High-dimensional variable selection for GLMs and survival models

Pazira, Hassan

High-dimensional Variable Selection for GLMs and Survival Models

Hassan Pazira

“Genius is 1% talent and 99% hard work.”
– Albert Einstein

High-dimensional variable selection for GLMs and survival models, PhD dissertation of Hassan Pazira.

This dissertation was financially supported by the MRI center Badr.

Printed by: Ridderprint, the Netherlands

ISBN: 978-90-367-9953-9 (Printed Book)
978-90-367-9952-2 (Ebook: PDF without DRM)

university of
groningen

High-dimensional Variable Selection for GLMs and Survival Models

PhD thesis

to obtain the degree of PhD at the
University of Groningen
on the authority of the
Rector Magnificus Prof. E. Sterken,
and in accordance with
the decision by the college of Deans.

This thesis will be defended in public on

Monday 10 July 2017 at 09:00 hours

by

Hassan Pazira

born on 16 July 1981
in Ghaemshahr, Iran

Supervisor
Prof. E. C. Wit

Assessment committee
Prof. E. R. van den Heuvel
Prof. Christine zu Eulenburg
Prof. Angelo Mineo

To my loving wife, *Saemeh*, and my beloved son, *Nikan*.

Contents

Chapter 1

Introduction and Basic Definitions

Contents

The work in this thesis is motivated by high-dimensional applications such as modelling phenotypes using gene expression data. Modeling gene expression data imposes a few challenges onto the traditional statistical methods. The most prominent is that the number of covariates (predictors) is generally much larger than the sample size (observations). High-dimensional gene expression data are increasingly used for modeling various clinical outcomes to facilitate disease diagnosis, disease prognosis, and prediction of treatment outcome.

Variable selection is an essential component of modern statistical data analysis. Starting with a large number of variables, possibly larger than the number of observations, the aim is to determine a smaller subset that includes the most important effects (Sparsity). Sparse inference in the past two decades has been dominated by methods that typically penalize convex likelihoods by functions of the parameters that happen to induce solutions with many zeros. The Least Absolute Shrinkage and Selection Operator (LASSO) [94] and other penalization approaches are all examples of methods that depending on some tuning parameter conveniently shrink estimates to exact zeros. Although the LASSO penalty induces sparsity, it is well known to suffer from possible inconsistent selection of variables.

In this thesis, we will approach sparsity directly from a likelihood point of view. The angle between the covariates and the tangent residual vector within the likelihood manifold provides a direct and scale-invariant way to assess the importance of the individual covariates. The idea is similar to the least angle regression (LARS) approach proposed by [29]. In the LARS method a multivariate solution path is defined by using the geometrical theory of the linear regression model. [13] proposed a method, called dgLARS, to introduce sparse inference for a generalized linear model (GLM) [60] based on the exponential families with canonical link. The basic idea underlying the dgLARS method is to use the differential geometrical structure of a GLM to generalize the LARS method.

In Chapter 2, we extend the dgLARS method to the high-dimensional GLMs based on the exponential dispersion (ED) models with arbitrary link functions. Moreover, we present an improved predictor-corrector (PC) algorithm to decrease the run times for computing the solution curve and implement it in R. A classical estimation of the unknown dispersion parameter ϕ based on high-dimensional feature space is proposed to make us able to do model selection. The AIC, BIC, and cross validation (CV) are adapted separately to select an optimal model and its corresponding optimal tuning parameter γ. The implementation works not only in the traditional setting of $p < n$, but also in the high-dimensional setting when $p > n$. The procedure proposed in this chapter is applied to the low- and high-dimensional datasets to illustrate the capacity of the extended dgLARS method.

It is known that in shrinkage situations the estimator of the dispersion parameter underestimates ϕ. For this, in Chapter 3, we focus on estimating the dispersion parameter in high-dimensional exponential dispersion GLMs and propose a new method which is more accurate than the classical estimator proposed in the previous chapter, and then we present an algorithm to improve the proposed estimator to obtain a more stable estimator. A numerical study is conducted to compare the proposed estimator with the classical one. The extended dgLARS method by means of the new dispersion estimate is applied to analyze both low- and high-dimensional diabetes datasets. The results of Chapter 2 and Chapter 3 can be found in [70].

Cancer survival is thought to be closely linked to the genomic constitution of the tumour. Discovering such signatures will be useful in the diagnosis of the patient and may be used for treatment decisions and perhaps even the development of new treatments. These studies rely on survival modelling to detect relevant factors that affect various event histories. However, genomic data are typically noisy and high-dimensional, often outstripping the number of patients included in the study. Regularized survival models have been proposed to deal with such scenarios. In Chapter 4, we suggest an alternative to the penalized inference methods, indeed we propose a principled method for sparse inference in relative risk survival models, based on differential geometrical analyses of the high-dimensional likelihood surface. The method is computationally fast and is implemented in the R-package **`dglars`**. The results of Chapter 4 can be found in [100].

Chapter 5 is devoted to introducing an implementation of the improved estimator of the dispersion parameter for high-dimensional generalized linear models, called General Refitted Cross-Validation (GRCV) estimator, with an implementation of the iterative algorithm for improving the proposed GRCV estimator to obtain a more stable and accurate estimator. A numerical study is conducted to compare the proposed estimator with the deviance, maximum likelihood and generalized Pearson estimators, proposed in [70]. The extended dgLARS method by means of the new dispersion estimator is applied to analyze both low- and high-dimensional diabetes datasets. Several dispersion parameter estimation methods and algorithms for computing the dgLARS solution curve, proposed in [13] and [70], are implemented in the new version of the R-package **dglars** [14]. The results of Chapter 5 can be found in [69].

We begin the thesis with some basic definitions and concepts. For this, the rest of the chapter is organized as follows; Section 1.1 is devoted to LARS and some other model selection methods. The basic concepts of differential Geometry will be discussed in Section 1.2. In Section 1.3 we explain GLMs based on the ED family and give a geometrical description of the GLMs. Section 1.4 is devoted to the survival models, and in the last section, Section 1.5, the structure of the thesis is presented.

1.1 Least Angle Regression and Previous Methods

In a variety of fields such as genomics, proteomics, drug discovery, fraud detection, and so on, the number of predictors (e.g., genes or proteins) is very large and may exceed the number of observations. Owing to the massive collection of predictor variables available in these datasets, model and variable selection have become important research topics in regression and classification. Model selection can produce interpretable models (i.e., parsimonious models that include only a subset of predictors) and provide accurate predictions.

In the past few decades, several approaches have been proposed to perform model and variable selection. Earlier developments include stepwise regression and all-subset selection. More recently, other methods such as the Least Absolute Shrinkage and Selection Operator (LASSO) [94] and stagewise regression [44] have been proposed.

[29] show that there are strong connections between these modern methods and a method they call least angle regression, and develop an algorithmic framework that includes all of these methods and provides a fast implementation, for which they use the term 'LARS'. LARS provides accurate variable selection and prediction. Moreover, it has also been shown that with some

slight modifications, LARS can efficiently generate the solutions for stagewise or LASSO problems, which further boosts LARS's popularity.

In the following, we explain LARS in Section 1.1.1 and compare it to modern procedures such as LASSO and forward stagewise regression methods in Section 1.1.2, but we will first very briefly review some model selection methods that are related to LARS:

***Stepwise and All-Subsets Regression*:** These methods (which are *pure variable selection* methods) focus on selecting variables for a model, rather than on how coefficients are estimated once variables are selected. In other words, they pick predictors and then estimate coefficients for those variables using standard criteria such as least-squares or maximum likelihood.

***Ridge Regression*:** This method is not concerned with variable selection (it uses all candidate predictors), and instead modifies how coefficients are estimated [45].

***LASSO*:** A variation of ridge regression that modifies coefficient estimation so as to reduce some coefficients to zero, effectively performing variable selection.

***Forward Stagewise Regression*:** An incremental version of stepwise regression that gives results very similar to LASSO.

***LARS*:** A method that connects all the methods.

1.1.1 Least Angle Regression (LARS)

LARS can be viewed as a version of stagewise that uses mathematical formulas to accelerate the computations. Rather than taking many tiny steps with the first variable, the appropriate number of steps is determined algebraically, until the second variable begins to enter the model. Then, rather than taking alternating steps between those two variables until a third variable enters the model, the method jumps right to the appropriate spot. Figure 1.1 shows this process in the case of 2 predictor variables, for linear regression. In this figure, O is the prediction based solely on an intercept. $\hat{Y} = \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2$ is the ordinary least-squares fit, the projection of the response vector Y onto the subspace spanned by X_1 and X_2. A is the forward stepwise fit after one step; the second step proceeds to $\hat{Y}$. Stagewise takes a number of tiny steps from O to B, then takes steps alternating between the X_1 and X_2 directions, eventually reaching D; if allowed to continue it would reach $\hat{Y}$. LARS jumps from O to B in one step, where B is the point such that $B\hat{Y}$ bisects the angle ABC. At the second step it jumps to $\hat{Y}$. LASSO follows a path from O to B, then from B to $\hat{Y}$. Here LARS agrees with LASSO and stagewise (as the step size → 0 for stagewise). In higher dimensions additional conditions are needed for exact agreement to hold.

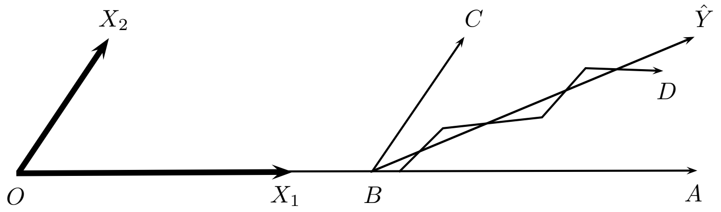


Figure 1.1: The LARS algorithm in the case of 2 predictors.

The first variable chosen is the one that has the smallest angle between the variable and the response variable; in Figure 1.1 the angle $\hat{Y}OX_1$ is smaller than $\hat{Y}OX_2$. We proceed in that direction as long as the angle between that predictor and the vector of residuals $Y - \xi X_1$ is smaller than the angle between other predictors and the residuals. Eventually the angle for another variable will equal this angle (once we reach point B in Figure 1.1), at which point we begin moving toward the direction of the least-squares fit based on both variables. In higher dimensions we will reach the point at which a third variable has an equal angle and will join the model, etc.

It has been shown that there is a correspondence between the geometric concept of angle and the statistical concept of correlation in a linear model. Expressed another way, the (absolute value of the) correlation between the residuals and the first predictor is greater than the (absolute) correlation for other predictors. As ξ increases, another variable will eventually have a correlation with the residuals equaling that of the active variable, and join the model as a second active variable. In higher dimensions additional variables will eventually join the model, when the correlation between all active variables and the residuals gradually drops to the levels of the additional variables.

There are three remarkable things about LARS. First is the speed: [29] note that "The entire sequence of LARS steps with $p < n$ variables requires $O(p^3 + np^2)$ computations - the cost of a least squares fit on p variables." Second is that the basic LARS algorithm, based on the geometry of angle bisection, can be used to efficiently fit LASSO and stagewise models, with certain modifications in higher dimensions [29]. This provides a fast and relatively simple way to fit LASSO and stagewise models. Third is the availability of a simple C_p statistic as a stopping criterion of the algorithm,

$$C_p = \hat{\sigma}^{-2} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 - n + 2k$$

where k is the number of steps and σ^2 is the estimated residual variance (estimated from the saturated model, assuming that $n > p$, or in some other way if $p \geq n$ in order to deal with overfitting). This is based on Theorem 3 in [29], which indicates that after k steps of LARS the degrees of freedom $\sum_{i=1}^{n} cov(\hat{\mu}_i, Y_i)/\sigma^2$ is approximately k. This provides a simple stopping rule, to stop after the number of steps k that minimizes the C_p statistic. Note that there are different definitions of degrees of freedom, and the one used here is appropriate for the C_p statistic, but this k does not measure other kinds of degrees of freedom.

1.1.2 Comparing LARS, LASSO and Stagewise

In general in higher dimensions native LARS and the least angle implementation of LASSO and stagewise give results that are similar but not identical. When they differ, LARS has a speed advantage, because LARS variables are added to the model, never removed. Hence it will reach the full least-squares solution, using all variables, in p steps. For LASSO, and to a greater extent for stagewise, variables can leave the model, and possibly re-enter later, multiple times. Hence they may take more than p steps to reach the full model (if $n > p$). [29] test the three procedures for the diabetes data using a quadratic model, consisting of the 10 main effects, 45 two-way interactions, and 9 squares (excluding the binary variable "*sex*"). LARS takes 64 steps to reach the full model, LASSO takes 103, and stagewise takes 255. Even in other situations, when stopping short of the saturated model, LARS has a speed advantage.

The three methods have interesting derivations. LASSO is regression with an l_1 penalty, a relatively simple concept; this is also known as a form of regularization in the machine learning community. Stagewise is closely related to boosting, or 'slow learning' in machine learning [29, 43]. LARS has a simpler interpretation than the original derivation; it can be viewed as a variation of Newton's method, which makes it easier to extend to some nonlinear models such as generalized linear models [82].

1.2 Elementary Differential Geometry

For information geometry the most important aspects of differential geometry are those which allow us to take problems from a variety of fields: statistics, information theory, and control theory; visualize them geometrically; and from this develop novel tools with which to extend and advance these fields. In this section we present an introduction to differential geometry from this point of view, and at the end of the section we present a geometric structure of statistical models.

1.2.1 Differentiable Manifolds

A *differentiable manifold* is a mathematical concept denoting a generalization/abstraction of geometric objects such as smooth curves and surfaces in an n-dimensional space. Intuitively, a manifold $\mathcal{S}$ is a "set with a coordinate system." Since $\mathcal{S}$ is a set, it has elements. It does not matter what these elements are (these elements are also called the *points* of $\mathcal{S}$.) $\mathcal{S}$ must also have a *coordinate system*. By this we mean a one-to-one mapping from S (or its subset) to $\mathcal{R}^n$, which allows us to specify each point in $\mathcal{S}$ using a vector of n real numbers (this vector is called the *coordinates* of the corresponding point). We call the natural number n the *dimension* of $\mathcal{S}$, and write $n = \dim \mathcal{S}$. We call a coordinate system that has $\mathcal{S}$ as its domain a global coordinate system.

Let $\mathcal{S}$ be a manifold and $\varphi : \mathcal{S} \to \mathcal{R}^n$ be a coordinate system for $\mathcal{S}$. Then φ maps each point p in $\mathcal{S}$ to n real numbers: $\varphi(p) = [\xi^1(p), \ldots, \xi^n(p)] = [\xi^1, \ldots, \xi^n]$. These are the coordinates of the point p. Each ξ^i may be viewed as a function $p \to \xi^i(p)$ which maps a point p to its i^{th} coordinate; we call these n functions $\xi^i : \mathcal{S} \to \mathcal{R} (i = 1, \ldots, n)$ the *coordinate functions*. We shall write the coordinate system φ in ways such as $\varphi = [\xi^1, \ldots, \xi^n] = [\xi^i]$.

Let $\psi = [\rho^i]$ be another coordinate system for $\mathcal{S}$. Then the same point $p \in \mathcal{S}$ has both the coordinates $[\xi^i(p)] = [\xi^i] \in \mathcal{R}^n$ with respect to the coordinate system φ, and the coordinates $[\rho^i(p)] = [\rho^i] \in \mathcal{R}^n$ with respect to the coordinate system ψ. The coordinates $[\rho^i]$ may be obtained from $[\xi^i]$ in the following way. First apply the inverse mapping φ^{-1} to $[\xi^i]$; this gives us a point p in $\mathcal{S}$. Then apply ψ to this point; this result is $[\rho^i]$. In other words, we apply the transformation on $\mathcal{R}^n$ given by

$$\psi \circ \varphi^{-1} : [\xi^1, \ldots, \xi^n] \mapsto [\rho^1, \ldots, \rho^n] \tag{1.1}$$

This is called the *coordinate transformation* from $\varphi = [\xi^i]$ to $\psi = [\rho^i]$.

Let $\mathcal{S}$ be a set. If there exists a set of coordinate systems $\mathcal{A}$ for $\mathcal{S}$ which satisfies the conditions (i) and (ii) below, we call $\mathcal{S}$ (more properly, $(\mathcal{S}, \mathcal{A})$) an n-dimensional **C**$^\infty$ *differentiable manifold*, or more simply, a *manifold*.

(i) Each element φ of $\mathcal{A}$ is a one-to-one mapping from $\mathcal{S}$ to some open subset of $\mathcal{R}^n$.

(ii) For all $\varphi \in \mathcal{A}$, given any one-to-one mapping ψ from $\mathcal{S}$ to $\mathcal{R}^n$, the following holds:

$$\psi \in \mathcal{A} \Longleftrightarrow \psi \circ \varphi^{-1} \text{ is a } \mathrm{C}^\infty \text{diffeomorphism.}$$

Here, by a **C**$^\infty$ *diffeomorphism* we mean that $\psi \circ \varphi^{-1}$ and its inverse $\varphi \circ \psi^{-1}$ are both C$^\infty$ (infinitely many times differentiable). From these conditions, and given the coordinate transformation described in Equation (1.1), it follows that we may take the partial derivative of the function $\rho^i = \rho^i(\xi^1, \ldots, \xi^n)$ with respect to its variable arguments as many times as needed, and that the same holds for $\xi^i = \xi^i(\rho^1, \ldots, \rho^n)$.

Let $\mathcal{S}$ be a manifold and φ be a coordinate system for $\mathcal{S}$. Let U be a subset of $\mathcal{S}$. If the image $\varphi(U)$ is an open subset of $\mathcal{R}^n$, then we say that U is an open subset of $\mathcal{S}$. From condition (ii) above, we see that this property is invariant over the choice of coordinate system φ. This allows us to consider $\mathcal{S}$ as a topological space.

1.2.2 Tangent Vectors and Tangent Spaces

The *tangent space* T_p at a point $p \in \mathcal{S}$ of a manifold $\mathcal{S}$ is intuitively the vector space obtained by locally linearizing $\mathcal{S}$ around p. Let $[\xi^i]$ be some coordinate system for $\mathcal{S}$, and let $\mathbf{e}_i$ denote the *tangent vector* which goes through point p and is parallel to the i^{th} coordinate curve (coordinate axis). By the i^{th} coordinate curve we mean the curve which is obtained by fixing the values of all ξ^j for $j \neq i$ and varying only the value of ξ^j. The n-dimensional space spanned by the n tangent vectors $\mathbf{e}_1, \ldots, \mathbf{e}_n$ is the tangent space T_p at point p. Let p' be a point "very close" to p, and let $[\xi^i]$ and $[\xi^i + d\xi^i]$ (where $d\xi^i$ is an infinitesimal) be the coordinates of p and p', respectively. Then the segment joining these two points may be described by $\overrightarrow{p\,p'} = d\xi^i\,\mathbf{e}_i$, an infinitesimal vector in T_p.

1.2.3 Submanifolds and Riemannian Metrics

Let $\mathcal{S}$ and $\mathcal{M}$ be manifolds, where $\mathcal{M}$ is a subset of $\mathcal{S}$. Let $[\xi^1, \ldots, \xi^n] = [\xi^i]$ and $[u^1, \ldots, u^m] = [u^a]$ be coordinate systems for $\mathcal{S}$ and $\mathcal{M}$, respectively, where $n = \dim \mathcal{S}$ and $m = \dim \mathcal{M}$. Below, we shall use the indices $i, j, k, \ldots$ over $\{1, \ldots, n\}$ for $\mathcal{S}$ and $a, b, c, \ldots$ over $\{1, \ldots, m\}$ for $\mathcal{M}$.

We call $\mathcal{M}$ a *submanifold* of $\mathcal{S}$ if the following conditions (i), (ii), and (iii) hold.

(i) The restriction $\xi^i|_{\mathcal{M}}$ of each ξ^i (: $\mathcal{S} \to \mathcal{R}$) to $\mathcal{M}$, is a C$^\infty$ function on $\mathcal{M}$.

(ii) Let $\mathcal{B}_a^i \overset{\text{def}}{=} \left(\frac{\partial \xi^i}{\partial u^a}\right)_p$ (more precisely, $\left(\frac{\partial \xi^i|_{\mathcal{M}}}{\partial u^a}\right)_p$) and $\mathcal{B}_a \overset{\text{def}}{=} [\mathcal{B}_a^1, \ldots, \mathcal{B}_a^n] \in \mathcal{R}^n$. Then for each point p in $\mathcal{M}$, $\{\mathcal{B}_1, \ldots, \mathcal{B}_m\}$ are linearly independent (hence $m \leq n$).

(iii) For any open subset $\mathcal{W}$ of $\mathcal{M}$, there exists $\mathcal{U}$, an open subset of $\mathcal{S}$, such that $\mathcal{W} = \mathcal{M} \cap \mathcal{U}$.

These conditions are independent of the choice of coordinate systems $[\xi^i]$ and $[u^a]$. Indeed, conditions (i) and (ii) mean that the embedding $\iota : \mathcal{M} \to \mathcal{S}$ denned by $\iota(p) = p$, $\forall p \in \mathcal{M}$, is a C$^\infty$ mapping and that its differential $(d\iota)_p$ is nondegenerate at each point p.

Let $\mathcal{S}$ be a manifold. For each point p in $\mathcal{S}$, let us assume that an inner product $\langle\ ,\ \rangle_p$ has been denned on the tangent space $T_P(\mathcal{S})$. In other words, for any tangent vectors D, $D' \in T_P(\mathcal{S})$ we have $\langle D, D' \rangle_p \in \mathcal{R}$, and the following hold.

- **Linearity**

$$\langle aD + bD', D'' \rangle_p = a\langle D, D'' \rangle_p + b\langle D', D'' \rangle_p\,, \quad (\forall a, b \in \mathcal{R}) \tag{1.2}$$

- **Symmetry**

$$\langle D, D' \rangle_p = \langle D', D \rangle_p, \tag{1.3}$$

- **Positive-definiteness**

$$\text{If } D \neq 0 \text{ then } \langle D, D \rangle_p > 0 \tag{1.4}$$

Note that $\langle\ ,\ \rangle_p \in [T_p(\mathcal{S})]^0_2$ since from Equations (1.2) and (1.3) we see that $\langle\ ,\ \rangle_p$ is a bilinear form. Hence the mapping from points p in $\mathcal{S}$ to their inner product on $T_P(\mathcal{S})$, say $g : p \mapsto \langle\ ,\ \rangle_p$, is a tensor field of covariant degree 2. We call this a (C$^\infty$) *Riemannian metric* on $\mathcal{S}$. Such a metric, g, is not naturally determined by the structure of $\mathcal{S}$ as a manifold; it is possible to consider an infinite number of Riemannian metrics on $\mathcal{S}$. Given a Riemannian metric g on $\mathcal{S}$, we call $\mathcal{S}$ (more precisely $(\mathcal{S}, g)$) a *Riemannian manifold*.

Also, the length $\|D\|$ of the tangent vector D is given by

$$\|D\|^2 = \langle D, D \rangle_p = g_{ij}(p) D^i D^j.$$

Another important property that we will make use of, is the following: two vectors are *orthogonal* if $\langle D, D' \rangle = 0$. The Schwarz inequality

$$\langle D, D' \rangle \leq \|D\| \|D'\|$$

allows the angle $0 \leq \vartheta \leq \pi$ between vectors to be defined by

$$\cos \vartheta = \frac{\langle D, D' \rangle}{\|D\| \|D'\|}.$$

1.2.4 The Geometric Structure of Statistical Models

Consider a family $\mathcal{S}$ of probability distributions on $\mathcal{X}$. Suppose each element of $\mathcal{S}$, a probability distribution, may be parameterized using n real-valued variables $[\xi^1, \ldots, \xi^n]$ so that

$$\mathcal{S} = \left\{ p_\xi = p(x; \xi) \mid \xi = [\xi^1, \ldots, \xi^n] \in \Xi \right\}, \tag{1.5}$$

where p is a probability density function on $\mathcal{X}$, Ξ is a subset of $\mathcal{R}^n$ and the mapping $\xi \mapsto p_\xi$ is injective. We call such $\mathcal{S}$ an n-dimensional *statistical model*, a *parametric model*, or simply a *model* on $\mathcal{X}$. We will often abbreviate Equation (1.5) as $\mathcal{S} = \{p_\xi\}$, and also use expression such as $p_\xi(x) = p(x;\xi)$ and $\mathcal{S} = \{p(x;\xi)\}$. When we say "a statistical model $\mathcal{S} = \{p_\xi\}$," there shall be cases in which we refer simply to the set S, and other cases in which we refer in addition to the parameterization $\xi \mapsto p_\xi$.

Let $\mathcal{S} = \{p_\xi \mid \xi \in \Xi\}$ be an n-dimensional statistical model. Given a point ξ $(\in, \Xi)$, the *Fisher information matrix* of $\mathcal{S}$ at ξ is the $n \times n$ matrix $G(\xi) = [g_{ij}(\xi)]$ where the $(i,j)^{\text{th}}$ element $g_{ij}(\xi)$ is defined by the equation below; in particular, when $n = 1$, we call this the *Fisher information*.

$$g_{ij}(\xi) \overset{\text{def}}{=} \mathrm{E}_\xi[\partial_i \ell_\xi \, \partial_j \ell_\xi] = \int \partial_i \ell(x;\xi) \, \partial_j \ell(x;\xi) \, p(x;\xi) \, dx, \tag{1.6}$$

where $\partial_i \overset{\text{def}}{=} \frac{\partial}{\partial \xi^i}$,

$$\ell_\xi(x) = \ell(x;\xi) = \log p(x;\xi). \tag{1.7}$$

We note that it is possible to write g_{ij} as

$$g_{ij}(\xi) = -\mathrm{E}_\xi[\partial_i \partial_j \ell_\xi].$$

Let $\mathcal{S} = \{p_\xi\}$ be an n-dimensional model, and consider the function $\Gamma^{(\alpha)}_{ij,k}$ which maps each point ξ to the following value:

$$\left(\Gamma^{(\alpha)}_{ij,k}\right)_\xi \overset{\text{def}}{=} \mathrm{E}_\xi\left[\left(\partial_i \partial_j \ell_\xi + \frac{1-\alpha}{2} \partial_i \ell_\xi \, \partial_j \ell_\xi\right)(\partial_k \ell_\xi)\right], \tag{1.8}$$

where α is some arbitrary real number. We have an affine connection $\nabla^{(\alpha)}$ on $\mathcal{S}$ defined by

$$\langle \nabla^{(\alpha)}_{\partial_i} \partial_j, \partial_k \rangle = \Gamma^{(\alpha)}_{ij,k}\,, \tag{1.9}$$

where $g = \langle\ ,\ \rangle$ is the Fisher metric. We call this $\nabla^{(\alpha)}$ the ***α**-connection*. The

α-connection is clearly a symmetric connection. We also have

$$\begin{aligned}\nabla^{(\alpha)} &= (1-\alpha)\,\nabla^{(0)} + \alpha\,\nabla^{(1)} \\ &= \frac{1+\alpha}{2}\,\nabla^{(1)} + \frac{1-\alpha}{2}\,\nabla^{(-1)}.\end{aligned} \tag{1.10}$$

In addition, for a submanifold $\mathcal{M}$ of $\mathcal{S}$, the α-connection on $\mathcal{M}$ is simply the projection with respect to g of the α-connection on $\mathcal{S}$.

Let us introduce now the notion of exponential family, which will be shown to have close relation to $\nabla^{(1)}$. In general, if an n-dimensional model $\mathcal{S} = \{p_\theta | \theta \in \Theta\}$ can be expressed in terms of functions $\{C, F_1, \ldots, F_n\}$ on $\mathcal{X}$ and a function ψ on Θ as

$$p(x;\theta) = \exp\left[C(x) + \sum_{i=1}^{n} \theta^i F_i(x) - \psi(\theta)\right], \tag{1.11}$$

then we say that $\mathcal{S}$ is an *exponential family*, and that the $[\theta^i]$ are its *natural* or its *canonical parameters*. From the normalization condition $\int p(x;\theta)dx = 1$ we obtain

$$\psi(\theta) = \log \int \exp\left[C(x) + \sum_{i=1}^{n} \theta^i F_i(x)\right] dx. \tag{1.12}$$

It is easy to see that the parametrization $\theta \mapsto p_\theta$ is one-to-one if and only if the $n+1$ functions $\{F_1, \ldots, F_n, 1\}$ are linearly independent, where 1 denotes the constant function which identically takes the value 1. For more details see [6].

1.3 Exponential Dispersion GLMs

This section is devoted to a brief review of the theory of dispersion models (DM) based primarily on Jørgensen's book [51], *The theory of dispersion models*. The dispersion models provide a rich class of one-dimensional parametric distributions for various data types, including those commonly considered in the GLM analysis. In effect, error distributions in the GLMs form a special subclass of the dispersion models, which are the *exponential dispersion* (ED) *models*. This means that the GLMs considered in [51] encompass a wider scope of GLMs than those outlined in McCullagh and Nelder's book [60], however, we will focus on only the ED models. Two special examples are the *von Mises* distribution for directional (circular or angular) data and the *simplex* distribution for compositional (or proportional) data, both of which are the dispersion models but not the exponential dispersion models. First, let's introduce the DM.

1.3.1 Dispersion Models

Mimicking the density of the normal distribution $N(\mu, \sigma^2)$, [51] defines a dispersion models by extending the Euclidean distance $(y-\mu)^2$, that measures the discrepancy between the observed y and the expected μ, to a general discrepancy function $d(y;\mu)$. It is found that many commonly used parametric distributions, such as Binomial, Poisson and Gamma, are included as special cases of this extension. Moreover, each of such distributions will be determined uniquely by the discrepancy function d, and the resulting distribution is fully parameterized by two parameters μ and σ^2.

A (*reproductive*) *dispersion model* DM(μ, σ^2) with *location parameter* μ and *dispersion parameter* σ^2 is a family of distributions whose probability density functions take the following form:

$$p(y;\mu,\sigma^2) = a(y;\sigma^2)\exp\left\{-\frac{1}{2\,\sigma^2}\,d(y;\mu)\right\}, \quad y \in \mathcal{C}, \tag{1.13}$$

where $\mu \in \Omega$, $\sigma^2 > 0$, and $a \geq 0$ is a suitable normalizing term that is independent of the μ. Usually, $\Omega \subseteq \mathcal{C} \subseteq \mathcal{R}$. The fact that the normalizing term a does not involve μ will allow to estimate μ (or $\boldsymbol{\beta}$ in the GLM setting) separately from estimating σ^2, which gives rise to great ease in the parameter estimation. This a nice property, known as the likelihood orthogonality, holds in the normal distribution, and is a feature in dispersion models.

A bivariate function $d(\cdot;\cdot)$ is called the *unit deviance* defined on $(y,\mu) \in \mathcal{C} \times \Omega$ if it satisfies the following two properties: i) It is zero when the observed y and the expected μ are equal, namely $d(y;y) = 0,\ \forall y \in \Omega$; ii) It is positive when the observed y and the expected μ are different, namely $d(y;\mu) > 0,\ \forall y \neq \mu$.

Furthermore, a unit deviance is called *regular* if function $d(y;\mu)$ is twice continuously differentiable with respect to (y,μ) on $\Omega \times \Omega$ and satisfies

$$\frac{\partial^2 d}{\partial \mu^2}(y;y) = \left.\frac{\partial^2 d}{\partial \mu^2}(y;\mu)\right|_{\mu=y} > 0, \quad \forall y \in \Omega.$$

Table 1.1: Unit deviance and variance functions of some dispersion models.

