Second Order Proximal Methods Applied to Elastic Net Penalised Vector Generalised Linear Models

Bertrand Nortier

A thesis submitted for the degree of
Master of Science by Research

University of Oxford
May 1, 2016

Abstract

The Vector Generalised Linear Model (VGLM) framework extends Generalised Linear Models (GLMs) to a large number of univariate and multivariate statistical models. The object of this thesis is to study the estimation of the maximum elastic net penalised log-likelihood of VGLM models. As the elastic net penalty has a separable non-differentiable part, second-order proximal methods are considered. For VGLMs, depending on the model, it may be more convenient to use the Fisher information matrix instead of the Hessian. Hence, we propose a proximal Fisher scoring method. Two examples are then investigated. The first example is an application of an elastic net penalised ordinal probit model to the prediction of mid-market price changes for tick-by-tick Limit Order Book data. The second example is an application of an Expectation Maximisation (EM) proximal Newton/Fisher scoring algorithm to variable selection for a bivariate Poisson regression model applied to health care data.

Contents

List of Symbols **3**

I Proximal Fisher Scoring for Vector Generalised Linear Models **7**

1 Introduction **8**

2 Literature **9**

3 Vector Generalised Linear Model with Elastic Net Penalty **10**
3.1 VGLM Framework 10
3.1.1 General Response 10
3.1.2 Log-likelihood and Fisher Scoring 11
3.1.3 Iteratively Reweighted Least Squares (IRLS) 12
3.2 Log-concavity, Log-convexity, Penalisation and Convention 12
3.3 Elastic Net Penalised VGLM 13

4 Second Order Proximal Methods Applied to Penalised VGLM **14**
4.1 Definitions 14
4.2 Interpretation 14
4.3 Proximal Newton-Type Methods and Proximal Fisher Scoring Algorithms 15
4.3.1 Proximal Newton-Type Method 15
4.3.2 Proximal Fisher Scoring 16
4.4 Convergence of Proximal Newton-Type Methods 16
4.5 Proximal Fisher Scoring in the Case of VGLMs and Equivalence with Penalised IRLS 17

5 Conclusion and Next Steps **18**

A Solving the Proximal Fisher Operator Using Coordinate Descent **20**

B Proximal Fisher Scoring Backtracking Line Search **21**

C Assumptions for Convergence **21**

II Regularised Ordinal Probit Applied to the Prediction of High Frequency Financial Data **22**

1 Introduction **23**

2 Literature Review **23**

3 Ordinal Probit Regularised with an Elastic Net: Model and Comparison of Estimation Methods **24**
3.1 Ordinal Probit 24
3.2 Calculating the Regularisation Path for the Penalised Ordinal Probit 25
3.3 Comparison of Different Estimation Methods 26

3.4 Example . . . 26

4 Application to High Frequency Financial Data 28
4.1 Characteristics of HFFD and Dataset . . . 28
4.2 Gain/Loss Function and Strategy . . . 29
4.3 Regression Settings . . . 29

5 Discussion and Future Steps 30

A Other Estimation Methods 33
A.1 Smooth Approximation of the Non-Differentiable Function . . . 33
A.2 Other Proximal Algorithms . . . 33
A.2.1 Solving the Inner Loop of the Proximal Fisher Scoring Using Quadratic Programming . . . 33
A.2.2 Forward Backward/Proximal Gradient/ISTA . . . 35
A.2.3 FISTA . . . 36

B Backtracking Line Search Algorithms 37
B.1 Backtracking Line Search for Smooth Approximation . . . 37
B.2 Backtracking Line Search for ISTA/FISTA . . . 37

C Multivariate Generalised Linear Models (MVGLM) 38

D Ordinal Probit Matrices in the MVGLM Framework 39

E Generated Example 40

F Positive Definite Matrix and Condition Number 41

G Gain/Loss Function and Strategy 41

III EM Proximal Newton/Fisher Scoring Method for Bivariate Poisson Regression with Application to Health Care Data 43

1 Introduction 44

2 Bivariate Poisson Regression and EM proximal Fisher Scoring 44
2.1 Bivariate Poisson Regression . . . 44
2.2 EM-Proximal Fisher Scoring Algorithm for Elastic Net Penalised Bivariate Poisson . 45

3 Application of the EM Proximal Newton/Fisher Scoring to Health Care Data 47
3.1 Data . . . 47
3.2 Penalised Regression . . . 47

4 Conclusion and Future Steps 48

A Score Vector and Fisher Information Matrix for a Poisson Regression 50

B Data 50

List of Symbols

α elastic net parameter, Part I, page 13

$\mathbf{B}$ matrix of coefficients of the covariates, Part I, page 10

$\boldsymbol{\beta}$ all coefficients of $\boldsymbol{\beta}_0$ and $\mathbf{B}$ re-written as a vector, Part I, page 10

$\boldsymbol{\beta}_0$ vector of intercepts, Part I, page 10

$\boldsymbol{\beta}_{prox}$ solution of the proximal operator for proximal Newton or proximal Fisher scoring, Part I, page 16

c smooth approximation constant, Part II, page 33

c_1 smooth approximation backtracking line search parameter, Part II, page 37

C closed convex set in $\mathbb{R}^n$, Part I, page 14

d number of points of the regularisation path, Part II, page 26

δ factor used to decrease the step in the backtracking line search for proximal Fisher scoring, quadratic approximation, ISTA and FISTA, Part I, page 21

∂ subdifferential, Part I, page 21

$\Delta\boldsymbol{\beta}$ $\Delta\boldsymbol{\beta} = \boldsymbol{\beta}_{prox} - \boldsymbol{\beta}$, Part I, page 16

$\mathbf{D}_i$ in the MVGLM model, derivative of the link function with respect to the linear predictors, Part II, page 38

$diag(.)$ $diag(vector)$ is the diagonal matrix with diagonal entries equal to the vector values and all other entries equal to 0, Part II, page 33

$E[.]$ expected value, Part I, page 11

$\mathbf{E}$ domain of $\boldsymbol{\beta}$, Part I, page 21

$\boldsymbol{\eta}_i$ vector of linear predictors, $i = 1, \ldots, n$, Part I, page 10

$\eta_{(j)i}$ linear predictor $j,\ j = 1, ..., M$ for observation $i,\ i = 1, ..., n$, Part I, page 10

f function equal to the negative log-likelihood plus elastic net penalty, Part I, page 13

$\mathbf{F}$ Fisher information matrix, Part I, page 11

f_1 function equal to the negative log-likelihood plus the ridge part of the elastic net penalty, Part I, page 13

f_2 lasso part of the elastic net penalty, Part I, page 13

$\mathbf{F}_{conv}$ Fisher information matrix of the negative log-likelihood, Part I, page 12

$\mathbf{F}_{f_1}$ Fisher information matrix of function f_1, Part I, page 13

$\mathbf{F}_i$ Fisher information matrix for observation $i,\ i = 1, \ldots, n$, Part I, page 11

$\mathbf{F}^{\lambda}$	Fisher information matrix of the concave log-likelihood minus the ridge penalty part of the elastic net penalty, Part I, page 13
g	conditional distribution of response $\mathbf{Y}_i,\ i = 1, ..., n$, Part I, page 10
$\mathbf{G}_a^{\lambda}$	$\mathbf{G}_a^{\lambda} = \left\{\mathbf{H}_{\mathbf{a}}^{\lambda}\boldsymbol{\beta}^{(\mathbf{a})} - \mathbf{U}_{\mathbf{a}}^{\lambda}\right\}$, Part I, page 16
γ_k	Poisson parameters of $W_0,\ W_1,\ W_2$, k=0, 1, 2, Part III, page 45
$\mathbf{G}_{\mathbf{F},\mathbf{a}}^{\lambda}$	$\mathbf{G}_{\mathbf{F},\mathbf{a}}^{\lambda} = \left\{\mathbf{F}_{\mathbf{a}}^{\lambda}\boldsymbol{\beta}^{(\mathbf{a})} + \mathbf{U}_{\mathbf{a}}^{\lambda}\right\}$, Part I, page 16
h	link function, Part I, page 10
$\mathbf{H}_{f_1}$	Hessian of function f_1, Part I, page 13
$\mathbf{H}^{\lambda}$	Hessian of the concave log-likelihood minus the ridge penalty part of the elastic net penalty, Part I, page 13
$\mathcal{I}$	indicator function. $\mathcal{I}(condition)$ is equal to 1 if *condition* is true, 0 otherwise, Part I, page 13
K	factor used in the backtracking line search for proximal Fisher scoring, Part I, page 21
κ	maximum condition number of the Fisher information matrix used as a constraint in Tong et al. [110], Part II, page 41
L	step parameter notation for proximal gradient, using the notation of Beck and Teboulle [11], Part II, page 35
l	log-likelihood, Part I, page 11
λ	regularisation parameter, Part I, page 13
λ_{max}	maximum lambda of the regularisation path, Part II, page 26
λ_{min}	minimum lambda of the regularisation path, Part II, page 26
l_{conv}	negative log-likelihood, Part I, page 12
l_i	log-likelihood for observation $i,\ i = 1, ..., n$, Part I, page 11
M	number of linear predictors, Part I, page 10
$\vee$	maximum of two elements, Part II, page 34
$\wedge$	minimum of two elements, Part II, page 34
$\boldsymbol{\mu}_i$	vector of means of the multivariate generalised linear model, Part II, page 38
n	number of observations, Part I, page 10
Ω	domain of function h, Part I, page 10
p	number of covariates, Part I, page 10
P_C	projection on closed convex set C, Part I, page 14
Φ	standard normal CDF, Part II, page 24
ϕ	standard normal PDF, Part II, page 40
$\pi_{(j)i}$	$\pi_{(j)i} = \Phi\left(\eta_{(j)i}\right) - \Phi\left(\eta_{(j-1)i}\right)$, Part II, page 24

$prox_f$ proximal mapping of function f, Part I, page 14

$prox_f^{\mathbf{H}}$ scaled proximal mapping using positive definite matrix $\mathbf{H}$ of function f, Part I, page 14

ψ dispersion parameter, Part I, page 10

Q number of responses, Part I, page 10

s_a FISTA accelerated step parameter, Part II, page 37

$\mathbf{\Sigma}_i$ in the MVGLM model, covariance matrix of the vector of responses $\mathbf{Y}_i$, Part II, page 38

$sign()$ sign function, Part I, page 13

$\succeq$ using the notation of Boyd and Vandenberghe, p.698 [19], if $\mathbf{H_1}, \mathbf{H_2}$ are two symmetric matrices, then $\mathbf{H_1} \succeq \mathbf{H_2}$ refers to the matrix inequality, i.e. ($\mathbf{H_1} - \mathbf{H_2}$) is positive semidefinite. If $\boldsymbol{\beta}$ is a vector, then $\boldsymbol{\beta} \succeq 0$ indicates that each element of the vector is positive, Part I, page 21

$\mathbf{T}$ partial identity matrix, Part I, page 13

t_a proximal Newton or proximal Fisher scoring step size, Part I, page 16

$\boldsymbol{\theta}_i$ vector of parameters of the multivariate generalised linear model, Part II, page 38

T_{jj} j^{th} diagonal element of matrix $\mathbf{T}$, Part I, page 21

^{T} transpose operator, Part I, page 10

$\mathbf{U}$ score vector, Part I, page 11

$\mathbf{U}_{conv}$ score vector of the negative log-likelihood, Part I, page 12

$\mathbf{U}_{f_1}$ score of function f_1, Part I, page 13

$\mathbf{U}_i$ score vector for observation i, Part I, page 11

$\mathbf{u}_i$ derivative of the log-likelihood with respect to the linear predictors observation $i, i = 1, \ldots, n$, Part I, page 11

$\mathbf{U}^{\lambda}$ score of the concave log-likelihood minus the ridge penalty part of the elastic net penalty, Part I, page 13

v_a first proximal gradient step parameter, using the forward-backward algorithm described in Combettes and Pesquet [29], Part II, page 35

w_a second proximal gradient step parameter, using the forward-backward algorithm described in Combettes and Pesquet [29], Part II, page 35

$\mathbf{W}_i$ working weights matrix $i = 1, \ldots, n$, Part I, page 11

W_k independent Poisson distributed random variables, k=0, 1, 2, Part III, page 45

w_k values of observations of W_0, W_1, W_2, k=0,1,2, Part III, page 45

$\mathbf{X}_i$ design matrix $i = 1, \ldots, n$, Part I, page 11

$\mathbf{x}_i$ vector of covariates, $i = 1, ..., n$, Part I, page 10

ξ_k eigenvalue or singular value of the Fisher information matrix after having floored the eigenvalues or singular values to ϵ_4, Part II, page 41

$\mathbf{Y}_i$ vector of responses, $i = 1, ..., n$, Part I, page 10

$\mathbf{y}_i$ vector of observations, $i = 1, ..., n$, Part I, page 10

$\mathbf{z}_i$ modified response (terminology used by Yee [130] for VGLMs) or working response (terminology used by Fahrmeir and Tutz [33], Tutz [113] for MVGLMs), $i = 1, ..., n$, Part I, page 12

Part I

Proximal Fisher Scoring for Vector Generalised Linear Models

1. Introduction

The aim of this document is to study methods used to compute the penalised maximum likelihood estimator of vector generalised linear models (VGLMs) (Yee and Wild [132], Yee [130]). The VGLM framework vastly extends generalised linear models (GLM) (Nelder and Wedderburn [79]) by modelling multiple correlated responses simultaneously and also by assuming that the distribution of each response may not be in the exponential family.

The main advantage of studying penalised VGLMs is that the VGLM framework encompasses many existing penalised univariate and multivariate statistical models that have been individually studied and a few core algorithms would suffice to estimate all these penalised models. Many of these penalised models contained in the penalised VGLM class are described in section 2.

In this document, the elastic net penalty (Zou and Hastie [142]) is investigated. The elastic net penalty is one of the most commonly used penalties. Similarly to the LASSO (Tibshirani [109]), the elastic net performs variable selection and shrinkage. An advantage of the elastic net over the LASSO is that it drives the coefficients of correlated covariates to similar values that is not the case for the LASSO.

The elastic net penalty is a combination of a ridge and a lasso penalty and, therefore, the sum of the VGLM log-likelihood and elastic net penalty is a non-differentiable function. Then, Fisher scoring, the traditional method that is used to estimate parameters via maximum likelihood estimation (MLE) for VGLMs, cannot be used. To deal with the optimisation of a non-differentiable function, proximal methods are used. Proximal methods are based on proximal operators (Moreau [77]). In some cases that include elastic net penalised (locally) concave VGLMs, where the function to optimise can be decomposed in a series of smooth (locally) convex functions and non-smooth (locally) convex functions, proximal methods may be used to find the minimum of this non-differentiable convex function. The two main types of proximal methods are first order proximal methods that only require the knowledge of the gradient of the smooth part of the function to optimise and second order methods that use both the first and second derivative (or a positive definite matrix that plays the role of a Hessian) of the smooth part of the function to optimise. The Fisher scoring algorithm used for non-penalised VGLMs consists in replacing the observed information matrix of the Newton-Raphson algorithm by the Fisher information matrix that is a positive semidefinite matrix (Newton-Raphson and Fisher scoring are identical only in cases where the observed information matrix is identical to the Fisher information matrix (for example in the case of "canonical link GLMs")). This availability of second order information for VGLMs can be used to accelerate the speed of convergence of the proximal algorithm by using a second order proximal method. Second order proximal methods have been studied in Lee et al. [69] who discuss proximal Newton-type methods. As Fisher scoring is a Newton-type method (see Kass and Vos pp.79-80 [56]), the framework defined by Lee et al. [69] can be applied.

Lee et al. [69] note that elastic net penalised generalised linear models (Friedman et al. [39]) and new elastic net by Yuan et al. [135] are special cases of the proximal Newton method. In fact, we here note that most methods used to estimate penalised models that fit in the penalised VGLM framework (section 2) are similar to proximal Newton or proximal Fisher scoring with a step of 1 and no backtracking line search. In the list of references given in section 2, proximal Newton/Fisher scoring or equivalent algorithms are referred to as: '(penalised) iteratively reweighted least squares', '(penalised) local quadratic approximation', 'regularised weighted least squares', 'Co-

ordinate Descent Newton'[1], 'second order approximation'. All these denominations can be grouped under the terminologies 'proximal Newton-type', 'proximal Newton' or 'proximal Fisher scoring'. The advantages of using proximal methods are that:

- First, we have some convergence results for proximal Newton-type methods, relying on the work of Lee et al. [68], [69].
- Second, based on their convergence proof, Lee et al. [68], [69] developed a novel sufficient decrease condition for backtracking line search.
- Last, there seems to be a lot of interest on proximal methods and by recognising the link between lasso penalised IRLS and proximal Fisher scoring, we can benefit from developments in proximal methods.

In the next section, we discuss the following points. First, in section 2, we give some of the penalised models that are part of the penalised VGLM class. Then in section 3.1, we give details of the penalised VGLM framework. Last in section 4, we discuss proximal Fisher scoring applied to elastic net penalised VGLMs.

2. Literature

As mentioned previously, the penalised VGLM framework generalises a series of regularised models that have been studied previously. They include:

- **Generalised linear models (GLMs)** (Fan and Li [34], Park and Hastie [91], Friedman et al. [39], Tutz chapter 6 [113], Dhurandar et al. [31]). GLMs have been studied especially in the case of the logistic regression (Lee et al. [70], Yuan et al. [135], Bian et al. [15]).
- **Multinomial regression** (Simon et al. [103], Tutz et al. [114], Mauerer et al. [75][2]).
- **Proportional odds/ordinal regression** (Lu and Zhang [73], Drießlein [32], Archer et al. [4], Hou [50]).
- **Continuation ratio model** (Archer and Williams [5]).
- **Count and zero inflated models** (Buu et al. [24], Zeng et al. [138], Wang et al. [123], Wang et al. [124]).
- **Cox proportional hazard model** (Yang and Zou [129], Simon et al. [104], Goeman [41]).
- **Accelerated failure model** (Khan and Shaw [61], Zhang et al. [139]).
- **Censored regression** (Ahmed et al. [2]).
- **Vector autoregressive model** (Furman [40]).

The penalised VGLM class contains not only the examples given previously but many others including multivariate categorical response models, cumulative generalised linear models, multivariate

[1] we solve the proximal operator using coordinate descent and therefore, this corresponds to the coordinate descent Newton as described in Yuan et al. [134], Bian et al. [15], [16].

[2] Tutz et al. [114] and Mauerer et al. [75] use the FISTA algorithm (Beck and Teboulle [11]) that is an accelerated first order proximal method. It can be seen as a proximal Newton-type method where we replace the Hessian approximation with a constant times the identity matrix.

survival regression, multivariate extreme value regressions, the Bradley-Terry model, copulas regressions, etc. (see Yee [131]). The work in this document therefore unifies the estimation of a vast array of penalised statistical models.

3. Vector Generalised Linear Model with Elastic Net Penalty

This section follows Yee p.3 [130]. The development of GLMs by Nelder and Wedderburn [79] came from the realisation that many different common statistical models were examples of a more general class where it is assumed that the distribution of the response is in the exponential family and the expected value of the response conditional on the knowledge of covariates is a non linear function of a linear combination of the covariates. Examples of GLMs include linear regression, Poisson regression, logistic regression, probit regression, multinomial logistic regression, etc.

Although the GLM framework has been very successful, two main limitations are first, that multiple responses can be modelled but only if the model can be decomposed into a series of univariate GLMs and second, that there is a large number of distributions that do not fall in the confines of the exponential family.

To address these two issues, Yee and Wild [132] and Yee [130] propose an extended framework: Vector Generalised Linear Models (VGLMs). In the VGLM framework, the conditional distribution of the responses is directly modelled as a non-linear function of a series of M different linear combinations of the covariates. The non-linear function defining the multivariate response is not specified. This is the reason why VGLMs can accommodate a much greater number of models than the GLM framework. A disadvantage of VGLMs is that the expressions of scores, Hessian and Fisher information matrices are more generic than in the case of GLMs as there is no assumption of exponential family.

In this part, the VGLM framework (Yee and Wild [132], Yee [130]) is first introduced (section 3.1), then we discuss some convention in the notation in section 3.2. Finally, the problem of the penalised VGLM framework is introduced in section 3.3.

3.1. VGLM Framework

In section 3.1.1, the VGLM framework is introduced. Then, in sections 3.1.2 and 3.1.3, the Fisher scoring and Iteratively Reweighted Least Squares for VGLMs are discussed.

3.1.1. General Response

The following description and notation follows mostly Yee [130], chapter 3. In the VGLM framework, we directly assume that the conditional distribution g of $\mathbf{Y}_i$ follows a given distribution h:

$$g\left(\mathbf{y}_i|\mathbf{x}_i, \boldsymbol{\beta}_0, \mathbf{B}, \psi\right) = h\left(\mathbf{y}_i, \boldsymbol{\eta}_{(1)i}, ..., \boldsymbol{\eta}_{(M)i}, \psi\right). \tag{3.1}$$

where:

- There are p covariates, Q responses, M linear predictors. There are M intercepts and therefore $(M + Mp)$ coefficients for the covariates + intercepts.
- $\mathbf{Y}_i$ is the response of size $[Q \times 1], i = 1, ..., n$. $\mathbf{y}_i$ is the observed response.
- h is a function: $\Omega \subseteq \mathbb{R}^{(Q+M+1)} \rightarrow (\mathbb{R}^+)$.

- ψ is a dispersion parameter.
- $\boldsymbol{\beta}_0 = \left(\beta_{(1)0}, ..., \beta_{(M)0}\right)^T$ is a vector of intercepts .
- $\mathbf{B} = \left(\boldsymbol{\beta}_{(1)}, ..., \boldsymbol{\beta}_{(M)}\right) = \begin{bmatrix} \beta_{(1)1} & \cdots & \beta_{(M)1} \\ \vdots & \ddots & \vdots \\ \beta_{(1)p} & \cdots & \beta_{(M)p} \end{bmatrix}$ of size $[p \times M]$ is the matrix of coefficients.
- $\boldsymbol{\beta} = \left(\beta_{(1)0}, \boldsymbol{\beta}_{(1)}^T, \beta_{(2)0}, \boldsymbol{\beta}_{(2)}^T, ..., \beta_{(M)0}, \boldsymbol{\beta}_{(M)}^T\right)$ are all the coefficients re-written as a vector.
- In part I and in the Appendix of part II, the elements of vector $\boldsymbol{\beta}$ will also be called β_j, $j = 1, ..., (M + Mp)$ or β_k, $k = 1, ..., (M + Mp)$ to simplify notations. Then $\beta_1 = \beta_{(1)0}, \beta_2 = \beta_{(1)1}, ..., \beta_p = \beta_{(1)p}, \beta_{p+1} = \beta_{(2)0}, ...$
- $\eta_{(j)i} = \beta_{(j)0} + \boldsymbol{\beta}_{(j)}^T \mathbf{x}_i = \beta_{(j)0} + \sum_{k=1}^{p} \beta_{(j)k} x_{ik}, \quad j = 1, ..., M.$
- $\boldsymbol{\eta}_i = \boldsymbol{\beta}_0 + \mathbf{B}^T \mathbf{x}_i$ is of size $[M \times 1]$.

