the unpenalised likelihood allows an infinite number of solutions. This $p > n$ case is very common in practice.

In the experiment shown in Table 1, the BIC works a bit better than the other methods, while in the experiment summarized in Table 2, QUT, BIC, and OFW are very close. In terms of computation time, QUT is the most interesting approach. Indeed, as explained in Section 2.3.3, this method allows to choose λ by executing the regression only a few times.

5 Conclusion

The present paper proposed a new estimator for sparse ordinal polytomous regression in a high dimensional setting together with a strategy for hyper-parameter calibration based on previous results from [17]. Performance of the method was assessed via extensive numerical experiments. The forthcoming report [9] will include further implementation details, and improvements, and additional numerical results on large real datasets.

References

1 Our python module. Accessed: 2018-05-11. URL: `https://github.com/SergeMOULIN/l1-penalised-ordinal-polytomous-regression-estimators`.

2 Hirotogu Akaike. Information theory and an extension of the maximum likelihood principle. In *Selected Papers of Hirotugu Akaike*, pages 199–213. Springer, 1998.

3 Sylvain Arlot, Alain Celisse, et al. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79, 2010.

4 Stephen Becker, Jérôme Bobin, and Emmanuel J Candès. Nesta: A fast and accurate first-order method for sparse recovery. *SIAM Journal on Imaging Sciences*, 4(1):1–39, 2011.

5 Alexandre Belloni, Victor Chernozhukov, and Lie Wang. Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika*, 98(4):791–806, 2011.

6 Peter J Bickel, Ya'acov Ritov, Alexandre B Tsybakov, et al. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009.

7 Emmanuel Candes, Terence Tao, et al. The dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6):2313–2351, 2007.

8 Emmanuel J Candès, Yaniv Plan, et al. Near-ideal model selection by $\ell 1$ minimization. *The Annals of Statistics*, 37(5A):2145–2177, 2009.

9 Stephane Chretien, Guyeux Christophe, and Serge Moulin. l1-penalised ordinal polytomous regression estimators. *arXiv preprint to be submitted*, 2018.

10 Stéphane Chrétien and Sébastien Darses. Sparse recovery with unknown variance: a lasso-type approach. *IEEE Transactions on Information Theory*, 60(7):3970–3988, 2014.

11 Stephane Chretien, Alex Gibberd, and Sandipan Roy. Hedging hyperparameter selection for basis pursuit. *arXiv preprint arXiv:1805.01870*, 2018.

12 Stephane Chretien, Christophe Guyeux, Michael Boyer-Guittaut, Regis Delage-Mouroux, and Francoise Descotes. Investigating gene expression array with outliers and missing data in bladder cancer. In *Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on*, pages 994–998. IEEE, 2015.

13 Stéphane Chrétien, Christophe Guyeux, Michael Boyer-Guittaut, Régis Delage-Mouroux, and Françoise Descôtes. Using the lasso for gene selection in bladder cancer data. *arXiv preprint arXiv:1504.05004*, 2015.

14 Stéphane Chrétien, Christophe Guyeux, Bastien Conesa, Régis Delage-Mouroux, Michèle Jouvenot, Philippe Huetz, and Françoise Descôtes. A bregman-proximal point algorithm for robust non-negative matrix factorization with possible missing values and outliers-application to gene expression analysis. *BMC bioinformatics*, 17(8):284, 2016.

15 Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics (NRL)*, 3(1-2):95–110, 1956.

16 Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

17 Caroline Giacobino, Sylvain Sardy, Jairo Diaz-Rodriguez, and Nick Hengartner. Quantile universal threshold for model selection. *arXiv preprint arXiv:1511.05433*, 2015.

18 Christopher Kennedy and Rachel Ward. Greedy variance estimation for the lasso. *arXiv preprint arXiv:1803.10878*, 2018.

19 Alan Miller. *Subset selection in regression*. CRC Press, 2002.

20 Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

21 Yurii Nesterov. A method of solving a convex programming problem with convergence rate o (1/k2). *Soviet Mathematics Doklady*, pages 372–376, 1983.

22 Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

23 Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

24 Robert Tibshirani. The lasso method for variable selection in the cox model. *Statistics in medicine*, 16(4):385–395, 1997.

25 Sara A Van de Geer et al. High-dimensional generalized linear models and the lasso. *The Annals of Statistics*, 36(2):614–645, 2008.