Distribution	$d(y;\mu)$	$\mathcal{C}$	Ω	V(μ)
Binomial	$2\left\{y\log\frac{y}{\mu}+(n-y)\log\frac{n-y}{n-\mu}\right\}$	$\{0,1,\dots,n\}$	$(0,1)$	$\mu\,(1-\mu)$
Gamma	$2\left(\frac{y}{\mu}-\log\frac{y}{\mu}-1\right)$	$(0,\infty)$	$(0,\infty)$	μ^2
Inverse Gaussian	$\frac{(y-\mu)^2}{y\,\mu^2}$	$(0,\infty)$	$(0,\infty)$	μ^3
Normal	$(y-\mu)^2$	$(-\infty,\infty)$	$(-\infty,\infty)$	1
Poisson	$2\,(y\log\frac{y}{\mu}-y+\mu)$	$\{0,1,\dots\}$	$(0,\infty)$	μ
Simplex	$\frac{(y-\mu)^2}{y(1-y)\mu^2(1-\mu)^2}$	$(0,1)$	$(0,1)$	$\mu^3(1-\mu)^3$
von Misses	$2\,\{1-\cos(y-\mu)\}$	$(0,2\pi)$	$(0,2\pi)$	1

For a regular unit deviance, the variance function is defined as follows. The *unit variance function* $V:\Omega\to(0,\infty)$ is

$$V(\mu)=\frac{2}{\frac{\partial^2 d}{\partial\mu^2}(y;\mu)\big|_{y=\mu}},\qquad \mu\in\Omega. \tag{1.14}$$

Some popular dispersion models are given in Table 1.1.

1.3.2 GLMs based on the Exponential Dispersion Models

The class of dispersion models contains two important subclasses, namely the *exponential dispersion* (ED) *models* and the *proper dispersion* (PD) *models*. The PD models are mostly of theoretical interest, so they are not discussed in this thesis. Readers may refer to [51] for relevant details.

This section focuses on the ED models, which have already been introduced at the beginning of Section 1.3 as a family of GLMs' error distributions. The family of ED models includes continuous distributions such as Normal, Gamma, and Inverse Gaussian, and discrete distributions such as Poisson, Binomial, Negative Binomial, among others.

According to [60], the random component of a GLM is specified by an exponential dispersion family density of the following form:

$$p(y;\theta,\phi)=\exp\left\{\frac{y\theta-b(\theta)}{a(\phi)}+c(y,\phi)\right\},\quad y\in\mathcal{C}, \tag{1.15}$$

with parameters $\theta\in\Theta\subseteq\mathcal{R}$ and $\phi\in\Phi\subseteq\mathcal{R}^+$, where $b(\cdot)$ is the cumulant generating function and $\mathcal{C}$ is the support of the density. It is known that the first derivative of the cumulant function $b(\cdot)$ gives the expectation of the distribution, namely $\mu = E(Y) = b'(\theta)$, where $b'(\theta) = \frac{\partial b(\theta)}{\partial \theta}$, and the variance of the distribution is $\mathrm{Var}(Y) = a(\phi)\, V(\mu)$. This mean-variance relationship is one of the key properties for the ED models, which will play an important role in the development of quasi-likelihood inference.

The systematic component of a GLM is then assumed to take the form:

$$g(\mu) = \mathbf{x}^\top \boldsymbol{\beta} = \beta_0 + \beta_1\, x_1 + \ldots + \beta_p\, x_p \tag{1.16}$$

where g is the link function, $\mathbf{x} = (1, x_1, \ldots, x_p)^\top$ is a $(p{+}1)$-dimensional vector of covariates, and $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^\top$ is a $(p{+}1)$-dimensional vector of regression coefficients. The *canonical* link function $g(\cdot)$ is such that $g(\mu) = \theta$, the canonical parameter. The primary statistical tasks include estimation and inference for $\boldsymbol{\beta}$.

To establish the connection of the ED model representation (1.15) to the DM, it is sufficient to show that expression (1.15) is a special form of (1.13). An advantage with the DM type of parametrization for the ED models is that both mean μ and dispersion parameters σ^2 are explicitly present in the density, whereas expression (1.15) hides the mean μ in the first order derivative $b'(\theta)$. In addition, having a density form similar to the normal enables us to easily borrow the classical normal regression theory to the development of regression analysis for non-normal data.

To reparametrize this density (1.15) by the mean μ and dispersion σ^2, denote $a(\phi) = \sigma^2$ and define the *mean value mapping*: $\tau : int(\Theta) \to \Omega$,

$$\tau(\theta) = b'(\theta) \equiv \mu,$$

where $int(\Theta)$ is the interior of the parameter space Θ. The mean mapping function $\tau(\theta)$ is strictly increasing and its inverse exists, denoted by $\theta = \tau^{-1}(\mu)$, $\mu \in \Omega$.

As a result, the density of an ED model in (1.15), denoted by $\mathrm{ED}(\mu, \sigma^2)$, can be expressed as of the DM form in (1.13) with the unit deviance function d given by

$$d(y; \mu) = 2\left[\sup_{\theta \in \Theta}\{\theta y - b(\theta)\} - y\, \tau^{-1}(\mu) + b(\tau^{-1}(\mu))\right], \tag{1.17}$$

and the normalizing term given by

$$a(y;\sigma^2) = c(y;\sigma^{-2}) \exp\left[\sigma^{-2} \sup_{\theta\in\Theta}\{\theta y - b(\theta)\}\right]. \tag{1.18}$$

Clearly, this d function (1.17) satisfies (i) $d(y;\mu) \geq 0$ for all $y \in \mathcal{C}$ and $\mu \in \Omega$, and (ii) $d(y;\mu)$ attains the minimum at $\mu = y$ because the supremum term is independent of μ. Thus, (1.17) gives a proper unit deviance function. Moreover, since it is continuously twice differentiable, it is also regular.

An important variant of the reproductive ED model representation is the so-called *additive exponential dispersion model*, denoted by $\text{ED}^*(\theta,\lambda)$, whose density takes the form

$$p^*(z;\theta,\lambda) = c^*(z;\lambda) \exp\{\theta\, z - \lambda\, b(\theta)\}, \quad z \in \mathcal{C}, \tag{1.19}$$

The Gamma and Inverse Gaussian distributions are members of the ED^* and ED families, respectively. Essentially the ED and ED^* representations are equivalent under the *duality transformation* that converts one form to the other. Suppose $Z \sim \text{ED}^*(\theta,\lambda)$ and $Y \sim \text{ED}(\mu,\sigma^2)$. Then, the duality transformation performs

$$Z \sim \text{ED}^*(\theta,\lambda) \Rightarrow Y = Z/\lambda \sim \text{ED}(\mu,\sigma^2), \text{ with } \mu = \tau(\theta), \text{ and } \sigma^2 = 1/\lambda;$$

$$Y \sim \text{ED}(\mu,\sigma^2) \Rightarrow Z = Y/\sigma^2 \sim \text{ED}^*(\theta,\lambda), \text{ with } \theta = \tau^{-1}(\mu), \text{ and } \lambda = 1/\sigma^2.$$

Consequently, the mean and variance of $\text{ED}^*(\theta,\lambda)$ are, respectively,

$$\mu^* = \text{E}(Z) = \lambda\,\tau(\theta), \text{ and } \text{Var}(Z) = \lambda\, V(\mu^*/\lambda).$$

An important property for these models is closure under the convolution operation.

***Convolution for the* ED^* *models*.** Assume $Z_1,\ldots,Z_n$ are independent and $Z_i \sim \text{ED}^*(\theta,\lambda_i), i = 1,\ldots,n$, then the sum follows still an ED^* model:

$$Z_+ = \sum_{i=1}^n Z_i \sim \text{ED}^*(\theta, \sum_{i=1}^n \lambda_i).$$

***Convolution for the* ED *models*.** Assume $Y_1,\ldots,Y_n$ are independent and

$Y_i \sim \mathrm{ED}(\mu, \frac{\sigma^2}{w_i})$, where w_is are certain positive weights. Let $w_+ = w_1 + \ldots + w_n$, then

$$\frac{1}{w_+}\sum_{i=1}^{n} w_i\, Y_i \sim \mathrm{ED}(\mu, \frac{\sigma^2}{w_+}).$$

We note that although the class of the ED models is closed under the convolution operation, it is in general not closed under scale transformation. That is, cY may not follow an ED model even if $Y \sim \mathrm{ED}(\mu, \sigma^2)$, for a constant c. However, a subclass of the ED models, termed as the *Tweedie class*, is closed under this type of scale transformation. Tweedie class is an important subclass of the ED models. Tweedie models are characterized by the unit variance functions in the form of the power function:

$$V_p(\mu) = \mu^p, \;\; \mu \in \Omega_p, \tag{1.20}$$

where $p \in R$ is a shape parameter. It is shown that the ED model with the power unit variance function (1.20) always exists except $0 < p < 1$. Special cases include the Normal ($p = 0$), Poisson ($p = 1$), Gamma ($p = 2$) and Inverse Gaussian ($p = 3$). Another interesting class of Tweedie GLMs is for values of p between 1 and 2. In this interval, closed form distribution functions do not exist, but Tweedies in this interval are compound Poisson distributions. (A compound Poisson random variable Y is the sum of N independent Gamma random variables where N follows a Poisson distribution and N and the Gamma random variates are independent.) A Tweedie model is denoted by $Y \sim Tw_p(\mu, \sigma^2)$ with mean μ and variance

$$\mathrm{Var}(Y) = \sigma^2\, \mu^p.$$

1.3.3 MLE in the Exponential Dispersion GLMs

This section is devoted to maximum likelihood estimation in the GLMs based on the ED models. Consider $(y_i, \mathbf{x}_i)$, $i = 1, \ldots n$, as a dataset where the y_is are i.i.d. realizations of Y_is according to $\mathrm{ED}(\mu_i, \sigma^2)$ and $g(\mu_i) = \mathbf{x}_i^\top \boldsymbol{\beta}$. Let $\mathbf{y} = (y_1, \ldots, y_n)^\top$ and $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)^\top$. The likelihood for the parameter vector $\boldsymbol{\theta} = (\boldsymbol{\beta}, \sigma^2)$ is given by

$$L(\boldsymbol{\theta}; \mathbf{y}) = \prod_{i=1}^{n} a(y_i; \sigma^2) \exp\left\{-\frac{1}{2\,\sigma^2}\, d(y_i; \mu_i)\right\}, \quad \boldsymbol{\beta} \in \mathcal{R}^{p+1},\ \sigma^2 > 0,$$

and the log-likelihood is then

$$\ell(\boldsymbol{\theta};\mathbf{y}) = \sum_{i=1}^{n} \log a(y_i;\sigma^2) - \frac{1}{2\,\sigma^2}\,\mathcal{D}(\mathbf{y};\boldsymbol{\mu}), \tag{1.21}$$

where $\mathcal{D}(\mathbf{y};\boldsymbol{\mu}) = \sum_{i=1}^{n} d(y_i;\mu_i)$ is the sum of deviances depending on $\boldsymbol{\beta}$ only and $\mu_i = \mu_i(\boldsymbol{\beta})$ is a nonlinear function in $\boldsymbol{\beta}$.

The *score function* for the regression coefficient $\boldsymbol{\beta}$ is

$$\begin{aligned} s(\mathbf{y};\boldsymbol{\beta}) = \frac{\partial \ell(\boldsymbol{\theta})}{\partial \boldsymbol{\beta}} &= -\frac{1}{2\,\sigma^2}\sum_{i=1}^{n} \frac{\partial d(y_i;\mu_i)}{\partial \mu_i}\,\frac{\partial \mu_i}{\partial \boldsymbol{\beta}} \\ &= \frac{1}{\sigma^2}\sum_{i=1}^{n} \mathbf{x}_i\,\frac{(y_i-\mu_i)}{g'(\mu_i)\,V(\mu_i)} \end{aligned} \tag{1.22}$$

because in this case

$$\frac{\partial d(y_i;\mu_i)}{\partial \mu_i} = -2\,\frac{(y_i-\mu_i)}{V(\mu_i)} \quad \text{and} \quad \frac{\partial \mu_i}{\partial \boldsymbol{\beta}} = \frac{\partial \mu_i}{\partial \eta_i}\frac{\partial \eta_i}{\partial \boldsymbol{\beta}} = \{g'(\mu_i)\}^{-1}\,\mathbf{x}_i$$

where $\eta_i = \mathbf{x}_i^\top \boldsymbol{\beta}$ is the i^{th} linear predictor, and $g'(\mu)$ is the first order derivative of the link function g w.r.t. μ.

Moreover, the score equation leading to the maximum likelihood estimate of $\boldsymbol{\beta}$ is

$$\sum_{i=1}^{n} \mathbf{x}_i\,\frac{(y_i-\mu_i)}{g'(\mu_i)\,V(\mu_i)} = \mathbf{0}, \tag{1.23}$$

such that it can be re-expressed in matrix form as

$$\mathbf{X}^\top\,\mathbf{W}^{-1}\,(\mathbf{y}-\boldsymbol{\mu}) = \mathbf{0},$$

where $\mathbf{X}$ is a $n \times (p+1)$ matrix with the i^{th} row being the $\mathbf{x}_i^\top$, and $\mathbf{W} = \text{diag}(w_1,\ldots,w_n)$ with $w_i = g'(\mu_i)\,V(\mu_i)$.

Note that this equation does not involve the dispersion parameter σ^2. Under some mild regularity conditions, the resulting ML estimator $\hat{\boldsymbol{\beta}}_n$, which is the solution to the score equation (1.23), is consistent

$$\hat{\boldsymbol{\beta}}_n \xrightarrow{p} \boldsymbol{\beta} \quad \text{as} \quad n \to \infty,$$

and asymptotically normal with mean 0 and covariance matrix $\mathcal{I}^{-1}(\boldsymbol{\theta})$. Here, $\mathcal{I}(\boldsymbol{\theta})$ is the *Fisher information matrix*, which is an $(p+1) \times (p+1)$ matrix, given by

$$\begin{aligned}
\mathcal{I}(\boldsymbol{\theta}) &= -\mathrm{E}\left\{\frac{\partial s(\mathbf{y};\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\right\} \\
&= \frac{1}{\sigma^2}\sum_{i=1}^{n} \mathbf{x}_i\, u_i^{-1} \mathbf{x}_i^\top \\
&= \mathbf{X}^\top\, \mathbf{U}^{-1}\mathbf{X}/\sigma^2, \qquad (1.24)
\end{aligned}$$

where $\mathbf{U}$ is a diagonal matrix with the i^{th} diagonal element u_i given by $u_i = \{g'(\mu_i)\}^2\, V(\mu_i)$. It consists of the elements $\mathcal{I}_{ij}(\boldsymbol{\theta}) = \mathrm{E}[\frac{\partial \ell(\boldsymbol{\theta};\mathbf{y})}{\partial \beta_i}\frac{\partial \ell(\boldsymbol{\theta};\mathbf{y})}{\partial \beta_j}]$. The Fisher information, which is the expected value of the observed information, gives information about the efficiency of the maximum likelihood. It determines the conditional correlation between β_i and β_j and we say that two parameters β_i and β_j are orthogonal if the element of the i^{th} row and j^{th} column of the Fisher information matrix is zero.

It is interesting to note that the choice of the canonical link function $g = \tau^{-1}(\cdot)$ simplifies both score function and Fisher information. Under the canonical link function, the score equation of an ED GLM is

$$\sum_{i=1}^{n} \mathbf{x}_i\,(y_i - \mu_i) = \mathbf{0}, \quad \text{or} \quad \mathbf{X}^\top\,(\mathbf{y} - \boldsymbol{\mu}) = \mathbf{0},$$

and the Fisher information takes the form

$$\mathcal{I}(\boldsymbol{\theta}) = \mathbf{X}^\top\, \mathbf{U}^{-1}\mathbf{X}/\sigma^2$$

where $\mathbf{U}$ is a diagonal matrix whose i^{th} diagonal element given by $u_i = 1/V(\mu_i)$. Because, in this case, $w_i = 1$ and $g'(\mu_i) = 1/V(\mu_i)$, the matrix $\mathbf{W}$ becomes the identity matrix and the matrix $\mathbf{U}$ is determined by the reciprocals of the variance functions.

When the dispersion parameter σ^2 is present in the model, the ML estimation for the dispersion parameter σ^2 can be derived similarly, if the normalizing term $a(y;\sigma^2)$ is simple enough to allow such a derivation, such as the case of the normal distribution. However, in many cases, the term $a(\cdot)$ has no closed form expression and its derivative w.r.t. σ^2 may appear too complicated to be numerically solvable. In this case, three methods have been suggested to acquire the estimation for σ^2. The first method, which is referred to as the *Jørgensen* estimator of the dispersion parameter, is

$$\hat{\sigma}_d^2 = \frac{1}{n}\mathcal{D}(\mathbf{y}; \hat{\boldsymbol{\mu}}) = \frac{1}{n}\sum_{i=1}^{n} d(y_i; \hat{\mu}_i), \tag{1.25}$$

where the index d stands for *deviance*. This estimator, in fact, is an average of the estimated unit deviances. However, this estimator is not, in general, unbiased even if the adjustment on the degrees of freedom, $n-(p+1)$, is made to replace n. Moreover, this formula is recommended when the dispersion parameter σ^2 is small, say less than 5. For more details about this estimator see [51] and [60].

Each ED model holds the so-called mean-variance relation, i.e., $\text{Var}(Y) = \sigma^2 V(\mu)$, which may be used to obtain a consistent estimator of the dispersion parameter σ^2 given as follows:

$$\hat{\sigma}_{P*}^2 = \frac{1}{n-p-1}\sum_{i=1}^{n}\frac{(y_i-\hat{\mu}_i)^2}{V(\hat{\mu}_i)}. \tag{1.26}$$

This second method, which utilizes a moment property, is referred to as the *Pearson* estimator of the dispersion parameter σ^2. The third method is the maximum likelihood (ML) method. The ML estimator of the dispersion parameter σ^2 is the solution of $\partial\ell(\hat{\boldsymbol{\beta}}, \sigma^2; \mathbf{y})/\partial\sigma^2 = 0$. More information about this estimator $\hat{\sigma}_{mle}^2$ can be found in [51] and [60].

1.3.4 A Differential Geometrical Description of the GLM

In this section we introduce the GLM from a differential geometric point of view. In our treatment, we rely heavily on [5], [53] and [6]. A differential geometric approach was also used in [98] to study non-linear models based on the exponential family. Essential aspects of differential and information geometry have been included in this section.

Under family (1.15), the joint probability density function of the random vector $\mathbf{Y}$ can be written as

$$p_{\mathbf{Y}}(\mathbf{y};\boldsymbol{\theta},\phi)=\prod_{i=1}^{n}p_{Y_i}(y_i;\theta_i,\phi), \tag{1.27}$$

where $\mathbf{Y}=(Y_1,Y_2,\ldots,Y_n)^\top$ is a random vector with independent components, Y_i is assumed to be a random variable with probability density function belonging to the family (1.15), and the canonical parameter $\boldsymbol{\theta}$ varies in the subset $\otimes_{i=1}^{n}\Theta_i=\Theta\subseteq\mathcal{R}^n$. As mentioned in Section 1.3.2, $\mathrm{E}(\mathbf{Y})=\boldsymbol{\mu}=(\tau(\theta_1),\ldots,\tau(\theta_n))^\top$, where $\tau(\theta_i)=b'(\theta_i)\equiv\mu_i$ is called mean value mapping, and $\mathrm{Var}(\mathbf{Y})=a(\phi)\,\mathbf{V}(\boldsymbol{\mu})$, where $\mathbf{V}(\boldsymbol{\mu})=\mathrm{diag}(V(\mu_1),\ldots,V(\mu_n))$ is an $n\times n$ diagonal matrix where $V(\mu_i)=b''(\theta_i)$ is called the variance function. From Section 1.3.2 we have $\tau:int(\Theta)\to\Omega$ so that $\tau(\cdot)$ is a one-to-one function, therefore $p_{\mathbf{Y}}(\mathbf{y};\boldsymbol{\theta},\phi)$ may be parameterized by $p_{\mathbf{Y}}(\mathbf{y};\boldsymbol{\mu},\phi)$, as described in Section 1.3.1 and 1.3.2. To simplify our notation we will assume that $\phi=1$ [53]. Assuming that Θ is open, the set

$$\mathcal{S}=\{p_{\mathbf{Y}}(\mathbf{y};\boldsymbol{\mu}):\boldsymbol{\mu}\in\Omega\} \tag{1.28}$$

is a minimal and regular exponential family of order n and can be treated as a differential manifold where the parameter vector $\boldsymbol{\mu}$ plays the role of a co-ordinate system [5]. The notion of differential manifold is necessary for extending the methods of differential calculus to a space that is more general than $\mathcal{R}^n$. For a rigorous definition of a differential manifold the reader is referred to [90] and [21]. It is worth noting that the results coming from differential geometry are not related to the chosen co-ordinate system, i.e., the parameterization that is used to specify the probability density function (1.15). This means that we could work with the differential manifold $\mathcal{S}$ using the parameter vector $\boldsymbol{\theta}$ as co-ordinate system. In this thesis we prefer to use definition (1.28) only because we believe that this makes the generalization of the LARS algorithm clearer.

Following [60], a Generalized Linear Model (GLM) is defined by means of a known function $g(\cdot)$, called link function, relating the expected value of each Y_i to the vector of covariates $\mathbf{x}_i=(1,x_{i1},\ldots,x_{ip})^\top$ by the identity

$$g\{\mathrm{E}(Y_i)\}=\eta_i=\mathbf{x}_i^\top\boldsymbol{\beta}$$

where η_i is called the i^{th} linear predictor and $\boldsymbol{\beta}=(\beta_0,\beta_1,\ldots,\beta_p)^\top$ is the vector of regression coefficients with the intercept and p parameters. In order to simplify our notation we let $\boldsymbol{\mu}(\boldsymbol{\beta}) = \{\mu_1(\boldsymbol{\beta}), \ldots, \mu_n(\boldsymbol{\beta})\}^\top$ where $\mu_i(\boldsymbol{\beta}) = g^{-1}(\mathbf{x}_i^\top \boldsymbol{\beta})$. Therefore, the probability density function can be written as $p_{\mathrm{Y}}(\mathbf{y}; \boldsymbol{\mu}(\boldsymbol{\beta}), \phi) = \prod_{i=1}^n p_{Y_i}(y_i; \mu_i(\boldsymbol{\beta}), \phi)$.

In order to study the geometrical structure of a GLM, we shall assume that $\boldsymbol{\beta} \to \{g^{-1}(\mathbf{x}_1^\top \boldsymbol{\beta}), \ldots, g^{-1}(\mathbf{x}_n^\top \boldsymbol{\beta})\}^\top = \boldsymbol{\mu}(\boldsymbol{\beta})$ is an embedding, this means that the set

$$\mathcal{M} = \{p_{\mathbf{Y}}(\mathbf{y}; \boldsymbol{\mu}(\boldsymbol{\beta})) \in \mathcal{S} : \boldsymbol{\beta} \in \mathcal{R}^{p+1}\}$$

is a $p+1$-dimensional submanifold of $\mathcal{S}$, which inherits the dualistic structure from its ambient space, then, as a simple consequence of theorem 3.5 in [6], $\mathcal{M}$ is a dually flat space only when we work with the canonical link function. To obtain a natural generalization of the equiangularity condition that was proposed by [29], it is necessary to introduce two fundamental notions on which Riemannian geometry is based: the notions of a tangent space and a Riemannian metric. To complete the differential geometric setting for the GLM, we shall assume that the usual regularity conditions hold [5, page 16]. Throughout this paper we use the convention that the indices i, j and k correspond to the quantities that are related to $\boldsymbol{\mu} \in \Omega$ whereas the indices l, m and q correspond to the quantities that are related to the coefficients $\boldsymbol{\beta} \in \mathcal{R}^{p+1}$ of our regression model.

Consider a double-differentiable curve, say $\boldsymbol{\mu} : \Gamma \to \Omega$, where Γ is the real interval $(-\delta, \delta)$ with $\delta > 0$. The tangent vector to the one-parametric family $p_{\mathbf{Y}}(\mathbf{y}; \boldsymbol{\mu}(\gamma))$ at $\boldsymbol{\mu} = \boldsymbol{\mu}(0)$ is defined as

$$v(\mathbf{Y}) = \left.\frac{d\ell(\boldsymbol{\mu}(\gamma); \mathbf{Y})}{d\gamma}\right|_{\gamma=0} = \sum_{i=1}^n d\mu_i(0)\, \partial_i \ell(\boldsymbol{\mu}; \mathbf{Y}), \tag{1.29}$$

where $d\mu_i(0) = d\mu_i(\gamma)/d\gamma|_{\gamma=0}$ and $\partial_i \ell(\boldsymbol{\mu}; \mathbf{Y}) = \partial \log\{p_{\mathbf{Y}}(\mathbf{Y}; \boldsymbol{\mu}(\gamma))\}/\partial\mu_i|_{\gamma=0}$. Roughly speaking, the tangent space of $\mathcal{S}$ at the point $p_{\mathbf{Y}}(\mathbf{y}; \boldsymbol{\mu})$ denoted by $T_{p(\boldsymbol{\mu})}\mathcal{S}$, is the set of all possible tangent vectors at $\boldsymbol{\mu} = \boldsymbol{\mu}(0)$. Formally, $T_{p(\boldsymbol{\mu})}\mathcal{S}$ is the vector space that is spanned by the n score functions $\partial_i \ell(\boldsymbol{\mu}; \mathbf{Y})$:

$$T_{p(\boldsymbol{\mu})}\mathcal{S} = \mathit{span}\{\partial_1 \ell(\boldsymbol{\mu}; \mathbf{Y}), \partial_2 \ell(\boldsymbol{\mu}; \mathbf{Y}), \ldots, \partial_n \ell(\boldsymbol{\mu}; \mathbf{Y})\}. \tag{1.30}$$

Under the regularity conditions cited above, $T_{p(\boldsymbol{\mu})}\mathcal{S}$ is a subspace of squared integrable random variables, in which elements $v(\mathbf{Y})$ satisfy the property $\mathrm{E}_{\boldsymbol{\mu}}\{v(\mathbf{Y})\} = 0$, where the expected value is computed with respect to $p_{\mathbf{Y}}(\mathbf{y}; \boldsymbol{\mu})$. As an application of the chain rule, it is easy to see that the definition of a tangent space does not depend on the chosen parameterization; in other words the tangent space can be defined as the vector space that is spanned by the n score functions $\partial_i^* \ell(\boldsymbol{\theta}; \mathbf{Y}) = \partial \log\{p_{\mathbf{Y}}(\mathbf{Y}; \boldsymbol{\theta}(\gamma))\}/\partial\theta_i|_{\gamma=0}$ where $\boldsymbol{\theta}(\gamma) = \theta(\boldsymbol{\mu}(\gamma))$. Using the terminology that was introduced in [97], $\partial_i \ell(\boldsymbol{\mu}; \mathbf{Y})$ are the natural bases of the tangent space when we choose $\boldsymbol{\mu}$ as co-ordinate system, whereas $\partial_i^* \ell(\boldsymbol{\theta}; \mathbf{Y})$ are the natural bases when $\boldsymbol{\theta}$ is used as the co-ordinate system.

Similarly, consider a double-differentiable curve $\boldsymbol{\beta} : \Gamma' \to \mathcal{R}^{p+1}$, with $\Gamma' = (-\delta', \delta')$ and $\delta' > 0$. The tangent vector to the one-parametric family $p_{\mathbf{Y}}(\mathbf{y}; \boldsymbol{\mu}(\boldsymbol{\beta}(\gamma)))$ at the point $\boldsymbol{\beta} = \boldsymbol{\beta}(0)$ is defined as

$$w(\mathbf{Y}) = \sum_{m=1}^{p} d\beta_m(0)\, \partial_m \ell(\boldsymbol{\beta}; \mathbf{Y}),$$

where $d\beta_m(0) = d\beta_m(\gamma)/d\gamma|_{\gamma=0}$ and $\partial_m \ell(\boldsymbol{\beta}; \mathbf{Y}) = \partial \log\{p_{\mathbf{Y}}(\mathbf{Y}; \boldsymbol{\mu}(\boldsymbol{\beta}(\gamma)))\}/\partial\beta_m|_{\gamma=0}$. Then, the tangent space of $\mathcal{M}$ at the point $p_{\mathbf{Y}}(\mathbf{y}; \boldsymbol{\mu}(\boldsymbol{\beta}))$ is

$$T_{p(\boldsymbol{\mu}(\boldsymbol{\beta}))}\mathcal{M} = span\{\partial_1 \ell(\boldsymbol{\beta}; \mathbf{Y}), \partial_2 \ell(\boldsymbol{\beta}; \mathbf{Y}), \ldots, \partial_n \ell(\boldsymbol{\beta}; \mathbf{Y})\}. \tag{1.31}$$

The definition of the inner product on each tangent space allows us to generalize the notion of angle between two curves, say $\boldsymbol{\mu}_1(\gamma)$ and $\boldsymbol{\mu}_2(\gamma)$, intersecting at $\boldsymbol{\mu}_1(0) = \boldsymbol{\mu}_2(0) = \boldsymbol{\mu}$, with tangent vectors belonging to $T_{p(\boldsymbol{\mu})}\mathcal{S}$, denoted by

$$v_1(\mathbf{Y}) = \sum_{i=1}^{n} d\mu_{1,i}(0)\, \partial_i \ell(\boldsymbol{\mu}; \mathbf{Y})$$

and

$$v_2(\mathbf{Y}) = \sum_{i=1}^{n} d\mu_{2,i}(0)\, \partial_i \ell(\boldsymbol{\mu}; \mathbf{Y})$$

respectively. When working with a parametric family of distributions, the inner product can be defined in a natural way [78], i.e.

$$\langle v_1(\mathbf{Y}), v_2(\mathbf{Y})\rangle_{p(\boldsymbol{\mu})} = \mathrm{E}_{\boldsymbol{\mu}}\{v_1(\mathbf{Y})\, v_2(\mathbf{Y})\} = d\boldsymbol{\mu}_1(0)^{\top}\mathcal{I}(\boldsymbol{\mu})d\boldsymbol{\mu}_2(0),$$

where $\mathcal{I}(\boldsymbol{\mu})$ is the Fisher information matrix for the mean parameter at point $\boldsymbol{\mu}$. In other words, the Fisher information defines a Riemannian metric by associating with each point of $\mathcal{S}$ an inner product on the tangent space. This Riemannian metric is also called the *information metric* [17]. Since $T_{p\{\boldsymbol{\mu}(\boldsymbol{\beta})\}}\mathcal{M}$ is a linear subspace of $T_{p\{\boldsymbol{\mu}(\boldsymbol{\beta})\}}\mathcal{S}$, the Fisher information also defines an inner product on $T_{p\{\boldsymbol{\mu}(\boldsymbol{\beta})\}}\mathcal{M}$. Therefore, we can define the inner product between a tangent vector $w(\mathbf{Y})$ of $T_{p\{\boldsymbol{\mu}(\boldsymbol{\beta})\}}\mathcal{M}$ and a tangent vector $v(\mathbf{Y})$ of $T_{p\{\boldsymbol{\mu}(\boldsymbol{\beta})\}}\mathcal{S}$, namely

$$\langle w(\mathbf{Y}), v(\mathbf{Y})\rangle_{p\{\boldsymbol{\mu}(\boldsymbol{\beta})\}} = \mathrm{E}_{\boldsymbol{\mu}(\boldsymbol{\beta})}\{v_1(\mathbf{Y})\, v_2(\mathbf{Y})\} = d\boldsymbol{\beta}(0)^\top \frac{\partial \boldsymbol{\mu}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}^\top \mathcal{I}\{\boldsymbol{\mu}(\boldsymbol{\beta})\} d\boldsymbol{\mu}(0),$$

where $\partial\boldsymbol{\mu}(\boldsymbol{\beta})/\partial\boldsymbol{\beta}$ is the Jacobian matrix of the vector function $\boldsymbol{\mu}(\boldsymbol{\beta})$.