Note that in the original framework by Yee [130], chapter 3, the vector of intercepts $\boldsymbol{\beta}_0$ is not considered separately from other parameters as Yee considers that the first column of $\mathbf{x}_i$ is a column of 1s.

3.1.2. Log-likelihood and Fisher Scoring

The following description and notation follows Yee [130], chapter 3. The log-likelihood is the sum of the individual log-likelihood for each observation:

$$l(\beta) = \sum_{i=1}^{n} l_i\left(\boldsymbol{\eta}_i\left(\boldsymbol{\beta}\right)\right) = \sum_{i=1}^{n} \log\left[h\left(\mathbf{y}_i, \boldsymbol{\eta}_i, \psi\right)\right]. \tag{3.2}$$

Note that the log-likelihood of each observation can also be weighted. The main algorithm to estimate the parameters in the VGLMs is the Fisher scoring algorithm. In the Fisher scoring algorithm, the observed information matrix that is used for the Newton-Raphson algorithm is replaced with the Fisher information matrix:

$$\boldsymbol{\beta}^{(a+1)} = \boldsymbol{\beta}^{(a)} + \left[\mathbf{F}\left(\boldsymbol{\beta}^{(a)}\right)\right]^{-1} \mathbf{U}\left(\boldsymbol{\beta}^{(a)}\right). \tag{3.3}$$

where:

- $\mathbf{F} = \sum_i \mathbf{F}_i, \mathbf{U} = \sum_i \mathbf{U}_i$.
- $\mathbf{F}_i$ is the Fisher information matrix of elements $\left(\mathbf{F}_i\right)_{(jk)} = -E\left[\frac{\partial^2 l_i}{\partial \beta_j \partial \beta_k}\right]$.
  $\mathbf{F}_i$ is of size $[(M + Mp) \times (M + Mp)]$.
- $\mathbf{U}_i$ is the score vector of elements $\left(\mathbf{U}_i\right)_j = \frac{\partial l_i}{\partial \beta_j}$ $\mathbf{U}_i$ is of size $[(M + Mp) \times 1]$.
- $\mathbf{F}_i = \mathbf{X}_i^T \mathbf{W}_i \mathbf{X}_i$.
- $\mathbf{X}_i = \left[\mathbf{I}_{M \times M} | \mathbf{x}_i^T \otimes \mathbf{I}_{M \times M}\right] = \begin{bmatrix} 1 & 0 & \cdots & 0 & \mathbf{x}_i^T & \mathbf{0}_{[1\times p]} & \cdots & \mathbf{0}_{[1\times p]} \\ 0 & 1 & \ddots & \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & \ddots & \ddots & \mathbf{0}_{[1\times p]} \\ 0 & \cdots & 0 & 1 & 0 & \cdots & \mathbf{0}_{[1\times p]} & \mathbf{x}_i^T \end{bmatrix}.$

- $\mathbf{X}_i$ is of size $[M \times (M + Mp)]$.
- $(\mathbf{W}_i)_{(jk)} = -E\left[\frac{\partial^2 l_i}{\partial \eta_{(j)i} \partial \eta_{(k)i}}\right]$ $\mathbf{W}_i$ is of size $[M \times M]$. $\mathbf{W}_i$ is the working weights matrix.
- $\mathbf{U}_i = \mathbf{X}_i^T \mathbf{u}_i = \mathbf{X}_i^T \begin{bmatrix} \frac{\partial l_i}{\partial \eta_{(1)i}} \\ \cdots \\ \cdots \\ \frac{\partial l_i}{\partial \eta_{(M)i}} \end{bmatrix}$.

3.1.3. Iteratively Reweighted Least Squares (IRLS)

The Fisher scoring algorithm for VGLMs is equivalent to the solution of a series of weighted least squares problems. This is referred to as Iteratively Reweighted Least Squares (IRLS):

$$\begin{aligned} \boldsymbol{\beta}^{(a+1)} &= \boldsymbol{\beta}^{(a)} + \left[\sum_{i=1}^{n} \mathbf{X}_i^T \mathbf{W}_i^{(a)} \mathbf{X}_i\right]^{-1} \sum_{i=1}^{n} \mathbf{X}_i^T \mathbf{W}_i^{(a)} \left(\mathbf{W}_i^{(a)}\right)^{-1} \mathbf{u}_i^{(a)} \\ &\Leftrightarrow \begin{cases} \boldsymbol{\beta}^{(a+1)} = \left[\sum_{i=1}^{n} \mathbf{X}_i^T \mathbf{W}_i^{(a)} \mathbf{X}_i\right]^{-1} \left[\sum_{i=1}^{n} \mathbf{X}_i^T \mathbf{W}_i^{(a)} \mathbf{z}_i^{(a)}\right] \\ \mathbf{z}_i^{(a)} = \left[\mathbf{X}_i \boldsymbol{\beta}^{(a)} + \left(\mathbf{W}_i^{(a)}\right)^{-1} \mathbf{u}_i^{(a)}\right] \end{cases} . \end{aligned} \tag{3.4}$$

We see that obtaining the estimated parameters via maximum likelihood consists in iteratively obtaining solutions to weighted least squares problems, using working weights matrices $\mathbf{W}_i^{(a)}$ as weights matrices and the modified response $\mathbf{z}_i^{(a)}$ as the response. Then from Yee, chapter 3 [130] the iterative weighted least square problem to solve can be written:

$$\boldsymbol{\beta}^{(a+1)} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \frac{1}{2n} \sum_i \left(\mathbf{z}_i^{(a)} - \mathbf{X}_i \boldsymbol{\beta}\right)^T \mathbf{W}_i^{(a)} \left(\mathbf{z}_i^{(a)} - \mathbf{X}_i \boldsymbol{\beta}\right) \right\}. \tag{3.5}$$

3.2. Log-concavity, Log-convexity, Penalisation and Convention

Wedderburn [126] showed that in the case of univariate generalised linear models, the likelihood is log-concave. In the case of VGLM, although many of the responses implemented are in the exponential family, as distribution h is not specified, there is the possibility to have non-exponentially distributed responses and/or non concave log-likelihoods. In this document, we assume that the likelihood is log-concave or at least locally log-concave. Note that as the Fisher scoring algorithm finds a zero of the score function, the algorithm would converge to a global minimum or maximum if the log-likelihood were convex or concave and would converge to a local minimum/maximum if the function were locally convex/concave.

Throughout this document, we use the following convention: the log-likelihood $l(\boldsymbol{\beta}) = \sum_i l_i(\boldsymbol{\beta})$ is concave. $\mathbf{F}$ is the Fisher information matrix of l. $\mathbf{U}$ is the score vector of l. We note that if we were to optimise $l_{conv} = -l$ associated with $\mathbf{F}_{conv} = -\mathbf{F}$ and $\mathbf{U}_{conv} = -\mathbf{U}$, then the Fisher scoring algorithm would identical as:

$$\boldsymbol{\beta}^{(a+1)} = \boldsymbol{\beta}^{(a)} + \left[-\mathbf{F}_{conv}\left(\boldsymbol{\beta}^{(a)}\right)\right]^{-1} \left[-\mathbf{U}_{conv}\left(\boldsymbol{\beta}^{(a)}\right)\right] = \boldsymbol{\beta}^{(a)} + \left[\mathbf{F}_{conv}\left(\boldsymbol{\beta}^{(a)}\right)\right]^{-1} \left[\mathbf{U}_{conv}\left(\boldsymbol{\beta}^{(a)}\right)\right].$$

In the next section, we briefly discuss penalising the log-likelihood and note that the (positive) penalty must be subtracted if the function is concave and added if the function is convex.

3.3. Elastic Net Penalised VGLM

From Tutz, chapter 6 [113], Park [90], Van Der Kooij, chapter 4 [116], amongst others, the penalised problem to solve is:

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= \underset{\boldsymbol{\beta}}{\operatorname{argmax}} \left(l\left(\boldsymbol{\beta}\right) - \frac{\lambda\left(1-\alpha\right)}{2}\boldsymbol{\beta}^T\mathbf{T}\boldsymbol{\beta} - \lambda\alpha\boldsymbol{\beta}^T\mathbf{T}sign\left(\boldsymbol{\beta}\right)\right) \\ &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(-l\left(\boldsymbol{\beta}\right) + \frac{\lambda\left(1-\alpha\right)}{2}\boldsymbol{\beta}^T\mathbf{T}\boldsymbol{\beta} + \lambda\alpha\boldsymbol{\beta}^T\mathbf{T}sign\left(\boldsymbol{\beta}\right)\right). \\ &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(f\left(\boldsymbol{\beta}\right)\right)\end{aligned} \tag{3.6}$$

where $sign(\boldsymbol{\beta}) = (sign(\beta_1), ..., sign(\beta_{M+Mp}))$ and $sign\left(\beta_j\right) = \mathcal{I}\left(\beta_j > 0\right) - \mathcal{I}\left(\beta_j < 0\right)$, $\mathcal{I}$ is the indicator function. Similarly to Tutz, p.234 [113] in the multinomial ridge regression case, we must define a partial identity, $\mathbf{T}$, that is a $[(M+Mp)\times(M+Mp)]$ matrix. $\mathbf{T}$ is an identity matrix where we have a zero coefficient for each intercept. The intercepts are gathered in vector $\boldsymbol{\beta}_0$:

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_{0,[(p+1)\times(p+1)]} & \mathbf{0}_{[(p+1)\times(p+1)]} & \cdots & \mathbf{0}_{[(p+1)\times(p+1)]} \\ \mathbf{0}_{[(p+1)\times(p+1)]} & \mathbf{I}_{0,[(p+1)\times(p+1)]} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0}_{[(p+1)\times(p+1)]} \\ \mathbf{0}_{[(p+1)\times(p+1)]} & \cdots & \mathbf{0}_{[(p+1)\times(p+1)]} & \mathbf{I}_{0,[(p+1)\times(p+1)]} \end{bmatrix}. \tag{3.7}$$

$\mathbf{I}_{0,[(p+1)\times(p+1)]}$ is an identity matrix with a first column of zero of size $(p+1)\times(p+1)$ as defined by Tutz p.234. Note that the definition above is different from matrix $\mathbf{T}$ defined in Tutz, p.234 [113].

We note that the penalised negative log-likelihood is the sum of three functions: the negative log-likelihood that is smooth and convex, the ridge penalty that is smooth and convex and the lasso penalty that is non-smooth and convex. From Combettes and Pesquet [29], the negative log-likelihood is grouped with the ridge penalty in function f_1 and the lasso is allocated to a second function f_2. Then the function to optimise is the sum of function f_1 that is smooth and convex and f_2 that is non-smooth and convex. This framework is exactly the setting that is appropriate for proximal methods:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(f_1\left(\boldsymbol{\beta}\right) + f_2\left(\boldsymbol{\beta}\right)\right); \quad f_1\left(\boldsymbol{\beta}\right) = \left[-l\left(\boldsymbol{\beta}\right) + \frac{\lambda\left(1-\alpha\right)}{2}\boldsymbol{\beta}^T\mathbf{T}\boldsymbol{\beta}\right]; \quad f_2\left(\boldsymbol{\beta}\right) = \lambda\alpha\boldsymbol{\beta}^T\mathbf{T}sign\left(\boldsymbol{\beta}\right). \tag{3.8}$$

When naming quantities such as Hessian, Fisher information matrix etc., we always refer to the original log-likelihood $l(\boldsymbol{\beta})$ rather than its opposite $-l(\boldsymbol{\beta})$. We use the following notation:

- $\mathbf{H}^{\lambda} = \frac{\mathbf{1}}{\mathbf{n}}\sum \mathbf{H_i} - \lambda\left(\mathbf{1}-\alpha\right)\mathbf{T}$ is the Hessian of the concave log-likelihood minus penalty.
- $\mathbf{F}^{\lambda} = -E\left[\frac{1}{n}\sum \mathbf{H_i} - \lambda\left(\mathbf{1}-\alpha\right)\mathbf{T}\right] = \frac{1}{n}\sum \mathbf{F_i} + \lambda\left(\mathbf{1}-\alpha\right)\mathbf{T}$ is the Fisher information matrix of the concave log-likelihood minus penalty. $E[\,]$ denotes the expected value.
- $\mathbf{U}^{\lambda} = \frac{\mathbf{1}}{\mathbf{n}}\sum \mathbf{U_i} - \lambda\left(\mathbf{1}-\alpha\right)\mathbf{T}\boldsymbol{\beta}$ is the score of the concave log-likelihood minus penalty.

Hence:

$$\mathbf{H}_{f_1} = -\mathbf{H}^{\lambda}; \quad \mathbf{F}_{f_1} = -\mathbf{F}^{\lambda}; \quad \mathbf{U}_{f_1} = -\mathbf{U}^{\lambda}. \tag{3.9}$$

Therefore, $\mathbf{H}, \mathbf{F}, \mathbf{U}$ are the Hessian, Fisher information matrix and score vector of the original log-likelihood $l(\boldsymbol{\beta})$ and not of function f_1 or function f_2.

Now that the problem to solve has been set and as it has been noted that this type of problems can be solved using proximal methods, proximal methods are introduced in the next section. Note that for the rest of this document, f_1 will refer to the smooth part of the convex function to optimise and f_2 will refer to the non-differentiable part of the convex function to optimise.

4. Second Order Proximal Methods Applied to Penalised VGLM

Proximal operators were first defined by Moreau [77], [78]. Proximal methods are convex optimisation methods based on proximal operators. They apply to functions that can be decomposed into a sum of convex smooth functions and the sum of convex non-smooth functions. Proximal methods are fast and can be run in parallel (see Parikh and Boyd [89]). In this section, we explain how proximal methods can be used to estimate penalised VGLMs. In 4.1, proximal operators are defined, in 4.2 an interpretation of proximal operators is given. In 4.3 proximal Newton-type and proximal Fisher scoring algorithms are discussed. Finally, in 4.4, there is a brief discussion on convergence.

4.1. Definitions

Definition 4.1 (Proximal Mapping)**.**
From Lee et al. [68], the proximal mapping of a convex function f at point $\boldsymbol{x}$ is:

$$prox_f(\boldsymbol{x}) = \underset{\boldsymbol{y}}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{x}\|^2 + f(\boldsymbol{y}) \right\}. \tag{4.1}$$

where $\|.\|$ is a norm.

Definition 4.2 (scaled proximal mapping)**.**
From Lee et al. [68]

$$prox_f^{\mathbf{H}}(\boldsymbol{x}) = \underset{\boldsymbol{y}}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{x}\|_{\mathbf{H}}^2 + f(\boldsymbol{y}) \right\} = \underset{\boldsymbol{y}}{\operatorname{argmin}} \left\{ \frac{1}{2} (\boldsymbol{y} - \boldsymbol{x})^T \mathbf{H} (\boldsymbol{y} - \boldsymbol{x}) + f(\boldsymbol{y}) \right\}. \tag{4.2}$$

where $\mathbf{H}$ is a positive definite matrix.

4.2. Interpretation

This section is a based on section (1.2) of Parikh et al. [89] p.124-126 and the interpretation of Combettes and Pesquet [29]. The proximal operator is an extension of the concept of projection onto a convex set. From Combettes and Pesquet [29], the projection of a point $\mathbf{x} \in \mathbb{R}^n$ onto a closed convex subset of $C \subset \mathbb{R}^n$ is the unique point $P_C(\mathbf{x}) \in C$ that minimises the distance between $\mathbf{x}$ and C for some distance d:

$$d_C(\mathbf{x}) = \underset{\mathbf{y} \in C}{\operatorname{argmin}} \|\mathbf{x} - \mathbf{y}\| = \|\mathbf{x} - P_C(\mathbf{x})\|. \tag{4.3}$$

Combettes and Pesquet [29] explain that the projection is equivalent to:

$$P_C(\mathbf{x}) = \underset{\mathbf{y} \in \mathbb{R}^n}{\operatorname{argmin}} \left\{ \imath(\mathbf{y} \in C) + \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|^2 \right\}. \tag{4.4}$$

where:

$$\imath(\mathbf{y} \in C) = \begin{cases} 0 & \mathbf{y} \in C \\ +\infty & \mathbf{y} \notin C \end{cases}. \tag{4.5}$$

The proximal operator is a generalisation of this concept where the indicator function is replaced by a generic convex function that is lower semi-continuous. We then obtain the definition:

$$prox_f(\mathbf{x}) = \underset{\mathbf{y}}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|^2 + f(\mathbf{y}) \right\}. \tag{4.6}$$

If the norm is the L_2 norm, then we obtain the classical proximal operator definition that uses the squared Euclidean distance:

$$prox_f(\mathbf{x}) = \underset{\mathbf{y}}{\operatorname{argmin}} \left\{ \frac{1}{2} (\mathbf{y} - \mathbf{x})^T (\mathbf{y} - \mathbf{x}) + f(\mathbf{y}) \right\}. \tag{4.7}$$

If the squared distance used is the squared Mahalanobis distance (see for example Grauman et al. [45]), calling the inverse covariance matrix $\mathbf{H}$, then we obtain the scaled proximal mapping operator defined in Lee et al. [68]:

$$prox_f^{\mathbf{H}}(\mathbf{x}) = \underset{\mathbf{y}}{\operatorname{argmin}} \left\{ \frac{1}{2} (\mathbf{y} - \mathbf{x})^T \mathbf{H} (\mathbf{y} - \mathbf{x}) + f(\mathbf{y}) \right\}. \tag{4.8}$$

Another interpretation of proximal operators is that the smooth part of the function to optimise is replaced by a local quadratic (or weighted quadratic) approximation around point $\mathbf{x}$ and the non-differentiable part is kept with no change. This sum of local quadratic (or weighted quadratic) approximation plus non-smooth part is then optimised.

For further information on proximal operators and methods, see Parikh and Boyd [89], Combettes and Pesquet [29], Becker and Fadili [12], Polson et al. [93], Bertsekas [14], Hastie et al. [47] chapter 5. In the next section, the proximal Newton algorithm is first introduced. This is followed by the proximal Fisher scoring that can be applied to the VGLM framework.

4.3. Proximal Newton-Type Methods and Proximal Fisher Scoring Algorithms

Proximal Newton-type methods have been defined in Lee et al. [68], [69]. Proximal Newton-type methods are based on scaled proximal mapping operators (equation (4.2)). In proximal Newton-type methods, the scaling matrix is any positive-definite matrix that has the same dimension as the Hessian/observed information matrix. Proximal Fisher scoring is defined as a proximal Newton-type algorithm where the scaling matrix used is the Fisher information matrix. Yee p.93, pp.277-278, p.538 [130] points out that the advantage of using the Fisher information matrix instead of the observed information matrix is that the Fisher information matrices and working weight matrices for each individual observation is positive semidefinite. Another reason to use the Fisher information matrix instead of the observed information matrix is simply that sometimes it is easier to compute the Fisher information matrix than the Hessian. In section 4.3.1 and 4.3.2, we introduce the proximal Newton and proximal Fisher scoring algorithm. Then in section 4.4, we discuss the convergence of the exact proximal Newton-type methods and inexact proximal Fisher scoring.

4.3.1. Proximal Newton-Type Method

As mentioned previously, the proximal Fisher scoring algorithm is a proximal Newton-type method. In this section, we discuss proximal Newton-type methods and, in the next section, the specific case of proximal Fisher scoring. From Lee et al. [68] (see also Lee et al. [69]), the proximal Newton-type iteration is based on the scaled proximal mapping operator. As previously discussed, matrix $\mathbf{H}$ and vector $\mathbf{U}$ will refer to the original likelihood. As the proximal Newton algorithm starts from the

assumption that the function is convex, $-\mathbf{H}$ and $-\mathbf{U}$ are used in the equations. Following Lee et al. [68], proximal Newton-type methods consist first, in calculating a Newton-type descent direction for the smooth function f_1. Then, if the function to optimise was only f_1, starting from point $\boldsymbol{\beta}^{(a)}$, the algorithm would descend to $\left(\boldsymbol{\beta}^{(a)} - \left[\mathbf{H}_{f_1}|_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(a)}}\right]^{-1}\left[\mathbf{U}_{f_1}|_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(a)}}\right]\right)$. The scaled proximal operator using nonsmooth function f_2 is then applied to this new point. Finally, the decrease amount is multiplied by a scaling factor $0 < t_a \leq 1$. This factor can be determined using backtracking line search. The main algorithm loop can be re-written as follows:

$$\begin{aligned}\boldsymbol{\beta}^{(a+1)} &= \boldsymbol{\beta}^{(a)} + t_a\left\{prox_{f_2}^{\mathbf{H}_{f_1}|_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(a)}}}\left(\boldsymbol{\beta}^{(a)} - \left[\mathbf{H}_{f_1}|_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(a)}}\right]^{-1}\left[\mathbf{U}_{f_1}|_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(a)}}\right]\right) - \boldsymbol{\beta}^{(a)}\right\} \\ &= \boldsymbol{\beta}^{(a)} + t_a\left\{\underset{\boldsymbol{\beta}}{\operatorname{argmin}}\left\{-\frac{1}{2}\boldsymbol{\beta}^T\mathbf{H}_a^\lambda\boldsymbol{\beta} + \left[\boldsymbol{\beta}^{(a)}\right]^T\mathbf{H}_a^\lambda\boldsymbol{\beta} - \left[\mathbf{U}_a^\lambda\right]^T\boldsymbol{\beta} + f_2\left(\boldsymbol{\beta}\right)\right\} - \boldsymbol{\beta}^{(a)}\right\}\end{aligned}. \tag{4.9}$$

Then the key step is to find the solution of the proximal operator that we call $\boldsymbol{\beta}_{prox}^{(a)}$:

$$\begin{aligned}\boldsymbol{\beta}_{prox}^{(a)} &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\left\{-\frac{1}{2}\boldsymbol{\beta}^T\mathbf{H}_a^\lambda\boldsymbol{\beta} + \left[\mathbf{G}_a^\lambda\right]^T\boldsymbol{\beta} + f_2\left(\boldsymbol{\beta}\right)\right\}. \\ \mathbf{G}_a^\lambda &= \left\{\mathbf{H}_{\mathbf{a}}^\lambda\boldsymbol{\beta}^{(\mathbf{a})} - \mathbf{U}_{\mathbf{a}}^\lambda\right\}.\end{aligned} \tag{4.10}$$

The following notation will also be used:

$$\Delta\boldsymbol{\beta}^{(a)} = \boldsymbol{\beta}_{prox}^{(a)} - \boldsymbol{\beta}^{(a)}. \tag{4.11}$$

Hence, the algorithm can be written as:

$$\boldsymbol{\beta}^{(a+1)} = \boldsymbol{\beta}^{(a)} + t_a\Delta\boldsymbol{\beta}^{(a)}. \tag{4.12}$$

4.3.2. Proximal Fisher Scoring

In the proximal Fisher scoring, we replace the observed information matrix by the Fisher information matrix. The advantage is that the Fisher information matrix is positive semidefinite and we may have analytic expressions of the Fisher information matrix for some of the models. In the proximal Fisher scoring algorithm, the proximal operator is:

$$\begin{aligned}\boldsymbol{\beta}_{prox}^{(a)} &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\left\{\frac{1}{2}\boldsymbol{\beta}^T\mathbf{F}_a^\lambda\boldsymbol{\beta} - \left[\mathbf{G}_{F,a}^\lambda\right]^T\boldsymbol{\beta} + f_2\left(\boldsymbol{\beta}\right)\right\}. \\ \mathbf{G}_{\mathbf{F},\mathbf{a}}^\lambda &= \left\{\mathbf{F}_{\mathbf{a}}^\lambda\boldsymbol{\beta}^{(\mathbf{a})} + \mathbf{U}_{\mathbf{a}}^\lambda\right\}.\end{aligned} \tag{4.13}$$

4.4. Convergence of Proximal Newton-Type Methods

The convergence for exact proximal Newton-type methods is proved in Lee et al. [69] under certain conditions given in Appendix C. Proximal Newton-type algorithms are deemed exact when each sub-problem is solved exactly and inexact when an error remains at each iteration.

Exact proximal Newton-type methods: Global convergence[3] for exact proximal Newton-type methods (that includes exact proximal Fisher scoring) is proved in Lee et al. [69]. A key result of Lee et al. is:

[3] convergence is deemed global when the algorithm converges to a stationary point for any starting point in the domain of definition of the function and convergence is deemed local when the algorithm converges to a stationary point as long as the starting point is sufficiently close to the stationary point.

Theorem 4.3 (Theorem 3.1 from Lee et al. [69]). *If f is a closed convex function and $\inf_\beta\{f(\beta)|\beta \in dom(f)\}$ is attained at β^*. If $H_a \succeq mI$, $m > 0$, and the proximal Newton step is solved exactly, then $\beta^{(a)}$ converges to an optimal solution starting from any $\beta_{start} \in dom(f)$.*

Lee et al. then prove the local convergence order of exact proximal Newton and quasi-Newton methods. The criterion necessary to prove the convergence order of quasi-Newton method (the Dennis-Moré criterion) is not verified in the case of proximal Fisher scoring. Therefore the convergence order of proximal Fisher scoring should to be investigated. Literature that could be used include Argyros and Magreñán [7] who discuss local convergence of proximal Gauss-Newton (Gauss-Newton method is closely related to Fisher scoring (see Osborne [84], Wang [120])) and articles that discuss the estimation of convergence order for smooth unconstrained or inequality constrained Newton-type methods or Fisher scoring (Osborne [83], Devidas and George [30], Ortega and Rheinboldt [82], Argyros [6], Wang [121]).

Inexact proximal Newton-type methods: Convergence of general inexact proximal Newton-type methods and inexact proximal Fisher scoring seems to have not yet been investigated. However, some specific cases have been studied. Lee et al. [69] investigate convergence speed of inexact proximal Newton. The case of inexact proximal quasi-Newton is discussed in Scheinberg and Tang [99]. Complexity of inexact proximal Fisher scoring may not have been investigated yet.

4.5. Proximal Fisher Scoring in the Case of VGLMs and Equivalence with Penalised IRLS

In this section, we apply the framework defined previously to the case of elastic net penalised VGLMs. First, the expressions of $\mathbf{U}_a^\lambda$ and $\mathbf{G}_{F,a}^\lambda$ in the case of elastic net penalised VGLMs are:

$$f_1(\boldsymbol{\beta}) = -l(\boldsymbol{\beta}) + \frac{\lambda(1-\alpha)}{2}\boldsymbol{\beta}^T\mathbf{T}\boldsymbol{\beta}. \tag{4.14}$$

$$\mathbf{U}_a^\lambda = \mathbf{U}_a - \lambda(1-\alpha)\mathbf{T}\boldsymbol{\beta}^{(a)} = \left(\frac{1}{n}\sum_{i=1}^n \mathbf{X}_i^T\mathbf{u}_i^{(a)}\right) - \lambda(1-\alpha)\mathbf{T}\boldsymbol{\beta}^{(a)}.$$

$$\begin{aligned}\mathbf{U}_a^\lambda &= -\left(\frac{1}{n}\sum_{i=1}^n \mathbf{X}_i^T\mathbf{W}_i^{(a)}\mathbf{X}_i + \lambda(1-\alpha)\mathbf{T}\right)\boldsymbol{\beta}^{(a)} + \frac{1}{n}\sum_{i=1}^n \mathbf{X}_i^T\mathbf{W}_i^{(a)}\mathbf{z}_i^{(a)} \\ &= -\mathbf{F}_a^\lambda\boldsymbol{\beta}^{(a)} + \frac{1}{n}\sum_{i=1}^n \mathbf{X}_i^T\mathbf{W}_i^{(a)}\mathbf{z}_i^{(a)}\end{aligned}. \tag{4.15}$$

We see that $\mathbf{G}_{F,a}$ in the case of VGLMs for an elastic net penalty is independent of λ and α as:

$$\mathbf{G}_{\mathbf{F},a}^\lambda = \left\{\mathbf{F}_a^\lambda\boldsymbol{\beta}^{(a)} + \mathbf{U}_a^\lambda\right\} = \frac{1}{n}\sum_{i=1}^n \mathbf{X}_i^T\mathbf{W}_i^{(a)}\mathbf{z}_i^{(a)}. \tag{4.16}$$

Then in the case of VGLMs, the proximal Fisher operator is:

$$\begin{aligned}\boldsymbol{\beta}_{prox}^{(a)} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\Bigg\{&\frac{1}{2n}\sum_i\left(\boldsymbol{\beta}^T\mathbf{X}_i^T\mathbf{W}_i^{(a)}\mathbf{X}_i\boldsymbol{\beta} - 2\left[\mathbf{z}_i^{(a)}\right]^T\mathbf{W}_i^{(a)}\mathbf{X}_i\boldsymbol{\beta}\right) \\ &+\frac{\lambda(1-\alpha)}{2}\boldsymbol{\beta}^T\mathbf{T}\boldsymbol{\beta} + \lambda\alpha\left[sign(\boldsymbol{\beta})\right]^T\mathbf{T}\boldsymbol{\beta}\Bigg\}\end{aligned}. \tag{4.17}$$

As mentioned previously, in proximal Newton-type methods, first, a Newton-type descent is applied to the smooth function f_1. Then a weighted local quadratic approximation is made around this point and the non-smooth function is added. This is equivalent to adding a penalty to the IRLS of equation (3.5). To see this, we note that the proximal Fisher operator can be re-written:

$$\begin{aligned}\boldsymbol{\beta}_{prox}^{(a)} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \Bigg\{ & \frac{1}{2n} \sum_i \left(\mathbf{z}_i^{(a)} - \mathbf{X}_i \boldsymbol{\beta} \right)^T \mathbf{W}_i^{(a)} \left(\mathbf{z}_i^{(a)} - \mathbf{X}_i \boldsymbol{\beta} \right) \\ & + \frac{\lambda (1-\alpha)}{2} \boldsymbol{\beta}^T \mathbf{T} \boldsymbol{\beta} + \lambda \alpha \left[sign\left(\boldsymbol{\beta} \right) \right]^T \mathbf{T} \boldsymbol{\beta} \Bigg\}\end{aligned} . \tag{4.18}$$

Similar expressions can be found for example in Friedman et al. [38], Friedman et al. [39], Breheny et al. [21], Breheny et al. [20] and Van der Kooij [116] chapter 4 in case of a univariate response. The proximal operator can be solved using coordinate descent (see Appendix A). The backtracking line search for the proximal Fisher scoring is given in Appendix B.

5. Conclusion and Next Steps

We presented a proximal Fisher scoring algorithm for elastic-net penalised Vector Generalised Linear Models. Our presentation has some limitations as there are a number of subjects that we have not investigated and that could be the object of further research.

- Elastic Irrepresentable Condition (EIC): From Yuan and Lin [136], a path is deemed consistent if it contains at least one point such that both the coefficient estimate and variable selection are consistent. Yuan and Lin [136] then gives the EIC that needs to be verified for such a point to exist. Preconditioning has been explored to alleviate this problem in the case of the linear model with LASSO (Jia et al. [52], Wauthier et al. [125]). Jia et al. [52] note that preconditioning could be investigated for IRLS/GLM.

- Path algorithms: It may be possible to design an algorithm that would use the estimated parameter values at a given λ to the next λ on the path. This could speed up the calculations. Approaches proposed recently include Yuan and Zou [137], Yu and Feng [133], Hu et al. [51], Augugliaro et al. [8]. It may also be useful to obtain the active set (see Keskar et al. [60], Solntsev et al. [106]).

- Extension to Vector Generalised Additive Models (VGAMs): VGAMs (Yee and Wild [132]) extend VGLMs by replacing the linear predictors by sums of nonlinear functions of each covariate. Penalised univariate Generalised Additive Models (GAMs) have been investigated in Avalos Fernandez [9], Wood and Marra [74], Chouldechova and Hastie [28].

In the next chapters we present two applications. In the first application, we implement an elastic net regularised ordinal probit and compare the convergence speed of various estimation methods. The regularised ordinal probit is then applied to variable selection for Limit Order Book data (LOBD). In the second application, we use proximal Fisher scoring as part of an EM algorithm used to estimate a regularised bivariate Poisson model. This model is applied to health care data.

Appendix for Part I

A. Solving the Proximal Fisher Operator Using Coordinate Descent

As pointed out by Wang [119], in the case where f_2 is a lasso-type penalty, equation (4.10) can be solved using a coordinate descent method described in Friedman et al. [38] or Wu and Lange [128]. To use coordinate descent, we follow Friedman et al. [38] equation (7): we try to find the optimal value of (4.10) component by component. For component j:

$$\boldsymbol{\beta}^{(a)}_{prox,j} = \underset{\beta_j}{\operatorname{argmin}} \left\{ \frac{1}{2}\boldsymbol{\beta}^T \mathbf{F}^{\lambda}_a \boldsymbol{\beta} - \left[\mathbf{G}^{\lambda}_a\right]^T \boldsymbol{\beta} + f_2\left(\boldsymbol{\beta}\right) \right\}.$$

let us call $\tilde{\boldsymbol{\beta}}^{(a)}\left(\beta_j\right)$ the function to optimise where all coordinates apart from the j^{th} is fixed.

We will note $\tilde{\beta}_k, k \neq j$ all blocked coordinates. Then following Friedman et al.:

$$\tilde{\boldsymbol{\beta}}^{(a)}\left(\beta_j\right) = \left\{ \frac{1}{2} \sum_{k=1,k\neq j}^{M+Mp} \sum_{l=1,l\neq j}^{M+Mp} \tilde{\beta}_k \tilde{\beta}_l F^{\lambda}_{a,kl} + \beta_j \sum_{k=1,k\neq j}^{M+Mp} \tilde{\beta}_k F^{\lambda}_{a,kj} + \frac{\beta_j^2}{2} F^{\lambda}_{a,jj} - \sum_{k=1,k\neq j}^{M+Mp} G^{\lambda}_{a,k}\tilde{\beta}_k \right.$$
$$\left. -G^{\lambda}_{a,j}\beta_j + \lambda\alpha \sum_{k=(M+1),k\neq j}^{M+Mp} T_{kk}\left|\tilde{\beta}_k\right| + \lambda\alpha T_{jj}\left|\beta_j\right| \right\}.$$

Then the subdifferential of the univariate function $\tilde{\boldsymbol{\beta}}^{(a)}\left(\beta_j\right)$ is:

$$\partial\tilde{\boldsymbol{\beta}}^{(a)}\left(\beta_j\right) = \sum_{k=1,k\neq j}^{M+Mp} \beta_k F^{\lambda}_{a,kj} + \beta_j F^{\lambda}_{a,jj} - G^{\lambda}_{a,j} + \lambda\alpha T_{jj}\partial\left(\left|\beta_j\right|\right).$$

Then (see for example Gordon and Tibshirani [43]):

$$0 \in \partial\tilde{\boldsymbol{\beta}}^{(a)}\left(\beta^{(a)}_{prox,j}\right) \Leftrightarrow 0 \in \left(\sum_{k=1,k\neq j}^{M+Mp} \beta_k F^{\lambda}_{a,kj} + \beta^{(a)}_{prox,j} F^{\lambda}_{a,jj} - G^{\lambda}_{a,j} + \lambda\alpha T_{jj}\partial\left(\left|\beta^{(a)}_{prox,j}\right|\right) \right).$$

Therefore:

if $T_{jj} = 0$:

$$\beta^{(a)}_{prox,j} = \frac{G^{\lambda}_{a,j} - \sum_{k=1,k\neq j}^{M+Mp} \beta_k F^{\lambda}_{a,kj}}{F^{\lambda}_{a,jj}}.$$

if $T_{jj} = 1$:

$$\begin{cases} \beta^{(a)}_{prox,j} = \frac{G^{\lambda}_{a,j} - \sum_{k=1,k\neq j}^{M+Mp} \beta_k F^{\lambda}_{a,kj} + \lambda\alpha}{F^{\lambda}_{a,jj}} & \text{if } \left(\left[G^{\lambda}_{a,j} - \sum_{k=1,k\neq j}^{M+Mp} \beta_k F^{\lambda}_{a,kj} \right] < -\lambda\alpha \right). \\ \beta^{(a)}_{prox,j} = 0 & \text{if } \left(\left| G^{\lambda}_{a,j} - \sum_{k=1,k\neq j}^{M+Mp} \beta_k F^{\lambda}_{a,kj} \right| \leq \lambda\alpha \right). \\ \beta^{(a)}_{prox,j} = \frac{G^{\lambda}_{a,j} - \sum_{k=1,k\neq j}^{M+Mp} \beta_k F^{\lambda}_{a,kj} - \lambda\alpha}{F^{\lambda}_{a,jj}} & \text{if } \left(\left[G^{\lambda}_{a,j} - \sum_{k=1,k\neq j}^{M+Mp} \beta_k F^{\lambda}_{a,kj} \right] > \lambda\alpha \right). \end{cases} \tag{A.1}$$

Convergence of the coordinate descent algorithm was proved by Tseng [111].

B. Proximal Fisher Scoring Backtracking Line Search

The algorithm below is directly taken from Lee et al. [68], equation (10) and (11). To determine t_a, we can use a backtracking line search:

$\boldsymbol{\Delta\beta}^{(a)} = \left(\boldsymbol{\beta}_{prox}^{(a)} - \boldsymbol{\beta}^{(a)}\right)$;
$K \in (0, 0.5)$;
$\delta \in (0, 1)$;
while
$f\left(\boldsymbol{\beta}^{(a)} + t_a \boldsymbol{\Delta\beta}^{(a)}\right) > f\left(\boldsymbol{\beta}^{(a)}\right) + K t_a \left[\left[\boldsymbol{\nabla} f_1\left(\boldsymbol{\beta}^{(a)}\right)\right]^T \boldsymbol{\Delta\beta}^{(a)} + f_2\left(\boldsymbol{\beta}^{(a)} + t_a \boldsymbol{\Delta\beta}^{(a)}\right) - f_2\left(\boldsymbol{\beta}^{(a)}\right)\right]$
do
| $t_a \leftarrow \delta \times t_a$
end

Algorithm 1: Backtracking linear search for proximal Fisher Scoring

We note that this algorithm is closely related to the Armijo condition used for the regular Fisher scoring. The only difference is that we take into account non-differentiable function f_2.

C. Assumptions for Convergence

We call $\mathbf{E}$ the domain of $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$.

1. f_1 is convex.
2. f_1 is twice continuously differentiable.
3. f_2 is convex, not necessarily differentiable.
4. $\mathbf{H}_a \succeq mI$, $m > 0$, and therefore positive definite.
5. The score of f_1 is Lipschitz: $\|\nabla f_1(\boldsymbol{\beta}_1) - \nabla f_1(\boldsymbol{\beta}_2)\| \leq L_1 \|\boldsymbol{\beta}_1 - \boldsymbol{\beta}_2\| \quad \forall \boldsymbol{\beta}_1, \boldsymbol{\beta}_2 \in \mathbf{E}$.
6. around the optimal solution $\boldsymbol{\beta}^*$, f_1 is strongly convex:
   $\nabla^2 f_1(\boldsymbol{\beta}) \succeq mI \quad \boldsymbol{\beta} \in N_\varepsilon(\boldsymbol{\beta}^*) := \{\boldsymbol{\beta} \in \mathbf{E}\ s.t. \|\boldsymbol{\beta} - \boldsymbol{\beta}^*\| \leq \varepsilon\}$.
7. $\nabla^2 f_1$ is Lipschitz around $\boldsymbol{\beta}^*$:
   $\left\|\nabla^2 f_1(\boldsymbol{\beta}_1) - \nabla^2 f_2(\boldsymbol{\beta}_2)\right\| \leq L_2 \|\boldsymbol{\beta}_1 - \boldsymbol{\beta}_2\|$, $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2 \in N_\varepsilon(\boldsymbol{\beta}^*) := \{\boldsymbol{\beta} \in \mathbf{E}\ s.t. \|\boldsymbol{\beta} - \boldsymbol{\beta}^*\| \leq \varepsilon\}$.

Part II

Regularised Ordinal Probit Applied to the Prediction of High Frequency Financial Data

1. Introduction

In this chapter, a first application of the penalised VGLM framework is given. As mentioned previously, the ordinal probit is an example of VGLM. In the first part of this chapter, we take a generated data example and compare the speed of convergence of the proximal Fisher scoring to several other methods. In the example taken, the smooth approximation (smooth approximation consists in approximating the L_1 penalty by a smooth function) converges in the smallest time interval and the smallest number of iterations. The proximal Fisher scoring (solved using coordinate descent or quadratic programming) comes second. However, for the smooth approximation, in the case of the generated data example taken, it seems that using a backtracking line search condition such as the Armijo condition (Appendix B.1) is important for convergence. Furthermore, the convergence seems to be sensitive to the parameters used for backtracking line search. This is not the case with proximal Fisher scoring. Most of the time, proximal Fisher scoring does not seem to require backtracking line search to converge and seems quite robust. In our example, other methods such as ISTA or FISTA appear experimentally to have a slower convergence than the proximal Fisher scoring algorithm (although in some cases, FISTA may be slightly faster).

In the second part of this chapter, we apply the regularised ordinal probit to variable selection for prediction of High Frequency Financial Data (HFFD). High Frequency Financial Data (HFFD) or Ultra-High Frequency Financial Data (UHFFD) is defined as financial data that has been recorded at high speed, using frequencies that can be as fast as one recording per microsecond. Two main types of HFFD are available: High Frequency Financial Transaction Data (HFFTD) and Limit Order Book Data (LOBD). HFFTD are recordings of financial transactions concluded electronically by market participants on a given financial contract. A Limit Order Book (LOB) is an electronic register that contains orders to buy or sell a given financial contract in a given quantity at a given price that cannot be executed immediately as prices requested are too far from current market prices. The main difference between HFFTD and LOBD is that much more data is available for a single contract with LOBD. Although there have been studies investigating variable selection for HFFD (see section 2), many studies were concerned with HFFTD. Variable selection was not a problem as the number of potential covariates was somewhat limited. In this document, we use the ordinal probit that had been proposed by Hausman et al. [48] to model price changes in HFFTD and apply it to the prediction of the mid-market price change of a LOBD. At UHF, price changes cannot be assumed to have a continuous distribution and are modelled as ordinal variables. Because there are many correlated covariates that can potentially predict mid-market price changes, we use an elastic net penalty to select variables. An example of a strategy and gain/loss function to decide when to predict a price change is then provided and used both to determine the regularisation parameters in the in-sample dataset and to calculate the performance of the prediction in the out-of-sample dataset.

2. Literature Review

Some previous studies have been using variable selection in the context of HFFD. Techniques that have been used include subset selection (Kercheval and Zhang [59], Panayi and Peters [87], Panayi et al. [88], Lam [67]), LASSO or elastic net penalty (Zheng and Moulines [141], Anane [3]). Other articles discussing prediction using LOBD include Palguna and Pollak [86] and Sirignano [105].

In this document, the elastic net penalty is used. The model that is regularised using the elastic

net penalty is the ordinal probit that was first proposed in the context of HFFTD by Hausman et al. [48]. Alternative models to the ordinal probit proposed in the context of HFFD that are not investigated here include the multinomial model (Russell and Engle [97]), the 'ADS' model (Rydberg and Shephard [98]), the 'ICH' model (Liesenfeld et al. [71] and Bien et al. [17]), the Skellam regression (Shahtahmassebi [101]). With regards to the penalised ordinal model itself, existing studies include Drießlein [32], Hou [50] and Archer et al. [4].

The rest of this chapter is organised as follows: in section 3.1, the ordinal probit regularised with an elastic net penalty is introduced, then there is a brief discussion of the method used to draw the regularisation path and the speed of convergence of the proximal Fisher scoring applied to the regularised ordinal probit is compared to other methods that can be used to solve the same problem. In section 4, having introduced HFFD, a strategy is used. The aim is to predict with the best possible percentage of success positive changes or negative changes of the mid-market price. An elastic net penalised 3-level ordinal model is used to select predictive variables.