Each Riemannian metric defines the notion of a geodesic, i.e. the generalization of a straight line in a differential geometric framework. Roughly speaking, a geodesic can be defined as the shortest path between two given points on a differential manifold. A geodesic is defined as the solution of a system of differential equations, the Euler–Lagrange equations, obtained from defining a connection on a differentiable manifold. In statistical theory a one-parametric family of connections plays a fundamental role, the so-called α-connections, denoted by ∇^α, that generalize the classical notion of a Levi–Civita connection, which is the special case that $\alpha = 0$. In the theory of information geometry, ∇^0 is also called the *information connection* since it is derived from the Fisher information. What is also important for what follows in this thesis is that $\mathcal{S}$ is a dually flat space, namely, it is flat with respect to the 1- and -1-connection. For more details of this dual geometry, the reader is referred to [6]. As shown in [97], associated with the -1-connection and each point $p_{\mathbf{Y}}(\mathbf{y};\boldsymbol{\mu}) \in \mathcal{S}$ there is a diffeomorphism between a neighbourhood of the origin in $T_{p(\boldsymbol{\mu})}\mathcal{S}$ and a neighbourhood of $p_{\mathbf{Y}}(\mathbf{y};\boldsymbol{\mu})$, called the -1-*exponential map*. The dual nature that exists between ∇^{-1} and ∇^1 defines the dual of the -1-exponential map, namely the so-called 1-exponential map. Since $\mathcal{S}$ is a dually flat space, the inverses of the two exponential maps are well defined on all $\mathcal{S}$ and for each $p_{\mathbf{Y}}(\mathbf{y};\boldsymbol{\mu})$. To complete the geometrical framework that is needed to generalize the LARS algorithm in next chapter, we consider the inverse of the -1-exponential map, which relates the observed response variable y to the tangent spaces. [97] defined what we call the *tangent residual vector*

$$\mathbf{r}(\boldsymbol{\mu}(\boldsymbol{\beta}), \mathbf{y}; \mathbf{Y}) = \sum_{i=1}^{n} \{y_i - \mu_i(\boldsymbol{\beta})\} \, \partial_i \ell(\boldsymbol{\mu}(\boldsymbol{\beta}); \mathbf{Y}) \tag{1.32}$$

where $\partial_i \ell(\boldsymbol{\mu}(\boldsymbol{\beta}); \mathbf{Y}) = \partial \ell(\boldsymbol{\mu}; \mathbf{Y}) / \partial \mu_i |_{\boldsymbol{\mu}=\boldsymbol{\mu}(\boldsymbol{\beta})}$. It is important to note that we define the tangent residual vector (1.32) with respect to both the fixed observations $\mathbf{y}$ and the random variable $\mathbf{Y}$, in such a way that it is a random variable with zero expected value and finite variance, and therefore $\mathbf{r}(\boldsymbol{\mu}(\boldsymbol{\beta}), \mathbf{y}; \mathbf{Y}) \in T_{p\{\boldsymbol{\mu}(\boldsymbol{\beta})\}}\mathcal{S}$. [97] showed that it is possible to give a differential geometric interpretation of the maximum likelihood estimator by using the tangent residual vector and the tangent space $T_{p\{\boldsymbol{\mu}(\hat{\boldsymbol{\beta}})\}}\mathcal{M}$, namely $\hat{\boldsymbol{\beta}}$ is the maximum likelihood estimate of $\boldsymbol{\beta}$ when the tangent residual vector is orthogonal to the tangent space $T_{p\{\boldsymbol{\mu}(\hat{\boldsymbol{\beta}})\}}\mathcal{M}$. It is worth noting that this statement is well defined even if $\mathbf{y}$ is not an element of the mean value parameter space Ω. In other words, the differential geometric description of the maximum likelihood estimator can be used even if the Kullback–Leibler divergence is not defined [97].

1.4 Survival Models

Survival analysis is a commonly-used method for the analysis of failure times such as death, mechanical failure, or credit default. Within this context, a failure is also referred to as an 'event'. Survival models can be used for the analysis of data which have three main characteristics: (1) the dependent variable or response is the waiting *time* until the occurrence of a well-defined event, (2) observations may be *censored*, in the sense that for some units the event of interest has not occurred at the time the data are analyzed, and (3) there are predictors or *explanatory* variables whose effect on the waiting time we wish to assess or control.

Let T be a non-negative random variable representing the waiting time until the occurrence of an event. For simplicity we will adopt the terminology of survival analysis, referring to the event of interest as *death* and to the waiting time as *survival* time, but the techniques to be studied have much wider applicability. They can be used, for example, to study age at marriage, the duration of marriage, the intervals between successive births, the duration of stay in a city or in a job, besides the length of life.

1.4.1 The Survival and Hazard Function

We will assume for now that T is a continuous random variable with probability density function (p.d.f.) $f(t)$ and cumulative distribution function (c.d.f.) $F(t) = \Pr\{T < t\}$, giving the probability that the event has occurred by duration t.

It will often be convenient to work with the complement of the c.d.f, the *survival* function

$$S(t) = \Pr\{T \geq t\} = 1 - F(t) = \int_t^\infty f(x)dx, \tag{1.33}$$

which gives the probability of being alive just before duration t, or more generally, the probability that the event of interest has not occurred by duration t.

An alternative characterization of the distribution of T is given by the *hazard function*, or instantaneous rate of occurrence of the event, defined as

$$\lambda(t) = \lim_{dt \to 0} \frac{\Pr\{t \leq T < t + dt \mid T \geq t\}}{dt}. \tag{1.34}$$

The numerator of this expression is the conditional probability that the event will occur in the interval $[t, t + dt)$ given that it has not occurred before, and the denominator is the width of the interval. Dividing one by the other we obtain a rate of event occurrence per unit of time. Taking the limit as the width of the interval goes down to zero, we obtain an instantaneous rate of occurrence. The conditional probability in the numerator may be written as the ratio of the joint probability that T is in the interval $[t, t + dt)$ and $T \geq t$ (which is, of course, the same as the probability that t is in the interval), to the probability of the condition $T \geq t$. The former may be written as $f(t)dt$ for small dt, while the latter is $S(t)$ by definition. Dividing by dt and passing to the limit gives the useful result

$$\lambda(t) = \frac{f(t)}{S(t)}, \tag{1.35}$$

which some authors give as a definition of the hazard function. In words, the rate of occurrence of the event at t equals the density of events at t, divided by the probability of surviving to that time without experiencing the event.

Note from Equation (1.33) that $-f(t)$ is the derivative of $S(t)$. This suggests rewriting Equation (1.35) as

$$\lambda(t) = -\frac{d}{dt} \log S(t).$$

If we now integrate from 0 to t and introduce the boundary condition $S(0) = 1$ (since the event is assumed not to have occurred at the beginning of the study), we can solve the above expression to obtain a formula for the probability of surviving to duration t as a function of the hazard at all durations up to t:

$$S(t) = \exp\{-\int_0^t \lambda(x)dx\} = \exp\{-\Lambda(t)\}, \tag{1.36}$$

where $\Lambda(t)$ is called the *cumulative hazard* (or cumulative risk).

These results show that the survival and hazard functions provide alternative but equivalent characterizations of the distribution of T. Given the survival function, we can always differentiate to obtain the density and then calculate the hazard using Equation (1.35). Given the hazard, we can always integrate to obtain the cumulative hazard and then exponentiate to obtain the survival function using Equation (1.36).

1.4.2 Censoring Mechanisms

The second distinguishing feature of survival analysis is censoring: the fact that for some units the event of interest has occurred and therefore we know the exact waiting time, whereas for others there is no precise knowledge about the survival time except that it falls in some interval.

There are several mechanisms that can lead to censored data. Under censoring of *Type I*, a sample of n units is followed for a fixed time t. The number of units experiencing the event, or the number of 'deaths', is random, but the total duration of the study is fixed. The fact that the duration is fixed may be an important practical advantage in designing a follow-up study.

In a simple generalization of this scheme, called *fixed censoring*, each unit has a potential maximum observation time t_i for $i = 1, \ldots, n$ which may differ from one case to the next but is nevertheless fixed in advance. The probability that unit i will be alive at the end of her observation time is $S(t_i)$, and the total number of deaths is again random.

Under censoring of *Type II*, a sample of n units is followed as long as necessary until d units have experienced the event. In this design the number of deaths d, which determines the precision of the study, is fixed in advance and can be used as a design parameter. Unfortunately, the total duration of the study is then random and cannot be known with certainty in advance.

In a more general scheme called *random censoring*, each unit has associated with it a potential censoring time C_i and a potential lifetime Z_i , which are assumed to the independent random variables. We observe $T_i = \min\{C_i, Z_i\}$, the minimum of the censoring and life times, and an indicator variable $\delta_i = \mathcal{I}(Z_i \leq C_i)$, often called *status*, that tells us whether observation terminated by death or by censoring.

All these schemes have in common the fact that the censoring mechanism is *non-informative* and they all lead to essentially the same likelihood function. The weakest assumption required to obtain this common likelihood is that the censoring of an observation should not provide any information regarding the prospects of survival of that particular unit beyond the censoring time. In fact, the basic assumption that we will make is simply this: all we know for an observation censored at duration t is that the lifetime exceeds t.

1.4.3 The Relative Risk Regression Models

The third distinguishing characteristic of survival models is the presence of covariates or explanatory variables that may affect survival time.

Let Z, C, and $\mathbf{X} = (X_1, X_2, \ldots, X_p)^\top$ denote the survival time, the censoring time, and their associated covariates, respectively, where p denotes the dimensionality of the covariate space. Correspondingly, denote by $T = \min\{Z, C\}$ the observed time and $\delta = \mathcal{I}(Z \leq C)$ the censoring indicator, as described in the previous section, Section 1.4.2. For simplicity we assume that Z and C are conditionally independent given $\mathbf{X}$ and that the censoring mechanism is non-informative. Our observed data set $\{(\mathbf{x}_i, t_i, \delta_i) : \mathbf{x}_i \in \mathbb{R}^p, t_i \in \mathbb{R}^+, \delta_i \in \{0, 1\}, i = 1, 2, \ldots, n\}$ is an independently and identically distributed random sample from a certain population $(\mathbf{X}, T, \delta)$. Define $\mathcal{C} = \{i : \delta_i = 0\}$ and $\mathcal{D} = \{i : \delta_i = 1\}$ to be the censored and uncensored index sets, respectively.

In the regression setting, the most mathematically tractable models are the relative risk models. These are based on the multiplicative intensity model, whereby the hazard modelled as

$$\lambda(t; \mathbf{x}_i(t)) = \lambda_0(t)\ \psi(\mathbf{x}_i(t); \boldsymbol{\beta}), \tag{1.37}$$

where $\lambda_0(t)$ is the *baseline hazard* function at time t, $\mathbf{x}_i(t) = (x_{i1}(t), \dots, x_{ip}(t))^\top$ is a vector of time-varying covariates belonging to individual i, $\boldsymbol{\beta}$ is a p-dimensional vector of unknown fixed parameters, and ψ is called the *relative risk* function, i.e., $\psi(\mathbf{x}(t); \boldsymbol{\beta}) > 0$ for each $\boldsymbol{\beta}$. Since this model has a nonparametric piece $\lambda_0(\cdot)$ and a parametric piece $\boldsymbol{\beta}$, it is called *semiparametric*.

This model is called the *relative risk* or *proportional hazards* model because there is an unchanging ratio of the hazard rate (or risk of the event) between individuals with parameter values $\mathbf{x}_i$ and $\mathbf{x}_j$.

Different choices for the relative risk function ψ are possible. We will focus here on the most common choice

$$\psi(\mathbf{x}_i(t); \boldsymbol{\beta}) = \exp\{\boldsymbol{\beta}^\top \mathbf{x}_i(t)\} = \exp\{\sum_{j=1}^{p} \beta_j x_{ij}(t)\}, \tag{1.38}$$

which assigns a constant proportional change to the hazard rate to each unit change in the covariate. This regression model is by far the most commonly used survival model in medical applications and is called the *Cox proportional hazards* regression model. A main reason why this model is so popular is that it gives stable estimates of regression coefficients and adjusted survival curves can be obtained for a wide variety of data situations. However, a disadvantage of the Cox proportional hazards models is that it tends to overestimate treatment effects on long survival. This has to do with the fact that its hazard is proportional.

Other relative risk models might be appropriate in certain settings. In the multidimensional-covariate setting we can define the *excess relative risk* model

$$\psi(\mathbf{x}_i(t); \boldsymbol{\beta}) = \prod_{j=1}^{p} (1 + \beta_j x_{ij}(t)). \tag{1.39}$$

This allows each covariate to contribute its own excess relative risk, independent of the others. Alternatively, we can define the *linear relative risk* function

$$\psi(\mathbf{x}_i(t);\boldsymbol{\beta}) = 1 + \boldsymbol{\beta}^\top \mathbf{x}_i(t) = 1 + \sum_{j=1}^{p} \beta_j x_{ij}(t). \tag{1.40}$$

1.4.4 The Likelihood Function

Suppose unit i is observed for a time t_i. If the unit died at t_i, its contribution to the likelihood function is the density at that duration, which can be written as the product of the survivor and hazard functions

$$\mathcal{L}_i(\boldsymbol{\beta}) = f(t_i;\mathbf{x}_i) = S(t_i;\mathbf{x}_i)\,\lambda(t_i;\mathbf{x}_i).$$

If the unit is still alive at t_i, all we know under non-informative censoring is that the lifetime exceeds t_i. The probability of this event is

$$\mathcal{L}_i(\boldsymbol{\beta}) = S(t_i;\mathbf{x}_i),$$

which becomes the contribution of a censored observation to the likelihood.

Note that both types of contribution share the survivor function $S(t_i;\mathbf{x}_i)$, because in both cases the unit lived up to time t_i. A death multiplies this contribution by the hazard $\lambda(t_i;\mathbf{x}_i)$, but a censored observation does not. We can write the two contributions in a single expression. To this end, let δ_i be a death indicator, taking the value one if unit i died and the value zero otherwise, and $\mathcal{R}(t)$ be the risk set right before the time t : $\mathcal{R}(t) = \{j : t_j \geq t\}$. Then the full likelihood function may be written as follows

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\beta}) &= \prod_{i=1}^{n} \mathcal{L}_i(\boldsymbol{\beta}) = \prod_{i\in\mathcal{D}} f(t_i;\mathbf{x}_i) \prod_{i\in\mathcal{C}} S(t_i;\mathbf{x}_i) \\
&= \prod_{i\in\mathcal{D}} \lambda(t_i;\mathbf{x}_i) \prod_{i=1}^{n} S(t_i;\mathbf{x}_i) = \prod_{i=1}^{n} \lambda(t_i;\mathbf{x}_i)^{\delta_i}\, S(t_i;\mathbf{x}_i) \\
&= \prod_{i=1}^{n} \left\{ \frac{\lambda(t_i;\mathbf{x}_i)}{\sum_{j\in\mathcal{R}(t_i)} \lambda(t_i;\mathbf{x}_j)} \right\}^{\delta_i} \left\{ \sum_{j\in\mathcal{R}(t_i)} \lambda(t_i;\mathbf{x}_j) \right\}^{\delta_i} S(t_i;\mathbf{x}_i),
\end{aligned}$$

by using the censoring indicator δ_i.

[29] argued that the first term in this product contained almost all of the information about $\boldsymbol{\beta}$, while the last two terms contained the information about $\lambda_0(\cdot)$, the baseline hazard.

We use the first term to estimate $\boldsymbol{\beta}$, so that we have the *partial* likelihood

$$\begin{aligned}\mathcal{L}_p(\boldsymbol{\beta}) &= \prod_{i=1}^{n} \left\{ \frac{\lambda(t_i; \mathbf{x}_i(t_i))}{\sum_{j \in \mathcal{R}(t_i)} \lambda(t_i; \mathbf{x}_j(t_i))} \right\}^{\delta_i} \\ &= \prod_{i \in \mathcal{D}} \frac{\lambda_0(t_i)\, \psi(\mathbf{x}_i(t_i); \boldsymbol{\beta})}{\sum_{j \in \mathcal{R}(t_i)} \lambda_0(t_i)\, \psi(\mathbf{x}_j(t_i); \boldsymbol{\beta})} \\ &= \prod_{i \in \mathcal{D}} \frac{\psi(\mathbf{x}_i(t_i); \boldsymbol{\beta})}{\sum_{j \in \mathcal{R}(t_i)} \psi(\mathbf{x}_j(t_i); \boldsymbol{\beta})}. \end{aligned} \tag{1.41}$$

The partial likelihood is useful because it involves only the parameters $\boldsymbol{\beta}$, isolating them from the nonparametric (and often less interesting) $\lambda_0(\cdot)$.

1.5 Structure of the Thesis

The structure of the thesis is as follows. In Chapter 2, we extend the dgLARS method to exponential dispersion GLMs with arbitrary link functions. Moreover, this chapter improves the standard PC algorithm used to estimate the dgLARS solution path.

Chapter 3 develops a new estimation method for the dispersion parameter in the exponential dispersion GLMs and proposes an iterative algorithm to stabilize the proposed estimator.

In Chapter 4, we introduce a principled sparse inference methodology for general relative risk survival models, and in last chapter (Chapter 5) we present an implementation of the improved estimator of the dispersion parameter for high-dimensional GLMs, called GRCV estimator, and also an implementation of the iterative algorithm to improve the proposed GRCV estimator to obtain a more stable and accurate estimator. They are implemented in the R-package **`dglars`**.

Chapter 2

Extended dgLARS for Exponential Dispersion GLMs with a New Algorithm

Contents

Abstract

A large class of modelling and prediction problems involve outcomes that belong to an exponential family distribution. Generalized linear models (GLMs) are a standard way of dealing with such situations. Even in high-dimensional feature spaces GLMs can be extended to deal with such situations. Penalized inference approaches, such as the ℓ_1 or SCAD, or extensions of least angle regression, such as dgLARS, have been proposed to deal with GLMs with high-dimensional feature spaces. Although the theory underlying these methods is in principle generic, the implementation has remained restricted to dispersion free models, such as the Poisson and logistic regression models. The aim of this chapter is to extend the differential geometric least angle regression method for high-dimensional GLMs to arbitrary exponential dispersion family distributions with arbitrary link functions. This entails, first, extending the predictor-corrector (PC) algorithm to arbitrary distributions and link functions, and second, proposing a classical estimator of the dispersion parameter. Furthermore, improvements to the computational algorithm lead to an important speed-up of the PC algorithm. Simulations provide supportive evidence concerning the proposed efficient algorithm for estimating coefficients. The resulting method has been implemented in the R-package **`dglars2`** (which will be merged with the original **`dglars`** package) and is shown to be an effective method for inference for arbitrary classes of GLMs.

Keywords: *High-dimensional inference; Generalized linear models; Least angle regression; Predictor-corrector algorithm; Dispersion parameter.*

2.1 Introduction

Nowadays, high-dimensional data problems, where the number of predictors is larger than the sample size, are becoming more common. In such scenarios, it is often sensible to assume that only a small number of predictors contributes to the response, i.e., that the underlying, generating model is sparse. With a sparse model we mean many elements equal to zero. Modern statistical methods for sparse regression models are usually based on using a penalty function to estimate a solution curve embedded in the parameter space and then to find the point that represents the best compromise between sparsity and predictive behaviour of the model. Some important examples are the least absolute shrinkage and selection operator (LASSO) estimator [94], the Smoothly Clipped Absolute Deviation (SCAD) method [31], the Dantzig selector [20], which was extended to generalized linear models (GLMs) in [48], and the MC+ penalty function introduced in [102], among others.

Differently from the methods cited above, [29] introduced a new method to select important variables in a linear regression model called least angle regression (LARS) which was extended to Generalized Linear Models (GLM) in [13] by using the differential geometry. This method, which does not require an explicit penalty function, has been called differential geometric LARS (dgLARS) because it is defined generalizing the geometrical ideas on which LARS is based. As underlined in [13], LARS is a proper likelihood method in its own right: it can be generalized to any model and its success does not depend on the arbitrary match of the constraint and the objective function, as is the case in penalized inference methods. In particular, using the differential geometric characterization of the classical signed Rao score test statistic, dgLARS gains important theoretical properties that are not shared by other methods. From a computational point of view, the dgLARS method essentially consists in the computation of the implicitly defined solution curve. In [13], this problem is solved by using a predictor-corrector (PC) algorithm.

Although the theory of the dgLARS method does not require restrictions on the dispersion parameter, the **dglars** package [9] is restricted to logistic and Poisson regression models, i.e., two specific GLMs with canonical link function and dispersion parameter is equal to one. Furthermore, the authors do not consider the problem of how to estimate the dispersion parameter in a high-dimensional setting. The aim of this chapter is to overcome this restriction and to define dgLARS for any generalized linear model with arbitrary link function. First, we extend the PC algorithm to GLMs with generic link function and unknown dispersion parameter; we also improve the algorithm by proposing a new method to reduce the number of solution points needed to approximate the dgLARS solution curve. As we shall show in the simulation study, the proposed algorithm outperforms the original PC algorithm previously implemented in **`dglars`** package. Second, we explicitly consider the problem of how to do inference on the dispersion parameter and we propose a moment method for estimating the dispersion parameter.

The chapter is organized as follows; In Section 2.2, we introduce the extended dgLARS method by giving some essential clues to the theory underlying a generalized linear model from a differential geometric point of view and present the general case of equations based on the class of the exponential family. In Section 2.3, we propose our improved predictor-corrector algorithm. In Section 2.4, firstly, we consider some model selection strategies that are commonly used to select the tuning parameter; secondly, we propose an estimator for dispersion parameter which can be used during the solution path; and thirdly, we present an estimator of the generalized degree of freedom for a general GLM. In Section 2.5, the simulation study is divided into two parts; first, we examine the performance of the extended dgLARS method, which uses the improved PC algorithm, and two other popular path-estimation methods; second, a comparison in terms of performance between the PC and improved PC algorithms is done. The application and data analysis based on continuous outcome with a canonical link function are described in Section 2.6.

2.2 Differential Geometric LARS for General GLM

The original LARS algorithm [29] defines a coefficient solution path for a linear regression model by sequentially adding variables to the solution curve. To make this section self contained, we briefly review the LARS method. Starting with only the intercept, the LARS algorithm finds the covariate that is most correlated with the response variable and proceeds in this direction by changing its associated linear parameter. The algorithm takes the largest step possible in the direction of this covariate until another covariate has as much correlation with the current residual as the current covariate. At that point the LARS algorithm proceeds in an equiangular direction between the two covariates until a new covariate earns its way into the equally most correlated set. Then it proceeds in the direction in which the residual makes an equal angle with the three covariates, and so on. [13] generalized these notions for GLMs by using differential geometry. The resulting defines a continuous solution path for GLM, with on the extreme of the path the maximum likelihood estimate of the coefficient vector and on the other side the intercept-only estimate. The aim of the method is to define a continuous model path with highest likelihood with the fewest number of variables. The reader interested in more of the differential geometric details of this method and its extensions is referred to [13, 10]. In this section, after a brief overview of GLMs, we derive the equations defining the dgLARS solution curve for a GLM with an arbitrary link function. Furthermore, we explicitly consider the role of the dispersion parameter and we shall show that it acts as a scale parameter of the tuning parameter γ.

2.2.1 An overview of GLMs: Terminology and Notation

Let $\mathbf{Y} = (Y_1, Y_2, \cdots, Y_n)^\top$ be an n-dimensional random vector with independent components. In what follows we shall assume that Y_i is a random variable with probability density function belonging to an exponential dispersion family [50, 51], i.e.,

$$p_{Y_i}(y_i; \theta_i, \phi) = \exp\left\{\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi)\right\}, \quad y_i \in \mathcal{Y}_i \subseteq \mathbb{R}, \tag{2.1}$$

where $\theta_i \in \Theta_i \subseteq \mathbb{R}$ is the canonical parameter, $\phi \in \Phi \subseteq \mathbb{R}^+$ is the dispersion parameter, and $a(.)$, $b(.)$ and $c(.,.)$ are given functions. In the following, we assume that each Θ_i is an open set and $a(\phi) = \phi$. We consider ϕ as an unknown parameter. The expected value of $\mathbf{Y}$ is related to the canonical parameter by $\boldsymbol{\mu} = \{\mu(\theta_1), \cdots, \mu(\theta_n)\}^\top$, where $\mu(\theta_i) = \frac{\partial b(\theta_i)}{\partial \theta_i}$ is called mean value mapping, and the variance of $\mathbf{Y}$ is related to its expected value by the identity Var($\mathbf{Y}$) $= \phi\mathbf{V}(\boldsymbol{\mu})$, where $\mathbf{V}(\boldsymbol{\mu}) = diag\{V(\mu_1), \ldots, V(\mu_n)\}$ is an $n \times n$ diagonal matrix with elements, called the variance functions, $V(\mu_i) = \frac{\partial^2 b(\theta_i)}{\partial \theta_i^2}$. Since μ_i is a reparameterization, model (2.1) can be also denoted as $p_{Y_i}(y_i; \mu_i, \phi)$.

Following [60], a Generalized Linear Model (GLM) is defined by means of a known function $g(\cdot)$, called link function, relating the expected value of each Y_i

to the vector of covariates $\mathbf{x}_i = (1, x_{i1}, \dots, x_{ip})^\top$ by the identity

$$g\{E(Y_i)\} = \eta_i = \mathbf{x}_i^\top \boldsymbol{\beta}$$

where η_i is called the i^{th} linear predictor and $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_p)^\top$ is the vector of regression coefficients. In order to simplify our notation we let $\boldsymbol{\mu}(\boldsymbol{\beta}) = \{\mu_1(\boldsymbol{\beta}), \dots, \mu_n(\boldsymbol{\beta})\}^\top$ where $\mu_i(\boldsymbol{\beta}) = g^{-1}(\mathbf{x}_i^\top \boldsymbol{\beta})$. Therefore, the joint probability density function can be written as $p_{\mathrm{Y}}(\mathbf{y}; \boldsymbol{\mu}(\boldsymbol{\beta}), \phi) = \prod_{i=1}^n p_{Y_i}(y_i; \mu_i(\boldsymbol{\beta}), \phi)$. For the remainder of this chapter we shall use $\ell(\boldsymbol{\beta}, \phi; \mathbf{y}) = \log p_{\mathrm{Y}}(\mathbf{y}; \boldsymbol{\mu}(\boldsymbol{\beta}), \phi)$ as notation for the log-likelihood function. From (2.1), the m^{th} score function is given as

$$\begin{aligned}
\partial_m \ell(\boldsymbol{\beta}, \phi; \mathbf{y}) &= \frac{\partial \ell(\boldsymbol{\beta}, \phi; \mathbf{y})}{\partial \beta_m} \\
&= \phi^{-1} \sum_{i=1}^n \frac{(y_i - \mu_i)}{V(\mu_i)} x_{im} \left(\frac{\partial \mu_i}{\partial \eta_i} \right) \\
&= \phi^{-1}\, \partial_m \ell(\boldsymbol{\beta}; \mathbf{y}), \qquad (2.2)
\end{aligned}$$

where $\mu_i = g^{-1}(\mathbf{x}_i^\top \boldsymbol{\beta})$, and the Fisher Information matrix has terms

$$\begin{aligned}
\mathcal{I}_{mn}(\boldsymbol{\beta}, \phi) &= E[\partial_m \ell(\boldsymbol{\beta}, \phi; \mathbf{y}) \cdot \partial_n \ell(\boldsymbol{\beta}, \phi; \mathbf{y})] \\
&= \phi^{-1} \sum_{i=1}^n \frac{x_{im}\, x_{in}}{V(\mu_i)} \left(\frac{\partial \mu_i}{\partial \eta_i} \right)^2 = \phi^{-1}\, \mathcal{I}_{mn}(\boldsymbol{\beta}), \qquad (2.3)
\end{aligned}$$

Using (2.2) and (2.3), we obtain expressions $r_m(\boldsymbol{\beta}, \phi)$ and $\partial_{mn} \ell(\boldsymbol{\beta}, \phi; \mathbf{y})$ to be used in Sections 2.2.2 and even 3.3.1 (in Chapter 3), respectively, as follows:

$$\begin{aligned}
\partial_{mn} \ell(\boldsymbol{\beta}, \phi; \mathbf{y}) &= \frac{\partial^2 \ell(\boldsymbol{\beta}, \phi; \mathbf{y})}{\partial \beta_m \partial \beta_n} \\
&= \phi^{-1} \sum_{i=1}^n \left\{ x_{im}\, x_{in}\, (y_i - \mu_i) \left[\frac{\partial^2 \theta_i}{\partial \mu_i^2} \cdot \left(\frac{\partial \mu_i}{\partial \eta_i} \right)^2 + \frac{\partial \theta_i}{\partial \mu_i} \cdot \frac{\partial^2 \mu_i}{\partial \eta_i^2} \right] \right. \\
&\left. - \frac{\partial \theta_i}{\partial \mu_i} \cdot \left(\frac{\partial \mu_i}{\partial \eta_i} \right)^2 \right\} \\
&= \phi^{-1} \sum_{i=1}^n \left\{ x_{im}\, x_{in}\, (y_i - \mu_i) \left(\frac{\partial^2 \theta_i}{\partial \mu_i^2} \cdot \left(\frac{\partial \mu_i}{\partial \eta_i} \right)^2 + \frac{\partial \theta_i}{\partial \mu_i} \cdot \frac{\partial^2 \mu_i}{\partial \eta_i^2} \right) \right\} \\
&- \mathcal{I}_{mn}(\boldsymbol{\beta}, \phi) \qquad (2.4)
\end{aligned}$$

where $\frac{\partial \theta_i}{\partial \mu_i} = \frac{1}{V(\mu_i)}$ and $\frac{\partial^2 \theta_i}{\partial \mu_i^2} = -\frac{\partial V(\mu_i)/\partial \mu_i}{V(\mu_i)^2}$. The Rao's score test statistic is given as

$$\begin{aligned} r_m(\boldsymbol{\beta}, \phi) &= \frac{\partial_m \ell(\boldsymbol{\beta}, \phi; \mathbf{y})}{\sqrt{\mathcal{I}_m(\boldsymbol{\beta}, \phi)}} \\ &= \phi^{-1/2} \, \frac{\sum_{i=1}^n \left\{ \frac{(y_i - \mu_i)\; x_{im}}{V(\mu_i)} \cdot \frac{\partial \mu_i}{\partial \eta_i} \right\}}{\left(\sum_{i=1}^n \left\{ \frac{x_{im}^2}{V(\mu_i)} \cdot \left(\frac{\partial \mu_i}{\partial \eta_i} \right)^2 \right\} \right)^{1/2}} \\ &= \phi^{-1/2} \, r_m(\boldsymbol{\beta}) \end{aligned} \tag{2.5}$$

where $\mathcal{I}_m(\boldsymbol{\beta}, \phi) = \mathcal{I}_{mm}(\boldsymbol{\beta}, \phi)$. The Rao's score test statistic helps to define $\rho_m(\boldsymbol{\beta}, \phi)$, the angle between the m^{th} basis function $\partial_m \ell(\boldsymbol{\beta}, \phi; \mathbf{Y})$ and the tangent residual vector $\boldsymbol{r}(\boldsymbol{\beta}, \phi, \mathbf{y}; \mathbf{Y}) = \sum_{i=1}^n (y_i - \mu_i) \, \frac{\partial \ell(\boldsymbol{\beta}, \phi; \mathbf{y})}{\partial \mu_i}$, defined as follows

$$\rho_m(\boldsymbol{\beta}, \phi) = \arccos \left[\frac{r_m(\boldsymbol{\beta}, \phi)}{\|\boldsymbol{r}(\boldsymbol{\beta}, \phi, \mathbf{y}; \mathbf{Y})\|_{p\{\boldsymbol{\mu}(\boldsymbol{\beta})\}}} \right], \tag{2.6}$$

where $\|\cdot\|_{p\{\boldsymbol{\mu}(\boldsymbol{\beta})\}}$ is the norm defined on the tangent space $\mathcal{T}_{p(\boldsymbol{\mu}(\boldsymbol{\beta}))}\mathcal{M}$, where the set $\mathcal{M}$ is a p-dimensional submanifold of the differential manifold $\mathcal{S}$ (for details about the $\mathcal{M}$ and $\mathcal{S}$ sets, see [13]). The angle will be used in Section 2.2.2 to define an extension of the least angle regression [29]. From (2.6), the Rao's score test statistic contains the same information as the angle $\rho_m(\boldsymbol{\beta}, \phi)$. Thereby we can define the dgLARS method with respect to the Rao's score test statistic rather than the angle as respects the smallest angle is equivalent to the largest Rao's score test statistic.

Gamma and Inverse Gaussian GLMs

The binomial, Poisson and Gaussian GLMs are by far the most commonly used, but there are a number of lesser known GLMs which are useful for particular types of data. The Gamma and Inverse Gaussian GLMs are intended for continuous and right-skewed responses. They are double-parameter GLMs and belong to the exponential dispersion (ED) family. The Gamma distribution is a member of the additive ED and the Inverse Gaussian distribution is a member of the reproductive ED [66]. We consider these two dispersion parameter models as follows; For Gamma family, we assume that $Y_i \sim G(\nu, \frac{\mu_i}{\nu})$ so that:

$$f_{Y_i}(y_i; \mu_i, \nu) = \exp\left\{\frac{-y_i \frac{1}{\mu_i} - \log(\mu_i)}{\frac{1}{\nu}} + \nu \log(y_i \nu) - \log(y_i \Gamma(\nu))\right\}, \quad y_i > 0,$$

then $E(Y_i) = -\frac{1}{\theta_i} = \mu_i$ and $\text{Var}(Y_i) = \phi\, V(\mu_i) = \frac{\mu_i^2}{\nu}$, where $\phi^{-1} = \nu$. We consider three of the most commonly used link functions: (i) the canonical link function, "*inverse*", $\eta_i = -\mu_i^{-1}$, (ii) "*log*", $\exp(\eta_i) = \mu_i$, and (iii) "*identity*", $\eta_i = \mu_i$. For Inverse Gaussian family, we assume that $Y_i \sim IG(\mu_i, \lambda)$ so that:

$$f_{Y_i}(y_i; \mu_i, \lambda) = \exp\left\{\frac{y_i(-\frac{1}{2\mu_i^2}) + 1/\mu_i}{1/\lambda} - \frac{\lambda}{2y_i} - \frac{1}{2}\log(\frac{2\pi y_i^3}{\lambda})\right\}, \quad y_i > 0,$$

then $E(Y_i) = \frac{1}{\sqrt{-2\theta_i}} = \mu_i$ and $\text{Var}(Y_i) = \phi\, V(\mu_i) = \frac{\mu_i^3}{\lambda}$, where $\phi^{-1} = \lambda$. We consider four of the most commonly used link functions: (i) the canonical link function, "*inverse-square*", $\eta_i = -0.5\mu_i^{-2}$, (ii) "*inverse*", $\eta_i = -\mu_i^{-1}$, (iii) "*log*", and (iv) "*identity*".

Table 2.1 shows all required equations for obtaining the dgLARS estimator based on the Gamma and Inverse Gaussian models with the most commonly used link functions.

2.2.2 The extended dgLARS Method

[13] showed that the dgLARS estimator follows naturally from a differential geometric interpretation of a GLM, generalizing the LARS method [29] using the angle between scores and tangent residual vector, as defined in (2.6). LARS and dgLARS algorithms define a coefficient solution curve by identifying the most important variables step by step and including them into the model at specific points of the path. The original algorithms took as starting point of the path the model with the intercept only. This is a sensible choice as it makes the model invariant under affine transformations of the response or the covariates. However, the choice of the starting point of the least angle approach can be used to incorporate prior information about which variables are expected to be part of the final model and which ones one does not want to make subject to selection. The extended dgLARS method allows for a set of covariates, possibly including the intercept, that are always part of the model. We define the set of

Table 2.1: Required Equations for obtaining extended dgLARS estimator based on Gamma (G) and Inverse Gaussian (IG) regressions.

Equations	$f_{Y_i}(y_i)$	$g(\mu_i) = \eta_i = \mathbf{x}_i^\top \boldsymbol{\beta}$			
		$-\frac{1}{2\mu_i^2}$ (canonical for IG)	$-\frac{1}{\mu_i}$ (canonical for G)	$\log(\mu_i)$	μ_i
$\partial_m \ell(\boldsymbol{\beta}, \phi; \mathbf{y})$	G	-	$\nu \sum_{i=1}^n (y_i - \mu_i)\, x_{im}$	$\nu \sum_{i=1}^n \frac{(y_i-\mu_i)}{\mu_i}\, x_{im}$	$\nu \sum_{i=1}^n \frac{(y_i-\mu_i)}{\mu_i^2}\, x_{im}$
	IG	$\lambda \sum_{i=1}^n (y_i - \mu_i)\, x_{im}$	$\lambda \sum_{i=1}^n \frac{(y_i-\mu_i)}{\mu_i}\, x_{im}$	$\lambda \sum_{i=1}^n \frac{(y_i-\mu_i)}{\mu_i^2}\, x_{im}$	$\lambda \sum_{i=1}^n \frac{(y_i-\mu_i)}{\mu_i^3}\, x_{im}$
$\partial_{mn} \ell(\boldsymbol{\beta}, \phi; \mathbf{y})$	G	-	$-\nu \sum_{i=1}^n x_{im}\, x_{in}\, \mu_i^2$	$-\nu \sum_{i=1}^n \frac{y_i}{\mu_i}\, x_{im}\, x_{in}$	$-\nu \sum_{i=1}^n \left(\frac{2y_i}{\mu_i^3} - \frac{1}{\mu_i^2} \right) x_{im}\, x_{in}$
	IG	$-\lambda \sum_{i=1}^n x_{im}\, x_{in}\, \mu_i^3$	$-\lambda \sum_{i=1}^n x_{im}\, x_{in}\, y_i$	$-\lambda \sum_{i=1}^n \left(\frac{2y_i}{\mu_i^2} - \frac{1}{\mu_i} \right) x_{im}\, x_{in}$	$-\lambda \sum_{i=1}^n \left(\frac{3y_i}{\mu_i^4} - \frac{2}{\mu_i^3} \right) x_{im}\, x_{in}$
$\mathcal{I}_{mn}(\boldsymbol{\beta}, \phi)$	G	-	$\nu \sum_{i=1}^n x_{im}\, x_{in}\, \mu_i^2$	$\nu \sum_{i=1}^n x_{im}\, x_{in}$	$\nu \sum_{i=1}^n \frac{x_{im}\, x_{in}}{\mu_i^2}$
	IG	$\lambda \sum_{i=1}^n x_{im}\, x_{in}\, \mu_i^3$	$\lambda \sum_{i=1}^n x_{im}\, x_{in}\, \mu_i$	$\lambda \sum_{i=1}^n \frac{x_{im}\, x_{in}}{\mu_i}$	$\lambda \sum_{i=1}^n \frac{x_{im}\, x_{in}}{\mu_i^3}$
$\partial_m \mathcal{I}_n(\boldsymbol{\beta}, \phi)$	G	-	$2\, \nu \sum_{i=1}^n x_{im}\, x_{in}^2\, \mu_i^3$	0	$-2\, \nu \sum_{i=1}^n \frac{x_{im}\, x_{in}^2}{\mu_i^3}$
	IG	$3\, \lambda \sum_{i=1}^n x_{im}\, x_{in}^2\, \mu_i^5$	$\lambda \sum_{i=1}^n x_{im}\, x_{in}^2\, \mu_i^2$	$-\lambda \sum_{i=1}^n \frac{x_{im}\, x_{in}^2}{\mu_i}$	$-3\, \lambda \sum_{i=1}^n \frac{x_{im}\, x_{in}^2}{\mu_i^4}$
$r_m(\boldsymbol{\beta}, \phi)$	G	-	$\sqrt{\nu}\, \frac{\sum_{i=1}^n (y_i-\mu_i)\, x_{im}}{\sqrt{\sum_{i=1}^n x_{im}^2\, \mu_i^2}}$	$\sqrt{\nu}\, \frac{\sum_{i=1}^n\, x_{im}(y_i-\mu_i)/\mu_i}{\sqrt{\sum_{i=1}^n x_{im}^2}}$	$\sqrt{\nu}\, \frac{\sum_{i=1}^n\, x_{im}(y_i-\mu_i)/\mu_i^2}{\sqrt{\sum_{i=1}^n x_{im}^2/\mu_i^2}}$
	IG	$\sqrt{\lambda}\, \frac{\sum_{i=1}^n (y_i-\mu_i)\, x_{im}}{\sqrt{\sum_{i=1}^n x_{im}^2\, \mu_i^3}}$	$\sqrt{\lambda}\, \frac{\sum_{i=1}^n\, x_{im}(y_i-\mu_i)/\mu_i}{\sqrt{\sum_{i=1}^n x_{im}^2\, \mu_i}}$	$\sqrt{\lambda}\, \frac{\sum_{i=1}^n\, x_{im}(y_i-\mu_i)/\mu_i^2}{\sqrt{\sum_{i=1}^n x_{im}^2/\mu_i}}$	$\sqrt{\lambda}\, \frac{\sum_{i=1}^n\, x_{im}(y_i-\mu_i)/\mu_i^3}{\sqrt{\sum_{i=1}^n x_{im}^2/\mu_i^3}}$
$\frac{\partial \mu_i}{\partial \eta_i}$		$(-2\eta_i)^{-1.5} = \mu_i^3$	$\eta_i^{-2} = \mu_i^2$	$\exp(\eta_i) = \mu_i$	1

$i = 1, \ldots, n$
$m, n = 1, \ldots, p$

the *protected variables* $\mathcal{P} = \{a_1^0, \ldots, a_b^0\}$, where $b = |\mathcal{P}| \leq \min(n, p+1)$ and a_j^0 is the index of the j^{th} protected variable. The idea is that variable a_j^0 is supposed to be of interest and should always be contained in the model during the path estimation procedure. The best example of a commonly protected variable is the intercept.

In the path estimation of the coefficients, we treat the protected variables in the set $\mathcal{P}$ differently from the other variables which are not protected, in the sense that the tangent residual vector is always orthogonal to the basis vector $\partial_j \ell(\hat{\boldsymbol{\beta}}(\gamma), \phi; \mathbf{Y})$ for $j \in \mathcal{P}$ at any stage (γ [1]) of the path algorithm $\hat{\boldsymbol{\beta}}(\gamma)$, and thereby by using (2.6) we have $r_{j\in\mathcal{P}}(\hat{\boldsymbol{\beta}}(\gamma), \phi) = \partial_{j\in\mathcal{P}} \ell(\hat{\boldsymbol{\beta}}(\gamma), \phi; \mathbf{Y}) = 0$. This means that at any stage of the path algorithm, the tangent residual vector contains only information on the non-protected variables denoted by $\mathcal{P}^c = \mathcal{A}(\gamma) \cup \mathcal{N}(\gamma)$, where $\mathcal{A}(\gamma) = \{a_1, \ldots, a_{k(\gamma)}\}$ is the *active set* and $\mathcal{N}(\gamma) = (\mathcal{P} \cup \mathcal{A}(\gamma))^c = \{a_1^c, \ldots, a_{h(\gamma)}^c\}$ is the *non-active set*. The numbers $k(\gamma) = |\mathcal{A}(\gamma)|$ and $h(\gamma) = |\mathcal{N}(\gamma)|$ are the number of included and non-included variables, respectively, in the model at location γ. Thus, we have $p + 1 = b + k(\gamma) + h(\gamma)$.

Let $\hat{\boldsymbol{\beta}}_{\mathbf{0}} = (\hat{\boldsymbol{\beta}}_{\mathcal{P}}, 0, \ldots, 0)^\top$ be the starting point, where $\hat{\boldsymbol{\beta}}_{\mathcal{P}} = (\hat{\beta}_{a_1^0}, \ldots, \hat{\beta}_{a_b^0})$ is the MLE of the protected variables and a zero for each $p + 1 - b$ non-protected variables $\{a_1, \ldots, a_{k(\gamma)}\} \cup \{a_1^c, \ldots, a_{h(\gamma)}^c\}$. Since at the beginning ($\gamma = \gamma_{max}$) the active set $\mathcal{A}(\gamma_{max})$ is empty ($k(\gamma_{max}) = 0$), we have $\mathcal{P}^c = \mathcal{N}(\gamma)$ and $h(\gamma_{max}) = p + 1 - b$. For a specified model (the model with the protected variables) with the starting point $\hat{\boldsymbol{\beta}}_{\mathbf{0}}$, we define γ_{max} to be the largest absolute value of the Rao's score statistic at $\hat{\boldsymbol{\beta}}_{\mathbf{0}}$, i.e.,

$$\gamma_{max} = \max_{m\in\mathcal{P}^c} \{|r_m(\hat{\boldsymbol{\beta}}_{\mathbf{0}})|\}.$$

Since the dispersion parameter in (2.2)-(2.6) is equal for any m, we can maximize $|r_{m\in\mathcal{P}^c}(\cdot)|$ (or minimize $\rho_{m\in\mathcal{P}^c}(\cdot)$) instead of $|r_{m\in\mathcal{P}^c}(\cdot, \phi)|$ (or $\rho_{m\in\mathcal{P}^c}(\cdot, \phi)$) in terms of m. The m^{th} variable which has the largest absolute value of $r_{m\in\mathcal{P}^c}(\hat{\boldsymbol{\beta}}_{\mathbf{0}})$ would make an excellent candidate for being included in the model. If we do not have any protected variables, $\hat{\boldsymbol{\beta}}_{\mathbf{0}} = (0, \ldots, 0)^\top$ can be used as the starting point, and in this case, $\boldsymbol{r}(\boldsymbol{\mu}(\mathbf{0}), \mathbf{y}; \mathbf{Y})$ is used to rank the covariates locally.

Before we define the dgLARS method, it can be described using Figure 2.1

[1] $\gamma \geq 0$ is a tuning parameter that controls the size of the coefficients. The increase of γ will shrink the coefficients closer to each other and to zero. In practice, it is usually determined by cross-validation.

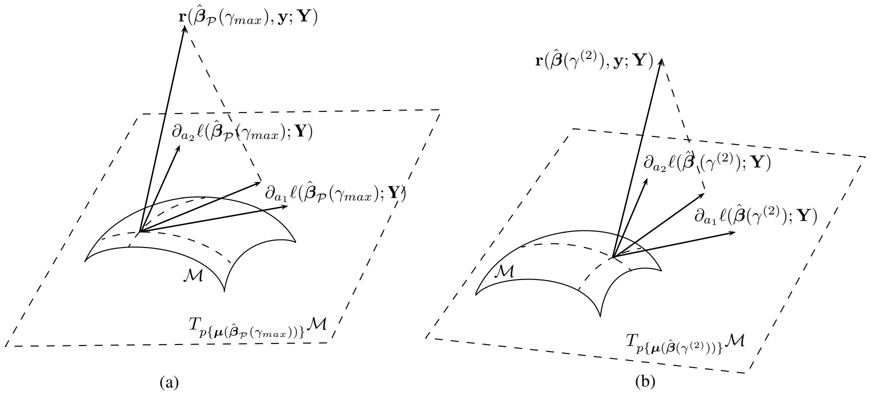

$\mathbf{r}(\hat{\boldsymbol{\beta}}_{\mathcal{P}}(\gamma_{max}), \mathbf{y}; \mathbf{Y})$
$\partial_{a_2}\ell(\hat{\boldsymbol{\beta}}_{\mathcal{P}}(\gamma_{max}); \mathbf{Y})$
$\partial_{a_1}\ell(\hat{\boldsymbol{\beta}}_{\mathcal{P}}(\gamma_{max}); \mathbf{Y})$
$\mathcal{M}$
$T_{p\{\boldsymbol{\mu}(\hat{\boldsymbol{\beta}}_{\mathcal{P}}(\gamma_{max}))\}}\mathcal{M}$
(a)
$\mathbf{r}(\hat{\boldsymbol{\beta}}(\gamma^{(2)}), \mathbf{y}; \mathbf{Y})$
$\partial_{a_2}\ell(\hat{\boldsymbol{\beta}}(\gamma^{(2)}); \mathbf{Y})$
$\partial_{a_1}\ell(\hat{\boldsymbol{\beta}}(\gamma^{(2)}); \mathbf{Y})$
$\mathcal{M}$
$T_{p\{\boldsymbol{\mu}(\hat{\boldsymbol{\beta}}(\gamma^{(2)}))\}}\mathcal{M}$
(b)

Figure 2.1: Differential geometrical description of the LARS algorithm with two covariates: (a) the first covariate X_{a_1} is found and included in the active set, where $\hat{\boldsymbol{\beta}}_{\mathcal{P}} = (\hat{\beta}_{a_1^0}, \ldots, \hat{\beta}_{a_b^0})$; (b) the generalized equiangularity condition (2.7) is satisfied for variables X_{a_1} and X_{a_2}.

in the following way. First the method selects the predictor, say X_{a_1}, whose basis vector $\partial_{a_1}\ell(\hat{\boldsymbol{\beta}}(\gamma_{max}); \mathbf{Y})$ has the smallest angle with the tangent residual vector, and includes it in the active set $\mathcal{A}(\gamma^{(1)}) = \{a_1\}$, where $\gamma^{(1)} = \gamma_{max}$. The solution curve, at this point $\gamma = \gamma^{(1)}$, $\hat{\boldsymbol{\beta}}(\gamma) = (\hat{\boldsymbol{\beta}}_{\mathcal{P}}(\gamma), \hat{\beta}_{a_1}(\gamma), 0, \ldots, 0)^\top$, where $\hat{\boldsymbol{\beta}}_{\mathcal{P}}(\gamma) = (\hat{\beta}_{a_1^0}(\gamma), \ldots, \hat{\beta}_{a_b^0}(\gamma))$, is chosen in such a way that the tangent residual vector is always orthogonal to the basis vectors $\partial_{j\in\mathcal{P}}\ell(\hat{\boldsymbol{\beta}}(\gamma); \mathbf{Y})$ of the tangent space $\mathcal{T}_{p(\boldsymbol{\mu}(\hat{\boldsymbol{\beta}}_{\mathcal{P}}(\gamma)))}\mathcal{M}$, while the direction of the curve $\hat{\boldsymbol{\beta}}(\gamma)$ is defined by the projection of the tangent residual vector onto the basis vector $\partial_{a_1}\ell(\hat{\boldsymbol{\beta}}(\gamma); \mathbf{Y})$. The curve $\hat{\boldsymbol{\beta}}(\gamma)$ continues as defined above until $\gamma = \gamma^{(2)}$, for which there exists a new predictor, say X_{a_2}, that satisfies the equiangularity condition, namely

$$\rho_{a_1}(\hat{\boldsymbol{\beta}}(\gamma^{(2)})) = \rho_{a_2}(\hat{\boldsymbol{\beta}}(\gamma^{(2)})). \tag{2.7}$$

At this point, X_{a_2} is included in the active set $\mathcal{A}(\gamma^{(2)}) = \{a_1, a_2\}$ and the curve $\hat{\boldsymbol{\beta}}(\gamma) = (\hat{\beta}_{a_1^0}(\gamma), \ldots, \hat{\beta}_{a_b^0}(\gamma), \hat{\beta}_{a_1}(\gamma), \hat{\beta}_{a_2}(\gamma), 0, \ldots, 0)^\top$ continues, such that the tangent residual vector is always orthogonal to the basis vectors $\partial_{j\in\mathcal{P}}\ell(\hat{\boldsymbol{\beta}}(\gamma); \mathbf{Y})$ and with direction defined by the tangent residual vector that bisects the angle between $\partial_{a_1}\ell(\hat{\boldsymbol{\beta}}(\gamma); \mathbf{Y})$ and $\partial_{a_2}\ell(\hat{\boldsymbol{\beta}}(\gamma); \mathbf{Y})$, as shown on the right side of Figure 2.1.

The extended dgLARS solution curve, which is denoted by $\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma) \subset \mathbb{R}^{b+k(\gamma)}$ where $\gamma \in [0, \gamma^{(1)}]$ and $0 \leqslant \gamma^{(p-b+1)} \leqslant \cdots \leqslant \gamma^{(2)} \leqslant \gamma^{(1)}$, is defined in the following way: for any $\gamma \in (\gamma^{(k+1)}, \gamma^{(k)}]$, the extended dgLARS estimator satisfies the following conditions:

$$\begin{aligned}
\mathcal{A}(\gamma) &= \{a_1, a_2, \cdots, a_{k(\gamma)}\}, & & \\
\mathcal{N}(\gamma) &= \{a_1^c, a_2^c, \cdots, a_{h(\gamma)}^c\}, & & \\
|r_{a_i}(\hat{\boldsymbol{\beta}}(\gamma))| &= |r_{a_j}(\hat{\boldsymbol{\beta}}(\gamma))| = \gamma\ , & & \forall a_i, a_j \in \mathcal{A}(\gamma)\ , \\
r_{a_i}(\hat{\boldsymbol{\beta}}(\gamma)) &= \upsilon_{a_i} \cdot \gamma, & & \forall a_i \in \mathcal{A}(\gamma)\ , \\
|r_{a_l^c}(\hat{\boldsymbol{\beta}}(\gamma))| &< |r_{a_i}(\hat{\boldsymbol{\beta}}(\gamma))| = \gamma, & & \forall a_l^c \in \mathcal{N}(\gamma) \text{ and } \forall a_i \in \mathcal{A}(\gamma),
\end{aligned} \tag{2.8}$$

where $\upsilon_{a_i} = \text{sign}\{r_{a_i}(\hat{\boldsymbol{\beta}}(\gamma))\}$, $k(\gamma) = |\mathcal{A}(\gamma)| = \#\{m : \hat{\beta}_m(\gamma) \neq 0\}$ and $h(\gamma) = |\mathcal{N}(\gamma)| = \#\{m : \hat{\beta}_m(\gamma) = 0\}$ are the number of covariates in the active and non-active sets, respectively, at location γ. The new covariate is included in the active set at $\gamma = \gamma^{(k+1)}$ when the following condition is satisfied:

$$\exists a_l^c \in \mathcal{N}(\gamma^{(k+1)}) : \quad |r_{a_l^c}(\hat{\boldsymbol{\beta}}(\gamma^{(k+1)}))| = |r_{a_i}(\hat{\boldsymbol{\beta}}(\gamma^{(k+1)}))|\ , \quad \forall a_i \in \mathcal{A}(\gamma^{(k+1)}). \tag{2.9}$$

It shows that the generalized equiangularity condition (2.8) does not depend on the value of the dispersion parameter. As noted before, the original **dglars** package [9] is developed only for Poisson and logistic regression models with canonical link function and $\phi = 1$. Although, the value of the dispersion parameter ϕ does not change the order of the variables included in the active set and also the solution path $\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)$, it is important to take it into consideration that it causes the achieved Rao's score statistic to be shrunk or expanded, since it affects the value of the log-likelihood function $\ell(\boldsymbol{\beta}, \phi; \mathbf{y})$. Therefore, the important point to note here is that the value of the dispersion parameter affects the value of various information criteria such as AIC or BIC, and that is why the estimation of the dispersion parameter is critically important, and will be dealt with in Section 2.4.

It is worth noting that in a high-dimensional setting, $n \leq p$, it is often assumed that the true model, $\mathcal{A}_0 = \{m : \beta_m \neq 0\}$, is sparse, i.e., the number of non-zero coefficients $|\mathcal{A}_0|$ is small (any number less than min$(n-1, p)$). In fact, the maximum number of variables that the dgLARS method can include in the active set is min$(n-1, p)$, namely $|\mathcal{A}| \leq$ min$(n-1, p)$. Hence, when $n \leq p$, the maximum number of non-zero coefficients selected by dgLARS method is min$(n-1, p) = n-1$, namely $|\mathcal{A}| \leq n-1$. It means that, when $n \leq p$, the dgLARS method does not consider the cases in which $n \leq |\mathcal{A}_0|$, thus, we assume that $|\mathcal{A}_0| < n$.

2.3 Improved Predictor-Corrector Algorithm

To compute the solution curve we can use the Predictor-Corrector (PC) algorithm [4], which explicitly finds a series of solutions by using the initial conditions (solutions at one extreme value of the parameter) and continuing to find the adjacent solutions on the basis of the current solutions. From a computational point of view, using the standard PC algorithm leads to an increase in the run times needed for computing the solution curve. In this section we propose an improved version of the PC algorithm to decrease the effects stemming from this problem for computing the solution curve. Using the improved PC algorithm leads to potential computational saving.

The PC method computes the exact coefficients at the values of γ at which the set of non-zero coefficients changes. This strategy yields a more accurate path in an efficient way than alternative methods and provides the exact order of the active set changes. Let us suppose that $k(\gamma)$ predictors are included in the active set $\mathcal{A}(\gamma) = \{a_1, \cdots, a_{k(\gamma)}\}$ at location γ, such that $\gamma \in (\gamma^{(k+1)}, \gamma^{(k)}]$ is a fixed value of the tuning parameter. The corresponding point of the solution curve will be denoted by $\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma) = (\hat{\boldsymbol{\beta}}_{\mathcal{P}}(\gamma), \hat{\beta}_{a_1}(\gamma), \ldots, \hat{\beta}_{a_{k(\gamma)}}(\gamma))^\top$ where $\hat{\boldsymbol{\beta}}_{\mathcal{P}}(\gamma) = (\hat{\beta}_{a_1^0}(\gamma), \ldots, \hat{\beta}_{a_b^0}(\gamma))$ where b is the number of protected variables. Using (2.8), the extended dgLARS solution curve $\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)$ satisfies the relationship

$$|r_{a_1}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))| = |r_{a_2}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))| = \cdots = |r_{a_{k(\gamma)}}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))|, \tag{2.10}$$

and is implicitly defined by the following system of $k(\gamma) + b$ non-linear equations:

$$\begin{cases} \partial_{a_1^0}\ell(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma); \mathbf{y}) & = & 0\,, \\ \vdots & & \vdots \\ \partial_{a_b^0}\ell(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma); \mathbf{y}) & = & 0\,, \\ r_{a_1}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)) & = & \upsilon_{a_1}\gamma\,, \\ \vdots & & \vdots \\ r_{a_{k(\gamma)}}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)) & = & \upsilon_{a_{k(\gamma)}}\gamma\,. \end{cases} \tag{2.11}$$

where $\upsilon_{a_i} = \text{sign}\{r_{a_i}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))\}$.