3. Ordinal Probit Regularised with an Elastic Net: Model and Comparison of Estimation Methods

The ordinal probit regression links an ordinal response with covariates. The ordinal probit can be seen as a continuous latent variable model where the latent variable is normally distributed. The observed response is a discretised version of the continuous latent variable. In this section, the ordinal probit is first introduced. Then, the regularisation path for an elastic net penalised ordinal probit is implemented using different methods. In the next section, we introduce the model with no penalty.

3.1. Ordinal Probit

As mentioned previously, the ordinal probit fits into the VGLM framework. It is convenient to give the expressions of the different quantities linked to the ordinal probit in the Multivariate Generalised Linear Model (MVGLM) framework (Fahrmeir and Tutz [33]). The MVGLM framework is a subfamily of VGLMs where the responses are assumed to be in the exponential family. The score vector, Fisher information matrix and working response matrices for the MVGLM framework are given in Appendix C. Response y_i can take $M+1$ different values, $1, ..., M+1$. Note that using the VGLM framework as described in part I, we assumed there were Q responses, M linear predictors and p covariates. In the case of the ordinal probit, the number of responses is equal to the number of linear predictors + 1, hence $Q = M+1$. This variable is modelled by assuming that there are $M+1$ binary variables $y_{i1}, ..., y_{i(M+1)}$ such that:

$$y_i = j \Leftrightarrow (y_{ij} = 1, y_{ik} = 0, k \neq j). \tag{3.1}$$

Then the joint distribution and marginal distribution of these $M+1$ variables is:

$$\begin{aligned}
&P\left[Y_{i1} = y_{i1}, Y_{i2} = y_{i2}, ..., Y_{i(M+1)} = y_{i(M+1)}\right] = \prod_{j=1}^{M+1} \pi_{(j)i}^{y_{ij}} \mathcal{I}\left(\sum_{k=1}^{M+1} y_{ik} = 1\right). \\
&P\left[Y_{ij} = y_{ij}\right] = \pi_{(j)i} = \Phi\left(\eta_{(j)i}\right) - \Phi\left(\eta_{(j-1)i}\right). \\
&\eta_{(j)i} = \beta_{(j)0} - \mathbf{B}^T \mathbf{X}_i. \\
&\text{By convention: } \beta_{(0)0} = -\infty; \quad \beta_{(M+1)0} = +\infty.
\end{aligned} \tag{3.2}$$

Note that because the values of $\beta_{(0)0}$ and $\beta_{(M+1)0}$ are set, vector $\boldsymbol{\beta}_0$ contains M coefficients. Matrix $\mathbf{B}$ is the matrix of coefficients. Because in the ordinal probit model the coefficients for the covariates

are the same for all the linear predictors $\eta_{(j)i}$, matrix $\mathbf{B}$ can be replaced by a single column of p coefficient for the p covariates $\mathbf{B} = \left(\overline{\beta}_{11}, ..., \overline{\beta}_{1p}\right)^T$. In the VGLM framework, this can be expressed by a matrix of constraints that forces the M columns of the $[p \times M]$ matrix $\mathbf{B}$ to be identical. Also note that by convention, for the ordinal probit, we use a negative sign in front of the term $\mathbf{B}^T\mathbf{X}_i$. This convention comes from the latent variable representation of the ordinal probit (if we start from the latent variable representation of the ordinal probit and rewrite it in the cumulative probability format, a negative sign appears). The design matrix $\mathbf{X}_i$ is :

$$X_i = \begin{bmatrix} & x_i^T \\ I_{M\times M}| & \vdots \\ & x_i^T \end{bmatrix}. \tag{3.3}$$

From Kedem and Fokianos [58] p.102, we note that:

$$\begin{cases} \pi_{(M+1)i} = \left(1 - \sum\limits_{k=1}^{M} \pi_{(k)i}\right) \\ y_{i(M+1)} = \left(1 - \sum\limits_{k=1}^{M} y_{ik}\right) \end{cases}. \tag{3.4}$$

Therefore, we can reduce the size of the system to $(M + p)$. We also note that $\mathcal{I}\left(\sum\limits_{k=1}^{M+1} y_{ik} = 1\right)$ is always equal to 1. Following Kedem and Fokianos [58] pp.101-106, the log-likelihood can be written as:

$$l\left(\boldsymbol{\beta}\right) = \sum_{i=1}^{n}\left(\sum_{j=1}^{M} y_{ij}\log\left(\frac{\pi_{(j)i}}{1 - \sum\limits_{j=1}^{M}\pi_{(k)i}}\right) + \log\left[\left(1 - \sum_{j=1}^{M} y_{ik}\right)\right]\right). \tag{3.5}$$

The expressions of the matrices necessary to estimate the maximum likelihood for the ordinal probit in the MVGLM framework are given in Appendix D.

3.2. Calculating the Regularisation Path for the Penalised Ordinal Probit

Regarding the different algorithms used to calculate the regularisation path (that includes proximal Fisher scoring), we use the following:

- As suggested by Friedman et al. [39], probabilities are floored at ε_1 and capped at $1 - \varepsilon_1$. We use $\varepsilon_1 = 1e - 10$.
- For the stopping criterion, following Hilbe [49] p.58, we use the change in deviance. The change in deviance between each loop is compared to a value ε_2. We use $\varepsilon_2 = 1e - 10$. A second criterion limiting the maximum number of loops is also added.
- As a partial identity times a regularisation parameter is added to the Fisher information matrix, the condition number of the Fisher information matrix + penalty matrix may be too large. To reduce this issue, we use a method described in Appendix F.

Then:

- We must initialise the vector of intercepts $\boldsymbol{\beta}_0$ to increasing values. For example, we can use values between $R_{min} = -\frac{R}{2}$ to $R_{max} = \frac{R}{2}$. Hence:

$$\Delta_R = \frac{R}{M-1}.$$

$$\boldsymbol{\beta}_{0,init} \leftarrow (R_{min}, R_{min} + \Delta_R, R_{min} + 2\Delta_R, ..., R_{max} - \Delta_R, R_{max}).$$

- To choose the grid values for λ where to calculate the regularisation path values, we follow Bühlmann et al. [22] p.38. We first choose λ_{min} and λ_{max} and d, the total number of points of the cross validation curve. Then Bühlmann et al. [22] define

$$S = \frac{\log(\lambda_{max}) - \log(\lambda_{min})}{d-1}.$$

  Then $\lambda_d = \lambda_{max}$, $\lambda_1 = \lambda_{min}$, $\lambda_{k-1} = \lambda_k \exp(-S)$.

3.3. Comparison of Different Estimation Methods

The proximal Fisher scoring algorithm is compared to other methods. The methods are detailed in Appendix A and are implemented in R. The backtracking line search algorithms used for these methods are detailed in Appendix B. The first method used for comparison purposes is a smooth approximation method from Oelker [80] where the non-differentiable $L1$ penalty is approximated by a smooth function (Appendix A.1). The other methods are other proximal methods or solution methods to the proximal Fisher scoring. In Appendix A.2.1, the method described consists in transforming the original problem into a problem that can be solved using a quadratic programming solver. Package 'quadprog', function 'solve.QP' (Turlach and Weingessel [112]) is used to solve the quadratic program. Last in Appendices A.2.2 and A.2.3 are described the ISTA and FISTA methods that are both proximal gradient method. The FISTA method (Beck and Teboulle [11]) is an accelerated first order method. These algorithms are compared to the proximal Fisher scoring in terms of convergence speed in the next section.

3.4. Example

We use a generated example described in Appendix E. We fix $\alpha = 0.5, \lambda = 1e-2$ (α is the elastic net parameter, λ the regularisation parameter). For the smooth approximation (see Appendix A.1) we use $c = 1e-5$. For the backtracking line search used for the proximal Fisher scoring solved using quadratic programming, we use parameters $K = 0.3$ and $\delta = 0.5$ (see Appendix B of Part I). For the smooth approximation backtracking line search (see Appendix B.1), we use $c_1 = 0.3$. We then calculate the squared error as the number of iteration increases. Results can be found on Figure 3.1. For the bottom chart, function 'get_nanotime', part of R package 'microbenchmark' (Mersmann et al. [76]) is used.

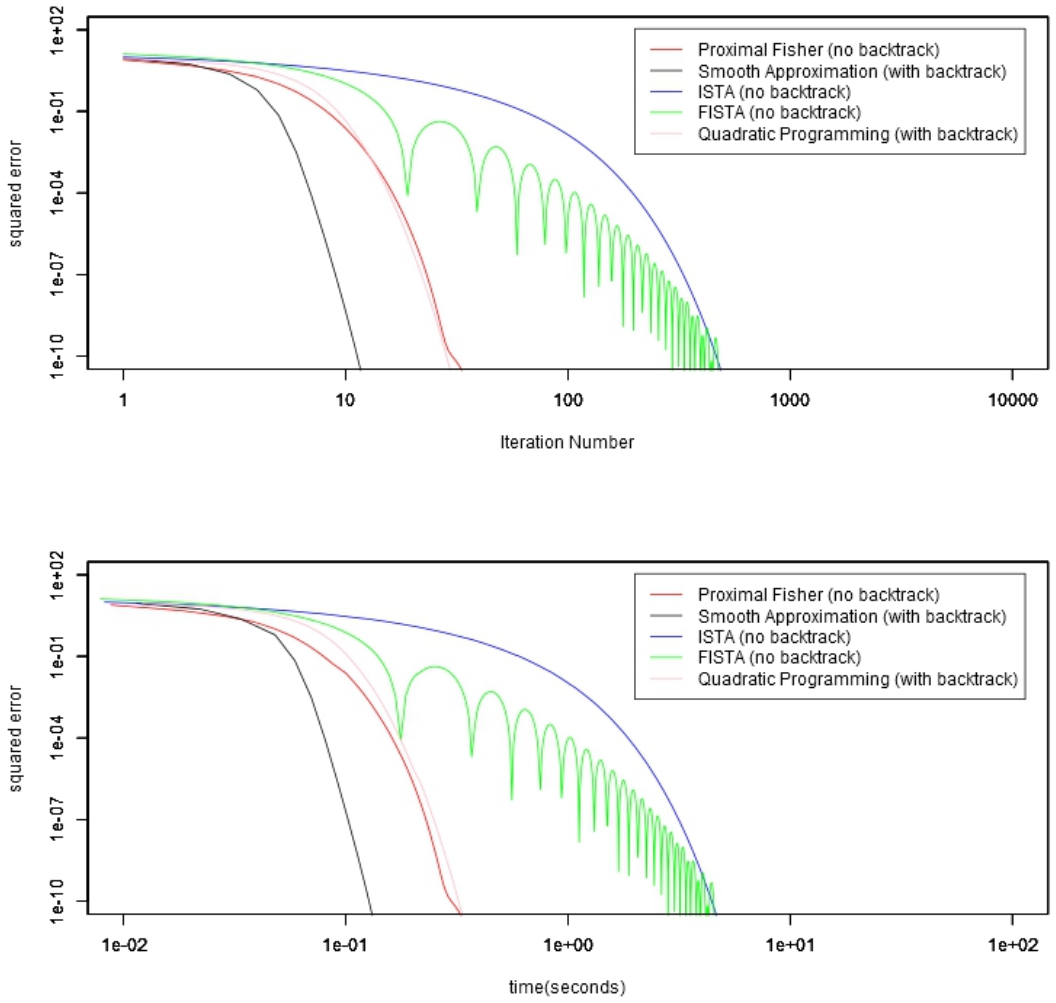

Figure 3.1: Convergence of the different algorithms for the generated example. In this specific case, backtracking line search is used for the smooth approximation and for the quadratic programming solution of the proximal Fisher scoring. This is because with this specific example with the specific values chosen, the algorithms would not have converged without backtracking line search. The top chart gives the squared error at each iteration. The bottom chart gives the squared error (y axis) vs time lapsed (x axis). In both charts, we see that the smooth approximation has the fastest convergence, followed by proximal Fisher scoring where the proximal operator is solved using coordinate descent or quadratic programming, followed by FISTA and ISTA. From this, it could be inferred that the smooth approximation is the appropriate method to use. However, although in general the smooth approximation converges the fastest, in our example, it requires backtracking line search to converge and is quite sensitive to the backtracking line search parameters. This is not the case with proximal Fisher scoring that converges often without backtracking line search and is robust.

4. Application to High Frequency Financial Data

The ordinal probit can be applied to the prediction of HFFD. First in section 4.1, characteristics of HFFD and a description of a dataset are given. The strategy is described in 4.2. A description of the regression is then given in section 4.3.

4.1. Characteristics of HFFD and Dataset

HFFD has several distinctive characteristics reported in the literature. Some of these characteristics are: first, the time series produced are very high frequency. The time interval between two sample times can be below 1 microsecond. Second, price changes are irregularly spaced in time as the recording is asynchronous. Third, price changes are ordinal variables. This is because prices cannot change by less than a fixed amount: the minimum tick size. (for further information on characteristics of LOB, see Gould et al. [44]).

The dataset was provided by the Oxford-Man Institute of Quantitative Finance. The dataset is an asynchronous recordings of a FTSE100 futures LOBD for 4.5 days from 27/10/2008 to 31/10/2008. Every time there is a change in one of the levels of LOB, the new state of the LOB is recorded. The data contains the 10 best levels where market participants are willing to buy (bids) and the 10 best levels where market participants are willing to sell (asks). The mid-market price is the average between the best bid and the best ask. A plot of an example mid-market price changes and histogram can be found on Figure 4.1. Prices changes are expressed in number of half minimum-tick size.

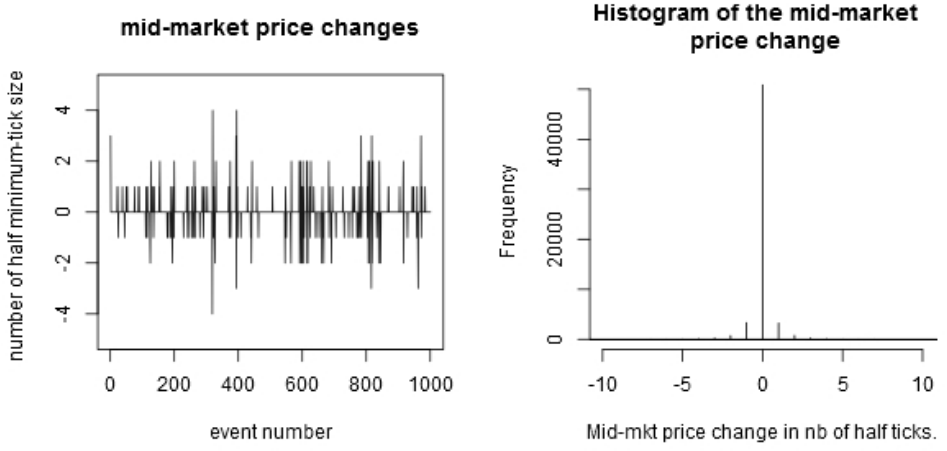

Figure 4.1: The chart on the left shows mid-market price changes. The chart on the right shows its histogram.

4.2. Gain/Loss Function and Strategy

The goal is to develop a strategy that predicts positive changes or negative changes of the mid-market price with a minimum number of incorrect predictions using an elastic net penalised ordinal probit. To minimise the number of incorrect predictions, a prediction is made only when the estimated probability of a price change (estimated using the ordinal probit) is high. We can express this by using a gain/loss function. The gain/loss function associates a positive weight w_+ if a prediction of an up or down move is made and is correct, w_0 if a prediction is made and there is no change and w_- if a prediction is made and the opposite change occurs. Based on these weights, a weighted expected gain or loss for the next mid-market price change can be estimated. The strategy consists in predicting an up or down move only if the expected weighted gain/loss for a negative change or if the expected weighted gain/loss for a positive change is positive. The strategy and gain/loss function are both used during cross validation, to calculate a realised gain/loss on the test set of each fold and in the out-of-sample dataset to determine the overall cumulative gain/loss. Further details about the strategy and gain/loss function are given in Appendix G. Note that Anane pp.73-74 [3] uses an elastic net penalised linear model to predict log returns of the mid-market price. The prediction is made only if the absolute value of the predicted change is greater than a given threshold. Similarly, Lam [67] uses a strategy where price changes are predicted only if the probability of a change is greater than a given threshold. In the strategy described in Appendix G, the weights play a similar role to these thresholds.

4.3. Regression Settings

We use the following setting for the regression:

- **Response:** The response is the change in mid-market price. The mid-market price is one of the fundamental quantities in finance. It is often used as the best representation of the current market price. In this document, we restrict price changes to 3 values: -1 (negative change in mid-market price), 0 (no change), +1 (positive change in mid-market price).
- **In and Out of Sample:** 5,000 datapoints are picked randomly from the first 10,000 datapoints and are used as the in-sample dataset. Then 5 folds are created and the 5,000 datapoints are used for cross validation. Last, for the out-of-sample, datapoints from 10,001 to 70,000 are used.
- **Covariates:** The different variables available in the dataset are:
  1. BestBid1, BestBid2,..., BestBid10 (BestBidi is the ith best bid).
  2. BestAsk1, BestAsk2,..., BestAsk10 (BestAski is the ith best ask).
  3. DepthBid1, DepthBid2,..., DepthBid10 (DepthBidi is the number of contracts available at price BestBidi).
  4. DepthAsk1, DepthAsk2,..., DepthAsk10 (DepthAski is the number of contracts available at price BestAski).
  5. time of LOB change.

  We use the following 410 covariates:

  1. Change in BestBid1, Change in BestBid2,..., Change in BestBid10 from (t-10) to (t-1) (100 covariates).
  2. Change in BestAsk1, Change in BestAsk2,..., Change in BestAsk10 from (t-10) to (t-1) (100 covariates).

3. DepthBid1, DepthBid2,..., DepthBid10 from (t-10) to (t-1) (100 covariates).
4. DepthAsk1, DepthAsk2,..., DepthAsk10 from (t-10) to (t-1) (100 covariates).
5. interchange duration for the past 10 changes (t-10) to (t-1) (10 covariates).

Figure 4.2 (next page) shows the results of the crossvalidation and the cumulative gain/loss in the out-of-sample.

5. Discussion and Future Steps

An application of proximal Fisher scoring was discussed. Proximal Fisher scoring was compared to several other methods using the example of a regularised ordinal probit model. Among the algorithms used and for the example taken, proximal Fisher scoring seems to be a good compromise between speed of convergence and stability. The regularised ordinal probit was then applied to prediction for the mid-market price in a Limit Order book Data. There are a number of limitations to this analysis that could be adressed:

- We here presented a single example of variable selection for LOBD for a specific dataset. An empirical criterion based on a strategy and a gain/loss function was used to determine the optimal value of the regularisation parameters α and λ. Further study would be required to test the validity of this criterion across different datasets.
- Several other methods could be implemented to be compared with proximal Fisher scoring and may be useful in cases where it may be difficult to use the proximal Fisher scoring.
  - The alternative transformation of the problem into a constrained quadratic program described at the end of Appendix A.2.1 could be solved using an interior point method as suggested by Koh et al. [65] in the case of a logistic regression.
  - As alternative to the smooth approximation, Generalised Iteratively Least Squares (GIRLS) (Bissantz et al. [18]) could be used. In GIRLS, the parameter used for the smooth approximation of the nondifferentiable function (called c in the smooth approximation (see Appendix A.1)) is reduced after each iteration.
  - Shi [102] proposes a stochastic proximal Newton method that has advantages as its convergence is independent of n and could be suited to large scale problems.
  - R package 'ordinalgmifs' (Archer et al. [4]) calculates the regularisation path using the forward stagewise method for ordinal models. This method could be compared to proximal Fisher scoring.
  - Trust region methods: Trust region methods are globally convergent and therefore may be useful for non-convex problems. The block coordinate descent-Newton trust region algorithm of Qin et al. [95] could be used.
  - In the augmented Lagrangian approach, a penalty is added to the Lagrangian function. The augmented Lagrangian function is then strictly convex. The ADMM method can then be used to solve the problem (see Qin and Goldfarb [94], Qin et al. [96]).
  - Patrinos et al. [92] propose accelerated second order proximal methods using conjugate gradient for cases where standard proximal Newton may not be applicable.
  - Wang and Leng [117] propose an alternative approach consisting in using a quadratic approximation of the log-likelihood around the MLE of the unpenalised likelihood.