When $\gamma = 0$, we obtain the maximum likelihood estimates of the subset of the parameter vector $\boldsymbol{\beta}$, denoted by $\hat{\boldsymbol{\beta}}_{\mathcal{A}}$, of the covariates in the active set. The point $\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma^{(k+1)})$ lies on the solution curve joining $\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma^{(k)})$ with $\hat{\boldsymbol{\beta}}_{\mathcal{A}}$. We define $\tilde{\boldsymbol{\varphi}}_{\mathcal{A}}(\gamma) = \boldsymbol{\varphi}_{\mathcal{A}}(\gamma) - \mathbf{v}_{\mathcal{A}}\gamma$, where $\boldsymbol{\varphi}_{\mathcal{A}}(\gamma) = (\partial_{a_1^0}\ell(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma);\mathbf{y}), \ldots, \partial_{a_b^0}\ell(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma);\mathbf{y}), r_{a_1}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)), \cdots, r_{a_{k(\gamma)}}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)))^\top$ and $\mathbf{v}_{\mathcal{A}} = (0, \ldots, 0, v_{a_1}, \ldots, v_{a_{k(\gamma)}})^\top$ starting with b zeros. By differentiating $\tilde{\boldsymbol{\varphi}}_{\mathcal{A}}(\gamma)$ with respect to γ, we can locally approximate the solution curve at $\gamma - \Delta\gamma$ by the following expression

$$\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma - \Delta\gamma) \approx \tilde{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma - \Delta\gamma) = \hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma) - \Delta\gamma \cdot \left(\frac{\partial \boldsymbol{\varphi}_{\mathcal{A}}(\gamma)}{\partial \hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)}\right)^{-1} \mathbf{v}_{\mathcal{A}}\,, \tag{2.12}$$

where $\Delta\gamma \in [0; \gamma - \gamma^{(k+1)}]$ and $\frac{\partial \boldsymbol{\varphi}_{\mathcal{A}}(\gamma)}{\partial \hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)}$ is the Jacobian matrix of the vector function $\boldsymbol{\varphi}_{\mathcal{A}}(\gamma)$ evaluated at the point $\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)$. Equation (2.12) with the step size given in (2.15) are used for the predictor step of the PC algorithm. In the corrector step, $\tilde{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma - \Delta\gamma)$ is used as starting point for the Newton-Raphson algorithm that is used to solve (2.11). For obtaining the Jacobian matrix we need $\partial_m r_n(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma), \phi)$, which is as follows:

$$\begin{aligned}\partial_m r_n(\boldsymbol{\beta}, \phi) &= \frac{\partial\, r_n(\boldsymbol{\beta}, \phi)}{\partial \beta_m} \\ &= \frac{\partial_{mn}\ell(\boldsymbol{\beta}, \phi; \mathbf{y})}{\sqrt{\mathcal{I}_n(\boldsymbol{\beta}, \phi)}} - \frac{1}{2}\,\frac{r_n(\boldsymbol{\beta}, \phi)\,\partial_m \mathcal{I}_n(\boldsymbol{\beta}, \phi)}{\mathcal{I}_n(\boldsymbol{\beta}, \phi)} = \phi^{-1}\,\partial_m r_n(\boldsymbol{\beta}),\end{aligned}$$

where $m, n \in \mathcal{A}$ and

$$\begin{aligned}\partial_m \mathcal{I}_n(\boldsymbol{\beta}, \phi) &= \frac{\partial\, \mathcal{I}_n(\boldsymbol{\beta}, \phi)}{\partial \beta_m} \\ &= \phi^{-1} \sum_{i=1}^{n} \left\{ \frac{x_{im}\, x_{in}^2}{V(\mu_i)} \left(2\,\frac{\partial \mu_i}{\partial \eta_i} \cdot \frac{\partial^2 \mu_i}{\partial \eta_i^2} - \frac{\partial V(\mu_i)/\partial \mu_i}{V(\mu_i)} \left(\frac{\partial \mu_i}{\partial \eta_i}\right)^3 \right) \right\} \\ &= \phi^{-1}\,\partial_m \mathcal{I}_n(\boldsymbol{\beta}).\end{aligned} \tag{2.13}$$

An efficient implementation of the PC method requires a suitable method to compute the smallest step size $\Delta\gamma$ that changes the active set of the non-zero coefficients. Using (2.9), we have a change in the active set when

$$\exists a_j^c \in \mathcal{N}(\gamma): \;\; |r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma - \Delta\gamma))| = |r_{a_i}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma - \Delta\gamma))|, \qquad \forall a_i \in \mathcal{A}(\gamma). \tag{2.14}$$

By expanding $r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))$ in a Taylor series around γ, and observing that the solution curve satisfies (2.11), expression (2.14) can be rewritten in the following way:

$$\exists a_j^c \in \mathcal{N}(\gamma): \;\; \left| r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)) - \frac{dr_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))}{d\gamma}\Delta\gamma \right| \approx \gamma - \Delta\gamma, \quad \forall a_i \in \mathcal{A}(\gamma)$$

where $\Delta\gamma \in [0; \gamma]$, then

$$r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)) \approx \frac{dr_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))}{d\gamma}\Delta\gamma + (\gamma - \Delta\gamma) = -\Delta\gamma\left(1 - \frac{dr_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))}{d\gamma}\right) + \gamma,$$

or

$$r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)) \approx \frac{dr_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))}{d\gamma}\Delta\gamma - (\gamma - \Delta\gamma) = \Delta\gamma\left(1 + \frac{dr_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))}{d\gamma}\right) - \gamma,$$

so that, they give two values for $\Delta\gamma$, namely

$$\Delta\gamma_1 = \frac{\gamma - r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))}{1 - \frac{dr_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))}{d\gamma}} \qquad \text{or} \qquad \Delta\gamma_2 = \frac{\gamma + r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))}{1 + \frac{dr_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))}{d\gamma}},$$

where

$$\begin{aligned}
\frac{dr_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))}{d\gamma} &= \frac{d}{d\gamma}\left(\frac{\partial_{a_j^c}\ell(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma);\mathbf{y})}{\sqrt{\mathcal{I}_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))}}\right) \\
&= \frac{\frac{d\,\partial_{a_j^c}\ell(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma);\mathbf{y})}{d\gamma} \cdot \mathcal{I}_{a_j^c}^{1/2}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)) - \partial_{a_j^c}\ell(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma);\mathbf{y}) \cdot \frac{d\,\mathcal{I}_{a_j^c}^{1/2}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))}{d\gamma}}{\mathcal{I}_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))} \\
&= \mathcal{I}_{a_j^c}^{-1/2}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)) \cdot \langle \partial_{a_i a_j^c}\ell(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma);\mathbf{y}), \frac{d\hat{\beta}_{a_i}(\gamma)}{d\gamma} \rangle
\end{aligned}$$

$$
\begin{aligned}
&- \frac{1}{2}\, r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)) \cdot \mathcal{I}_{a_j^c}^{-1}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)) \cdot \langle \partial_{a_i} \mathcal{I}_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)), \frac{d\hat{\beta}_{a_i}(\gamma)}{d\gamma} \rangle \\
&= \frac{\sum_{a_i \in \mathcal{A}(\gamma)} \left\{ \partial_{a_i a_j^c} \ell(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma); \mathbf{y}) \cdot \frac{d\hat{\beta}_{a_i}(\gamma)}{d\gamma} \right\}}{\mathcal{I}_{a_j^c}^{1/2}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))} \\
&- \frac{1}{2}\, \frac{r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)) \cdot \sum_{a_i \in \mathcal{A}} \left\{ \partial_{a_i} \mathcal{I}_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)) \cdot \frac{d\hat{\beta}_{a_i}(\gamma)}{d\gamma} \right\}}{\mathcal{I}_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))} \\
&= \sum_{a_i \in \mathcal{A}(\gamma)} \left\{ \frac{d\hat{\beta}_{a_i}(\gamma)}{d\gamma} \left[\frac{\partial_{a_i a_j^c} \ell(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma); \mathbf{y})}{\mathcal{I}_{a_j^c}^{1/2}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))} - \frac{1}{2}\, \frac{r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)) \cdot \partial_{a_i} \mathcal{I}_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))}{\mathcal{I}_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))} \right] \right\},
\end{aligned}
$$

where $\langle \cdot, \cdot \rangle$ is an inner product, $\partial_{a_i} \mathcal{I}_{a_j^c}(\boldsymbol{\beta})$ is given by (2.13), and $\frac{d\hat{\beta}_{a_i}(\gamma)}{d\gamma}$ is an element of the matrix of $\frac{d\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)}{d\gamma} = \left(\frac{\partial \boldsymbol{\varphi}_{\mathcal{A}}(\gamma)}{\partial \hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)} \right)^{-1} \mathbf{v}_{\mathcal{A}}$. For each $a_j^c \in \mathcal{N}(\gamma)$ we have a value for $\Delta\gamma^{a_j^c}$ as follows

$$
\Delta\gamma^{a_j^c} = \begin{cases} \Delta\gamma_1 & \text{if } \; 0 \le \Delta\gamma_1 \le \gamma; \\ \Delta\gamma_2 & \text{if } \; o.w. \end{cases}
$$

and from the set of $\Delta\gamma^{a_j^c}$s, $\{\Delta\gamma^{a_j^c},\ a_j^c \in \mathcal{N}(\gamma)\}$, we consider the smallest value of this set as a optimal value for the step size. It can be shown by the following expression

$$
\Delta\gamma^{opt} = \min \left\{ \Delta\gamma^{a_j^c} \mid a_j^c \in \mathcal{N}(\gamma) \right\}. \tag{2.15}
$$

The main problem of the original PC algorithm is related to the number of arithmetic operations needed to compute the Euler predictor, which requires the inversion of an adequate Jacobian matrix. From a computational point of view, using the PC algorithm leads to an increase in the run times needed to compute the solution curve. To improve the PC algorithm we propose a method to reduce the number of steps, thereby greatly reducing the computational burden because of reducing the number of points of the solution curve.

Since the optimal step size is based on a local approximation, we also include an exclusion step for removing incorrectly included variables in the model. When an incorrect variable is included in the model after the corrector step, we have that there is a non-active variable such that the absolute value of the corresponding Rao score test statistic is greater than γ. To adjust the step size in the case of incorrectly including certain variables in the active set, [13] reduced the optimal step size from the previous step, $\triangle\gamma^{opt}$, by using a small positive constant ε and then the inclusion step is redone until the correct variable is joined to the model. They proposed a half of $\Delta\gamma^{opt}$ for ε as a possible choice. [13, 11] and [9] used a contractor factor cf, which is a fixed value, (i.e., $\gamma_{cf} = \gamma_{old} - \Delta\gamma$, where $\gamma_{old} = \gamma_{new} + \triangle\gamma^{opt}$ and $\Delta\gamma = \Delta\gamma^{opt} \cdot cf$), where $cf = 0.5$ as a default. In this case, this method acts like a *Bisection* method. However, the predicted root, γ_{cf}, may be closer to γ_{new}, or γ_{old}, than the mid-point between them. The poor convergence of the Bisection method as well as its poor adaptability to higher dimensions (i.e., systems of two or more non-linear equations) motivate the use of better techniques. In this case, we apply the method of Regula-Falsi (or False-Position), which always converges, for more details see [76] and [99]. The Regula-Falsi method uses the information about the function, $h(.)$, to arrive at γ_{rf}, while in the case of the Bisection method finding γ is a *static* procedure since for a given γ_{new} and γ_{old}, it gives *identical* γ_{cf}, no matter what the function we wish to solve.

The Regula-Falsi method draws a secant from $h(\gamma_{new})$ to $h(\gamma_{old})$, and estimates the root as where it crosses the γ-axis, so that in our case $h(\gamma) = r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)) - \upsilon_{a_j^c} \cdot \gamma$ where $\upsilon_{a_j^c} = \text{sign}\{r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma_{new}))\}$ and $a_j^c \in \mathcal{N}(\gamma)$. From (2.8), we have that $h(\gamma) = r_{a_i}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)) - \upsilon_{a_i}\gamma = 0$ for all $a_i \in \mathcal{A}(\gamma)$. Indeed, after the corrector step, when there is a non-active variable such that the absolute value of the corresponding Rao score test statistic is greater than γ, we want to find an exact point, γ_{rf}, which is very close or even equal to the true point, called the transition point, that changes the active set, so that at the end, it reduces the number of the points of the solution curve.

For applying the Regula-Falsi method to find the root of the equation $h(\gamma_{rf}) = 0$, let us suppose that k predictors are included in the active set, such that $\gamma_{new} < \gamma^{(k)}$. After the corrector step, when $\exists a_j^c \in \mathcal{N}(\gamma_{new})$ such that $|r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma_{new}))| > \gamma_{new}$, we find an γ_{rf} in the interval $[\gamma_{new}, \gamma_{old}]$, where $\gamma_{old} = \gamma_{new} + \triangle\gamma^{opt}$, which is given by the intersection of the γ-axis and the straight line passing through $(\gamma_{new}, r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma_{new})) - \upsilon_{a_j^c} \cdot \gamma_{new})$ and $(\gamma_{old}, r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma_{old})) - \upsilon_{a_j^c} \cdot \gamma_{old})$. It is easy to

Table 2.2: Pseudo-code of the improved PC algorithm to compute the solution curve defined by the extended dgLARS method for a model with the protected variables.

```
Step  Algorithm
1     First compute $\hat{\boldsymbol{\beta}}_{\mathcal{P}} = (\hat{\beta}_{a_1^0}, \dots, \hat{\beta}_{a_b^0})$
2     $\mathcal{A} \leftarrow \arg\max_{a_j^c \in \mathcal{N}(\gamma)} \{|r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{P}})|\}$ and $\gamma \leftarrow |r_{a_1}(\hat{\boldsymbol{\beta}}_{\mathcal{P}})|$
3     Repeat
4       Use (2.15) to compute $\triangle\gamma^{opt}$ and set $\triangle\gamma \leftarrow \triangle\gamma^{opt}$ and $\gamma \leftarrow \gamma - \triangle\gamma^{opt}$
5       Use (2.12) to compute $\tilde{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)$   (predictor step)
6       Use $\tilde{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)$ as starting point to solve system (2.11)  (corrector step)
7       For all $a_j^c \in \mathcal{N}(\gamma)$ compute $r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))$
8       If $\exists N \subset \mathcal{N}(\gamma)$ such that $\left|r_{a_l^{c*}}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))\right| > \gamma$ for all $a_l^{c*} \in N$, then
9         use (2.16) to compute $\gamma_{rf}^{(l)}$ and set $\gamma_{rf} \leftarrow \max_{l}\{\gamma_{rf}^{(l)}\}$
10        first set $\triangle\gamma \leftarrow \triangle\gamma^{opt} - (\gamma_{rf} - \gamma)$ and then $\gamma \leftarrow \gamma_{rf}$, and go to step 5
11      If $\exists a_j^c \in \mathcal{N}(\gamma)$ such that $\left|r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))\right| = \left|r_{a_i}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))\right|$ for all $a_i \in \mathcal{A}(\gamma)$, then
12        update $\mathcal{A}(\gamma)$ and $\mathcal{N}(\gamma)$
13    Until convergence criterion rule is met
```

verify that the root γ_{rf} is given by

$$\gamma_{rf} = \frac{\gamma_{new}\, r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma_{old})) - \gamma_{old}\, r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma_{new}))}{r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma_{old})) - r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma_{new})) + v_{a_j^c} \cdot (\gamma_{new} - \gamma_{old})}, \qquad \forall a_j^c \in \mathcal{N}(\gamma_{new}). \tag{2.16}$$

Then, we first set $\triangle\gamma = \triangle\gamma^{opt} - (\gamma_{rf} - \gamma_{new})$ and then $\gamma = \gamma_{rf}$, to be able to go to the predictor step.

If at γ_{new} there exists a set $N(\gamma_{new}) \subset \mathcal{N}(\gamma_{new})$ such that $|r_{a_l^{c*}}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma_{new}))| > \gamma_{new}$ for all $a_l^{c*} \in N(\gamma_{new})$, the equation (2.16) gives a vector with an element of $\gamma_{rf}^{(l)}$, so that we consider $\gamma_{rf} = \max_{l}\{\gamma_{rf}^{(l)}\}$, and if $\max_{l}\{\gamma_{rf}^{(l)}\}$ is greater than γ_{old}, then we consider $\gamma_{rf} = \gamma_{old}$. When the Newton-Raphson algorithm does not converge, the step size is reduced by the contractor factor cf, and then the predictor and corrector steps are repeated. In Table 2.2 we report the pseudo-code of the improved PC algorithm that was proposed in this section for a model with the protected variables $\{a_1^0, \dots a_b^0\}$.

In Section 2.5, we examine the performance of the proposed PC algorithm and compare it with the original PC algorithm. In the next section, we consider some model selection strategies to select the tuning parameter, and also we propose an estimator for the dispersion parameter.

2.4 Model Selection

Model selection is a process of seeking the model in a set of candidate models that gives the best balance between model fit and complexity [18]. In the literature, selection criteria are usually classified into two categories: consistent (e.g., the Bayesian information criterion (BIC) [84]) and efficient (e.g., the Akaike information criterion (AIC) [3], and the k-fold cross-validation (CV) [42]). A consistent criterion identifies the true model with a probability that approaches 1 in large samples when a set of candidate models contains the true model. An efficient criterion selects the model so that its average squared error is asymptotically equivalent to the minimum offered by the candidate models when the true model is approximated by a family of candidate models. Detailed discussions on efficiency and consistency can be found in [86, 87], [57], [85], [61], and [8].

[91] shows that the AIC is asymptotically equivalent to Leave-One-Out CV. Both of these criteria are based on the Kullback–Leibler information criteria [56]. While the BIC, which is based on the Bayesian posterior probability, is asymptotically equivalent to v-fold CV, where $v = n[1 - 1/(\log(n) - 1)]$. Actually, it is well-known that CV on the original models behaves somewhere between AIC and BIC, depending on the data splitting ratio [85]. In Section 2.6, we will compare the performance of these three criteria when the extended dgLARS method is involved as a variable selection method. The dgLARS approach involves the choice of a tuning parameter for variable selection. The selection of the tuning parameter γ is critically important because it determines the dimension of the selected model. A proper tuning parameter can improve the efficiency and accuracy for variable selection [22]. As an all-round option, the k-fold CV has always been a popular choice, especially in the early years. In the present chapter, we use the k-fold CV deviance for the extended dgLARS, so that, data are randomly split into k arbitrary equal-sized subsets $L_1, L_2, \dots, L_k$ and each subset L_v, $v = 1, \dots, k$, is used as an validation data set $L_v = (\mathbf{y}_{n_v}^{(v)}, \mathbf{X}_{n_v \times p}^{(v)})$ consisting of

n_v sample points (and its complement L_v^c is the v^{th} training data set consisting of the remaining n_t observations, where $n_v + n_t = n$) to evaluate the performance of each of the models fitted to the remaining $(k-1)/k$ of the data, L_v^c. The unscaled residual deviance $\mathcal{D}(.,.)$ of the predictions on the validation data set L_v is computed and averaged for the k validation subsets;

$$CV(\gamma) = \frac{1}{k}\sum_{v=1}^{k} \mathcal{D}(\mathbf{y}^{(v)}, \hat{\boldsymbol{\mu}}^{(v)}), \tag{2.17}$$

where $\hat{\boldsymbol{\mu}}^{(v)} = g^{-1}(\mathbf{X}^{(v)}\hat{\boldsymbol{\beta}}_{\mathcal{A}_v}(\gamma))$ and $\hat{\boldsymbol{\beta}}_{\mathcal{A}_v}(\gamma)$ is selected by L_v. The idea will be to select the model with the lowest average CV deviance.

Classical information criteria such as the AIC and BIC can also be used. We use the $AIC(\gamma)$ and $BIC(\gamma)$ for the extended dgLARS written as

$$AIC(\gamma) = -2\ell(\boldsymbol{\beta}_{\mathcal{A}}(\gamma), \phi; \mathbf{y}) + 2\left(k(\gamma) + 1\right), \tag{2.18}$$

and

$$BIC(\gamma) = -2\ell(\boldsymbol{\beta}_{\mathcal{A}}(\gamma), \phi; \mathbf{y}) + \log(n)(k(\gamma) + 1), \tag{2.19}$$

where $k(\gamma) = |\mathcal{A}(\gamma)|$ is an appropriate degree of freedom that measures complexity of the model with the tuning parameter γ. As it can be seen, the selection criteria (2.18) and (2.19) rely heavily on the dispersion parameter which has an important impact on them. Since the log-likelihood function $\ell(\boldsymbol{\beta}(\gamma), \phi; \mathbf{y})$ depends on the dispersion parameter, an estimator is needed in order to evaluate these criteria, and as a result $k(\gamma)$ becomes $k(\gamma) + 1$ in the penalty term [101]. In the next section, Section 2.4.1, an moment estimator of the dispersion parameter is presented.

In what follows we will use the selected tuning parameters $\hat{\gamma}_{_{AIC}}$, $\hat{\gamma}_{_{BIC}}$ and $\hat{\gamma}_{_{CV}}$, where

$$\begin{aligned}
\hat{\gamma}_{_{AIC}} &= \underset{\gamma \in \mathbb{R}^+}{\text{argmin}}\; AIC(\gamma),\\
\hat{\gamma}_{_{BIC}} &= \underset{\gamma \in \mathbb{R}^+}{\text{argmin}}\; BIC(\gamma),\\
\hat{\gamma}_{_{CV}} &= \underset{\gamma \in \mathbb{R}^+}{\text{argmin}}\; CV(\gamma).
\end{aligned}$$

2.4.1 Path Estimation of Dispersion Parameter

Since in practice the dispersion parameter ϕ is often unknown, in this chapter we consider ϕ as an unknown parameter which is the same for all Y_i. As we mentioned above, by estimating the dispersion parameter, the solution path $\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)$ is not changed, although the value of the log-likelihood function $\ell(\boldsymbol{\beta}, \phi; \mathbf{y})$ is changed and so considerations about the selection of the optimal model are going to be importantly affected.

There are three classical methods to estimate ϕ: Deviance, Pearson and Maximum Likelihood (ML) estimators. The Deviance estimator is $\hat{\phi}_d = \mathcal{D}(\mathbf{y}, \hat{\boldsymbol{\mu}})/(n-p-1)$, where $\mathcal{D}(\mathbf{y}, \hat{\boldsymbol{\mu}}) = \phi\mathcal{D}(\mathbf{y}, \hat{\boldsymbol{\mu}}, \phi) = -2\phi(\ell(\hat{\boldsymbol{\mu}}, \phi; \mathbf{y}) - \ell(\mathbf{y}, \phi; \mathbf{y}))$ is the unscaled residual deviance. The ML estimator of ϕ, $\hat{\phi}_{mle}$, is the solution of $\partial\ell(\hat{\boldsymbol{\mu}}, \phi; \mathbf{y})/\partial\phi = 0$; For instance, the ML estimators for the Gamma and Inverse Gaussian distributions are $\hat{\phi}_{mle,G} \approx 2\mathcal{D}_{G}/\{n + \sqrt{(n^2 + 2n\mathcal{D}_{G}/3)}\}$ and $\hat{\phi}_{mle,IG} = \mathcal{D}_{IG}/n$, where $\mathcal{D}_{G} = 2\sum_{i=1}^{n}\{\log(\hat{\mu}_i/y_i) + (y_i - \hat{\mu}_i)/\hat{\mu}_i\}$ and $\mathcal{D}_{IG} = \sum_{i=1}^{n}(y_i - \hat{\mu}_i)^2/(y_i\,\hat{\mu}_i^2)$ are the unscaled residual deviance $\mathcal{D}(\mathbf{y}, \hat{\boldsymbol{\mu}})$ for the Gamma and Inverse Gaussian distributions, respectively [23]. [60] note for the Gamma case that both the Deviance ($\hat{\phi}_{d,G}$) and MLE ($\hat{\phi}_{mle,G}$) are sensitive to rounding errors (the difference between the calculated approximation of a number and its exact mathematical value) and model error (deviance from the chosen model) in very small observations and in fact deviance is infinite if any component of $\mathbf{y}$ is zero. Commonly used estimates of the unknown dispersion parameter are the Pearson statistic or the modification of [34], who proposed a first order linear correction term to Pearson's statistic. [60] recommend the use of an approximately unbiased estimate, Pearson method, $\hat{\phi}_{P*} = \frac{\mathcal{X}_P^2}{n-p-1} = \frac{1}{n-p-1}\sum_{i=1}^{n}\frac{(y_i - \hat{\mu}_i)^2}{V(\hat{\mu}_i)}$, where $\mathcal{X}_P^2$ is the Pearson's statistic, $V(.)$ is the variance function, and $\hat{\mu}_i = g^{-1}(\mathbf{x}_i^\top\hat{\boldsymbol{\beta}})$. [62] shows numerically that the choice of estimator can give quite different results in the Gamma case and that $\hat{\phi}_{P*}$ is more robust against model error. Since we can use $\hat{\phi}_{P*}$ only for $n > p$, in the high-dimensional setting ($p \geq n$) we define the dispersion estimator $\hat{\phi}_{P}(\gamma)$ at $\gamma \in [0, \gamma_{max}]$ by the Pearson-like dispersion estimator, as proposed by [101] and [96];

$$\hat{\phi}_{P}(\gamma) = \frac{1}{n - k(\gamma)}\sum_{i=1}^{n}\frac{(y_i - g^{-1}(\mathbf{x}_i^\top\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)))^2}{V(g^{-1}(\mathbf{x}_i^\top\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)))}, \tag{2.20}$$

where $k(\gamma) = |\mathcal{A}(\gamma)| = \#\{j : \hat{\beta}_j(\gamma) \neq 0\}$ such that $\hat{\beta}_j(\gamma)$ is the element of the extended dgLARS estimator $\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)$. Note that, since the estimator $\hat{\phi}_{P}(\gamma)$ depends on γ, we can apply it into the improved PC algorithm in order to calculate the value of the information criteria such as AIC and BIC at each point of the solution path (γ), so that $AIC(\gamma)$ and $BIC(\gamma)$ can be found in (2.18) and (2.19).

2.4.2 Generalized Degree of freedom

The concept of degrees of freedom, which is often used for measurement of model complexity, plays an important role in the theory of linear regression models. This concept is involved in various model selection criteria such as the AIC and BIC. Within the classical theory of linear regression models, it is well known that the degrees of freedom are equal to the number of covariates but for non-linear modelling procedures this equivalence is not satisfied. Generalized degrees of freedom (GDF) is a generic measure of model complexity for any modeling procedure. It accounts for the cost due to both model selection and parameter estimation. For the dgLARS estimator, [13] proposed the notion of generalized degrees of freedom (GDF) to define an adaptive model selection criterion. The authors showed that the cardinality of the active set, $k(\gamma) = |\mathcal{A}(\gamma)|$, is a biased estimator of the generalized degrees of freedom when the model is a logistic regression model, and also proposed a possible estimator of the GDF when it is possible to compute the MLE of the considered GLM. In general, gdf(γ) is a function of the tuning parameter γ, so that gdf$(0) \approx p$. This estimator for a general GLM is given by

$$\widehat{\text{gdf}}(\gamma) = \text{tr}\{J_{\mathcal{A}}^{-1}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))\ I_{\mathcal{A}}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma), \hat{\boldsymbol{\beta}}_{\mathcal{A}}(0))\}, \tag{2.21}$$

where $J_{\mathcal{A}}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))$ is the unscaled observed Fisher Information matrix evaluated at the point $\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)$ which has elements

$$J_{a_j a_k}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)) = \sum_{i=1}^{n} \frac{x_{ia_j}\ x_{ia_k}}{V(\mu_i)} \left\{ \left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2 + (y_i - \mu_i) \left(\frac{\partial V(\mu_i)/\partial \mu_i}{V(\mu_i)} \left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2 - \frac{\partial^2 \mu_i}{\partial \eta_i^2} \right) \right\},$$

and $I_{\mathcal{A}}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma), \hat{\boldsymbol{\beta}}_{\mathcal{A}}(0))$ is an unscaled matrix with elements

$$I_{a_j a_k}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma), \hat{\boldsymbol{\beta}}_{\mathcal{A}}(0)) = \sum_{i=1}^{n} x_{ia_j}\ x_{ia_k}\ \frac{V(\mu_i(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(0)))}{V(\mu_i(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)))^2} \left(\frac{\partial \mu_i(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))}{\partial \eta_i} \right)^2,$$

where $\mu_i(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(0))$ is the maximum likelihood estimate of $\mu_i(\boldsymbol{\beta})$, and $\eta_i = g(\mu_i(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)))$. Note that, the proposed estimator (2.21) does not depend on ϕ. In general, $\widehat{\text{gdf}}(\gamma)$ is different from $k(\gamma)$. It can be used, instead of $k(\gamma)$, in the penalty term of (2.18) and (2.19) to have alternative criteria, namely, $AIC^*(\gamma) = -2\ell(\boldsymbol{\beta}_{\mathcal{A}}(\gamma), \phi; \mathbf{y}) + 2\,(\widehat{\text{gdf}}(\gamma) + 1)$ and $BIC^*(\gamma) = -2\ell(\boldsymbol{\beta}_{\mathcal{A}}(\gamma), \phi; \mathbf{y}) + \log(n)(\widehat{\text{gdf}}(\gamma) + 1)$.

2.5 Simulation Studies

The simulation studies focus on two subjects: The first one is to examine the performance of the extended dgLARS method (which uses the IPC algorithm) and two other popular path-estimation methods (which use the original PC algorithm); The second one is to investigate the performance between the original PC and improved PC algorithms.

In this section, we compare the behavior of the extended dgLARS method obtained by using the improved PC algorithm (by a new package [2]) with two of the most popular sparse GLM packages; **`dglars`**: the dgLARS method obtained by using the PC algorithm [9], and **`glmpath`**: the L_1 Regularization Path method obtained by using the PC algorithm developed by [67]. The **`dglars`** package is available for the binomial and Poisson families with the canonical link function, and the **`glmpath`** package is available for the Gaussian, binomial and Poisson families with the canonical link function. To make the results comparable, the simulation study is based on a *Logistic* regression model (binomial family with *logit* link), with sample size $n = (50, 200)$ and three different values of p, namely $p = (10, 100, 500)$. The large values of p are useful to study the behavior of the methods in a high dimensional setting. The study is based on three different configurations of the covariance structure of the p predictors, such that $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{n^*}$ are sampled from an $N(\mathbf{0}, \Sigma)$ distribution, where the diagonal elements of Σ are 1 and the off-diagonal elements follow $corr(X_i; X_j) = \rho^{|i-j|}$, where X_i and X_j are the i^{th} and j^{th} covariates respectively, $i \neq j$ and $\rho = (0, 0.5, 0.75)$. Only the first five predictors are used to simulate the binary response variable. The intercept term is equal to one and the non-zero coefficients are equal to two. We simulate $n^* = 100$ data sets and let the algorithms compute the entire path of the coefficient estimates.

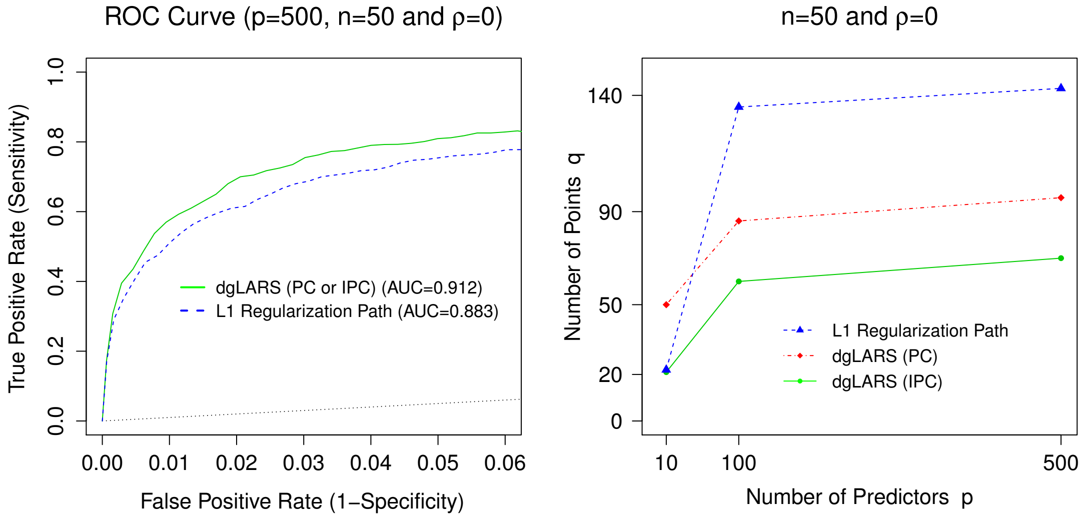
[2] This package is being merged with the original package **`dglars`**.

Table 2.3: Results of the simulation study based on the Logistic regression model; For each p, n and ρ we report the mean number of the points of the entire solution curve (q) and the area under the ROC curve (AUC). Bold values identify the lowest q for each scenario.

			dgLARS (IPC)*		dgLARS (PC)*		glmpath	
p	n	ρ	q	AUC	q	AUC	q	AUC
10	50	0	**21.06**	0.969	49.04	0.969	22.95	0.968
		0.5	**21.96**	0.970	44.59	0.970	27.78	0.968
		0.75	**22.39**	0.927	41.05	0.927	30.53	0.935
	200	0	**17.99**	1.000	46.65	1.000	18.53	1.000
		0.5	**18.61**	1.000	47.13	1.000	19.48	1.000
		0.75	**19.68**	0.999	45.67	0.999	19.68	0.999
100	50	0	**59.66**	0.955	84.87	0.955	106.3	0.944
		0.5	**51.00**	0.969	69.12	0.969	93.42	0.964
		0.75	**42.15**	0.930	56.24	0.930	83.32	0.930
	200	0	**125.5**	1.000	187.2	1.000	392.0	1.000
		0.5	**107.1**	1.000	155.9	1.000	527.1	1.000
		0.75	**96.33**	1.000	143.1	1.000	846.2	1.000
500	50	0	**70.23**	0.912	93.16	0.912	128.7	0.883
		0.5	**62.78**	0.952	77.78	0.952	119.0	0.941
		0.75	**53.12**	0.916	63.91	0.916	111.5	0.905
	200	0	**171.2**	1.000	212.1	1.000	322.7	1.000
		0.5	**139.7**	1.000	174.2	1.000	273.3	1.000
		0.75	**116.9**	1.000	145.9	1.000	248.7	1.000

* The dgLARS (PC) refers to the predictor-corrector implementation of [13], whereas dgLARS (IPC) refers to the improved predictor-corrector algorithm proposed in the present chapter.

In Table 2.3 we report the mean number of the points of the whole solution curve (q) and the area under the receiver operating characteristic (ROC) curve (AUC, average AUC over 100 simulations), as the performance measure. A higher AUC indicates a better performance. The results show that, in the dgLARS method with both the original PC (PC) and improved PC (IPC) algorithms, when the number of predictors is sufficiently large, the mean number of the points of the solution curve (q) decreases as the correlation (ρ) increases. However, for the L_1 Regularization Path method, when $n < p$, q decreases as ρ increases, and when $n > p$ then q increases as ρ decreases. The dgLARS method



(a)

(b)

Figure 2.2: (a) ROC curve, (b) the mean number of the points of the solution curve q computed by the dgLARS method with the PC and IPC algorithms, and the L_1 Regularization Path method from the simulation study based on the Logistic regression model with $n = 50$ and $\rho = 0$.

obtained by using the IPC algorithm, in all scenarios, has the lowest q identified by the bold values, which leads to potential computational saving.