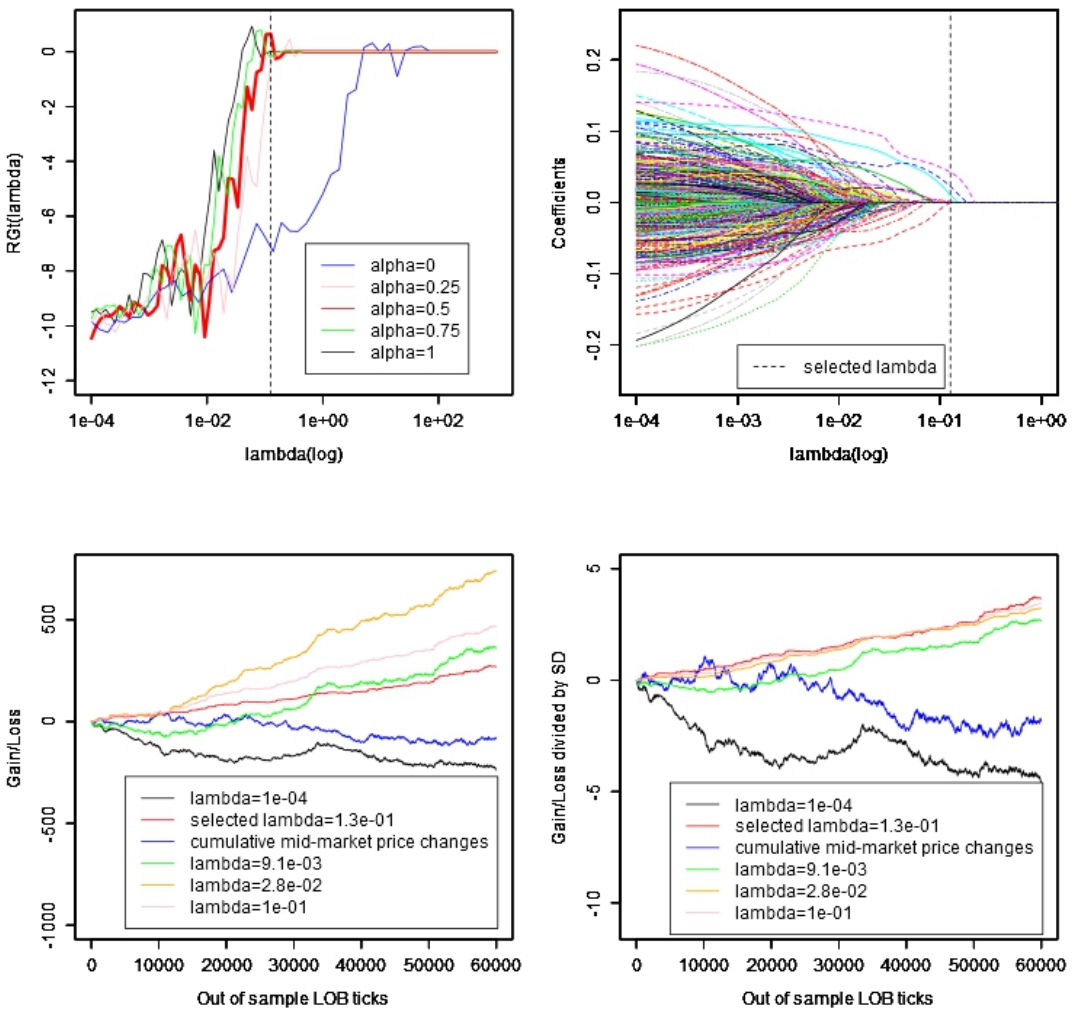

Figure 4.2: The top left chart represents the average realised gain in the in-sample over the 5 cross validated samples for $\alpha = 0, 0.25, 0.5, 0.75, 1$. The peaks for the 5 curves are comparable. To obtain a parcimonious model, any value different from zero would be appropriate. The value of 0.5 for α is then arbitrarily used (red curve). The top right chart represents the regularisation path for the 410 covariates. Only four covariates are selected: the change in BestBid1 (t-10), change in BestAsk1 (t-10), change in BestAsk3 (t-10), change in BestAsk4 (t-10). The bottom left chart represents the cumulative gains and losses using $w_- = -1.2, w_0 = -0.05, w_+ = 1$ with 60,000 out-of-sample datapoints. The blue curve is the actual mid-market price change. The red curve represents the cumulative gains and losses using the selected lambda. The black curve are cumulative gains and losses if all 410 covariates are used. Other curves show gains and losses with other values of λ. Hence the selected λ does not necessarily bring the highest cumulative gains. We note that although cumulative gains are lower with the selected λ than with some other values of λ, the standard deviation seems to be lower. To take this into account and better compare the different curves, we divide each cumulative gain curve from the standard deviation of the whole cumulative gain path. This is what is shown on the bottom right figure. We note that if the whole cumulative gain path is scaled by its standard deviation, the performance with different λ values and the chosen λ are comparable.

Appendix for Part II

A. Other Estimation Methods

In this section, we discuss the methods used to estimate the elastic net penalised log-likelihood that are implemented in R to compare with proximal Fisher scoring. In terms of implementation, the main difference between these methods is that first order methods only require the availability of the gradient whilst second order methods require the Hessian, Fisher information matrix or an approximation of one of these matrices. A summary of several methods that can be used for non-smooth optimisation can also be found in Hastie et al. [47] chapter 5.

A.1. Smooth Approximation of the Non-Differentiable Function

Oelker [80], Oelker and Tutz [81] based on Koch [63] and Ulbricht [115] propose to approximate the non-differentiable penalty with a smooth function. The non-smooth part of the penalised log-likelihood is:

$$f_2(\boldsymbol{\beta}) = \lambda\alpha \sum_j T_{jj}|\beta_j|. \tag{A.1}$$

From Oelker and Tutz [81] we replace the $|\beta_j|$ by:

$$\sqrt{\beta_j^2 + c}. \tag{A.2}$$

Where c is a constant. Then we have:

$$\begin{aligned} f_2^{approx}(\boldsymbol{\beta}) &= \lambda\alpha \sum_j T_{jj}\sqrt{\beta_j^2 + c}. \\ \frac{\partial f_2^{approx}(\boldsymbol{\beta})}{\partial \beta_j} &= \frac{\lambda\alpha T_{jj}\beta_j}{\sqrt{\beta_j^2 + c}}. \\ \frac{\partial^2 f_2^{approx}(\boldsymbol{\beta})}{\partial \beta_j^2} &= \frac{\lambda\alpha T_{jj}}{\sqrt{\beta_j^2 + c}}\left(1 - \frac{\beta_j}{\beta_j^2 + c}\right). \end{aligned} \tag{A.3}$$

Oelker and Tutz [81] use the approximation: $\frac{\partial^2 f_2^{approx}(\boldsymbol{\beta})}{\partial \beta_j^2} \approx \frac{\lambda\alpha T_{jj}}{\sqrt{\beta_j^2 + c}}$. We will not use this approximation but the exact second derivative. Then we simply use the Fisher scoring algorithm where:

$$\begin{aligned} \mathbf{F}_i^{\lambda approx} &= \mathbf{F}_i + \lambda(1-\alpha)\mathbf{T} + \lambda\alpha\mathbf{T} \times diag\left(\frac{\partial^2 f_2^{approx}(\boldsymbol{\beta})}{\partial \beta_1^2}, ..., \frac{\partial^2 f_2^{approx}(\boldsymbol{\beta})}{\partial \beta_{(M+Mp)}^2}\right). \\ \mathbf{U}_i^{\lambda approx} &= \mathbf{U}_i - \lambda(1-\alpha)\mathbf{T}\boldsymbol{\beta} - \lambda\alpha\mathbf{T} \times \left(\frac{\partial f_2^{approx}(\boldsymbol{\beta})}{\partial \beta_1}, ..., \frac{\partial f_2^{approx}(\boldsymbol{\beta})}{\partial \beta_{(M+Mp)}}\right)^T. \end{aligned} \tag{A.4}$$

A.2. Other Proximal Algorithms

A.2.1. Solving the Inner Loop of the Proximal Fisher Scoring Using Quadratic Programming

This section is based on Schmidt [100], Laber and Zhou [66], Figueiredo et al. [36], Bach et al. [10] and Buša [23]. As noted in Osborne et al. [85], the LASSO problem has two equivalent forms: a

constrained regression form and a penalised regression form. Tibshirani [109] noted that the LASSO path can be calculated by a decomposition method where the vector of parameters $\boldsymbol{\beta}$ is decomposed into a positive part $\boldsymbol{\beta}^+$ and a negative part $\boldsymbol{\beta}^-$. Tibshirani applied this decomposition to the constrained regression form of the LASSO problem to solve it. This technique cannot be used in the case of the elastic net problem. However, as in Bach et al. [10] or Schmidt [100], we can also apply the decomposition into a negative part to the penalised regression form of the problem. From equation (4.13) of part I, the problem to solve is:

$$\boldsymbol{\beta}_{opt}^{(a)} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \frac{1}{2} \boldsymbol{\beta}^T \mathbf{F}_a^{\lambda} \boldsymbol{\beta} - \left[\mathbf{G}_{F,a}^{\lambda} \right]^T \boldsymbol{\beta} + \lambda \alpha \boldsymbol{\beta}^T \mathbf{T} sign\left(\boldsymbol{\beta}\right) \right\}. \tag{A.5}$$

From Bach et al. [10], Schmidt [100], we use the following transformation:

$$\tilde{\boldsymbol{\beta}} = \begin{pmatrix} \boldsymbol{\beta}^+ \\ \boldsymbol{\beta}^- \end{pmatrix} \quad \boldsymbol{\beta}^+ = \left(\boldsymbol{\beta} \vee \mathbf{0}\right) \quad \boldsymbol{\beta}^- = -\left(\boldsymbol{\beta} \wedge \mathbf{0}\right). \tag{A.6}$$

$$\boldsymbol{\beta} = \left(\boldsymbol{\beta}^+ - \boldsymbol{\beta}^-\right). \tag{A.7}$$

Similarly to Laber and Zhou [66], Figueiredo et al. [36], the problem can be re-written:

$$\begin{aligned} \boldsymbol{\beta}_{opt}^{(a)} = \underset{\tilde{\boldsymbol{\beta}}}{\operatorname{argmin}} \Bigg\{ & \frac{1}{2} \tilde{\boldsymbol{\beta}}^T \begin{bmatrix} \mathbf{F}_{\mathbf{a}}^{\lambda} & -\mathbf{F}_{\mathbf{a}}^{\lambda} \\ -\mathbf{F}_{\mathbf{a}}^{\lambda} & \mathbf{F}_{\mathbf{a}}^{\lambda} \end{bmatrix} \tilde{\boldsymbol{\beta}} - \left(\left\{ \left[\mathbf{G}_{\mathbf{F},\mathbf{a}}^{\lambda} \right]^T \right\}; \left\{ - \left[\mathbf{G}_{\mathbf{F},\mathbf{a}}^{\lambda} \right]^T \right\} \right) \tilde{\boldsymbol{\beta}} \\ & + \lambda \alpha \mathbf{1}_{[2(M \times (1+p)) \times 1]}^T \begin{bmatrix} \mathbf{T} & \mathbf{0} \\ \mathbf{0} & \mathbf{T} \end{bmatrix} \tilde{\boldsymbol{\beta}} \Bigg\}. \end{aligned} \tag{A.8}$$

$$\tilde{\boldsymbol{\beta}} \succeq \mathbf{0}.$$

$$\tilde{\boldsymbol{\beta}} = \left(\left(\boldsymbol{\beta}^+\right)^T; \left(\boldsymbol{\beta}^-\right)^T \right)^T.$$

Where:

$$\mathbf{G}_{\mathbf{F},a}^{\lambda} = \frac{1}{n} \sum_i \mathbf{X}_i^T \mathbf{W}_i^{(a)} \mathbf{z}_i^{(a)}.$$

$$\mathbf{F}_a^{\lambda} = \frac{1}{n} \sum_i \mathbf{X}_i^T \mathbf{W}_i^{(a)} \mathbf{X}_i + \lambda \left(1 - \alpha\right) \mathbf{T}.$$

$$\tilde{\mathbf{F}}_a^{\lambda} = \begin{bmatrix} \mathbf{F}_{\mathbf{a}}^{\lambda} & -\mathbf{F}_{\mathbf{a}}^{\lambda} \\ -\mathbf{F}_{\mathbf{a}}^{\lambda} & \mathbf{F}_{\mathbf{a}}^{\lambda} \end{bmatrix}. \tag{A.9}$$

Buša [23] proves that if $\mathbf{F}_a^{\lambda}$ is positive definite, then $\tilde{\mathbf{F}}_a^{\lambda}$ is positive semi-definite and $\left(\tilde{\mathbf{F}}_a^{\lambda} + \varepsilon \mathbf{I}_{[2(M+Mp)] \times [2(M+Mp)]}\right)$ is positive definite for some $\varepsilon > 0$. We can therefore add a small element to the diagonal of $\tilde{\mathbf{F}}_a^{\lambda}$ to make sure the matrix is positive definite. As suggested by Laber and Zhou [66], we can use function 'solve.QP', part of package 'quadprog' (Turlach and Weingessel [112]) to solve this quadratic program.

An alternative method presented for example in Koh et al. [65], consists in adding a new variable vector $\boldsymbol{\zeta}$, replacing the Lasso penalty $\lambda \alpha \boldsymbol{\beta}^T \mathbf{T} sign\left(\boldsymbol{\beta}\right)$ by $\lambda \alpha \mathbf{1}^T \mathbf{T} \boldsymbol{\zeta}$ with constraint:

$$-\zeta_j \mathcal{I}\left(T_{jj} = 1\right) \leq \beta_j \mathcal{I}\left(T_{jj} = 1\right) \leq \zeta_j \mathcal{I}\left(T_{jj} = 1\right) \quad j \in \{1, ..., (M + Mp)\}.$$

This form of the problem is then solved using an interior point method in Koh et al. [65].

A.2.2. Forward Backward/Proximal Gradient/ISTA

Forward backward split/proximal gradient/ISTA are different names for first order proximal algorithm. We here describe the most general form of the algorithm as can be found in Combettes and Pesquet [29]. Other variants that can be found in the literature differ from the algorithm below by coefficients v_a or w_a. From Combettes and Pesquet [29], the constant step forward backward algorithm (algorithm 3.2) is:

initialise ψ_0 to a value
$\varepsilon \in]0, (1 \wedge 1/\psi_0)[$;
while *condition* **do**
  $v_a \in [\varepsilon, 3/2 - \varepsilon]$
  $w_a \in [\varepsilon, 1]$
  $\boldsymbol{\beta}^{(a+1)} = \boldsymbol{\beta}^{(a)} + w_a \left[prox_{v_a f_2}\{\boldsymbol{\beta}^{(a)} - v_a \boldsymbol{\nabla} f_1\left(\boldsymbol{\beta}^{(a)}\right)\} - \boldsymbol{\beta}^{(a)} \right]$
end

Algorithm 2: algorithm 3.2 from Combettes and Pesquet

Then:

- if $w_a = 1$, $v_a = t_a$, we obtain the classical proximal gradient as described for example in Lee et al. [69].
- If $w_a = v_a$, then we obtain the proximal Newton algorithm described in Lee et al. [68] [69] where the Inverse Hessian is replaced by $\frac{1}{t_a} I_{(M+Mp)(M+Mp)}$, where I_{ab} is the identity matrix of size $a \times b$.
- If $v_a = constant = \psi_0^{-1}$, $w_a \in [\varepsilon, 3/2 - \varepsilon]$ then we obtain the constant step forward backward algorithm (algorithm 3.4 from Combettes and Pesquet [29]).
- Taking $w_a = 1$, we note that as $v_a > 0$, we can divide the expression inside the argmin function without changing the result:

$$
\begin{aligned}
\boldsymbol{\beta}^{(a+1)} &= \left[prox_{v_a f_2}\left\{ \boldsymbol{\beta}^{(a)} - v_a \boldsymbol{\nabla} f_1\left(\boldsymbol{\beta}^{(a)}\right)\right\}\right] \\
&= \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(\frac{1}{2}\left(\boldsymbol{\beta} - \boldsymbol{\beta}^{(a)} + v_a \boldsymbol{\nabla} f_1\left(\boldsymbol{\beta}^{(a)}\right)\right)^T \left(\boldsymbol{\beta} - \boldsymbol{\beta}^{(a)} + v_a \boldsymbol{\nabla} f_1\left(\boldsymbol{\beta}^{(a)}\right)\right) + v_a f_2\left(\boldsymbol{\beta}\right) \right) \\
&= \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(\frac{1}{v_a}\left[\frac{1}{2}\left(\boldsymbol{\beta} - \boldsymbol{\beta}^{(a)} + v_a \boldsymbol{\nabla} f_1\left(\boldsymbol{\beta}^{(a)}\right)\right)^T \left(\boldsymbol{\beta} - \boldsymbol{\beta}^{(a)} + v_a \boldsymbol{\nabla} f_1\left(\boldsymbol{\beta}^{(a)}\right)\right) + v_a f_2\left(\boldsymbol{\beta}\right) \right] \right) \\
&= \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(\frac{1}{2v_a}\left(\boldsymbol{\beta} - \boldsymbol{\beta}^{(a)} + v_a \boldsymbol{\nabla} f_1\left(\boldsymbol{\beta}^{(a)}\right)\right)^T \left(\boldsymbol{\beta} - \boldsymbol{\beta}^{(a)} + v_a \boldsymbol{\nabla} f_1\left(\boldsymbol{\beta}^{(a)}\right)\right) + f_2\left(\boldsymbol{\beta}\right) \right).
\end{aligned}
$$

Then taking $v_a = \frac{1}{L}, L > 0$

$$
\boldsymbol{\beta}^{(a+1)} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(\frac{L}{2}\left(\boldsymbol{\beta} - \boldsymbol{\beta}^{(a)} + \frac{\boldsymbol{\nabla} f_1\left(\boldsymbol{\beta}^{(a)}\right)}{L}\right)^T \left(\boldsymbol{\beta} - \boldsymbol{\beta}^{(a)} + \frac{\boldsymbol{\nabla} f_1\left(\boldsymbol{\beta}^{(a)}\right)}{L}\right) + f_2\left(\boldsymbol{\beta}\right) \right). \tag{A.10}
$$

This expression is the expression of the proximal gradient/ISTA that can be found in Beck and Teboulle [11].

Then in the VGLM case, for algorithm 3.2 from Combettes and Pesquet, we obtain:

$$\boldsymbol{\beta}_{prox}^{(a)} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(\frac{1}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} - [\boldsymbol{\beta}^{(a)}]^T\boldsymbol{\beta} + v_a \left[\boldsymbol{\nabla} f_1\left(\boldsymbol{\beta}^{(a)}\right)\right]^T \boldsymbol{\beta} + v_a f_2\left(\boldsymbol{\beta}\right) \right).$$

where:

$f_2\left(\boldsymbol{\beta}\right) = \lambda\alpha \left[sign\left(\boldsymbol{\beta}\right)\right]^T \mathbf{T}\boldsymbol{\beta}.$

therefore:

$$\begin{aligned} &\text{if } T_{jj} = 0 : \\ &\beta_{prox,j}^{(a)} = \beta_j^{(a)} + v_a U_{a,j}^{\lambda}. \end{aligned} \tag{A.11}$$

$$\text{if } T_{jj} = 1 :$$

$$\beta_{prox,j}^{(a)} = \begin{cases} \beta_j^{(a)} + v_a U_{a,j}^{\lambda} + v_a\lambda\alpha & \text{if } \left(\beta_j^{(a)} + v_a U_{a,j}^{\lambda}\right) < -v_a\lambda\alpha. \\ 0 & \text{if } \left|\beta_j^{(a)} + v_a U_{a,j}^{\lambda}\right| \leq v_a\lambda\alpha. \\ \beta_j^{(a)} + v_a U_{a,j}^{\lambda} - v_a\lambda\alpha & \text{if } \left(\beta_j^{(a)} + v_a U_{a,j}^{\lambda}\right) > v_a\lambda\alpha. \end{cases}$$

We implement the classical proximal gradient algorithm with $w_a = 1$ and determine step v_a using backtracking line search.

A.2.3. FISTA

Tutz et al. [114] use the FISTA algorithm to estimate a lasso-regularised multinomial logit. The FISTA algorithm given by Beck and Teboulle [11] starts as a proximal forward backward step described in Combettes and Pesquet [29], algorithm 3.2 with $v_a = \frac{1}{L}$ and $w_a = 1$.

$$\begin{aligned} \boldsymbol{\beta}_{prox}^{(a)} &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(\frac{L}{2} \left(\boldsymbol{\beta} - \boldsymbol{\beta}^{(a)} + \frac{1}{L}\nabla f_1\left(\boldsymbol{\beta}\right)|_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(a)}} \right)^T \left(\boldsymbol{\beta} - \boldsymbol{\beta}^{(a)} + \frac{1}{L}\nabla f_1\left(\boldsymbol{\beta}\right)|_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(a)}} \right) + f_2\left(\boldsymbol{\beta}\right) \right) \\ &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(\frac{1}{L} \left\{ \frac{L}{2} \left(\boldsymbol{\beta} - \boldsymbol{\beta}^{(a)} + \frac{1}{L}\nabla f_1\left(\boldsymbol{\beta}\right)|_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(a)}} \right)^T \left(\boldsymbol{\beta} - \boldsymbol{\beta}^{(a)} + \frac{1}{L}\nabla f_1\left(\boldsymbol{\beta}\right)|_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(a)}} \right) + f_2\left(\boldsymbol{\beta}\right) \right\} \right) \\ &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(\frac{1}{2} \left(\boldsymbol{\beta} - \boldsymbol{\beta}^{(a)} + \frac{1}{L}\nabla f_1\left(\boldsymbol{\beta}\right)|_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(a)}} \right)^T \left(\boldsymbol{\beta} - \boldsymbol{\beta}^{(a)} + \frac{1}{L}\nabla f_1\left(\boldsymbol{\beta}\right)|_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(a)}} \right) + \frac{1}{L} f_2\left(\boldsymbol{\beta}\right) \right) \\ &= prox_{\frac{f_2}{L}} \left(\boldsymbol{\beta}^{(a)} - \frac{1}{L}\nabla f_1\left(\boldsymbol{\beta}\right)|_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(a)}} \right). \end{aligned} \tag{A.12}$$

Then in the VGLM case, the first part of the algorithm is the same as in the case of ISTA where v_a is replaced by $\frac{1}{L}$:

$$\begin{aligned}
&\text{if } T_{jj} = 0 : \\
&\qquad \beta_{prox,j}^{(a)} = \beta_j^{(a)} + \frac{U_{a,j}^{\lambda}}{L}. \\
&\text{if } T_{jj} = 1 : \\
&\qquad \beta_{prox,j}^{(a)} = \begin{cases} \beta_j^{(a)} + \frac{U_{a,j}^{\lambda}}{L} + \frac{\lambda\alpha}{L} & \text{if } \left[\beta_j^{(a)} + \frac{U_{a,j}^{\lambda}}{L}\right] < -\frac{\lambda\alpha}{L}. \\ 0 & \text{if } \left|\beta_j^{(a)} + \frac{U_{a,j}^{\lambda}}{L}\right| \leq \frac{\lambda\alpha}{L}. \\ \beta_j^{(a)} + \frac{U_{a,j}^{\lambda}}{L} - \frac{\lambda\alpha}{L} & \text{if } \left[\beta_j^{(a)} + \frac{U_{a,j}^{\lambda}}{L}\right] > \frac{\lambda\alpha}{L}. \end{cases}
\end{aligned} \tag{A.13}$$

Then we add an extra step:

$$\begin{aligned}
s_{a+1} &= \frac{1 + \sqrt{1 + 4s_a^2}}{2}. \\
\boldsymbol{\beta}^{(a+1)} &= \boldsymbol{\beta}_{opt,j}^{(a)} + \left(\frac{s_a - 1}{s_{a+1}}\right)\left(\boldsymbol{\beta}_{prox,j}^{(a)} - \boldsymbol{\beta}_{prox,j}^{(a-1)}\right).
\end{aligned} \tag{A.14}$$

B. Backtracking Line Search Algorithms

In this section, we give a backtracking line search condition for smooth approximation (Armijo condition) and ISTA/FISTA. As in Nocedal and Wright [127], we define $\mathbf{p}_a^{\lambda}$ as the search direction. In the case of gradient methods, $\mathbf{p}_a^{\lambda} = -\mathbf{U}_a^{\lambda}$ is the gradient. In the case of Newton methods, $\mathbf{p}_a^{\lambda} = -\left(\mathbf{H}_a^{\lambda}\right)^{-1}\mathbf{U}_a^{\lambda}$. In the case of Fisher scoring, $\mathbf{p}_a^{\lambda} = \left(\mathbf{F}_a^{\lambda}\right)^{-1}\mathbf{U}_a^{\lambda}$.

B.1. Backtracking Line Search for Smooth Approximation

We use the Armijo condition for our backtracking line search (Nocedal and Wright [127] p.33):

$$f(\boldsymbol{\beta}^{(a)} + t_a\mathbf{p}_a^{\lambda}) \leq f(\boldsymbol{\beta}^{(a)}) + c_1 t_a(\nabla f(\boldsymbol{\beta}^{(a)}))^T\mathbf{p}_a^{\lambda}. \tag{B.1}$$

Where $c_1 \in (0, 1)$ is a constant. Other conditions that could have been used are the Armijo-Wolfe or Goldstein conditions.

B.2. Backtracking Line Search for ISTA/FISTA

For ISTA/FISTA, we use the backtracking line search for the proximal gradient that can be found in Beck and Teboulle [11]. The algorithm consists in finding t_a such that the following condition is

verified:

$$
\begin{aligned}
&f_1\left(\boldsymbol{\beta}_{opt}^{(a)}\left(t_a\right)\right)-f_1\left(\boldsymbol{\beta}^{(a)}\right)\leq\left(\boldsymbol{\beta}_{opt}^{(a)}\left(t_a\right)-\boldsymbol{\beta}^{(a)}\right)^T\left[\boldsymbol{\nabla}f_1\left(\boldsymbol{\beta}^{(a)}\right)+\frac{1}{2t_a}\left(\boldsymbol{\beta}_{opt}^{(a)}\left(t_a\right)-\boldsymbol{\beta}^{(a)}\right)\right].\\
&\boldsymbol{\beta}_{\beta opt}^{(a)}\left(t_a\right)=\underset{\boldsymbol{\beta}}{\operatorname{argmin}}\left(\frac{1}{2}\left(\boldsymbol{\beta}-\boldsymbol{\beta}^{(a)}+t_a\boldsymbol{\nabla}f_1\left(\boldsymbol{\beta}^{(a)}\right)\right)^T\left(\boldsymbol{\beta}-\boldsymbol{\beta}^{(a)}+t_a\boldsymbol{\nabla}f_1\left(\boldsymbol{\beta}^{(a)}\right)\right)+f_2\left(\boldsymbol{\beta}\right)\right).\\
&\boldsymbol{\nabla}f_1\left(\boldsymbol{\beta}^{(a)}\right)=\left(\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{X}_i^T\left(\mathbf{W}_i^{(a)}\right)\mathbf{X}_i+\lambda\left(1-\alpha\right)\mathbf{T}\right)\boldsymbol{\beta}^{(a)}-\frac{1}{n}\sum_{i=1}^{n}\mathbf{X}_i^T\left(\mathbf{W}_i^{(a)}\right)\mathbf{z}_i^{(a)}\right)=-\mathbf{U}_a^{\lambda}.\\
&f_1\left(\boldsymbol{\beta}\right)=-l\left(\boldsymbol{\beta}\right)+\frac{\lambda\left(1-\alpha\right)}{2}\boldsymbol{\beta}^T\mathbf{T}\boldsymbol{\beta}.\\
&f_2\left(\boldsymbol{\beta}\right)=\lambda\alpha\left[sign\left(\boldsymbol{\beta}\right)\right]^T\mathbf{T}\boldsymbol{\beta}.
\end{aligned}
\tag{B.2}
$$

We start from a given value of t_a and then iterate $t_a \leftarrow \delta t_a$, where $0 < \delta < 1$ until the condition is verified.

C. Multivariate Generalised Linear Models (MVGLM)

The MVGLM framework is a sub-family of VGLMs where the distribution of the responses $\mathbf{Y}_i$ is assumed to be in the exponential family (Fahrmeir and Tutz [33] p.76):

$$
g\left(\mathbf{y}_i|\boldsymbol{\theta}_i,\psi\right)=\exp\left\{\frac{\mathbf{y}_i^T\boldsymbol{\theta}_i-b\left(\boldsymbol{\theta}_i\right)}{\psi}+e\left(\mathbf{y}_i,\psi\right)\right\}. \tag{C.1}
$$

where $b(.)$ and $e(.,.)$ are functions, ψ is a dispersion parameter.
Then, similarly to the GLM case, the expected value of the responses conditional on the covariates is assumed to be a non-linear function of a linear combination of the covariates.

$$
E[\mathbf{Y}_i|\mathbf{x}_i]=\boldsymbol{\mu}_i=h(\mathbf{x}_i\boldsymbol{\beta}). \tag{C.2}
$$

The advantage of using the MVGLM framework over the VGLM framework is that in case the model used falls in the exponential family, more detailed expressions can be used for the score and Fisher information matrices. As described in Yee [130], chapter 3, Fahrmeir and Tutz [33] p.76 and p.105, Kedem and Fokianos [58] pp.101-106 and Tutz [113] pp.63-66, in the MVGLM, the expressions of the score vector and Fisher information matrix are:

- Score vector:

$$
\begin{aligned}
\mathbf{U}_i&=\frac{\partial l_i}{\partial\boldsymbol{\beta}}=\frac{\partial l_i}{\partial\boldsymbol{\theta}_i^T}\frac{\partial\boldsymbol{\theta}_i}{\partial\mathbf{h}^T}\frac{\partial\mathbf{h}}{\partial\boldsymbol{\eta}_i^T}\frac{\partial\boldsymbol{\eta}_i}{\partial\boldsymbol{\beta}}\\
&=\mathbf{X_i^T}\mathbf{D_i}\boldsymbol{\Sigma}_\mathbf{i}^{-\mathbf{1}}\left[\mathbf{y_i}-\boldsymbol{\mu_i}\right]=\mathbf{X_i^T}\mathbf{W_i}\mathbf{D_i^{-T}}\left[\mathbf{y_i}-\boldsymbol{\mu_i}\right]=\mathbf{X_i^T}\mathbf{u_i}.
\end{aligned}
\tag{C.3}
$$

  where:

$$
\mathbf{W_i}=\mathbf{D_i}\boldsymbol{\Sigma}_\mathbf{i}^{-\mathbf{1}}\mathbf{D_i^T}. \tag{C.4}
$$

$$
\mathbf{D_i}=\begin{bmatrix}\frac{\partial[h_1]_i}{\partial\eta_{(1)i}} & \cdots & \frac{\partial[h_Q]_i}{\partial\eta_{(1)i}}\\ \vdots & \ddots & \vdots\\ \frac{\partial[h_1]_i}{\partial\eta_{(M)i}} & \cdots & \frac{\partial[h_Q]_i}{\partial\eta_{(M)i}}\end{bmatrix}. \tag{C.5}
$$

(C.6)

and:

$$\boldsymbol{\Sigma}_i = covar\left[\mathbf{Y_i}, \mathbf{Y_i}\right] \text{ of size } [Q \times Q]. \tag{C.7}$$

- The expression of the Fisher information matrix is the same as in the case of VGLM as:

$$\begin{aligned}
\mathbf{F}_i &= -E\left[\frac{\partial^2 l_i}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T}\right] = +E\left[\frac{\partial l_i}{\partial \boldsymbol{\beta}}\left(\frac{\partial l_i}{\partial \boldsymbol{\beta}}\right)^T\right] \\
&= E\left[\mathbf{X_i^T D_i \Sigma_i^{-1}}\left[\mathbf{y_i} - \boldsymbol{\mu_i}\right]\left[\mathbf{y_i} - \boldsymbol{\mu_i}\right]^{\mathbf{T}} \mathbf{\Sigma_i^{-1} D_i^T X_i}\right] \\
&= \mathbf{X_i^T D_i \Sigma_i^{-1} D_i^T X_i} = \mathbf{X_i^T W_i X_i}.
\end{aligned} \tag{C.8}$$

- Last, the working response can also be calculated using matrix $\mathbf{D}_i$ and the difference between the observed response and the expected response:

$$\mathbf{z}_i = \left(\mathbf{X}_i\boldsymbol{\beta} + \left(\mathbf{W}_i\right)^{-1}\mathbf{u}_i\right) = \left(\mathbf{X}_i\boldsymbol{\beta} + \left[\mathbf{D}_i\right]^{-T}\left[\mathbf{y}_i - \boldsymbol{\mu}_i\right]\right). \tag{C.9}$$

D. Ordinal Probit Matrices in the MVGLM Framework

Based on Kedem and Fokianos [58], Tutz [113], Agresti [1], the expressions of the working weights matrix, working response vector, covariance matrix, inverse covariance matrix in the MVGLM frame-

work are:

$$\mathbf{W}_i = \mathbf{D}_i \mathbf{\Sigma}_i^{-1} \mathbf{D}_i^T.$$

$$\mathbf{z}_i = \mathbf{X}_i \boldsymbol{\beta} + \mathbf{D}_i^{-T} \left[\mathbf{y}_i - \boldsymbol{\pi}_i \right].$$

$$\mathbf{\Sigma}_i = \begin{bmatrix} \left[\pi_{(1)i}\left(1-\pi_{(1)i}\right)\right] & -\pi_{(1)i}\pi_{(2)i} & \cdots & -\pi_{(1)i}\pi_{(M)i} \\ -\pi_{(2)i}\pi_{(1)i} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & -\pi_{(M-1)i}\pi_{(M)i} \\ -\pi_{(M)i}\pi_{(1)i} & \cdots & -\pi_{(M)i}\pi_{(M-1)i} & \left[\pi_{(M)i}\left(1-\pi_{(M)i}\right)\right] \end{bmatrix}.$$

$$\mathbf{\Sigma}_i^{-1} = diag\left(\frac{1}{\pi_{(1)i}}, \frac{1}{\pi_{(2)i}}, ..., \frac{1}{\pi_{(M)i}}\right) + \frac{1}{1 - \sum_{j=1}^{M} \pi_{(j)i}} \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}.$$

$$\mathbf{D}_i = \begin{bmatrix} \phi\left(\eta_{(1)i}\right) & -\phi\left(\eta_{(1)i}\right) & 0 & \cdots & 0 \\ 0 & \phi\left(\eta_{(2)i}\right) & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -\phi\left(\eta_{(M-1)i}\right) \\ 0 & \cdots & \cdots & 0 & \phi\left(\eta_{(M)i}\right) \end{bmatrix}. \tag{D.1}$$

$$\mathbf{D}_i^{-1} = \begin{bmatrix} \frac{1}{\phi\left(\eta_{(1)i}\right)} & \frac{1}{\phi\left(\eta_{(2)i}\right)} & \frac{1}{\phi\left(\eta_{(3)i}\right)} & \cdots & \frac{1}{\phi\left(\eta_{(M)i}\right)} \\ 0 & \frac{1}{\phi\left(\eta_{(2)i}\right)} & \vdots & \ddots & \vdots \\ \vdots & \ddots & \frac{1}{\phi\left(\eta_{(3)i}\right)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \frac{1}{\phi\left(\eta_{(M)i}\right)} \\ 0 & \cdots & \cdots & 0 & \frac{1}{\phi\left(\eta_{(M)i}\right)} \end{bmatrix}.$$

where ϕ is the normal distribution.

E. Generated Example

The example is generated as follows:

- There are $p = 5$ covariates. The 5 covariates are standard normal random variables.
- There are $n = 30$ samples.
- Response Y is an ordinal variable that can take 4 values (0,1,2,3)
- There is a latent variable Y^*. We calculate y^* as follows: $y_i^* \leftarrow x_{i1} - 3x_{i2} + x_{i3} + \varepsilon_i$, where ε_i follows a standard normal distribution.
- Then $y_i \leftarrow \mathcal{I}(y_i^* \geq (-1)) + \mathcal{I}(y_i^* \geq 0.5) + \mathcal{I}(y_i^* \geq 3)$

F. Positive Definite Matrix and Condition Number

The Fisher information matrix of the penalised log-likelihood may not be invertible. One reason for this is that in the case of an elastic-net penalised VGLM model, a partial identity matrix multiplied by a regularisation parameter is added to the Fisher information matrix of the VGLM model. The zeros of the partial identity matrix correspond to the intercepts that should not be penalised. When the regularisation parameter λ is very large, the matrix may not be invertible as it may have a large condition number. We then use the following method:

- First, we either calculate the eigenvalue or svd decomposition of the Fisher information matrix of the penalised log-likelihood, using R functions 'eigen' or 'svd' and floor every eigen value/singular value at ϵ_4.
- If the matrix is still not invertible, we use a technique proposed by Tong et al. [110]. The technique consists in finding the closest matrix to the Fisher information matrix that verifies a given condition number κ. Tong et al. propose to determine the closest approximate matrix that verifies the condition number constraint. They define the 'closest' matrix as the matrix that minimises the Frobenius norm of the difference between the original matrix and the approximate matrix. Matrix $\mathbf{F}$ is an $(M+Mp)\times(M+Mp)$ matrix. Tong et al. [110] showed that the minimum can be determined by running two loops using numbers M_1 and M_2, where $1 \leq M_1 \leq (M+Mp-1)$ and $1 \leq M_2 \leq (M+Mp-M_1)$. We then calculate:

$$u^*(M_1, M_2) = \frac{\kappa\left(\sum_{m=1}^{M_1}\xi_m\right) + \left(\sum_{m=M+Mp-M_2+1}^{M+Mp}\xi_m\right)}{\kappa^2 M_1 + M_2}. \tag{F.1}$$

  Where the ξ_m are the eigenvalues or singular values of the Fisher information matrix ranked from the largest (ξ_1) to the smallest (ξ_{M+Mp}) after we made sure that no value is less than ϵ_4. We use $\epsilon_4 = 1e-5$. κ is the condition number. Then we find the values of M_1 and M_2 that minimise:

$$\begin{aligned}&\min_{1\leq M_1\leq(M+Mp-1),1\leq M_2\leq(M+Mp-M_1)}\left(\left\|\mathbf{F}-\tilde{\mathbf{F}}\right\|^2_{Frobenius}(M_1,M_2)\right)\\ &=\min_{M_1,M_2}\left(\sum_{m=1}^{M_1}\left\{\xi_m-\kappa\left[u^*(M_1,M_2)\right]\right\}^2+\sum_{m=M+Mp-M_2+1}^{M+Mp}\left\{\xi_m-\left[u^*(M_1,M_2)\right]\right\}^2\right).\end{aligned} \tag{F.2}$$

  Then to obtain matrix $\tilde{\mathbf{F}}$, we replace the eigenvalues/singular values by:

$$\tilde{\xi}_m = \left\{\left(u^*(M_1^*, M_2^*) \vee \xi_m\right) \wedge \left(\kappa u^*(M_1^*, M_2^*)\right)\right\}. \tag{F.3}$$

G. Gain/Loss Function and Strategy

The strategy and gain/loss function can be expressed as follows. At each time step, first for given regularisation parameters α and λ, calculate the probabilities of a positive change, no change and negative change. Then based on these probabilities, calculate the weighted expected gain of a positive change and of a negative change. The strategy consists in predicting only when the weighted expected

gain of a positive change or the weighted expected gain of a negative change is positive. Last, we can calculate the realised gain/loss based on the weighted expected gains/losses and the observation of mid-market price changes. The algorithm below applies to both cross validation (in-sample) and to the calculation of the cumulative gain and loss in the out-of-sample part of the dataset. If applied during the cross validation, then n is the size of one of the folds. If in the out-of-sample, then n is the size of the out-of-sample part of the dataset.

for *t in 1:(n-1)* **do**

Calculate the estimated probabilities of a positive change, negative change, no change with ordinal probit:

$\hat{P}(Y_{t+1} = +1|X_t, \lambda)$, $\hat{P}(Y_{t+1} = 0|X_t, \lambda)$, $\hat{P}(Y_{t+1} = -1|X_t, \lambda)$ using estimated ordinal probit using λ.

Calculate the expected gains for a positive change prediction and negative change prediction:

$E_t[G^{up}_{t+1}(\lambda)] = w_+\hat{P}(Y_{t+1} = +1|X_t, \lambda) + w_0\hat{P}(Y_{t+1} = 0|X_t, \lambda) + w_-\hat{P}(Y_{t+1} = -1|X_t, \lambda)$

$E_t[G^{down}_{t+1}(\lambda)] = w_+\hat{P}(Y_{t+1} = -1|X_t, \lambda) + w_0\hat{P}(Y_{t+1} = 0|X_t, \lambda) + w_-\hat{P}(Y_{t+1} = +1|X_t, \lambda)$

Calculate the realised gain or loss:

$$\begin{aligned} RG_{t+1}(\lambda) =& \mathcal{I}(E_t[G^{down}_{t+1}(\lambda)] > 0) \times [w_+\mathcal{I}(y_{t+1} = -1) + w_0\mathcal{I}(y_{t+1} = 0) + w_-\mathcal{I}(y_{t+1} = +1)] \\ &+ \mathcal{I}(E_t[G^{up}_{t+1}(\lambda)] > 0) \times [w_+\mathcal{I}(y_{t+1} = 1) + w_0\mathcal{I}(y_{t+1} = 0) + w_-\mathcal{I}(y_{t+1} = -1)] \end{aligned}$$

end

Algorithm 3: Implementation of strategy. $\mathcal{I}(condition)$ is the indicator function that is equal to 1 if the condition is verified, 0 otherwise. To simplify, we dropped the elastic net parameter α from the notation.

Part III

EM Proximal Newton/Fisher Scoring Method for Bivariate Poisson Regression with Application to Health Care Data

1. Introduction

As a second application of proximal Fisher scoring, we revisit an example given by Karlis and Ntzoufras [55]. Karlis and Ntzoufras propose a bivariate Poisson regression model where the parameters are estimated with the Expectation-Maximisation (EM) algorithm. The EM algorithm is used because a bivariate Poisson can be modelled as a sum of three independent Poisson latent random variables. As an example, Karlis and Ntzoufras apply the model to health care data originally used by Cameron and Trivedi [26], [25] and show that a model with a covariance that depends on a covariate fits better than a model with a covariance that is fixed or a model with no covariance. Karlis and Ntzoufras use a single covariate for the covariance.

In this document, we use regularisation to find the subset of covariates that influence the two observed counts and their covariance. The bivariate Poisson is penalised with an elastic net penalty. Because the penalty is not differentiable, the maximum penalised likelihood estimator of the parameters is determined using an EM algorithm where, in the M step, the Fisher scoring is replaced with a proximal Newton/Fisher scoring. This method is referred to as 'EM proximal Newton/Fisher scoring'.

Note that the problem solved, algorithm used and methodology used in this part are related to several articles proposed for penalised zero inflated count regression. In particular, the methodology used is almost identical to the one used in Wang et al. [123], Wang et al. [122] and Tang et al. [108] (the method used is similar to an EM proximal Newton/Fisher scoring with a constant step of 1). Other related articles that solve penalised zero inflated Poisson models include Buu et al. [24] (proximal Fisher scoring type algorithm solved using LARS), Zeng et al. [138] (quadratic approximation for adaptive LASSO), Su et al. [107] (quadratic approximation for multiple inflated Poisson), Wang et al. [124] (penalised IRLS solved using coordinate descent).

In section 2, the bivariate Poisson regression and an EM-proximal Newton/Fisher scoring algorithm are introduced. In section 3, the data from the example of Karlis and Ntzoufras is described and the algorithm described previously is used to select variables.

2. Bivariate Poisson Regression and EM proximal Fisher Scoring

The bivariate Poisson regression is used to model two correlated count variables. It can be constructed with a trivariate reduction method. This method, that can be found in Kocherlakota and Kocherlakota [64] chapter 4, consists in assuming that each count observation is the sum of two independent Poisson latent random variables. Only three Poisson latent random variables are used as one of the latent variables is common to both responses and drives the covariance of the count observations. In section 2.1, the bivariate Poisson regression is introduced. The elastic net penalty is used to penalise the log-likelihood. In section 2.2, the EM proximal Fisher algorithm is described.