Note that since the dgLARS method obtained by using the improved PC and original PC algorithms compute the same solution curve, their ROC curves and then the values of their AUC are equal, as it can be seen in the corresponding AUC columns of the dgLARS (IPC) and dgLARS (PC). The AUC value of the dgLARS (PC or IPC) method is always greater or equal than the L_1 Regularization Path method. In fact, without depending on p, when the sample size n is small, the dgLARS method has a greater AUC value, and when the sample size is large the AUC value of all methods are equal to one. In other words, when n is sufficiently large without considering the number of predictors ($p > n$ or $p < n$) the value of AUC for the methods is 1.

In Figure 2.2(a) we show the ROC curves ($1 -$ specificity versus sensitivity, computed by averaging over the 100 simulations) corresponding to the dgLARS (by using any of the PC or IPC algorithms) and L_1 Regularization Path methods with $p = 500$, $n = 50$ and $\rho = 0$ based on the Logistic regression model. Also, in

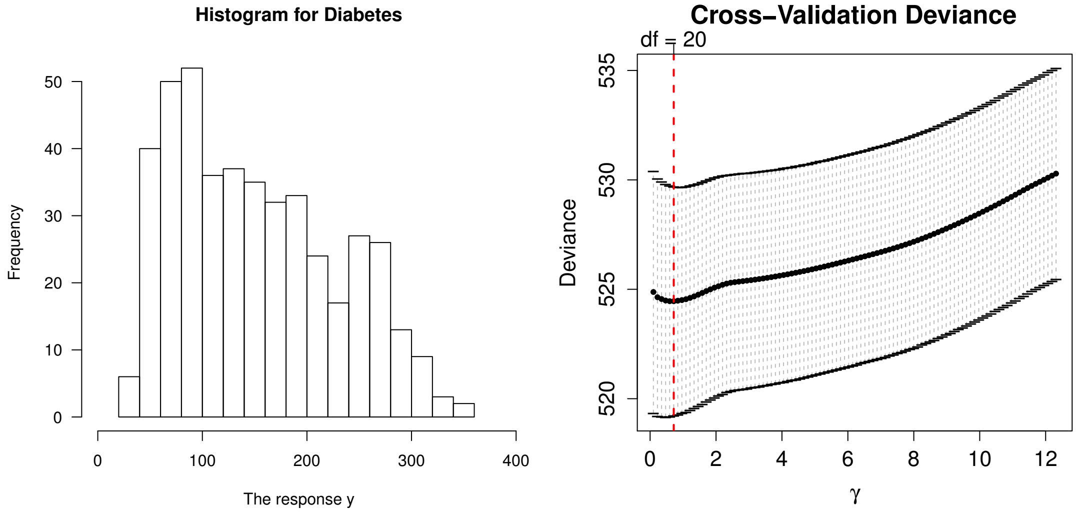
Figure 2.2(b), the mean number of the points of the solution curve (q), computed for these three algorithms, are showed as a function of $p = (10, 100, 500)$ with $n = 50$ and $\rho = 0$. What we mentioned above about q can be clearly seen in this figure.

However, the results related to the number of the covariates included in the final model is not reported for the sake of brevity, we point out that the dgLARS method selects sparser models than the L_1 Regularization Path method. At the end of this section, it should be mentioned that the dgLARS method does not use explicitly a penalized function, so that this method is based on a theory completely different from the L_1 Regularization Path method (L_1-penalized MLE) implemented in the **`glmpath`** package.

2.6 Application to a Diabetes Dataset

In this section we consider the benchmark *diabetes* data used in [29] and [46], among others. The response y is a quantitative measure of disease progression for patients with diabetes one year later. The data includes 10 baseline measurements for each patient, such as *age*, *sex* (gender, which is binary), *bmi* (body mass index), *map* (mean arterial blood pressure), and six blood serum measurements: *ldl* (high-density lipoprotein), *hdl* (low-density lipoprotein), *ltg* (lamotrigine), *glu* (glucose), *tc* (triglyceride) and *tch* (total cholesterol), in addition to 45 interactions and 9 quadratic terms, for a total of 64 variables for each patient, so that this data has $n = 442$ observations on $p = 64$ variables. The aim of the study is to identify which of the covariates are important factors in disease progression. Since the original diabetes data is a low-dimensional data ($p = 64$), we add a thousand noise variables to the original data to also have a high-dimensional dataset with $p = 1064$. These low- and high-dimensional diabetes data can be found in our package.

In the recent literature, variable selection techniques, such as LARS and Spike and Slab, were used in a linear regression model applied to this diabetes data. While we spot from Figure 2.3(a) that, surprisingly, the response y is markedly right-skewed which can arise from a non-normal distribution, for example, a Gamma (or Inverse Gaussian) distribution. Therefore, we fit a Gamma regression model with the *inverse* canonical link function ($\eta_i = -\frac{1}{\mu_i}$) for the (low- and high-dimensional) diabetes data and use the extended dgLARS



(a) (b)

Figure 2.3: (a) Histogram of the response y for the diabetes data. (b) Plot of the 10-fold cross-validation deviance computed for the low-dimensional diabetes data with $p = 64$.

method by means of the proposed algorithm (IPC). Moreover, we let the algorithm to estimate the dispersion parameter in each step by the Pearson moment estimator (2.20).

The results based on the low- and high-dimensional diabetes data is reported in the next two subsections.

2.6.1 Low-dimensional Diabetes Data

For the low-dimensional scenario, when $p < n$, as mentioned above, we consider the benchmark diabetes data ($n = 442, p = 64$) used in [29], which is a dataset included in the **lars** package. At the beginning, we apply several methods such as LARS [29], LASSO [94], Elastic Net [104], L_1 Regularization Path [68], and Spike and Slab [46] by using the **lars** [41], **elasticnet** [103], **glmpath** [67] and **spikeslab** [47] packages for analyzing the low-dimensional diabetes data and then compare them to the results of our method. The top 15 selected predictors for the LARS, LASSO, Elastic Net and L_1 Regularization Path models are exactly the same and are in the order 3, 9, 4, 7, 37, 20, 19, 12, 22, 28, 2, 10, 27, 11 and 30. Moreover, the sequence of the top 15 selected variables for the Spike and Slab algorithm is 3, 9, 4, 7, 2, 20, 37, 19, 27, 12, 22, 11, 30, 10

and 28. We can see that, in all models the top 4 variables (3, 9, 4, and 7) are the same and importantly, all models have the same selected variables just in the different order.

To identify and rank the most important variables, by the dgLARS Gamma regression model, we use three variable selection methods; cross-validation deviance (CV), AIC and BIC. First, we use a tenfold cross-validation to obtain the tuning parameter (γ) of the dgLARS Gamma regression model. Figure 2.3(b) shows the 10-fold cross-validation deviance curve as a function of the tuning parameter (γ), where the vertical red dashed line shows the optimal value of γ, which is $\hat{\gamma}_{_{CV}} = 0.72$, with the number of non-zero estimated coefficients, which is $|\mathcal{A}_{_{CV}}| = 20$, where $\mathcal{A}_{_{CV}} = \mathcal{P} \cup \mathcal{A}(\hat{\gamma}_{_{CV}}) = \{m : \hat{\beta}_m(\hat{\gamma}_{_{CV}}) \neq 0 , m = 0, 1, \ldots, p\}$. Since we consider the protected variables set $\mathcal{P}$ contains only the intercept, $|\mathcal{P}| = b = 1$. Second, by means of the BIC criterion the dgLARS method estimates a Gamma regression model with a high level of sparsity, so that $\gamma_{_{BIC}} = 1.60$ and only with $|\mathcal{A}_{_{BIC}}| = |\mathcal{P} \cup \mathcal{A}(\hat{\gamma}_{_{BIC}})| = 10$ covariates (i.e.,the intercept plus a subset of 9 parameters) are found to influence disease progression. While by using the AIC criterion, $\gamma_{_{AIC}} = 0.47$ and the number of non-zero estimated coefficients is $|\mathcal{A}_{_{AIC}}| = |\mathcal{P} \cup \mathcal{A}(\hat{\gamma}_{_{AIC}})| = 24$.

In Table 2.4, we report the sequence of the top 24 variables and their parameter estimates obtained using the dgLARS Gamma method with the inverse canonical link function, based on all three variable selection methods. In interpreting the table, we note that the selected variables are those having non-zero coefficient estimates. When we compare the results of the dgLARS Gamma method to the previous results obtained using other algorithms, we find out the remarkable results. From Table 2.4 we can see that, the top 4 variables (3, 9, 4, and 7) are the top 4 from the previous results obtained using other algorithms. While all previous algorithms select the covariates $37, 12, 22$, and 27 in the top 15 variables, our proposed method does not select them even among the top 20 variables. Instead, the dgLARS Gamma method select four other variables $60, 46, 18$, and 42, identified by the bold values, in the top 15 variables, namely 3, 9, 4, 7, 20, 60, 2, 46, 18, 10, 42, 28, 11, 19, and 30. As a result, the dgLARS method based on a *Gamma* model, with the canonical link function, finds out that the variables "$hdl : ltg$", "$map : hdl$", "$ltg\hat{}2$" and "$bmi : ltg$" ($60, 46, 18$, and 42) are more important factor in disease progression than the variables "$bmi : map$", "$bmi\hat{}2$", "$age : map$" and "$age : ltg$" ($37, 12, 22$, and 27).

Table 2.4: A list of the top 24 selected variables and their parameter estimates obtained using dgLARS Gamma method (with inverse canonical link, $\eta_i = -\frac{1}{\mu_i}$) for low-dimensional diabetes data. In each criterion, variables selected are those having non-zero coefficient estimates; $|\mathcal{A}_{CV}| = 20$, $|\mathcal{A}_{AIC}| = 24$ and $|\mathcal{A}_{BIC}| = 10$.

	Variable		Coefficient Estimate		
Step	Name	Number	CV	AIC	BIC
1	bmi	3	0.0182	0.0187	0.0171
2	ltg	9	0.0262	0.0278	0.0205
3	map	4	0.0129	0.0136	0.0101
4	hdl	7	-0.0145	-0.0159	-0.0105
5	$age : sex$	20	0.0067	0.0069	0.0042
6	$hdl : ltg$	**60**	0.0046	0.0053	0.0032
7	sex	2	-0.0090	-0.0113	-0.0035
8	$map : hdl$	**46**	0.0040	0.0052	0.0011
9	ltg^2	**18**	-0.0053	-0.0067	-0.0015
10	glu	10	0.0001	-0.0001	0
11	$bmi : ltg$	**42**	-0.0026	-0.0032	0
12	$age : glu$	28	0.0008	0.0010	0
13	age^2	11	0.0021	0.0027	0
14	glu^2	19	0.0016	0.0012	0
15	$sex : map$	30	0.0012	0.0020	0
16	$sex : ltg$	35	0.0007	0.0015	0
17	$sex : bmi$	29	0.0006	0.0015	0
18	$bmi : hdl$	40	0.0004	0.0015	0
19	$age : ldl$	24	-0.0002	-0.0014	0
20	$tch : glu$	63	0	0.0013	0
21	$age : ltg$	27	0	0.0009	0
22	$tc : tch$	52	0	-0.0005	0
23	$sex : glu$	36	0	0.0001	0
24	$map : ltg$	25	0	0	0

One point should be mentioned here is that the number of the points of the solution curve (q) for this data set by using the original PC and improved PC algorithms are 302 and 111, respectively.

2.6.2 High-dimensional Diabetes Data

For a p larger than n setup, we expanded the original diabetes data to become $n = 442$ and $p = 1064$, so that the 1000 additional variables are in reality just noise. We fit a Gamma regression model with the inverse canonical link function for this high-dimensional data and use the extended dgLARS method by means of the improved PC algorithm. The algorithm, in each step, uses the moment estimator of the dispersion parameter given in (2.20).

Like Section 2.6.1, we consider three criteria; Firstly, Figure 2.4(a) shows the 10-fold cross-validation deviance curve in which the optimal value of the tuning parameter is $\hat{\gamma}_{CV} = 1.69$, with the number of non-zero estimated coefficients, which is $|\mathcal{A}_{CV}| = 59$ (with the intercept). Secondly, by the BIC model selection criterion the dgLARS method estimates a Gamma regression model with a high level of sparsity, so that $\hat{\gamma}_{BIC} = 2.99$ and $|\mathcal{A}_{BIC}| = 6$ covariates (i.e.,the intercept plus a subset of 5 parameters) are found to influence disease progression. While by the AIC model selection criterion, $\hat{\gamma}_{AIC} = 1.13$ and the number of non-zero estimated coefficients is $|\mathcal{A}_{AIC}| = 129$.

Figure 2.4 displays the dgLARS Gamma solution path, the Rao score path and the CV, AIC and BIC criteria obtained using the improved PC algorithm and the full data. In addition, we report the sequence of the 25 selected variables and their parameter estimates based on all three criteria in Table 2.5. In interpreting the table, we note that variables starting with "$n.$" are noise variables and the rest are the original variables.

Using Figure 2.4 and Table 2.5 we can see that, while only 4 variables $(3, 9, 4$ and $7)$ have path-profiles that clearly stand out in all three criteria, significantly these variables are the top 4 from our previous analysis obtained using the low-dimensional data (Section 2.6.1). It is interesting that 3 other non-noise variables, "$age : sex$", "$hdl : ltg$" and "sex" (with variable numbers: $20, 60$ and 2) are in the top 25 variables. Regardless of the criteria used, when we inspected the first 100 variables selected by the improved PC algorithm, we found that 12 were from the original 64 variables, and 7 were from the top 25 variable from Table 2.5. This demonstrates stability of the improved PC algorithm even in ultra-high dimensional problems.

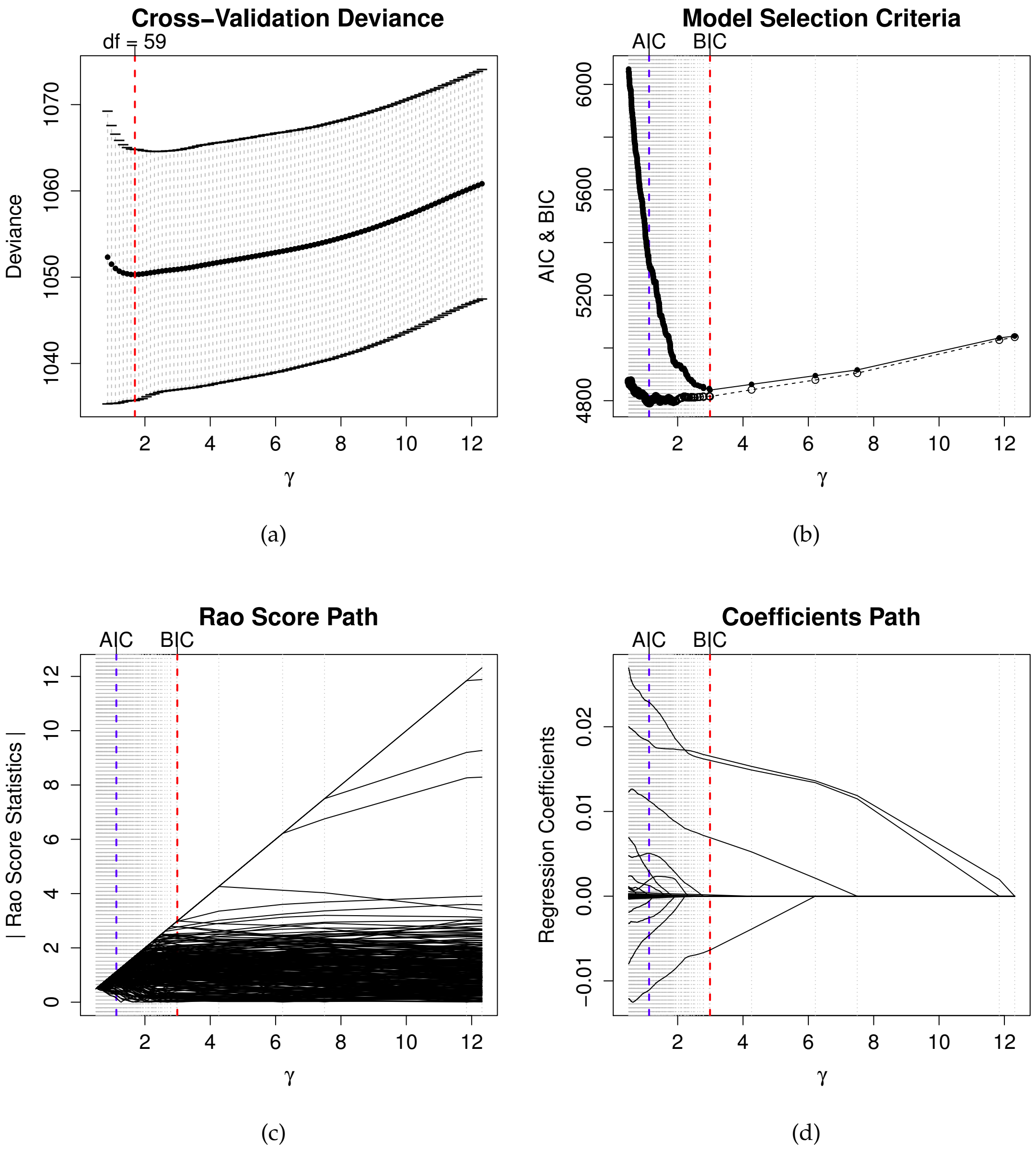
In the meantime, for this data set the number of the points of the solution curve by using the original PC and improved PC algorithms are 1358 and 570, respectively.

Table 2.5: The top 25 variables and their parameter estimates obtained using the dgLARS Gamma method (with inverse canonical link, $\eta_i = -\frac{1}{\mu_i}$) for high-dimensional diabetes data. In each criterion, variables selected are those having non-zero coefficient estimates; $|\mathcal{A}_{CV}| = 59$, $|\mathcal{A}_{AIC}| = 129$ and $|\mathcal{A}_{BIC}| = 6$.

	Variable		Coefficient Estimate		
Step	Name	Number	CV	AIC	BIC
1	bmi	3	0.0174	0.0183	0.0165
2	ltg	9	0.0200	0.0229	0.0160
3	map	4	0.0095	0.0113	0.0069
4	hdl	7	-0.0093	-0.0111	-0.0064
5	$n.312$	376	0.0003	0.0004	0.0002
6	$n.545$	609	0.0002	0.0002	0
7	$n.423$	487	0.0001	0.0001	0
8	$n.543$	607	-0.0001	-0.0002	0
90	$age : sex$	20	0.0037	0.0051	0
10	$n.969$	1033	-0.0001	-0.0001	0
11	$n.62$	126	-0.0001	-0.0002	0
12	$n.347$	411	0.0001	0.0001	0
13	$n.636$	700	-0.0001	-0.0002	0
14	$n.54$	118	0.0001	0.0001	0
15	$n.71$	135	-0.0001	-0.0001	0
16	$n.954$	1018	0.0001	0.0001	0
17	$n.160$	224	0.0001	0.0001	0
18	$hdl : ltg$	60	0.0024	0.0019	0
19	$n.689$	753	-0.0001	-0.0001	0
20	$n.988$	1052	0.0001	0.0001	0
21	$n.337$	401	-0.0001	-0.0002	0
22	$n.612$	676	-0.0001	-0.0001	0
23	$n.404$	468	0.0001	0.0001	0
24	sex	2	-0.0026	-0.0047	0
25	$n.635$	699	0.0160	-0.0001	0

2.7 Conclusions

In this chapter we extended the dgLARS method for a GLM to a larger class of the exponential family, namely the exponential dispersion family (when the dispersion parameter, ϕ, is unknown), and obtained the general framework of the dgLARS estimator for general GLM with general link function. We implemented explicitly the method for Gamma with the canonical and non-canonical link functions. To estimate the dispersion parameter in high-dimensional fea-



(a)

(b)

(c)

(d)

Figure 2.4: (a) Plot of the 10-fold cross-validation deviance, (b) Model selection criteria, (c) Rao score statistics path, (d) Regression coefficients path for the dgLARS Gamma regression model for the high-dimensional diabetes data with $p = 1000$ noise variables.

ture space, we presented a moment estimator which can be used during the solution path. Moreover, we proposed an improved version of the predictor-corrector algorithm to compute the solution curve. The improved PC algorithm allows the dgLARS method to be implemented using less steps, greatly reducing the computational burden because of reducing the number of points of the solution curve. The method was compared well with two well-known methods. The results show that the improved PC algorithm is better and quicker than the original PC algorithm, and now the dgLARS method can be used for a variety of distributions with different types of the canonical and non-canonical link functions. A more stable and accurate estimate of the dispersion parameter for high-dimensional GLMs will be addressed in future work.

Chapter 3

An Estimation Method of Dispersion Parameter for High-dimensional GLMs

Contents

Abstract

In recent years, several methods have been developed to model non-normal outcomes for high-dimensional feature space; for regression models based on the exponential models, important examples are the ℓ_1-penalized estimator for Generalize Linear Models [36] and the dgLARS method [13]. Although the theory underlying these methods is generic, the application is restricted to some specific models such as the Poisson regression model or the logistic regression model in which the dispersion parameter is equal to one ($\phi = 1$). In previous chapter, we extended the least angle regression method for high-dimensional GLMs to arbitrary exponential dispersion family distributions using the IPC algorithm to compute the dgLARS solution curve and presented a classical moment estimator of dispersion parameter. In this chapter, we develop a new method to make high-dimensional inference on the dispersion parameter of the exponential family. Moreover, we propose an iterative algorithm to improve the accuracy of the new proposed method. Simulation studies provide supportive evidence concerning the proposed efficient algorithm for estimating dispersion parameter. The resulting method has been implemented in the R-package **`dglars2`** (which will be merged with the original **`dglars`** package).

Keywords: *High-dimensional Generalized linear models; Differential geometry; Least angle regression; Improved predictor-corrector algorithm; Dispersion Parameter.*

3.1 Introduction

In recent statistical literature, many variable selection techniques for high-dimensional statistical models are based on the penalized likelihood approach. Some important examples are the least absolute shrinkage and selection operator (LASSO) estimator [94], the Smoothly Clipped Absolute Deviation (SCAD) method [31], the Dantzig selector [20], which was extended to generalized linear models (GLMs) in [48], and the MC+ penalty function introduced in [102], among others. Also, the R package **penalized** (Goeman 2010a,b) is a package for fitting possibly high dimensional penalized regression models. In this package, the algorithm uses gradient ascent, and the available models are: Poisson, logistic, linear and cox. Friedman *et al.* (2010a,b), in the **glmnet** package, developed fast algorithms for estimation of generalized linear models with convex penalties. The models include linear regression, two-class logistic regression, and multinomial regression problems while the penalties include ℓ_1 (the LASSO), ℓ_2 (ridge regression) and mixtures of the two (the elastic net). The algorithms use cyclical coordinate descent (CCD) method, computed along a regularization path. In the **glmpath** package, Park and Hastie (2007b), provided a path-following algorithm for ℓ_1 regularized generalized linear models and Cox proportional hazards model. The algorithm uses predictor-corrector (PC) method to compute the entire regularization path for generalized linear models with ℓ_1 penalty. The distribution of y to be used in the model, in this package, must be binomial, Gaussian, or Poisson. For each one, the *canonical* link function is used; logit for binomial, identity for Gaussian, and log for Poisson distribution.

[29] introduced a new method to select important variables in a linear regression model called least angle regression (LARS). [13] proposed a new approach based on the differential geometrical representation of a GLM. The method, which does not require an explicit penalty function, has been called differential geometric LARS (dgLARS) because it is defined generalizing the geometrical ideas on which LARS is based. Although the theory of the dgLARS method does not require restrictions on the dispersion parameter, the **dglars** package [9] restricted to logistic and Poisson regression models, i.e., two specific GLMs with canonical link function and dispersion parameter is equal to one. Furthermore, the authors do not consider the problem of how to estimate the dispersion

parameter in a high-dimensional setting.

As mentioned above, most high-dimensional inferences are limited to binomial, Poisson or Gaussian inference with canonical link function. Therefore, in previous chapter, we considered high-dimensional inference for general GLM with general link function. For this reason, we extended the dgLARS method for a GLM to a larger class of the exponential family, namely the *exponential dispersion family* (EDF), when the dispersion parameter ϕ is unknown, and obtain the general framework of the dgLARS estimator for arbitrary GLM with arbitrary link function. We implemented explicitly the method for Gamma and Inverse Gaussian with a variety of link functions. Moreover, we proposed an improved version of the Predictor-Corrector (PC) algorithm , called Improved Predictor-Corrector (IPC), to compute the solution curve.

In previous chapter it was shown that, although the value of the dispersion parameter ϕ does not change the order of the variables included in the active set and also the solution path $\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)$, it is important to take it into consideration that it causes the achieved Rao's score statistic to be shrunk or expanded, because it affects the value of the log-likelihood function $\ell(\boldsymbol{\beta}, \phi; \mathbf{y})$. Therefore, the value of the dispersion parameter can affect the value of various information criteria such as AIC or BIC, and that is why the estimation of the dispersion parameter is critically important. In previous chapter, we explicitly considered the problem of how to do inference on the dispersion parameter and we proposed a moment estimator for high-dimensional generalized linear model to estimate the dispersion parameter. In this chapter, we deal with the dgLARS method for a high-dimensional exponential dispersion GLMs by using the IPC algorithm and develop a new method to make high-dimensional inference on the dispersion parameter of the exponential family. Moreover, we propose an iterative algorithm to improve the accuracy of the new proposed method.

The chapter is organized as follows; In Section 3.2, we briefly introduce the extended dgLARS method for high-dimensional exponential dispersion GLMs, the IPC algorithm and the moment estimator of dispersion parameter explained in the previous chapter. In Section 3.3, we focus on the estimation of the dispersion parameter and propose a new method to do high-dimensional inference on it, and then we propose an iterative algorithm to achieve a more stable and accurate estimation. In Section 3.4, we investigate how well the new estimator of the dispersion parameter based on the proposed iterative algorithm behaves by using the simulation studies. The application and data analysis based on continuous outcome with a non-canonical link function are described in Section 3.5.

3.2 An overview of the Extended dgLARS

To make this paper self-contained, after an overview of Generalized Linear Models (GLMs), we briefly introduce, firstly, the dgLARS method for high-dimensional exponential dispersion GLMs, secondly, the IPC algorithm, and thirdly the moment estimator of dispersion parameter explained in the previous chapter.

Let $\mathbf{Y} = (Y_1, Y_2, \cdots, Y_n)^T$ be a n-dimensional random vector with independent components. In what follows we shall assume that Y_i is a random variable with probability density function belonging to an exponential dispersion family [50, 51], i.e.,

$$p_{Y_i}(y_i; \theta_i, \phi) = \exp\left\{(y_i\theta_i - b(\theta_i))/a(\phi) + c(y_i, \phi)\right\}, \quad y_i \in \mathcal{Y}_i \subseteq \mathbb{R}, \tag{3.1}$$

where $\theta_i \in \Theta_i \subseteq \mathbb{R}$ is the canonical parameter, $\phi \in \Phi \subseteq \mathbb{R}^+$ is the dispersion parameter, and $a(.)$, $b(.)$ and $c(.,.)$ are given functions. In the following, we assume that each Θ_i is an open set and $a(\phi) = \phi$. We consider ϕ as an unknown parameter. The expected value of $\mathbf{Y}$ is related to the canonical parameter by $\boldsymbol{\mu} = (\mu(\theta_1), \cdots, \mu(\theta_n))^T$, where $\mu(\theta_i) = \frac{\partial b(\theta_i)}{\partial \theta_i}$ is called mean value mapping, and the variance of $\mathbf{Y}$ is related to its expected value by the identity $\text{Var}(\mathbf{Y}) = \phi\mathbf{V}(\boldsymbol{\mu})$, where $\mathbf{V}(\boldsymbol{\mu})$ is an $n \times n$ diagonal matrix with elements $\mathbf{V}(\mu_i) = \frac{\partial^2 b(\theta_i)}{\partial \theta_i^2}$, and is called the variance function. Since μ_i is a reparameterization, the model (3.1) can be also denoted as $p_{Y_i}(y_i; \mu_i, \phi)$. A GLM is defined by means of a known function $g(\cdot)$, called link function, relating the expected value of each Y_i to the vector of covariates $\mathbf{x}_i = (1, x_{i1}, \ldots, x_{ip})^\top$ by the identity $g\{E(Y_i)\} = \eta_i = \mathbf{x}_i^\top\boldsymbol{\beta}$ where η_i is called the i^{th} linear predictor and $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^\top$ is the vector of regression coefficients with the intercept and p parameters. In order to simplify our notation we let $\boldsymbol{\mu}(\boldsymbol{\beta}) = \{\mu_1(\boldsymbol{\beta}), \ldots, \mu_n(\boldsymbol{\beta})\}^\top$ where $\mu_i(\boldsymbol{\beta}) = g^{-1}(\mathbf{x}_i^\top\boldsymbol{\beta})$. For the remainder of this chapter we shall use $\ell(\boldsymbol{\beta}, \phi; \mathbf{y}) = \log p_{\mathrm{Y}}(\mathbf{y}; \boldsymbol{\mu}(\boldsymbol{\beta}), \phi)$ as notation for the log-likelihood function, where $p_{\mathrm{Y}}(\mathbf{y}; \boldsymbol{\mu}(\boldsymbol{\beta}), \phi) = \prod_{i=1}^n p_{Y_i}(y_i; \mu_i(\boldsymbol{\beta}), \phi)$.

The Rao's score test statistic is given as

$$r_m(\boldsymbol{\beta}, \phi) = \frac{\partial_m \ell(\boldsymbol{\beta}, \phi; \mathbf{y})}{\sqrt{\mathcal{I}_m(\boldsymbol{\beta}, \phi)}}, \tag{3.2}$$

where $\mathcal{I}_m(\boldsymbol{\beta}, \phi)$ and $\partial_m \ell(\boldsymbol{\beta}, \phi; \mathbf{y})$ can be found in Chapter 2 (in Section 2.2.1). The Rao's score test statistic helps to define $\rho_m(\boldsymbol{\beta}, \phi)$, the angle between the m^{th} basis function $\partial_m \ell(\boldsymbol{\beta}, \phi; \mathbf{Y})$ and the tangent residual vector $\boldsymbol{r}(\boldsymbol{\beta}, \phi, \mathbf{y}; \mathbf{Y}) = \sum_{i=1}^{n}(y_i - \mu_i)\ \frac{\partial \ell(\boldsymbol{\beta}, \phi; \mathbf{y})}{\partial \mu_i}$, given by

$$\rho_m(\boldsymbol{\beta}, \phi) = \arccos\left[\frac{r_m(\boldsymbol{\beta}, \phi)}{\|\boldsymbol{r}(\boldsymbol{\beta}, \phi, \mathbf{y}; \mathbf{Y})\|_{p(\boldsymbol{\mu}(\boldsymbol{\beta}))}}\right], \tag{3.3}$$

where $\|\cdot\|_{p(\boldsymbol{\mu}(\boldsymbol{\beta}))}$ is the norm defined on the tangent space $\mathcal{T}_{p(\boldsymbol{\mu}(\boldsymbol{\beta}))}\mathcal{M}$, with the set $\mathcal{M}$ is a p-dimensional submanifold of the differential manifold $\mathcal{S}$. For more details the reader is referred to Chapters 1 and 2.