2.1. Bivariate Poisson Regression

This section is based on Kocherlakota and Kocherlakota [64] chapter 4, Karlis and Ntzoufras [55], Karlis [53]. We assume that we want to model two correlated count random variables Y_1 and Y_2. We

assume that these two random variables are linear combinations of three independent latent random variables W_0, W_1, W_2 that are Poisson distributed:

$$P(W_k = w_k|\gamma_k) = \frac{\exp(-\gamma_k)\gamma_k^{w_k}}{w_k!} \quad k = 0, 1, 2. \tag{2.1}$$

The correlated random variables Y_1 and Y_2 are combinations of the latent random variables:

$$\begin{aligned} Y_1 &= W_1 + W_0. \\ Y_2 &= W_2 + W_0. \end{aligned} \tag{2.2}$$

Then the expected value, covariance and correlation between the observable random variables are:

$$\begin{aligned} E[Y_k|\gamma_0, \gamma_k] &= \gamma_0 + \gamma_k \quad k = 1, 2. \\ Covar[Y_1, Y_2|\gamma_0, \gamma_1, \gamma_2] &= \gamma_0. \\ Corr[Y_1, Y_2|\gamma_0, \gamma_1, \gamma_2] &= \frac{\gamma_0}{\sqrt{(\gamma_0 + \gamma_1)(\gamma_0 + \gamma_2)}}. \end{aligned} \tag{2.3}$$

The joint probability of Y_1 and Y_2 is:

$$\begin{aligned} &BP\left(y_1, y_2|\gamma_1, \gamma_2, \gamma_0\right) = P\left(Y_1 = y_1, Y_2 = y_2|\gamma_1, \gamma_2, \gamma_0\right) = \\ &\exp\{-\left(\gamma_1 + \gamma_2 + \gamma_0\right)\}\frac{\gamma_1^{y_1}}{y_1!}\frac{\gamma_2^{y_2}}{y_2!}\sum_{l=0}^{\min(y_1,y_2)}\binom{y_1}{l}\binom{y_2}{l}l!\left(\frac{\gamma_0}{\gamma_1\gamma_2}\right)^l. \end{aligned} \tag{2.4}$$

See Kawamura [57] for a derivation. To introduce covariates, it is assumed that γ_k is a function of the covariates:

$$\gamma_k = \exp(\beta_{(k)0} - \mathbf{X}^T\boldsymbol{\beta}_{(k)1:p_k}) \quad k = 0, 1, 2. \tag{2.5}$$

Note that here in the context of VGLMs, $M = 3$. However, we use numbering 0, 1, 2. Also note that we may also have used a positive sign for the coefficients of the covariates. Then, based on the bivariate Poisson regression, the elastic net penalised log-likelihood is:

$$\begin{aligned} l\left(\boldsymbol{\beta}_{(0)}, \boldsymbol{\beta}_{(1)}, \boldsymbol{\beta}_{(2)}\right) = &\frac{1}{n}\sum_{i=1}^{n}\log\left(BP\left(y_{1,i}, y_{2,i}|\boldsymbol{\beta}_{(0)}, \boldsymbol{\beta}_{(1)}, \boldsymbol{\beta}_{(2)}\right)\right) \\ &- \frac{\lambda_0\left(1-\alpha\right)}{2}\boldsymbol{\beta}_{(0)}^T\mathbf{I}_{0,0}\boldsymbol{\beta}_{(0)} - \lambda_0\alpha\boldsymbol{\beta}_{(0)}^T\mathbf{I}_{0,0}sign\left(\boldsymbol{\beta}_{(0)}\right) \\ &- \frac{\lambda_1\left(1-\alpha\right)}{2}\boldsymbol{\beta}_{(1)}^T\mathbf{I}_{0,1}\boldsymbol{\beta}_{(1)} - \lambda_1\alpha\boldsymbol{\beta}_{(1)}^T\mathbf{I}_{0,1}sign\left(\boldsymbol{\beta}_{(1)}\right) \\ &- \frac{\lambda_2\left(1-\alpha\right)}{2}\boldsymbol{\beta}_{(2)}^T\mathbf{I}_{0,2}\boldsymbol{\beta}_{(2)} - \lambda_2\alpha\boldsymbol{\beta}_{(2)}^T\mathbf{I}_{0,2}sign\left(\boldsymbol{\beta}_{(2)}\right). \end{aligned} \tag{2.6}$$

where $\mathbf{I}_{0,k}, k = 0, 1, 2$ is a partial identity matrix of size $(p_k+1)\times(p_k+1)$ where the first column is a column of 0s. In the next section, the EM-proximal Fisher scoring is used to maximise log-likelihood (2.6).

2.2. EM-Proximal Fisher Scoring Algorithm for Elastic Net Penalised Bivariate Poisson

From Karlis and Ntzoufras [55], to optimise the log-likelihood described previously, as the model has three latent variables W_0, W_1, W_2, we can use the EM algorithm. The algorithm consists first in calculating the expected value of W_0 (E-step). Once the expected value of W_0 is known then the value of the two other latent variables W_1 and W_2 can be deduced from the observations y_1 and

y_2 (as $Y_1 = W_0 + W_1$, $Y_2 = W_0 + W_2$). Then the coefficients for the three independent Poisson regressions composing the bivariate Poisson can be estimated (M-step). Using the description of the EM algorithm from Bermudez and Karlis [13] and similarly to the EM algorithm for regularised zero inflated Poisson as described in Wang et al. [123], the E-step is:

E-Step

The expected value of W_0 at the a^{th} iteration is:

$$\begin{aligned} w_{0,i}^{(a)} &= E\left[W_{0,i}|Y_{1,i}=y_{1,i}, Y_{2,i}=y_{2,i}, \gamma_0^{(a)}, \gamma_1^{(a)}, \gamma_2^{(a)}\right] \\ &= \begin{cases} \gamma_0^{(a)} \frac{BP\left(y_{1,i}-1, y_{2,i}-1|\gamma_0^{(a)}, \gamma_1^{(a)}, \gamma_2^{(a)}\right)}{BP\left(y_{1,i}, y_{2,i}|\gamma_0^{(a)}, \gamma_1^{(a)}, \gamma_2^{(a)}\right)} & \text{if } y_{1,i}>0, y_{2,i}>0 \cdot \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{2.7}$$

$i = 1, ..., n$.
For further explanation, see Bermudez and Karlis [13].

M-step

In the case of the EM-proximal Fisher scoring, we now estimate three sets of parameters using $w_1^{(a)} = y_1 - w_0^{(a)}$,$w_2^{(a)} = y_2 - w_0^{(a)}$:

$$\boldsymbol{\beta}_{(1)}^{(a+1)} = \boldsymbol{\beta}_{(1)}^{(a)} + t_a \left\{ \underset{\overline{\boldsymbol{\beta}}}{\operatorname{argmin}} \left(\frac{1}{2} \left[\overline{\boldsymbol{\beta}}\right]^T \left[\mathbf{F}_{1,a}^{\lambda_1}\left(\mathbf{y}_1 - \mathbf{w}_0^{(a)}\right)\right] \overline{\boldsymbol{\beta}} - \left[\mathbf{G}_{1,F,a}^{\lambda_1}\left(\mathbf{y}_1 - \mathbf{w}_0^{(a)}\right)\right]^T \overline{\boldsymbol{\beta}} + \lambda_1 \alpha \left[\overline{\boldsymbol{\beta}}\right]^T \mathbf{I}_{0,1} sign\left(\overline{\boldsymbol{\beta}}\right) \right) - \boldsymbol{\beta}_{(1)}^{(a)} \right\}.$$

$$\boldsymbol{\beta}_{(2)}^{(a+1)} = \boldsymbol{\beta}_{(2)}^{(a)} + t_a \left\{ \underset{\tilde{\boldsymbol{\beta}}}{\operatorname{argmin}} \left(\frac{1}{2} \left[\tilde{\boldsymbol{\beta}}\right]^T \left[\mathbf{F}_{2,a}^{\lambda_2}\left(\mathbf{y}_2 - \mathbf{w}_0^{(a)}\right)\right] \tilde{\boldsymbol{\beta}} - \left[\mathbf{G}_{2,F,a}^{\lambda_2}\left(\mathbf{y}_2 - \mathbf{w}_0^{(a)}\right)\right]^T \tilde{\boldsymbol{\beta}} + \lambda_2 \alpha \left[\tilde{\boldsymbol{\beta}}\right]^T \mathbf{I}_{0,2} sign\left(\tilde{\boldsymbol{\beta}}\right) \right) - \boldsymbol{\beta}_{(2)}^{(a)} \right\}.$$

$$\boldsymbol{\beta}_{(0)}^{(a+1)} = \boldsymbol{\beta}_{(0)}^{(a)} + t_a \left\{ \underset{\check{\boldsymbol{\beta}}}{\operatorname{argmin}} \left(\frac{1}{2} \left[\check{\boldsymbol{\beta}}\right]^T \left[\mathbf{F}_{0,a}^{\lambda_0}\left(\mathbf{w}_0^{(a)}\right)\right] \check{\boldsymbol{\beta}} - \left[\mathbf{G}_{0,F,a}^{\lambda_0}\left(\mathbf{w}_0^{(a)}\right)\right]^T \check{\boldsymbol{\beta}} + \lambda_0 \alpha \left[\check{\boldsymbol{\beta}}\right]^T \mathbf{I}_{0,0} sign\left(\check{\boldsymbol{\beta}}\right) \right) - \boldsymbol{\beta}_{(0)}^{(a)} \right\}.$$

$$\mathbf{F}_{k,a}^{\lambda_k}(\mathbf{y}) = \mathbf{F}_{k,a}(\mathbf{y}) + \lambda_k (1-\alpha) \mathbf{I}_{0,k} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{F}_{i,k,a}(\mathbf{y}_i) + \lambda_k (1-\alpha) \mathbf{I}_{0,k} \qquad k = 0, 1, 2.$$

$$\mathbf{U}_{k,F,a}^{\lambda_k}(\mathbf{y}) = \mathbf{U}_{k,a}(\mathbf{y}) - \lambda_k (1-\alpha) \mathbf{I}_{0,k} \boldsymbol{\beta}_{(k)}^{(a)} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{U}_{i,k,a}(\mathbf{y}_i) - \lambda_k (1-\alpha) \mathbf{I}_{0,k} \boldsymbol{\beta}_{(k)}^{(a)}.$$

$$\mathbf{G}_{k,F,a}^{\lambda_k}(\mathbf{y}) = \left[\boldsymbol{\beta}_{(k)}^{(a)}\right]^T \mathbf{F}_{k,a}(\mathbf{y}) + \mathbf{U}_{k,F,a}^{\lambda_k}(\mathbf{y}) = \left[\boldsymbol{\beta}_{(k)}^{(a)}\right]^T \frac{1}{n} \sum_{i=1}^{n} \mathbf{F}_{i,k,a}(\mathbf{y}_i) + \mathbf{U}_{k,F,a}^{\lambda_k}(\mathbf{y}).$$

$\mathbf{F}_{k,a}(\mathbf{y})$ is the Fisher information matrix of the
log-likelihood of Poisson regression calculated with a response $\mathbf{y}$.
$\mathbf{U}_{k,a}(\mathbf{y})$ is the score vector of the
log-likelihood of a Poisson regression calculated with a response $\mathbf{y}$.

$$\boldsymbol{\beta}_{(k)} = \left(\beta_{(k)0}; -\boldsymbol{\beta}_{(k)1:p_k}^T\right)^T. \tag{2.8}$$

The number of covariates for $\mathbf{W}_0$, $\mathbf{W}_1$ and $\mathbf{W}_2$ is p_0, p_1, p_2. $\overline{\boldsymbol{\beta}}$ is a vector of parameters to optimise of size p_1. $\tilde{\boldsymbol{\beta}}$ is a vector of parameters to optimise of size p_2. $\check{\boldsymbol{\beta}}$ is a vector of parameters to optimise of size p_0. Similarly to the EM algorithm proposed by Wang et al. [123], [122], we use several regularisation parameters $\lambda_0, \lambda_1, \lambda_2$. The code is written in R. To calculate bivariate Poisson

probabilities, function 'bivpois' created by Karlis and Ntzoufras is used[4] The Fisher information matrix and score vectors for a Poisson regression can be found in Appendix A. In the next section, an example application of the EM-proximal Newton/Fisher scoring is discussed.

3. Application of the EM Proximal Newton/Fisher Scoring to Health Care Data

First, in section 3.1, the data used is described, then in section 3.2, the EM algorithm and cross validation procedures and described and some results are presented.

3.1. Data

From Cameron and Trivedi [26], the data is the result of a health care survey in Australia for 1977-1978. The data used in this document was obtained from the online archive of package 'bivpois' created by Karlis and Ntzoufras [54]. The table in Appendix B gives the two responses of the regression and all the covariates used in this document. See Cameron and Trivedi [26] for further data description.

3.2. Penalised Regression

In the example taken by Karlis and Ntzoufras, the two responses used are the number of visits to a doctor in the past 2 weeks (y_1) and the number of prescribed medications used in the past 2 days (y_2). Karlis and Ntzoufras use three covariates for γ_1 and γ_2 (gender, age, income) and one covariate for γ_0 (gender). We note that other covariates are available in the dataset that could be used to explain the variance of the responses. We use the elastic net penalised bivariate Poisson to select the variables that influence γ_1, γ_2 and γ_0. We use the following setup:

- To limit the calculation, we use 5 covariates: 'gender','age','income','illness','health score'.
- Instead of using the full dataset, 500 samples from the original dataset are used to reduce the computation time.
- The Bayesian Information Criterion (BIC) is used to assess the fit of the model. Zou et al. [143], Wang et al. [118] showed that the number of non-zero coefficients post-lasso was an unbiased estimator of the effective degrees of freedom. This is the criterion chosen by Wang et al. [123] in the case of the EM for zero inflated Poisson.
- The BIC is calculated on a grid of points $(\lambda_0, \lambda_1, \lambda_2)$. We note $BIC(\lambda_0, \lambda_1, \lambda_2)$ to indicate the BIC estimated at a particular combination of penalties for γ_0, γ_1 and γ_2.
- To reduce the computation time, we assume that $\lambda_0 = \lambda_1$ and therefore, the grid is a two-dimensional grid.
- We use a grid of 10×10 points for (λ_0, λ_2). We start with fixed upper and lower bounds for the grid. After the first estimation, we can lower the upper bound and raise the lower bound of the grid to obtain more accurate estimations.
- To limit the number of calculations, the elastic net parameter α is set at 0.5.

Covariates selected are:

[4] package 'bivpois' was not available on the CRAN repository but was retrieved online on the website of Ioannis Ntzoufras (address given in reference [54]).

	covariates selected
γ_0	age, income, illness
γ_1	age, illness, health score
γ_2	gender, age, income, illness

We note that four out of five covariates are selected for γ_2. This could indicate we might want to use a different selection criterion for choosing the regularisation parameters. Note that the algorithm may be more difficult to use with larger systems. As function $BIC(\lambda_0, \lambda_1, \lambda_2)$ is not convex, there may be several points of the grid with similar BIC value. The chosen gridpoint based on the lowest BIC may then change completely from one step of the EM algorithm to the next. Therefore, we may not be able to rely on early estimations of the BIC values on the grid to choose the gridpoint with the lowest BIC and we may have to wait until the EM algorithm has converged to a high degree of accuracy.

4. Conclusion and Future Steps

We used an EM-proximal Newton/Fisher scoring algorithm for a bivariate Poisson model and applied it to an example proposed by Karlis and Ntzoufras based on data provided by Cameron and Trivedi. By using regularisation, we selected covariates that influence the different parameters of the model, including the covariance between the responses. This analysis has a number of limitations that could be addressed:

- Parallel processing: The regularisation parameter selection could be run in parallel. We could then increase the degree of accuracy of the grid and potentially calculate a 3 dimensional grid.
- Selection criterion: A number of alternatives to BIC have been proposed to select variables and may be investigated to obtain a model that is more sparse. For example, the use of the Generalised Information Criterion (GIC) could be investigated (see Fan and Tang [35], Kim et al. [62], Zhang et al. [140]. We could use the comparisons from Flynn et al. [37], Chand [27] and Liu [72]).
- Post estimation confidence interval: We would like to test our coefficients post-selection. However, estimation post-lasso is not an established theory and is an active area of research. This is why Goeman et al. [42] specifically chose not to provide standard errors in their package 'penalized'. A summary different research axis can be found in Hastie et al. [47].
- Zero inflation: Karlis and Ntzoufras [55] note that a zero inflated model may be a better fit to the data.
- Limited correlation: Gurmu and Elder [46] note that the possible correlations of the bivariate Poisson are limited in range. They propose a model with an unrestricted correlation and this model could be used to replace the trivariate Poisson decomposition.

Acknowledgements

I would like to thank my supervisor Arnaud Doucet, my examiners Dino Sejdinovic and Thomas Yee as well as Vimal Balasubramaniam, Francesca Brusa, Coralia Cartis, Rémy Cottet, Yves-Laurent Kom Samo, Harald Oberhauser, Gareth Peters, Johannes Ruf, Galen Sher, Pedro Vitória, James Wolter for comments. I would also like to thank the Oxford-Man Institute of Quantitative Finance for research support.

Appendix for Part III

A. Score Vector and Fisher Information Matrix for a Poisson Regression

$$
\begin{aligned}
l(\boldsymbol{\beta}) &= \sum_i y_i \left(\beta_0 - \mathbf{X}_i^T \boldsymbol{\beta}_{1:p}\right) - \exp\left(\beta_0 - \mathbf{X}_i^T \boldsymbol{\beta}_{1:p}\right) - \log(y_i!). \\
\frac{\partial l}{\partial \beta_0} &= \sum_i \left(y_i - \exp\left(\beta_0 - \mathbf{X}_i^T \boldsymbol{\beta}_{1:p}\right)\right). \\
\frac{\partial l}{\partial \boldsymbol{\beta}_{1:p}} &= -\frac{\partial l}{\partial \beta_0}\mathbf{X}_i. \\
\frac{\partial^2 l}{\partial \beta_0^2} &= \sum_i \left[-\exp\left(\beta_0 - \mathbf{X}_i^T \boldsymbol{\beta}_{1:p}\right)\right]. \\
\frac{\partial^2 l}{\partial \beta_0 \partial \boldsymbol{\beta}_{1:p}} &= -\frac{\partial^2 l}{\partial \beta_0^2}\mathbf{X}_i. \\
\frac{\partial^2 l}{\partial \boldsymbol{\beta}_{1:p} \partial \boldsymbol{\beta}_{1:p}^T} &= \frac{\partial^2 l}{\partial \beta_0^2}\mathbf{X}_i \mathbf{X}_i^T. \\
\mathbf{U} &= \begin{bmatrix} \frac{\partial l}{\partial \beta_0}. \\ \frac{\partial l}{\partial \boldsymbol{\beta}_{1:p}} \end{bmatrix}.
\end{aligned}
\tag{A.1}
$$

$$
\mathbf{F} = \begin{bmatrix} -\frac{\partial^2 l}{\partial \beta_0^2} & -\frac{\partial^2 l}{\partial \beta_0 \partial \boldsymbol{\beta}_{1:p}^T} \\ -\frac{\partial^2 l}{\partial \beta_0 \partial \boldsymbol{\beta}_{1:p}} & -\frac{\partial^2 l}{\partial \boldsymbol{\beta}_{1:p} \partial \boldsymbol{\beta}_{1:p}^T} \end{bmatrix}.
$$

B. Data

The table below is based on Cameron and Trivedi [26], Cameron and Trivedi [25], table 3.2 and Karlis and Ntzoufras [55]:

	variable	**description**
	gender	1 for female, 0 for male
	age	in years divided by 100
	income	in tens of thousands of dollars
	illness	nb of illnesses in the past 2 weeks
	health score	health questionnaire score
y1	doctor consultations	nb of consultations with doctor or specialist in past 2 weeks
y2	prescribed medications	nb of prescribed medications used in past 2 days

Bibliography

[1] Alan Agresti. *Categorical Data Analysis, Third Edition*. John Wiley & Sons, 2013.

[2] Syed Ejaz Ahmed, Shakhawat Hossain, and Kjell A Doksum. LASSO and Shrinkage Estimation in Weibull Censored Regression Models. *Journal of Statistical Planning and Inference*, 142(6):1273–1284, 2012.

[3] Marouane Anane. *Une Approche Mathématique de l'Investissement Boursier*. PhD thesis, Ecole Centrale Paris, 2015.

[4] Kellie J Archer, Jiayi Hou, Qing Zhou, Kyle Ferber, John G Layne, and Amanda E Gentry. ordinalgmifs: An R Package for Ordinal Regression in High-dimensional Data Settings. *Cancer Informatics*, 13:187, 2014.

[5] Kellie J. Archer and André A.A. Williams. L1 Penalized Continuation Ratio Models for Ordinal Response Prediction using High-Dimensional Datasets. *Statistics in Medicine*, 31(14):1464–1474, 2012.

[6] Ioannis K Argyros. *Convergence and Applications of Newton-type Iterations*. Springer Science & Business Media, 2008.

[7] Ioannis K Argyros and Á Alberto Magreñán. Local Convergence Analysis of Proximal Gauss-Newton Method for Penalized Nonlinear Least Squares Problems. *Applied Mathematics and Computation*, 241:401–408, 2014.

[8] Luigi Augugliaro, Angelo M Mineo, and Ernst C Wit. dglars: An R Package to Estimate Sparse Generalized Linear Models. *Journal of Statistical Software*, 59(8):1–40, 2014.

[9] Marta Avalos Fernandez. Modèles Additifs Parcimonieux, PhD thesis. *Université de Technologie de Compiègne-UTC*, 2004.

[10] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Convex Optimization with Sparsity-inducing Norms. *Optimization for Machine Learning*, pages 19–53, 2012.

[11] Amir Beck and Marc Teboulle. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[12] Stephen Becker and Jalal Fadili. A Quasi-Newton Proximal Splitting Method. In *Advances in Neural Information Processing Systems*, pages 2618–2626, 2012.

[13] Lluís Bermudez and Dimitris Karlis. Mixture of Bivariate Poisson Regression Models with an Application to Insurance. XREAP2011-10, available at SSRN:`http://ssrn.com/abstract=1884898`, 2011.

[14] Dimitri P Bertsekas. Incremental Gradient, Subgradient, and Proximal Methods for Convex Optimization: A Survey. *Optimization for Machine Learning*, 2010:1–38, 2011.

[15] Yatao Bian, Xiong Li, Mingqi Cao, and Yuncai Liu. Bundle CDN: A Highly Parallelized Approach for Large-Scale l1-Regularized Logistic Regression. In *Machine Learning and Knowledge Discovery in Databases*, pages 81–95. Springer, 2013.

[16] Yatao Bian, Xiong Li, and Yuncai Liu. Parallel Coordinate Descent Newton for Large Scale L1 Regularized Minimization. arXiv preprint, `http://arxiv.org/abs/1306.4080`, 2013.

[17] Katarzyna Bien, Ingmar Nolte, and Winfried Pohlmeier. An Inflated Multivariate Integer Count Hurdle Model: an Application to Bid and Ask Quote Dynamics. *Journal of Applied Econometrics*, 26(4):669–707, 2011.

[18] Nicolai Bissantz, Lutz Dümbgen, Axel Munk, and Bernd Stratmann. Convergence Analysis of Generalized Iteratively Reweighted Least Squares Algorithms on Convex Function Spaces. *SIAM Journal on Optimization*, 19(4):1828–1845, 2009.

[19] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[20] Patrick Breheny and Jian Huang. Penalized Methods for Bi-Level Variable Selection. *Statistics and its Interface*, 2(3):369, 2009.

[21] Patrick Breheny and Jian Huang. Coordinate Descent Algorithms for Nonconvex Penalized Regression, with Applications to Biological Feature Selection. *The Annals of Applied Statistics*, 5(1):232, 2011.

[22] Peter Bühlmann and Sara Van De Geer. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer Science & Business Media, 2011.

[23] Ján Buša. Solving Quadratic Programming Problem with Linear Constraints Containing Absolute Values. *Acta Electrotechnica et Informatica*, 12(3):11–18, 2012.

[24] Anne Buu, Norman J Johnson, Runze Li, and Xianming Tan. New Variable Selection Methods for Zero-Inflated Count Data with Applications to the Substance Abuse Field. *Statistics in Medicine*, 30(18):2326–2340, 2011.

[25] A Colin Cameron and Pravin K Trivedi. *Regression Analysis of Count Data*, volume 53. Cambridge University Press, 2013.