From (3.3), the Rao's score test statistic contains the same information as the angle $\rho_m(\boldsymbol{\beta}, \phi)$. Thereby we can define the dgLARS method with respect to the Rao's score test statistic rather than the angle as respects the smallest angle is equivalent to the largest Rao's score test statistic.

3.2.1 The Extended dgLARS Method

[13] showed that the dgLARS estimator follows naturally from a differential geometric interpretation of a GLM, generalizing the LARS method introduced in [29] using the angle between scores and tangent residual vector as defined in (3.3). The LARS and dgLARS algorithms define a coefficient solution curve by identifying the most important variables step by step and including them into the model at specific points of the path. The original algorithms took as starting point of the path algorithm the model with the intercept only. This is a sensible choice as it makes the model invariant under affine transformations of the response or the covariates. However, the choice of the starting point of the least angle approach can be used to incorporate prior information about which variables are expected to be part of the final model and which ones one does not want to make subject to selection. The extended dgLARS method allows for a set of covariates, possibly including the intercept, that are always part of the model. We define the set of the *protected variables* $\mathcal{P} = \{a_1^0, \ldots, a_b^0\}$, where

$b = |\mathcal{P}| \leq p + 1$ and a_j^0 is the index of the j^{th} protected variable. The idea is that variable a_j^0 is supposed to be of interest and should always be contained in the model during the path estimation procedure. The best example of a commonly protected variable is the intercept.

The extended dgLARS solution curve, which is denoted by $\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma) \subset \mathbb{R}^{b+k(\gamma)}$ where $\gamma \in [0, \gamma^{(1)}]$ and $0 \leqslant \gamma^{(p-b+1)} \leqslant \cdots \leqslant \gamma^{(2)} \leqslant \gamma^{(1)}$, is defined in the following way: For any $\gamma \in (\gamma^{(k+1)}, \gamma^{(k)}]$, the extended dgLARS estimator satisfies the following conditions:

$$\begin{aligned}
\mathcal{A}(\gamma) &= \{a_1, a_2, \cdots, a_{k(\gamma)}\}, & & \\
\mathcal{N}(\gamma) &= (\mathcal{P} \cup \mathcal{A}(\gamma))^c = \{a_1^c, a_2^c, \cdots, a_{h(\gamma)}^c\}, & & \\
|r_{a_i}(\hat{\boldsymbol{\beta}}(\gamma))| &= |r_{a_j}(\hat{\boldsymbol{\beta}}(\gamma))| = \gamma, & & \forall a_i, a_j \in \mathcal{A}(\gamma), \\
|r_{a_l^c}(\hat{\boldsymbol{\beta}}(\gamma))| &< |r_{a_i}(\hat{\boldsymbol{\beta}}(\gamma))| = \gamma, & & \forall a_l^c \in \mathcal{N}(\gamma) \text{ and } \forall a_i \in \mathcal{A}(\gamma),
\end{aligned} \tag{3.4}$$

where $k(\gamma) = |\mathcal{A}(\gamma)| = \#\{m : \hat{\beta}_m(\gamma) \neq 0\}$ and $h(\gamma) = |\mathcal{N}(\gamma)| = \#\{m : \hat{\beta}_m(\gamma) = 0\}$ are the number of covariates in the active and non-active sets, respectively, at location γ. The new covariate is included in the active set at $\gamma = \gamma^{(k+1)}$ when the following condition is satisfied:

$$\exists a_l^c \in \mathcal{N}(\gamma^{(k+1)}) : \quad |r_{a_l^c}(\hat{\boldsymbol{\beta}}(\gamma^{(k+1)}))| = |r_{a_i}(\hat{\boldsymbol{\beta}}(\gamma^{(k+1)}))|\,, \qquad \forall a_i \in \mathcal{A}(\gamma^{(k+1)}). \tag{3.5}$$

It shows that the generalized equiangularity condition (3.4) does not depend on the value of the dispersion parameter.

3.2.2 Improved Predictor-Corrector Algorithm

From a computational point of view, using the PC algorithm leads to an increase in the run times needed for computing the solution curve. In previous chapter, we proposed an improved version of the PC algorithm to decrease the effects stemming from this problem for computing the solution curve. To make this chapter self-contained, we briefly review the IPC algorithm, for more details see Section 2.3 in Chapter 2. Let us suppose that $k(\gamma)$ predictors are included in the active set $\mathcal{A}(\gamma) = \{a_1, \cdots, a_{k(\gamma)}\}$ at location γ, such that $\gamma \in (\gamma^{(k+1)}, \gamma^{(k)}]$ is a fixed value of the tuning parameter. The corresponding point of the solution curve will be denoted by $\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma) = (\hat{\boldsymbol{\beta}}_{\mathcal{P}}(\gamma), \hat{\beta}_{a_1}(\gamma), \ldots, \hat{\beta}_{a_{k(\gamma)}}(\gamma))^T$ where $\hat{\boldsymbol{\beta}}_{\mathcal{P}}(\gamma) = (\hat{\beta}_{a_1^0}(\gamma), \ldots, \hat{\beta}_{a_b^0}(\gamma))$ where b is the number of protected variables.

Using (3.4), the extended dgLARS solution curve $\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)$ satisfies the relationship

$$|r_{a_1}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))| = |r_{a_2}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))| = \cdots = |r_{a_{k(\gamma)}}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))|, \tag{3.6}$$

and is implicitly defined by the following system of $k(\gamma) + b$ non-linear equations:

$$\begin{cases} r_{a_1^0}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)) & = & 0\,, \\ \vdots & & \vdots \\ r_{a_b^0}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)) & = & 0\,, \\ r_{a_1}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)) & = & \upsilon_{a_1}\gamma\,, \\ \vdots & & \vdots \\ r_{a_{k(\gamma)}}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)) & = & \upsilon_{a_{k(\gamma)}}\gamma\,. \end{cases} \tag{3.7}$$

where $\upsilon_{a_i} = \text{sign}\{r_{a_i}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma^{(k)}))\}$. We define $\tilde{\boldsymbol{\varphi}}_{\mathcal{A}}(\gamma) = \boldsymbol{\varphi}_{\mathcal{A}}(\gamma) - \mathbf{v}_{\mathcal{A}}\gamma$, where $\boldsymbol{\varphi}_{\mathcal{A}}(\gamma) = (\partial_{a_1^0}\ell(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma); \mathbf{y}), \ldots, \partial_{a_b^0}\ell(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma); \mathbf{y}), r_{a_1}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)), \cdots, r_{a_{k(\gamma)}}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)))^T$ and $\mathbf{v}_{\mathcal{A}} = (0, \ldots, 0, \upsilon_{a_1}, \ldots, \upsilon_{a_{k(\gamma)}})^T$ starting with b zeros. We can locally approximate the solution curve at $\gamma - \Delta\gamma$ by

$$\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma - \Delta\gamma) \approx \tilde{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma - \Delta\gamma) = \hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma) - \Delta\gamma \cdot \left(\frac{\partial \boldsymbol{\varphi}_{\mathcal{A}}(\gamma)}{\partial \hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)}\right)^{-1} \mathbf{v}_{\mathcal{A}}\,, \tag{3.8}$$

where $\Delta\gamma \in [0; \gamma - \gamma^{(k+1)}]$ and $\frac{\partial \boldsymbol{\varphi}_{\mathcal{A}}(\gamma)}{\partial \hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)}$ is the Jacobian matrix of the vector function $\boldsymbol{\varphi}_{\mathcal{A}}(\gamma)$ evaluated at the point $\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)$. In the corrector step, $\tilde{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma - \Delta\gamma)$ is used as starting point for the Newton-Raphson algorithm that is used to solve (3.7). The smallest step size that changes the active set of the non-zero coefficients is

$$\Delta\gamma^{opt} = \min\left\{\Delta\gamma^{a_j^c} \mid a_j^c \in \mathcal{N}(\gamma)\right\}. \tag{3.9}$$

where $\Delta\gamma^{a_j^c} = \Delta\gamma_1$ if $0 \leq \Delta\gamma_1 \leq \gamma$ and $\Delta\gamma^{a_j^c} = \Delta\gamma_2$ otherwise, where $\Delta\gamma_1$ and $\Delta\gamma_2$ are given in Section 2.3 in the previous chapter. Equation (3.8) with the step size given in (3.9) is used for the predictor step of the IPC algorithm.

Since the optimal step size is based on a local approximation, we also include an exclusion step for removing incorrectly included variables in the model. When an incorrect variable is included in the model after the corrector step, we have that there is a non-active variable such that the absolute value of the corresponding Rao score test statistic is greater than γ. To adjust the step size in the case of incorrectly including certain variables in the active set, we apply the method of Regula-Falsi which uses the information about the function, say $h(.)$, to arrive at γ_{rf}. In our case, the function can be $h(\gamma) = r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)) - \upsilon_{a_j^c} \cdot \gamma$ where $\upsilon_{a_j^c} = \text{sign}\{r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma_{new}))\}$ and $a_j^c \in \mathcal{N}(\gamma)$. While in the case of the Bisection method finding γ is a *static* procedure because for a given γ_{new} and γ_{old}, it gives identical γ_{cf}, no matter what the function we wish to solve. The Regula-Falsi method draws a secant from $h(\gamma_{new})$ to $h(\gamma_{old})$, and estimates the root as where it crosses the γ-axis. Let us obtain a closed form expression for the transition point γ_{rf}. For applying the Regula-Falsi method to find the root of the equation $h(\gamma_{rf}) = 0$, let us suppose that k predictors are included in the active set, such that $\gamma_{new} < \gamma^{(k)}$. After the corrector step, when $\exists a_j^c \in \mathcal{N}(\gamma_{new})$ such that $|r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma_{new}))| > \gamma_{new}$, we find an γ_{rf} in the interval $[\gamma_{new}, \gamma_{old}]$, where $\gamma_{old} = \gamma_{new} + \triangle\gamma^{opt}$, which is given by the intersection of the γ-axis and the straight line passing through $(\gamma_{new}, r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma_{new})) - \upsilon_{a_j^c} \cdot \gamma_{new})$ and $(\gamma_{old}, r_{a_j^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma_{old}) - \upsilon_{a_j^c} \cdot \gamma_{old})$. It is easy to verify that the root γ_{rf} is given by

$$\gamma_{rf} = \frac{\gamma_{new}\, r_{a_j^c}(\hat{\boldsymbol{\beta}}(\gamma_{old})) - \gamma_{old}\, r_{a_j^c}(\hat{\boldsymbol{\beta}}(\gamma_{new}))}{r_{a_j^c}(\hat{\boldsymbol{\beta}}(\gamma_{old})) - r_{a_j^c}(\hat{\boldsymbol{\beta}}(\gamma_{new})) + \upsilon_{a_j^c} \cdot (\gamma_{new} - \gamma_{old})}, \qquad \forall a_j^c \in \mathcal{N}(\gamma_{new}). \tag{3.10}$$

Then, we first set $\triangle\gamma = \triangle\gamma^{opt} - (\gamma_{rf} - \gamma_{new})$ and then $\gamma = \gamma_{rf}$, to be able to go to the predictor step.

3.2.3 Selection of the Tuning Parameter

The dgLARS approach involves the choice of a tuning parameter for variable selection. The selection of the tuning parameter γ is critically important because it determines the dimension of the selected model. A proper tuning parameter can improve the efficiency and accuracy for variable selection. Henceforth, we will use the selected tuning parameters

$$\begin{aligned}\hat{\gamma}_{AIC} &= \underset{\gamma \in \mathbb{R}^+}{\text{argmin}}\; AIC(\gamma),\\ \hat{\gamma}_{BIC} &= \underset{\gamma \in \mathbb{R}^+}{\text{argmin}}\; BIC(\gamma),\end{aligned}$$

$$\hat{\gamma}_{CV} = \underset{\gamma \in \mathbb{R}^+}{\text{argmin}}\, CV(\gamma),$$

where $AIC(\gamma)$, $BIC(\gamma)$ and the k-fold CV deviance for the extended dgLARS can be found in Equations (2.18), (2.19) and (2.17) in Chapter 2. In the next section, a moment estimator of the dispersion parameter is presented.

3.2.4 Moment Estimation of Dispersion

Although, the value of the dispersion parameter ϕ does not change the order of the variables included in the active set and also the solution path $\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)$, it is important to take it into consideration that it causes the achieved Rao's score statistic to be shrunk or expanded, since it affects the value of the log-likelihood function $\ell(\boldsymbol{\beta}, \phi; \mathbf{y})$. Therefore, the important point to note here is that the value of the dispersion parameter affects the value of various information criteria such as AIC or BIC, and so considerations about the selection of the optimal model are going to be importantly affected.

Commonly used estimates of the dispersion parameter are the Pearson statistic. Since we can not use the Pearson method in the high-dimensional setting ($p \geq n$), we define the generalized Pearson dispersion estimator $\hat{\phi}_{P}(\gamma)$ at $\gamma \in [0, \gamma_{max}]$ by

$$\hat{\phi}_{P}(\gamma) = \frac{1}{n - k(\gamma)} \sum_{i=1}^{n} \frac{(y_i - g^{-1}(\mathbf{x}_i^\top \hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)))^2}{V(g^{-1}(\mathbf{x}_i^\top \hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)))}, \tag{3.11}$$

where $k(\gamma) = |\mathcal{A}(\gamma)| = \#\{j : \hat{\beta}_j(\gamma) \neq 0\}$ such that $\hat{\beta}_j(\gamma)$ is the element of the extended dgLARS estimator $\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)$. Note that, since the estimator $\hat{\phi}_{P}(\gamma)$ depends on γ, we can apply it into the improved PC algorithm in order to calculate the value of the information criteria such as AIC and BIC at each point of the solution path (γ).

3.3 A Stable Estimation of the Dispersion Parameter

In Section 3.2.4, we defined a Pearson-type path estimator of the dispersion parameter ϕ. Combined with model selection in Section 3.2.3 this could be used to estimate ϕ overall, but it is known that in shrinkage situations this underestimates ϕ. In this section, we first propose an improved estimator of the dispersion parameter for high-dimensional generalized linear models, called General

Refitted Cross-Validation (GRCV) estimator. Then, we present an algorithm to improve the proposed GRCV estimator to obtain a more stable and accurate estimator based on the GRCV estimator.

3.3.1 General Refitted Cross-Validation Estimator

[30] introduced a two-stage refitted procedure for estimating the dispersion parameter in a linear regression model (variance in linear model) via a data splitting technique called refitted cross-validation (RCV), to attenuate the influence of irrelevant variables with high spurious correlations in the linear models. The RCV estimator is accurate and stable, and insensitive to model selection considerations and the size of the model selected.

For generalized linear models, we propose a general refitted procedure called general refitted cross-validation (GRCV) which is based on four stages. The idea of the GRCV method is as follows; We split the data $(\mathbf{y}_n, \mathbf{X}_{n\times p})$ randomly into two halves $(\mathbf{y}^{(1)}_{n_1}, \mathbf{X}^{(1)}_{n_1\times p})$ and $(\mathbf{y}^{(2)}_{n_2}, \mathbf{X}^{(2)}_{n_2\times p})$, where $n_1 + n_2 = n$. Without loss of generality, for notational simplicity, we assume that the sample size n is even [1], and $n_1 = n_2 = n/2$. In the first stage, our high dimensional variable selection method, extended dgLARS, is applied to these two data sets separately, to estimate whole solution path, which yields $\hat{\boldsymbol{\beta}}_{\mathcal{A}_i}(\gamma)$ selected by $(\mathbf{y}^{(i)}, \mathbf{X}^{(i)})$ where $|\mathcal{A}_i| \le \min(\frac{n}{2} - 1, p)$, $\gamma \in [0, \gamma_{max}]$, and $i = 1, 2$. In the second stage, by using the Pearson-like dispersion estimate (3.11) on the two data sets separately, $\hat{\phi}^{(i)}_P(\gamma)$ where $i = 1, 2$, we determine two small subsets of selected variables $\hat{\mathcal{A}}_i$ where $\hat{\mathcal{A}}_i \subseteq \mathcal{A}_i$ and $i = 1, 2$, by model selection tools such as the AIC, on each data set. Although all three criteria mentioned in the present chapter are available in our package, we recommend using the AIC criterion because the goal is to have a accurate prediction in the third stage [2]. In the third stage, the MLE method is applied to each subset of the data with the variables selected by another subset of the data, namely $(\mathbf{y}^{(2)}, \mathbf{X}^{(2)}_{\hat{\mathcal{A}}_1})$ and $(\mathbf{y}^{(1)}, \mathbf{X}^{(1)}_{\hat{\mathcal{A}}_2})$, to re-estimate the coefficient $\boldsymbol{\beta}$. Since the MLE may not always exist in GLMs, in this stage we propose to use the dgLARS method to estimate the coefficients based on the selected variables, $\hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}_1}(\gamma_0)$ and $\hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}_2}(\gamma_0)$, where γ_0 is close to zero, because the dgLARS estimate $\hat{\boldsymbol{\beta}}_{\mathcal{A}}(0)$ is equal to the MLE of $\boldsymbol{\beta}_{\mathcal{A}}$. Therefore, we apply MLE to the first subset of the data with the variables selected by the sec-

[1] If n is odd, we can consider $|n_1 - n_2| = 1$, and then we randomly apply one of the member of the larger data set to the smaller data set to both have the same dimension, $n_1 = n_2 = n/2$.

ond subset of the data $(\mathbf{y}^{(1)}, \mathbf{X}^{(1)}_{\hat{\mathcal{A}}_2})$ to obtain $\hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}_2}(0)$, and similarly, we use MLE again for the second data set with the set of important variables selected by the first data set $(\mathbf{y}^{(2)}, \mathbf{X}^{(2)}_{\hat{\mathcal{A}}_1})$ to obtain $\hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}_1}(0)$. The refitting in the third stage is fundamental to reduce the influence of the spurious variables in the second stage of variable selection. Finally, in the fourth stage, we estimate ϕ by averaging the two following estimators on the two data sets $(\mathbf{y}^{(2)}, \mathbf{X}^{(2)}_{\hat{\mathcal{A}}_1})$ and $(\mathbf{y}^{(1)}, \mathbf{X}^{(1)}_{\hat{\mathcal{A}}_2})$;

$$\hat{\phi}_1(\hat{\mathcal{A}}_2) = \frac{1}{\frac{n}{2} - |\hat{\mathcal{A}}_2|} \sum_{i=1}^{\frac{n}{2}} \frac{\left(y_i^{(1)} - g^{-1}\left((\mathbf{x}_{i,\hat{\mathcal{A}}_2}^{(1)\top}\hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}_2}(0)\right)\right)^2}{V\left(g^{-1}\left(\mathbf{x}_{i,\hat{\mathcal{A}}_2}^{(1)\top}\hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}_2}(0)\right)\right)},$$

and

$$\hat{\phi}_2(\hat{\mathcal{A}}_1) = \frac{1}{\frac{n}{2} - |\hat{\mathcal{A}}_1|} \sum_{i=1}^{\frac{n}{2}} \frac{\left(y_i^{(2)} - g^{-1}\left(\mathbf{x}_{i,\hat{\mathcal{A}}_1}^{(2)\top}\hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}_1}(0)\right)\right)^2}{V\left(g^{-1}\left(\mathbf{x}_{i,\hat{\mathcal{A}}_1}^{(2)\top}\hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}_1}(0)\right)\right)},$$

where $\mathbf{x}^{(l)}_{i,\hat{\mathcal{A}}_j}$ is the i^{th} row of the l^{th} subset of the data $\mathbf{X}^{(l)}_{\hat{\mathcal{A}}_j}$, $|\hat{\mathcal{A}}_j| = \#\{k : (\hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}_j}(\gamma))_k \neq 0\}$, $\hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}_j}(\gamma)$ is the extended dgLARS estimator at γ, so that $\gamma \in [0, \gamma_{max}]$, and $\hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}_j}(0)$ is the MLE estimator. The GRCV estimator is just the average of these two estimators:

$$\hat{\phi}_{GRCV}(\hat{\mathcal{A}}_1, \hat{\mathcal{A}}_2) = \frac{\hat{\phi}_1(\hat{\mathcal{A}}_2) + \hat{\phi}_2(\hat{\mathcal{A}}_1)}{2}. \tag{3.12}$$

In this procedure, although $\hat{A}_1$ includes some extra unimportant variables besides the important variables, these extra variables will play minor roles when we estimate ϕ by using the second data set along with refitting since they are just some random unrelated variables over the second data set. Furthermore, even when some important variables are missed in the second stage of model selection, they have a good chance of being well approximated by the other variables selected in the second stage to reduce modeling biases. It should be mentioned that, by applying a variable selection tool, the GRCV estimator is sensitive to the model selection tool and the size of the model selected.

In the meantime, we can extend the GRCV technique to get a more accurate estimator. The first extension is to use a k-fold data splitting technique rather than twofold splitting. We can divide the data into k groups and select the model with all groups except one, which is used to estimate the dispersion with refitting. Although there are now more data in the second stage, there are only $n = k$ data points in the third stage for refitting. This means that the number of variables that are selected in the second stage should be much less than $n = k$. That is why we use $k = 2$. The second extension is using a repeated data splitting procedure; since there are many ways to split the data randomly, many GRCV estimators can be obtained. To reduce the influence of the randomness in the data splitting we may take the average of the resulting estimators. For an extensive review of the RCV method, for the linear models, the reader is referred to [32] and [30].

3.3.2 An Iterative GRCV Algorithm

In Section 3.3.1, we proposed the GRCV estimator $\hat{\phi}_{GRCV}$ to estimate ϕ. In this section, we show how the GRCV estimator can be improved to have numerically more stable and accurate behavior. We propose an iterative algorithm which at convergence will also result in more stable and accurate model selection behavior. This algorithm yields a new estimate for ϕ which we call it the MGRCV estimate.

As mentioned in Section 3.3.1 to obtain the GRCV estimate, in the second stage we need to calculate the value of the AIC, BIC or some k-fold CV criteria which depend on the unknown dispersion parameter itself. Hence, the dispersion parameter has to be estimated and for this we used the Pearson-type estimator $\hat{\phi}_P(\gamma)$ given in (3.11) inside the extended dgLARS method during the calculation of the solution path. To improve the accuracy of the estimator $\hat{\phi}_{GRCV}$, we propose an algorithm which repeats the process of finding the GRCV estimate iteratively, such that for the $(k+1)^{\text{th}}$ iteration the k^{th} GRCV estimate ($\hat{\phi}^{(k)}_{GRCV}$) is used to compute the new $(k+1)^{\text{th}}$ GRCV estimate ($\hat{\phi}^{(k+1)}_{GRCV}$), and so on. Therefore, by using this algorithm, the GRCV estimator uses the Pearson-type estimate inside its process only for the first time, and after that the algorithm applies the obtained GRCV estimates instead of the Pearson-type estimate inside the extended dgLARS algorithm. Since the estimate contains some random variation due to the random CV splits, D_1 and D_2, the algorithm will not numerically converge, one in practice simply needs to define a maximal number of iterations T (which should not be too large). Therefore we propose as fi-

Table 3.1: Pseudo code for the iterative algorithm to stabilize the GRCV estimator with T iterations.

```
$pearson \leftarrow 1$
$grcv.vec \leftarrow 0$
$i \leftarrow 1$
while $i \leq T$
   split the data into two random groups: $D_1$ and $D_2$
   apply the extended dgLARS to $D_1$ and $D_2$ separately to obtain whole solution paths $\hat{\boldsymbol{\beta}}_{\mathcal{A}_1}(\gamma)$ and $\hat{\boldsymbol{\beta}}_{\mathcal{A}_2}(\gamma)$   (first stage)
   if $pearson = 1$ then
      use (3.11) to compute $\hat{\phi}_P^{(1)}(\gamma)$ and $\hat{\phi}_P^{(2)}(\gamma)$ for $D_1$ and $D_2$
      use $\hat{\phi}_P^{(1)}(\gamma)$ and $\hat{\phi}_P^{(2)}(\gamma)$ to do model selection* on $D_1$ and $D_2$, respectively, to obtain $\hat{\mathcal{A}}_1$ and $\hat{\mathcal{A}}_2$   (second stage)
      $pearson \leftarrow 0$
   else
      use $\hat{\phi}_{GRCV}(\hat{\mathcal{A}}_1, \hat{\mathcal{A}}_2)$ for model selection* on each $D_1$ and $D_2$ to obtain $\hat{\mathcal{A}}_1$ and $\hat{\mathcal{A}}_2$   (second stage)
   end if
   apply again extended dgLARS to $D_1$ and $D_2$ separately to obtain $\hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}_1}(0)$ and $\hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}_2}(0)$   (third stage)
   use (5.13) to compute $\hat{\phi}_{GRCV}(\hat{\mathcal{A}}_1, \hat{\mathcal{A}}_2)$   (fourth stage)
   $grcv.vec[\,i\,] \leftarrow \hat{\phi}_{GRCV}(\hat{\mathcal{A}}_1, \hat{\mathcal{A}}_2)$
   $i \leftarrow i + 1$
end while
$\hat{\phi}_{MGRCV} \leftarrow$ median( $grcv.vec$ )
use $\hat{\phi}_{MGRCV}$ to do model selection
```

* The AIC or BIC criteria.

nal GRCV estimate the median of the T GRCV estimates, for which we call it MGRCV estimate, $\hat{\phi}_{MGRCV} =$ median$\{\hat{\phi}_{GRCV}^{(1)}, \ldots, \hat{\phi}_{GRCV}^{(T)}\}$. The MGRCV estimate $\hat{\phi}_{MGRCV}$ is more stable and accurate than the first estimate $\hat{\phi}_{GRCV}^{(1)}$. Finally, the overall model selection is performed using $\hat{\phi}_{MGRCV}$.

Table 3.1 shows how this algorithm works. It should be mentioned that, $\hat{\phi}_P^{(1)}(\gamma)$ and $\hat{\phi}_P^{(2)}(\gamma)$ are vectors of the estimates calculated during the solution path, while $\hat{\phi}_{GRCV}(\hat{\mathcal{A}}_1, \hat{\mathcal{A}}_2)$ is a fixed number. In order to investigate the performance of the algorithm we test it on simulated data in Section 3.4.1.

3.4 Simulation Studies

The simulation studies are divided into two parts: the studies on the extended dgLARS method and the GRCV estimator. The first part is devoted to examining the performance of the extended dgLARS method, which uses the improved PC algorithm, and two other popular path-estimation methods. The second part is devoted to investigating the performance of the GRCV estimator based on the iterative GRCV algorithm.

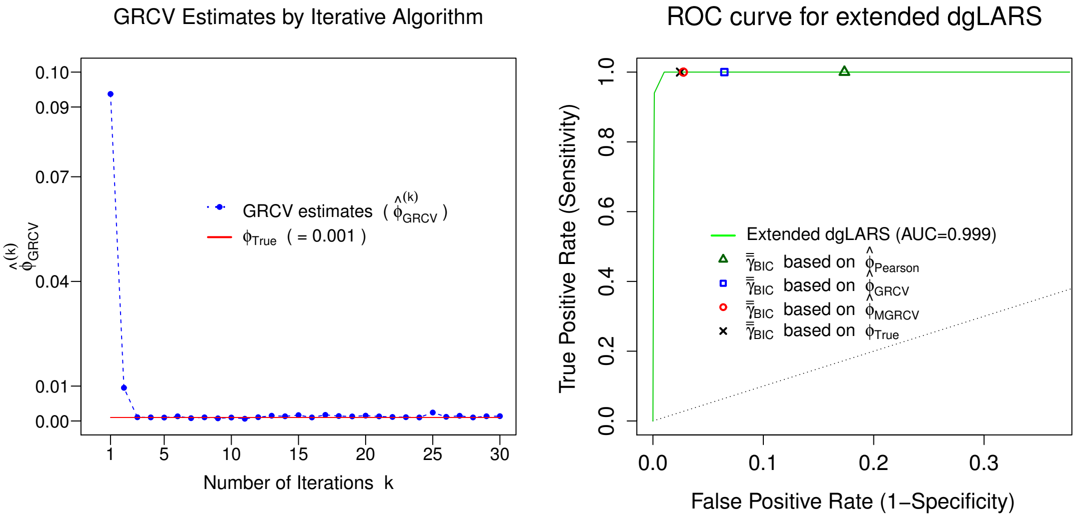
3.4.1 Comparing Dispersion Estimators

This section is divided into two parts; First, in order to show how the GRCV estimator of ϕ and its proposed algorithm work, one simple, but illustrative, example which is a part of a simulation study is presented. Second, we compare the performance of the three dispersion estimators; Pearson ($\hat{\phi}_P$), GRCV ($\hat{\phi}_{GRCV}$) and MGRCV ($\hat{\phi}_{MGRCV}$, the median of the estimators obtained from the iterative GRCV algorithm).

In this simulation study, high-dimensional data are generated according to a Gamma regression model with a non-canonical *log* link, with the shape parameter equal to $\nu = \phi^{-1} = 10^3$ and the scale parameter $\frac{\mu_i}{\nu}$, where $\mu_i = \exp(\mathbf{x}_i^\top \boldsymbol{\beta})$ and $\mathbf{x}_i^\top = (1, x_{i1}, \ldots, x_{ip})$ is as i^{th} row of the design matrix $\mathbf{X}_{n\times(p+1)}$ in which the first column is a column of all ones and the sample size n is 40 and $p = 100$ ($p > n$). We simulate 50 data sets $(\mathbf{y}_1, \mathbf{X}_1), \ldots, (\mathbf{y}_{50}, \mathbf{X}_{50})$, such that $\mathbf{X}_i$ is sampled from an $N(\mathbf{0}, \Sigma)$ distribution, where the diagonal elements of Σ are 1 and the off-diagonal elements are 0, and only the first two predictors ($d = 2$) are used to simulate the response variable $\mathbf{y}_i$,

$$\boldsymbol{\beta} = (\underbrace{0}_{Intercept}, \underbrace{1, 2}_{2}, \underbrace{0, \ldots, 0}_{98}).$$

We show the result of the simulation study in two pictures (a) and (b) in Figure 3.1. Figure 3.1(a) displays the procedure of obtaining the GRCV estimates $\hat{\phi}^{(k)}_{GRCV}$, where $k = (1, 2, \ldots, 30)$, by using the iterative GRCV algorithm, described in Table 3.1, with only the first data set $(\mathbf{y}_1, \mathbf{X}_1)$. The values of the



(a) (b)

Figure 3.1: (a) GRCV estimates, $\hat{\phi}^{(k)}_{GRCV}$, produced by the iterative GRCV algorithm based on a simulated data set from Gamma model. (b) ROC curve of the extended dgLARS method computed by averaging over the 50 simulations along with some selected tuning parameters.

30 GRCV estimates, $\{\hat{\phi}^{(1)}_{GRCV}, \ldots, \hat{\phi}^{(30)}_{GRCV}\}$, computed by the iterative GRCV algorithm, are showed as a function of the number of iterations k. What we mentioned in Section 3.3.2 can be clearly seen in this figure. It can be seen that, after two iterations, the estimate appears to have improved significantly and converges to the true value of the dispersion parameter $\phi_{True} = 0.001$, so that the median of the GRCV estimates, $\hat{\phi}_{MGRCV}$, is 0.0012. It shows that the proposed iterative algorithm can improve the accuracy of the GRCV estimator.