[26] A Colin Cameron, Pravin K Trivedi, Frank Milne, and John Piggott. A Microeconometric Model of the Demand for Health Care and Health Insurance in Australia. *The Review of Economic Studies*, 55(1):85–106, 1988.

[27] Sohail Chand. On Tuning Parameter Selection of Lasso-Type Methods-a Monte Carlo Study. In *Proceedings of 2012 9th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pages 120–129. IEEE, 2012.

[28] Alexandra Chouldechova and Trevor Hastie. Generalized Additive Model Selection. arXiv preprint, `http://arxiv.org/abs/1506.03850`, 2015.

[29] Patrick L Combettes and Jean-Christophe Pesquet. Proximal Splitting Methods in Signal Processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer, 2011.

[30] Meenakshi Devidas and E Olusegun George. Monotonic Algorithms for Maximum Likelihood Estimation in Generalized Linear Models. *Sankhyā: The Indian Journal of Statistics, Series B*, pages 382–396, 1999.

[31] Amit Dhurandhar and Marek Petrik. Efficient and Accurate Methods for Updating Generalized Linear Models with Multiple Feature Additions. *The Journal of Machine Learning Research*, 15(1):2607–2627, 2014.

[32] David Drießlein. Penalisierungsansätze in Ordinalen Regressionsmodellen, Bsc Thesis, Ludwig Maximilian Universität Munich, 2013.

[33] Ludwig Fahrmeir and Gerhard Tutz. *Multivariate Statistical Modelling Based on Generalized Linear Models, Second Edition*. Springer Verlag, 2001.

[34] Jianqing Fan and Runze Li. Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

[35] Yingying Fan and Cheng Yong Tang. Tuning Parameter Selection in High Dimensional Penalized Likelihood. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(3):531–552, 2013.

[36] Mário A.T. Figueiredo, Robert D Nowak, and Stephen J Wright. Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):586–597, 2007.

[37] Cheryl Flynn, Clifford M Hurvich, and Jeffrey S Simonoff. Efficiency and Consistency for Regularization Parameter Selection in Penalized Regression: Asymptotics and Finite-Sample Corrections. *NYU Working Paper No. 2451/31317*, 2011.

[38] Jerome Friedman, Trevor Hastie, Holger Höfling, and Robert Tibshirani. Pathwise Coordinate Optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

[39] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*, 33(1):1, 2010.

[40] Yoel Furman. VAR Estimation with the Adaptive Elastic Net. available at SSRN: `http://ssrn.com/abstract=2456510`, 2014.

[41] Jelle J Goeman. L1 Penalized Estimation in the Cox Proportional Hazards Model. *Biometrical Journal*, 52(1):70–84, 2010.

[42] Jelle J Goeman, Rosa J Meijer, and Nimisha Chaturvedi. L1 and L2 Penalized Regression Models (R package 'penalized' user guide). `https://cran.r-project.org/web/packages/penalized/`, 2014.

[43] Geoff Gordon and Ryan Tibshirani. Subgradient Method, Optimization 10-725/36-725. `https://www.cs.cmu.edu/~ggordon/10725-F12/slides/06-sg-method.pdf`, 2012.

[44] Martin D Gould, Mason A Porter, Stacy Williams, Mark McDonald, Daniel J Fenn, and Sam D Howison. Limit Order Books. *Quantitative Finance*, 13(11):1709–1742, 2013.

[45] Kristen Grauman and Rob Fergus. Learning Binary Hash Codes for Large-Scale Image Search. In *Machine Learning for Computer Vision*, pages 49–87. Springer, 2013.

[46] Shiferaw Gurmu and John Elder. A Bivariate Zero-Inflated Count Data Regression Model with Unrestricted Correlation. *Economics Letters*, 100(2):245–248, 2008.

[47] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press, 2015.

[48] Jerry A Hausman, Andrew W Lo, and A Craig MacKinlay. An Ordered Probit Analysis of Transaction Stock Prices. *Journal of Financial Economics*, 31(3):319–379, 1992.

[49] Joseph M Hilbe. *Logistic Regression Models*. CRC Press, 2011.

[50] Jiayi Hou. *Regularization Method for Predicting an Ordinal Response using Longitudinal High-Dimensional Genomic Data*. PhD thesis, Virginia Commonwealth University, 2015.

[51] Yue Hu, Eric Chi, and Genevera I Allen. ADMM Algorithmic Regularization Paths for Sparse Statistical Machine Learning. arXiv preprint, `http://arxiv.org/abs/1504.06637`, 2015.

[52] Jinzhu Jia and Karl Rohe. Preconditioning the Lasso for Sign Consistency. *Electronic Journal of Statistics*, 9:1150–1172, 2015.

[53] Dimitris Karlis. An EM Algorithm for Multivariate Poisson Distribution and Related Models. *Journal of Applied Statistics*, 30(1):63–77, 2003.

[54] Dimitris Karlis and Ioannis Ntzoufras. Bivariate Poisson and Diagonal Inflated Bivariate Poisson Regression Models in R. Package 'bivpois' and data files. `http://www.stat-athens.aueb.gr/~jbn/papers/paper14.htm`.

[55] Dimitris Karlis and Ioannis Ntzoufras. Bivariate Poisson and Diagonal Inflated Bivariate Poisson Regression Models in R. *Journal of Statistical Software*, 14(10):1–36, 2005.

[56] Robert E. Kass and Paul W. Vos. *Geometrical Foundations of Asymptotic Inference*. Wiley-Interscience, 1997.

[57] Kazutomo Kawamura. The Structure of Bivariate Poisson Distribution. In *Kodai Mathematical Seminar Reports*, volume 25, pages 246–256, 1973.

[58] Benjamin Kedem and Konstantinos Fokianos. *Regression Models for Time Series Analysis*. John Wiley & Sons, 2002.

[59] Alec N Kercheval and Yuan Zhang. Modelling High-Frequency Limit Order Book Dynamics with Support Vector Machines. *Quantitative Finance*, 15(8):1315–1329, 2015.

[60] Nitish Shirish Keskar, Jorge Nocedal, Figen Oztoprak, and Andreas Waechter. A Second-Order Method for Convex l1-Regularized Optimization with Active Set Prediction. arXiv preprint, `http://arxiv.org/abs/1505.04315`, 2015.

[61] Md Hasinur Rahaman Khan and J Ewart H Shaw. Variable Selection for Survival Data with a Class of Adaptive Elastic Net Techniques. *Statistics and Computing*, pages 1–17, 2013.

[62] Yongdai Kim, Sunghoon Kwon, and Hosik Choi. Consistent Model Selection Criteria on High Dimensions. *The Journal of Machine Learning Research*, 13(1):1037–1057, 2012.

[63] Inge Koch. On the Asymptotic Performance of Median Smoothers in Image Analysis and Nonparametric Regression. *The Annals of Statistics*, pages 1648–1666, 1996.

[64] Subrahmaniam Kocherlakota and Kathleen Kocherlakota. *Bivariate Discrete Distributions*. Wiley, 1992.

[65] Kwangmoo Koh, Seung-Jean Kim, and Stephen P Boyd. An Interior-Point Method for Large-Scale l1-Regularized Logistic Regression. *Journal of Machine learning research*, 8(8):1519–1555, 2007.

[66] Eric B Laber and Hua Zhou. ST810: Advanced Computing, Lecture 9: Quadratic Programming, Department of Statistics, North Carolina State University. `http://www.stat.ncsu.edu/people/zhou/courses/st810/notes/lect09QP.pdf`, 2013.

[67] Ting Pong Lam. Building Directional Signals with Machine Learning. Msc Thesis (unpublished), University of Oxford, 2015.

[68] Jason D Lee, Yuekai Sun, and Michael A Saunders. Proximal Newton-Type Methods for Convex Optimization. In *Advances in Neural Information Processing Systems*, pages 836–844, 2012.

[69] Jason D Lee, Yuekai Sun, and Michael A Saunders. Proximal Newton-Type Methods for Minimizing Composite Functions. *SIAM Journal on Optimization*, 24(3):1420–1443, 2014.

[70] Su-In Lee, Honglak Lee, Pieter Abbeel, and Andrew Y Ng. Efficient l1 Regularized Logistic Regression. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 401, 2006.

[71] Roman Liesenfeld, Ingmar Nolte, and Winfried Pohlmeier. Modelling Financial Transaction Price Movements: a Dynamic Integer Count Data Model. *Empirical Economics*, 30(4):795–825, 2006.

[72] Yang Liu. *Improving the Accuracy of Variable Selection Using the Whole Solution Path*. PhD thesis, Bowling Green State University, 2015.

[73] Wenbin Lu and Hao H Zhang. Variable Selection for Proportional Odds Model. *Statistics in Medicine*, 26(20):3771–3781, 2007.

[74] Giampiero Marra and Simon N Wood. Practical Variable Selection for Generalized Additive Models. *Computational Statistics & Data Analysis*, 55(7):2372–2387, 2011.

[75] Ingrid Mauerer, Wolfgang Pößnecker, Paul W Thurner, and Gerhard Tutz. Modeling Electoral Choices in Multiparty Systems with High-Dimensional Data: A Regularized Selection of Parameters using the Lasso Approach. *Journal of choice modelling*, 16:23–42, 2015.

[76] Olaf Mersmann, Claudia Beleites, Rainer Hurling, and Ari Friedman. microbenchmark: Sub Microsecond Accurate Timing Functions. `http://cran.r-project.org/web/packages/microbenchmark/index.html`, 2013.

[77] Jean-Jacques Moreau. Fonctions Convexes Duales et Points Proximaux dans un Espace Hilbertien. *Comptes-Rendus de l'Académie des Sciences de Paris Série A, Mathématiques*, 255:2897–2899, 1962.

[78] Jean-Jacques Moreau. Proximité et Dualité dans un Espace Hilbertien. *Bulletin de la Société Mathématique de France*, 93:273–299, 1965.

[79] John A. Nelder and Robert W.M. Wedderburn. Generalized Linear Models. *Journal of the Royal Statistical Society: Series A*, 135(3):370–384, 1972.

[80] Margret-Ruth Oelker. *Penalized Regression for Discrete Structures*. PhD thesis, Ludwig-Maximilians-Universität München, 2015.

[81] Margret-Ruth Oelker and Gerhard Tutz. A Uniform Framework for the Combination of Penalties in Generalized Structured Models. *Advances in Data Analysis and Classification*, pages 1–24, 2015.

[82] James M Ortega and Werner C Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*, volume 30. Siam, 1970.

[83] Michael R Osborne. Fisher's Method of Scoring. *International Statistical Review/Revue Internationale de Statistique*, pages 99–117, 1992.

[84] Michael R Osborne. Least Squares Methods in Maximum Likelihood Problems. *Optimization Methods and Software*, 21(6):943–959, 2006.

[85] Michael R Osborne, Brett Presnell, and Berwin A Turlach. On the LASSO and its Dual. *Journal of Computational and Graphical Statistics*, 9(2):319–337, 2000.

[86] Deepan Palguna and Ilya Pollak. Non-Parametric Prediction of the Mid-Price Dynamics in a Limit Order Book. In *Information Theory Workshop (ITW), 2013 IEEE*, pages 1–5. IEEE, 2013.

[87] Efstathios Panayi and Gareth W. Peters. Survival Models for the Duration of Bid-Ask Spread Deviations. In *Computational Intelligence for Financial Engineering & Economics (CIFEr), 2104 IEEE Conference on*, pages 9–16. IEEE, 2014.

[88] Efstathios Panayi, Gareth W. Peters, Jon Danielsson, and Jean-Pierre Zigrand. Designating Market Maker Behaviour in Limit Order Book Markets. available at SSRN: `http://ssrn.com/abstract=2646649`, 2015.

[89] Neal Parikh and Stephen Boyd. Proximal Algorithms. *Foundations and Trends in optimization*, 1(3):123–231, 2013.

[90] Mee Young Park. *Generalized Linear Models with Regularization*. PhD thesis, Stanford University, 2006.

[91] Mee Young Park and Trevor Hastie. L1-Regularization Path Algorithm for Generalized Linear Models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):659–677, 2007.

[92] Panagiotis Patrinos, Lorenzo Stella, and Alberto Bemporad. Forward-Backward Truncated Newton Methods for Convex Composite Optimization. arXiv preprint, `http://arxiv.org/abs/1402.6655`, 2014.

[93] Nicholas G Polson, James G Scott, Brandon T Willard, et al. Proximal Algorithms in Statistics and Machine Learning. *Statistical Science*, 30(4):559–581, 2015.

[94] Zhiwei Qin and Donald Goldfarb. Structured Sparsity via Alternating Direction Methods. *The Journal of Machine Learning Research*, 13(1):1435–1468, 2012.

[95] Zhiwei Qin, Katya Scheinberg, and Donald Goldfarb. Efficient Block-Coordinate Descent Algorithms for the Group Lasso. *Mathematical Programming Computation*, 5(2):143–169, 2013.

[96] Zhiwei Qin, Xiaocheng Tang, Ioannis Akrotirianakis, and Amit Chakraborty. HIPAD-A Hybrid Interior-Point Alternating Direction Algorithm for Knowledge-Based SVM and Feature Selection. In *Learning and Intelligent Optimization*, pages 324–340. Springer, 2014.

[97] Jeffrey Russell and Robert F Engle. Econometric Analysis of Discrete-Valued Irregularly-Spaced Financial Transactions Data Using a New Autoregressive Conditional Multinomial Model. *University of California San Diego Discussion Paper 98-10*, 1998.

[98] Tina Hviid Rydberg and Neil Shephard. Dynamics of Trade-by-Trade Price Movements: Decomposition and Models. *Journal of Financial Econometrics*, 1(1):2–25, 2003.

[99] Katya Scheinberg and Xiaocheng Tang. Practical Inexact Proximal Quasi-Newton Method with Global Complexity Analysis. arXiv preprint, `http://arxiv.org/abs/1311.6547`, 2013.

[100] Mark Schmidt. Least Squares Optimization with l1-Norm Regularization, CS542B Project Report. `https://www.cs.ubc.ca/~schmidtm/Documents/2005_Notes_Lasso.pdf`, 2005.

[101] Golnaz Shahtahmassebi. *Bayesian Modelling of Ultra High-Frequency Financial Data*. PhD thesis, School of Computing and Mathematics, Faculty of Science and Technology, University of Plymouth, 2011.

[102] Ziqiang Shi. Proximal Stochastic Newton-type Gradient Descent Methods for Minimizing Regularized Finite Sums. arXiv preprint, `http://arxiv.org/abs/1409.2979`, 2014.

[103] Noah Simon, Jerome Friedman, and Trevor Hastie. A Blockwise Descent Algorithm for Group-Penalized Multiresponse and Multinomial Regression. arXiv preprint, `http://arxiv.org/abs/1311.6529`, 2013.

[104] Noah Simon, Jerome Friedman, Trevor Hastie, Rob Tibshirani, et al. Regularization Paths for Coxs Proportional Hazards Model via Coordinate Descent. *Journal of Statistical Software*, 39(5):1–13, 2011.

[105] Justin A. Sirignano. Neural Networks for Limit Order Books. `http://arxiv.org/pdf/1601.01987.pdf`, 2015.

[106] Stefan Solntsev, Jorge Nocedal, and Richard H Byrd. An Algorithm for Quadratic l1-Regularized Optimization with a Flexible Active-Set Strategy. *Optimization Methods and Software*, pages 1–25, 2014.

[107] Xiaogang Su, Juanjuan Fan, Richard A Levine, Xianming Tan, and Arvind Tripathi. Multiple-Inflation Poisson Model with l1 Regularization. *Statistica Sinica*, 23:1071–1090, 2013.

[108] Yanlin Tang, Liya Xiang, and Zhongyi Zhu. Risk Factor Selection in Rate Making: EM Adaptive LASSO for Zero-Inflated Poisson Regression Models. *Risk Analysis*, 34(6):1112–1127, 2014.

[109] Robert Tibshirani. Regression Shrinkage and Selection via the LASSO. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[110] Jun Tong, Qinghua Guo, Sheng Tong, Jiangtao Xi, and Yanguang Yu. Condition Number-Constrained Matrix Approximation with Applications to Signal Estimation in Communication Systems. *Signal Processing Letters, IEEE*, 21(8):990–993, 2014.

[111] Paul Tseng. Convergence of a Block Coordinate Descent Method for Nondifferentiable Minimization. *Journal of Optimization Theory and Applications*, 109(3):475–494, 2001.

[112] Berwin A Turlach and Andreas Weingessel. Quadprog: Functions to Solve Quadratic Programming Problems. `https://cran.r-project.org/web/packages/quadprog/index.html`, 2007.

[113] Gerhard Tutz. *Regression for Categorical Data*. Cambridge University Press, 2011.

[114] Gerhard Tutz, Wolfgang Pößnecker, and Lorenz Uhlmann. Variable Selection in General Multinomial Logit Models. *Computational Statistics & Data Analysis*, 82:207–222, 2015.

[115] Jan Ulbricht. *Variable Selection in Generalized Linear Models*. Phd Thesis, Ludwig Maximilian University, 2010.

[116] Anita J. Van der Kooij. *Prediction Accuracy and Stability of Regression with Optimal Scaling Transformations*. PhD thesis, University of Leiden, 2007.

[117] Hansheng Wang and Chenlei Leng. Unified LASSO Estimation by Least Squares Approximation. *Journal of the American Statistical Association*, 102(479), 2007.

[118] Hansheng Wang, Runze Li, and Chih-Ling Tsai. Tuning Parameter Selectors for the Smoothly Clipped Absolute Deviation Method. *Biometrika*, 94(3):553–568, 2007.

[119] Hao Wang. Coordinate Descent Algorithm for Covariance Graphical Lasso. *Statistics and Computing*, 24(4):521–529, 2014.

[120] Yong Wang. Maximum Likelihood Computation Based on the Fisher Scoring and Gauss-Newton Quadratic Approximations. *Computational Statistics & Data Analysis*, 51(8):3776–3787, 2007.

[121] Yong Wang. The Constrained Fisher Scoring Method for Maximum Likelihood Computation of a Nonparametric Mixing Distribution. *Computational Statistics*, 24(1):67–81, 2009.

[122] Zhu Wang, Shuangge Ma, and Ching-Yun Wang. Variable Selection for Zero-Inflated and Overdispersed Data with Application to Health Care Demand in Germany. *Biometrical Journal*, 57(5):867–884, 2015.

[123] Zhu Wang, Shuangge Ma, Ching-Yun Wang, Michael Zappitelli, Prasad Devarajan, and Chirag Parikh. EM for Regularized Zero-Inflated Regression Models with Applications to Postoperative Morbidity after Cardiac Surgery in Children. *Statistics in Medicine*, 33(29):5192–5208, 2014.

[124] Zhu Wang, Shuangge Ma, Michael Zappitelli, Chirag Parikh, Ching-Yun Wang, and Prasad Devarajan. Penalized Count Data Regression with Application to Hospital Stay after Pediatric Cardiac Surgery. *Statistical Methods in Medical Research*, 2014.

[125] Fabian L Wauthier, Nebojsa Jojic, and Michael I Jordan. A Comparative Framework for Preconditioned Lasso Algorithms. In *Advances in Neural Information Processing Systems*, pages 1061–1069, 2013.

[126] Robert W.M. Wedderburn. On the Existence and Uniqueness of the Maximum Likelihood Estimates for Certain Generalized Linear Models. *Biometrika*, 63(1):27–32, 1976.

[127] Stephen J Wright and Jorge Nocedal. *Numerical Optimization, Second Edition*. Springer New York, 2006.

[128] Tong Tong Wu and Kenneth Lange. Coordinate Descent Algorithms for Lasso Penalized Regression. *The Annals of Applied Statistics*, pages 224–244, 2008.

[129] Yi Yang and Hui Zou. A Cocktail Algorithm for Solving the Elastic Net Penalized Cox's Regression in High Dimensions. *Statistics and its Interface*, 6(2):167–173, 2012.

[130] Thomas W. Yee. *Vector Generalized Linear and Additive Models with an Implementation in R*. Springer, New York, NY, USA, 2015.

[131] Thomas W. Yee. The VGAM Package for R. `https://www.stat.auckland.ac.nz/~yee/VGAM/`, 2016.

[132] Thomas W. Yee and Chris J. Wild. Vector Generalized Additive Models. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 481–493, 1996.

[133] Yi Yu and Yang Feng. APPLE: Approximate Path for Penalized Likelihood Estimators. *Statistics and Computing*, 24(5):803–819, 2014.

[134] Guo-Xun Yuan, Kai-Wei Chang, Cho-Jui Hsieh, and Chih-Jen Lin. A Comparison of Optimization Methods and Software for Large-Scale l1-Regularized Linear Classification. *The Journal of Machine Learning Research*, 11:3183–3234, 2010.

[135] Guo-Xun Yuan, Chia-Hua Ho, and Chih-Jen Lin. An Improved GLMNET for L1-Regularized Logistic Regression. *The Journal of Machine Learning Research*, 13(1):1999–2030, 2012.

[136] Ming Yuan and Yi Lin. On the Non-Negative Garrotte Estimator. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):143–161, 2007.

[137] Ming Yuan and Hui Zou. Efficient Global Approximation of Generalized Nonlinear l1-Regularized Solution Paths and its Applications. *Journal of the American Statistical Association*, 2012.

[138] Ping Zeng, Yongyue Wei, Yang Zhao, Jin Liu, Liya Liu, Ruyang Zhang, Jianwei Gou, Shuiping Huang, and Feng Chen. Variable Selection Approach for Zero-Inflated Count Data via Adaptive Lasso. *Journal of Applied Statistics*, 41(4):879–894, 2014.

[139] Qingzhao Zhang, Sanguo Zhang, Jin Liu, Jian Huang, and Shuangge Ma. Penalized Integrative Analysis under the Accelerated Failure Time Model. *Statistica Sinica preprint SS-14-194R2*, 2015.

[140] Yiyun Zhang, Runze Li, and Chih-Ling Tsai. Regularization Parameter Selections via Generalized Information Criterion. *Journal of the American Statistical Association*, 105(489):312–323, 2010.

[141] Ban Zheng, Eric Moulines, and Frédéric Abergel. Price Jump Prediction in Limit Order Book. available at SSRN: `http://ssrn.com/abstract=2026454`, 2012.

[142] Hui Zou and Trevor Hastie. Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

[143] Hui Zou, Trevor Hastie, and Robert Tibshirani. On the Degrees of Freedom of the Lasso. *The Annals of Statistics*, 35(5):2173–2192, 2007.