In Figure 3.1(b), we plot the ROC curve (computed by averaging over the 50 simulations) corresponding to the extended dgLARS method and present the area under the ROC curve (average AUC over 50 simulations). As seen in the figure, the average AUC is 0.999 which means that the accuracy of the model selected by the extended dgLARS method is quite high. We have reported this result for low- and high-dimensional datasets in the previous section (in Table 2.3).

Moreover, on the ROC curve in Figure 3.1(b), we also show the average values of the tuning parameter selected by the BIC criterion $\bar{\hat{\gamma}}_{BIC}$ (computed by averaging $\hat{\gamma}_{BIC}$ over 50 simulations) by means of the dispersion estimators $\hat{\phi}_P$, $\hat{\phi}_{GRCV}$ and $\hat{\phi}_{MGRCV}$, and also the true dispersion parameter ϕ_{True}. As [2] noted, when $d \ll n$, where d (is 2 here) is the number of parameters in the true mode, then the BIC criterion is appropriate. That is why we prefer $\hat{\gamma}_{BIC}$ to $\hat{\gamma}_{AIC}$ and $\hat{\gamma}_{CV}$. We use (2.19) in which the number of non-zero estimated coefficients $k(\gamma)$ is used as the degree of freedom to calculate the values of the BIC criterion. The same results are obtained if we use the BIC based on the $\widehat{\text{gdf}}(\gamma)$, because the same final model is identified in both cases (this result is not reported for the sake of brevity).

The point on the ROC curve in the most upper left corner has the highest sensitivity and specificity. A higher sensitivity and specificity indicates superior performance among the tuning parameters obtained by different dispersion estimators. Our results demonstrate that all three final models selected by the chosen tuning parameter $\bar{\hat{\gamma}}_{BIC}$, obtained by the three dispersion estimators $\hat{\phi}_P$, $\hat{\phi}_{GRCV}$ and $\hat{\phi}_{MGRCV}$, have the highest sensitivity (100%), while the specificities of them are 83%, 93% and 97%, respectively. Although these final models selected by means of the three dispersion estimators have a high sensitivity and specificity, the model selected by means of the MGRCV estimator $\hat{\phi}_{MGRCV}$ has the best performance. That means, the dispersion estimator $\hat{\phi}_{MGRCV}$ is a good compromise between specificity and sensitivity. The results also show that our proposed GRCV estimator has a better performance than the Pearson estimator. In addition, since the MGRCV estimate $\hat{\phi}_{MGRCV}$ has a better performance than the GRCV estimate $\hat{\phi}_{GRCV}$, the iterative GRCV algorithm can improve the GRCV estimate to have a more stable and accurate estimate, which proves our claim in Section 3.3.2.

As a result, the results indicate that the extended dgLARS method with $\hat{\phi}_{MGRCV}$ provides a highly specific and sensitive model for high-dimensional GLMs.

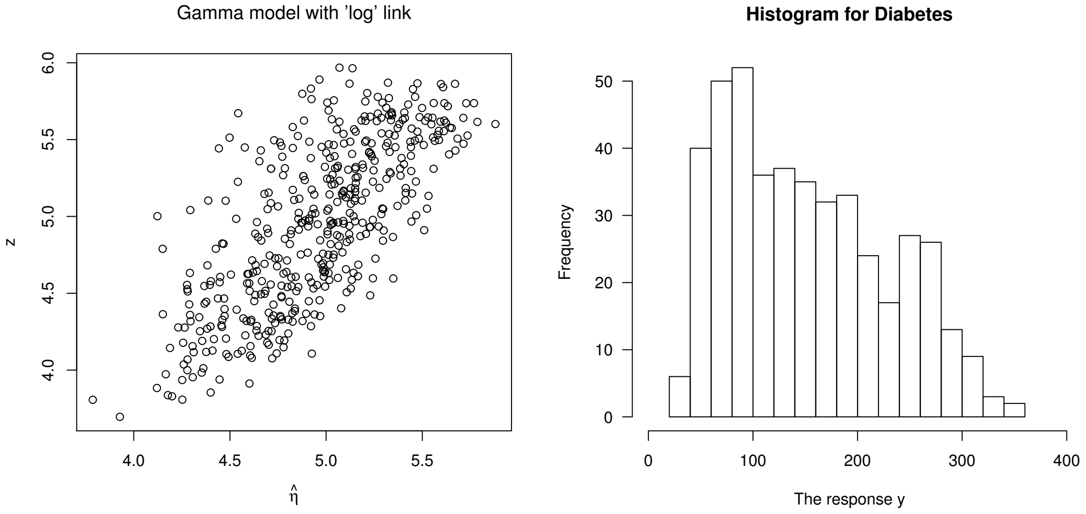
3.5 Application to Real Data

In this section we consider the benchmark *diabetes* data used in [29] and [46], among others. The response y is a quantitative measure of disease progression for patients with diabetes one year later. The data includes 10 baseline measurements for each patient, such as *age*, *sex* (gender, which is binary), *bmi* (body mass index), *map* (mean arterial blood pressure), and six blood serum measurements: *ldl* (high-density lipoprotein), *hdl* (low-density lipoprotein), *ltg* (lamotrigine), *glu* (glucose), *tc* (triglyceride) and *tch* (total cholesterol), in addition to 45 interactions and 9 quadratic terms, for a total of 64 variables for each patient, so that this data has $n = 442$ observations on $p = 64$ variables. The aim of the study is to identify which of the covariates are important factors in disease progression. Since the original diabetes data is a low-dimensional data ($p = 64$), we add a thousand noise variables to the original data to also have a high-dimensional dataset with $p = 1064$. These low- and high-dimensional diabetes data can be found in our package.

In the recent literature, variable selection techniques, such as LARS and Spike and Slab, were used in a linear regression model applied to this diabetes data. While we spot from Figure 3.2(a) that, surprisingly, the response y is markedly right-skewed which can arise from a non-normal distribution, for example, a Gamma (or Inverse Gaussian) distribution. Therefore, we fit a Gamma regression model for the (low- and high-dimensional) diabetes data and use the extended dgLARS method by means of the proposed algorithm (IPC). According to the results of the previous section (Section 3.4.1), the MGRCV estimate $\hat{\phi}_{MGRCV}$ is applied as the dispersion estimator to the data.

Since we do not have prior information on the link function, before analyzing we have to choose between three of the most commonly used link functions *inverse*, *log* and *identity*. Therefore, for each of the low- and high-dimensional diabetes data, we fit the Gamma model with these three link functions and then choose the most suitable link function in two ways. First, we plot the adjusted dependent variable $\mathbf{z} = \hat{\boldsymbol{\eta}} + (\mathbf{y} - \hat{\boldsymbol{\mu}})(\partial\boldsymbol{\eta}/\partial\boldsymbol{\mu})$ against the estimated linear predictor $\hat{\boldsymbol{\eta}} = \mathbf{X}\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)$, suggested by [60], where $\hat{\boldsymbol{\mu}} = g^{-1}(\mathbf{X}\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))$ is the fitted value, $\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)$ is the extended dgLARS estimator at γ, and $\partial\boldsymbol{\eta}/\partial\boldsymbol{\mu}$ can be found in Table 2.1 in Chapter 2. The plot should be linear, departure from linear suggests a poor choice of link function [58]. Second, after fitting these three models (the Gamma model with the three link functions), we choose the best model by comparing the BIC values to see which link function would be more suitable for the data.

The results based on the low- and high-dimensional diabetes data are reported in Sections 3.5.1 and 3.5.2, respectively.



(a) (b)

Figure 3.2: (a) Histogram of the response y for the diabetes data. (b) Plot of $\mathbf{z}$ versus $\hat{\boldsymbol{\eta}}$ with the *log* link function, computed for the low-dimensional diabetes data, $p = 64$.

3.5.1 Low-dimensional Real Data

For the low-dimensional scenario, when $p < n$, we consider the diabetes data with $n = 442$ and $p = 64$ used in [29]. For this dataset, we plotted the adjusted dependent variable $\mathbf{z}$ versus the estimated linear predictor $\hat{\boldsymbol{\eta}}$ for the Gamma model with the *inverse*, *log* and *identity* link functions, but for the sake of brevity we only show the plot related to the *log* link (Figure 3.2(b)). The plots illustrate that while there are scatter in all three plots, there are no overt departure from linearity and hence no obvious evidence of the poor choice of these link functions. In addition, the results (not reported) show that, the model with the *log* link performs the best among these models with BIC of 4806, and the model with the *identity* link (with BIC 4814) fits better than the model with the *inverse* canonical link (with BIC 4829). Finally, we find out that the *log* link function is the most suitable link for the low-dimensional diabetes data and we choose it, in the following, as the selected link function.

We first apply a number of variable selection methods such as LARS [29], LASSO [94], Ridge [45], Elastic Net [104], and Spike and Slab [46] by using the **`lars`** [41], **`glmnet`** [35] and **`spikeslab`** [47] packages, and then compare the results to the results obtained from the proposed dgLARS method implemented by our package. Note that, for the dgLARS method we use the *Gamma* family in our package, while this family is not available in other packages, so that we fit the *Gaussian* family to the data to be able to use these packages.

The top 20 selected variables obtained by these algorithms (without considering any model selection criterion) are reported on Table 3.2, where we used $type$ =*'lar'* and $type$ =*'lasso'* in the **`lars`** package for the LARS and LASSO methods, respectively, and for the Ridge and Elastic Net methods we used $\alpha = 0.001$ and $\alpha = 0.5$ in the **`glmnet`** package, respectively. For the Spike and Slab method we considered $set.seed(112358)$ in the **`spikeslab`** package, and for the dgLARS method we fitted the Gamma model with the *log* link and also the canonical *inverse* link, so that for this dataset we calculated the dispersion estimates based on each link function as $\hat{\phi}^{log}_{MGRCV} = 0.140$ and $\hat{\phi}^{inverse}_{MGRCV} = 0.145$.

When we compare the results of the dgLARS Gamma method to the results obtained from other algorithms, we find out the remarkable results. From Table 3.2 we can see that, the variables selected by the LARS, LASSO and Elastic Net methods are the same, and almost in all models the first 4 variables $(3, 9, 4$ and $7)$ are the same. Moreover, importantly, all models (except the dgLARS) have the same selected variables just in the different order. While all algorithms (except the dgLARS) select the covariates $12, 27, 33$ and 52 in the first 20 variables, our proposed algorithm does not select them among the top 20 variables. Instead, the dgLARS algorithm by the Gamma model selects several new other variables (indicated in bold in Table 3.2) which none of the other algorithms do. For instance, the variables $60, 18$ and 25 are selected into the first 20 selected variables by the dgLARS Gamma model with the *log* link function, and the variables $60, 18, 42, 35$ and 40 are selected when the link function is the *inverse*. As a result, the extended dgLARS method based on a *Gamma* model, with the log link function, finds out that the variables "hdl : ltg", "$ltg\hat{}2$" and "map : ltg" $(60, 18,$ and $25)$ are more important factor in disease progression than the variables "$bmi\hat{}2$", "age : ltg", "sex : hdl" and "tc : tch" $(12, 27, 33$ and $52)$.

To identify and rank the most important variables, by the dgLARS Gamma regression model with the *log* link function, we use three model selection criteria; cross-validation deviance (CV), AIC and BIC, so that in Table 3.3, we report the sequence of the top 20 variables and their parameter estimates obtained based on all three model selection criteria. In interpreting the table,

Table 3.2: The sequences of the top 20 predictors selected by the LARS, LASSO, Ridge, Elastic Net, Spike and Slab and dgLARS algorithms obtained for low-dimensional diabetes data.

Algorithm	Selected Variables																			
LARS	3	9	4	7	37	20	19	12	22	28	2	10	27	11	30	46	33	52	24	29
LASSO	3	9	4	7	37	20	19	12	22	28	2	10	27	11	30	46	33	52	24	29
Ridge	3	9	4	8	7	10	12	5	1	6	13	43	24	37	19	63	64	16	39	17
Elastic Net	3	9	4	7	37	12	20	19	10	22	28	2	27	30	11	52	46	33	24	29
Spike and Slab	3	9	4	7	2	20	37	19	12	27	52	11	10	22	63	30	24	58	43	5
dgLARS (log)	3	9	4	7	20	2	28	**60**	11	46	19	29	**18**	30	22	10	37	24	58	**25**
dgLARS ($inverse$)	3	9	4	7	20	**60**	2	46	**18**	10	**42**	28	11	19	30	**35**	29	**40**	24	63

we note that the selected variables are those having non-zero coefficient estimates. First, we use a tenfold cross-validation to obtain the tuning parameter (γ) of the dgLARS Gamma model. Figure 3.3(a) shows the 10-fold cross-validation deviance curve as a function of the tuning parameter (γ), where the vertical red dashed line shows the optimal value of γ, which is $\hat{\gamma}_{_{CV}} = 1.011$, with the number of non-zero estimated coefficients, which is $|\mathcal{A}_{_{CV}}| = 16$, where $\mathcal{A}_{_{CV}} = \mathcal{P} \cup \mathcal{A}(\hat{\gamma}_{_{CV}}) = \{m \,:\, \hat{\beta}_m(\hat{\gamma}_{_{CV}}) \neq 0\,, m = 0, 1, \ldots, p\}$. Since we consider the protected variables set $\mathcal{P}$ contains only the intercept, $|\mathcal{P}| = b = 1$. Second, by means of the BIC criterion the dgLARS method estimates a Gamma regression model with a high level of sparsity, so that only $|\mathcal{A}_{_{BIC}}| = |\mathcal{P} \cup \mathcal{A}(\hat{\gamma}_{_{BIC}})| = 9$ covariates (i.e.,the intercept plus a subset of 8 parameters) are found to influence disease progression, where $\hat{\gamma}_{_{BIC}} = 1.87$. While by using the AIC criterion the number of non-zero estimated coefficients is $|\mathcal{A}_{_{AIC}}| = |\mathcal{P} \cup \mathcal{A}(\hat{\gamma}_{_{AIC}})| = 16$, where $\hat{\gamma}_{_{AIC}} = 0.98$ with AIC 4000.

One point should be mentioned here is that, the number of the points of the solution curve (q) for this low-dimensional data set by using the original PC and improved PC algorithms are 121 and 82, respectively, which shows that the improved algorithm works faster than the original one.

3.5.2 High-dimensional Real Data

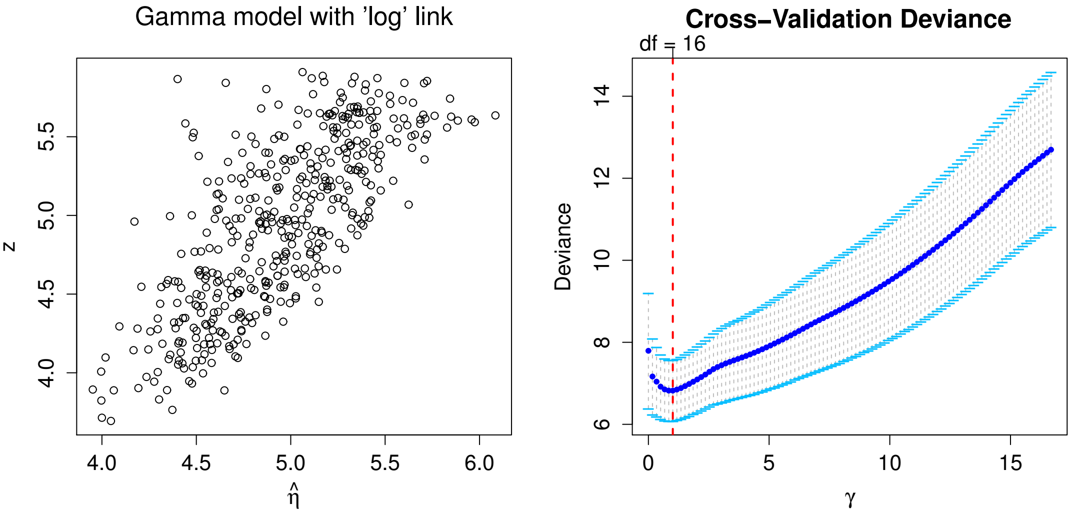
For a p larger than n setup, we expanded the original diabetes data to become $n = 442$ and $p = 1064$, so that the 1000 additional variables are in reality just noise. We fit a Gamma regression model for this high-dimensional data and use the extended dgLARS method by means of the proposed algorithm (IPC). For the high-dimensional diabetes data, based on the plots of the adjusted de-

Table 3.3: A list of the top 20 selected variables and their parameter estimates obtained using the dgLARS Gamma method (with log link, $\eta_i = \log \mu_i$) for low-dimensional diabetes data. In each criterion, variables selected are those having non-zero coefficient estimates; $|\mathcal{A}_{CV}| = 16$, $|\mathcal{A}_{AIC}| = 16$ and $|\mathcal{A}_{BIC}| = 9$.

	Variable		Coefficient Estimate		
Step	Name	Number	CV	AIC	BIC
1	bmi	3	3.0757	3.0783	2.9998
2	ltg	9	3.4997	3.5071	3.2909
3	map	4	1.9033	1.9181	1.5009
4	hdl	7	-1.7297	-1.7416	-1.3879
5	$age : sex$	20	0.9493	0.9551	0.6846
6	sex	2	-1.2282	-1.2489	-0.6400
7	$age : glu$	28	0.2091	0.2109	0.1542
8	$hdl : ltg$	**60**	0.4284	0.4355	0.1377
9	$age\hat{}2$	11	0.2715	0.2815	0
10	$map : hdl$	46	0.2929	0.3077	0
11	$glu\hat{}2$	19	0.2497	0.2599	0
12	$sex : bmi$	29	0.1282	0.1380	0
13	$ltg\hat{}2$	**18**	0.0021	-0.1202	0
14	$sex : map$	30	0.1087	0.1206	0
15	$age : map$	22	0.0091	0.0116	0
16	glu	10	0	0	0
17	$bmi : map$	37	0	0	0
18	$age : ldl$	24	0	0	0
19	$ldl : glu$	58	0	0	0
20	$map : ltg$	**25**	0	0	0

pendent variable $\mathbf{z}$ versus the estimated linear predictor $\hat{\boldsymbol{\eta}}$ (not shown here except for the *log* link, Figure 3.3(b)), we obtained the same results for all three considered link functions, but based on the BIC values (not reported here) we chose the Gamma model with the *log* link function as the best model. Moreover, for this dataset we calculated the dispersion estimate based on this model by using the MGRCV estimator $\hat{\phi}_{MGRCV} = 0.147$.

Figure 3.4 consists of four images which are outputs of our package. The figure displays the dgLARS Gamma solution path, the Rao score path and the CV, AIC and BIC criteria obtained using the improved PC algorithm and the full data. Like Section 3.5.1, we consider three criteria; Firstly, Figure 3.4(a) shows the 10-fold cross-validation deviance curve in which the optimal value of the tuning parameter is $\hat{\gamma}_{CV} = 1.77$, with the number of non-zero estimated



(a) (b)

Figure 3.3: (a) Plot of the 10-fold cross-validation deviance computed for the low-dimensional diabetes data with $p = 64$. (b) Plot of **z** against $\hat{\boldsymbol{\eta}}$ computed with the high-dimensional diabetes data, $p = 1064$, when the link functions is *log*.

coefficients, which is $|\mathcal{A}_{CV}| = |\mathcal{P} \cup \mathcal{A}(\hat{\gamma}_{CV})| = 57$, where $\mathcal{P}$ contains only the intercept. Secondly, by the BIC model selection criterion the dgLARS method estimates a Gamma regression model with a high level of sparsity, so that $\hat{\gamma}_{BIC} = 2.76$ with BIC of 4817 and $|\mathcal{A}_{BIC}| = 11$ covariates (i.e., the intercept plus a subset $(\mathcal{A}(\hat{\gamma}_{BIC}))$ of 10 parameters) are found to influence disease progression. While by the AIC model selection criterion, $\hat{\gamma}_{AIC} = 1.79$ (with AIC of 4760) and the number of non-zero estimated coefficients is $|\mathcal{A}_{AIC}| = 53$ (i.e., the subset $\mathcal{A}(\hat{\gamma}_{AIC})$ has 52 covariates).

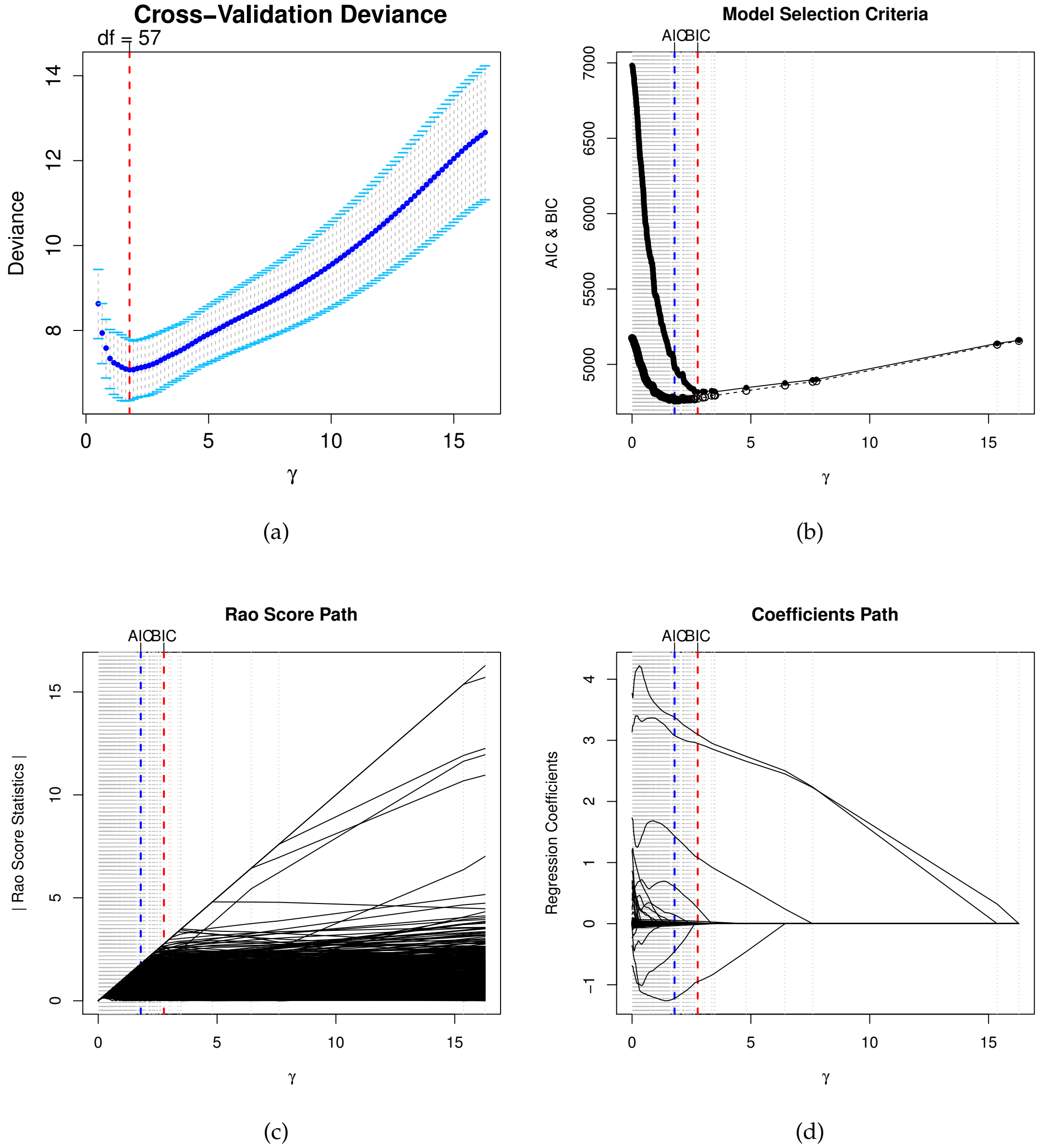
In addition, we report the sequence of the 25 selected variables and their parameter estimates based on all three criteria in Table 3.4. In interpreting the table, we note that variables starting with "$n.$" are noise variables and the rest are the original variables.

Using Figure 3.4 and Table 3.4 we can see that, while only 4 variables $(3, 9, 4$ and $7)$ have path-profiles that clearly stand out in all three criteria, significantly these variables are the top 4 from our previous analysis obtained using the low-dimensional data (Section 3.5.1). It is interesting that 3 other non-noise variables, "$age : sex$", "sex" and "$age : glu$" (with variable numbers: $20, 2$ and 28)

Table 3.4: The top 25 variables and their parameter estimates obtained using the dgLARS Gamma method (with log link) for high-dimensional diabetes data. In each criterion, variables selected are those having non-zero coefficient estimates; $|\mathcal{A}_{CV}| = 57$, $|\mathcal{A}_{AIC}| = 53$ and $|\mathcal{A}_{BIC}| = 11$.

	Variable		Coefficient Estimate		
Step	Name	Number	CV	AIC	BIC
1	bmi	3	3.0794	3.0762	2.9489
2	ltg	9	3.3787	3.3746	3.0971
3	map	4	1.4391	1.4337	1.0788
4	hdl	7	-1.2253	-1.2228	-0.9491
5	$n.312$	376	0.0551	0.0548	0.0387
6	$n.545$	609	0.0320	0.0318	0.0155
7	$n.543$	607	-0.0341	-0.0338	-0.0142
8	$age : sex$	20	0.6080	0.6033	0.2545
9	$n.423$	487	0.0113	0.0112	0.0034
10	$n.770$	834	0.0177	0.0175	0.0036
11	$n.657$	721	0.0115	0.0113	0
12	sex	2	-0.4608	-0.4550	0
13	$n.636$	700	-0.0170	-0.0167	0
14	$n.283$	347	0.0124	0.0123	0
15	$n.337$	401	-0.0162	-0.0160	0
16	$n.404$	468	0.0090	0.0088	0
17	$n.62$	126	-0.0121	-0.0118	0
18	$n.988$	1052	0.0089	0.0086	0
19	$age : glu$	28	0.1465	0.1440	0
20	$n.71$	135	-0.0090	-0.0088	0
21	$n.160$	224	-0.0083	0.0083	0
22	$n.635$	699	-0.0080	-0.0087	0
23	$n.466$	530	-0.0085	-0.0083	0
24	$n.612$	676	-0.0084	-0.0082	0
25	$n.969$	1033	-0.0045	-0.0045	0

are in the top 25 variables, so that in Table 3.3, they have the variable number: $5, 6$ and 7, respectively, and along with "bmi","ltg","map" and "hdl" are the first 7 variables in Table 3.3. Regardless of the criteria used, when we inspected the first 100 variables selected by the improved PC algorithm, we found that 8 were from the original 64 variables, and 7 were from the top 25 variable from Table 3.4. This demonstrates stability of the improved PC algorithm even in ultra-high dimensional problems.



(a) (b)

(c) (d)

Figure 3.4: (a) Plot of the 10-fold cross-validation deviance, (b) Model selection criteria, (c) Rao score statistics path, (d) Regression coefficients path for the dgLARS Gamma regression model for the high-dimensional diabetes data with $p = 1000$ noise variables.

At the end, the number of the points of the solution curve for this data set by using the original PC and improved PC algorithms are 482 and 465, respectively.

3.6 Conclusions

In this chapter we extended the dgLARS method for a GLM to a larger class of the exponential family, namely the *exponential dispersion family* (when the dispersion parameter, ϕ, is unknown), and obtained the general framework of the dgLARS estimator for general GLM with general link function. We implemented explicitly the method for Gamma and Inverse Gaussian with a variety of link functions. To estimate the dispersion parameter we first presented a classical estimator which can be used during the solution path, and then proposed a new method to do high-dimensional inference on the dispersion parameter. We also proposed an iterative algorithm that produces a more stable and accurate estimation. Moreover, we proposed an improved version of the predictor-corrector (PC) algorithm to compute the solution curve. The improved PC algorithm allows the dgLARS method to be implemented using less steps, greatly reducing the computational burden because of reducing the number of points of the solution curve. The method was compared well with some well-known methods where can be used. The results show that the improved PC algorithm is better and quicker than the original PC algorithm, and now the dgLARS method can be used for a variety of distributions with different types of the canonical and non-canonical link functions.

Chapter 4

Sparse Relative Risk Regression Models

Contents

Abstract

Clinical studies where patients are routinely screened for many genomic features are becoming more routine. In principle, this holds the promise of being able to find genomic signatures for a particular disease. In particular, cancer survival is thought to be closely linked to the genomic constitution of the tumour. Discovering such signatures will be useful in the diagnosis of the patient, may be used for treatment decisions and, perhaps, even the development of new treatments. However, genomic data are typically noisy and high-dimensional, not rarely outstripping the number of patients included in the study. Regularized survival models have been proposed to deal with such scenarios. These methods typically induce sparsity by means of a coincidental match of the geometry of the convex likelihood and (near) non-convex regularizer. The disadvantages of such methods are that they are typically non-invariant to scale changes of the covariates, they struggle with highly correlated covariates and they have a practical problem of determining the amount of regularization. In this chapter we propose a method for sparse inference in relative risk regression models based only on the likelihood, closely related to least angle regression. The method is computationally fast and is implemented in the R-package **`dglars`**.

Keywords: *Relative risk regression models; Survival analysis; Gene expression data; High-dimensional data; Sparsity; dgLARS.*

4.1 Introduction

Advances in genomic technologies have meant that many new clinical studies in cancer survival include a variety of genomic measurements, ranging from gene expression to SNP data. Studying the relationship between survival and genomic markers can be useful for a variety of reasons. If a genomic signature can be found, then patients can be given more accurate survival information. Furthermore, treatment and care may be adjusted to the prospects of an individual patient. Eventually, the genomic signature combined with information from other studies may be used to identify drug targets. We will focus on four recent studies of cancer survival for four different tumours. Our aim is to find a reproducible sparse predictor for cancer survival.

Sparse inference in the past two decades has been dominated by methods that penalize typically convex likelihoods by functions of the parameters that happen to induce solutions with many zeros. The lasso [94], elastic net [105], l_0 [80] and the SCAD [31] penalties are examples of such penalties that, depending on some tuning parameter, conveniently shrink estimates to exact zeros. Also in survival analysis these methods have been introduced. [95] applied the lasso penalty to the Cox proportional hazards model. [40], [89] and [39] suggested important computational improvements to make the calculation of the lasso path in the Cox proportional hazards model more efficient. Although the lasso penalty induces sparsity, it is well known to suffer from possible inconsistent selection of variables. [19] implemented the SCAD penalty for the Cox model. This method enjoys the property of sparsistency, i.e., in an appropriate asymptotic sense it selects first the true variables before selecting the incorrect ones.

Whereas penalized inference is convenient, justification of the penalty is somewhat problematic. Interpreting the solution as a Bayesian MAP estimator with a particular prior on the parameters seems to merely reformulate the problem, rather than solving it. Furthermore, the methods suffer from being not invariant under scale transformations of the explanatory variables. This means that measuring, e.g., height in centimeters or inches can and probably will result in dramatically different answers. Therefore, most penalized regression methods start their exposition by assuming that the variables are appropriately renormalized. This is clearly a merely algorithmic device and simply begs the question of invariance. Clearly the strongest argument in favour of some of these methods are their asymptotic properties. Nevertheless, what this means in the small sample settings encountered in practice is also problematic.

In this chapter, we will approach sparsity directly from a likelihood point of view. The angle between the covariates and the tangent residual vector within the likelihood manifold provides a direct and scale-invariant way to assess the importance of the individual covariates. The idea is similar to the least angle regression approach proposed by [29]. However, rather than using it as a computational device for obtaining lasso solutions, we view the method in its own right as in [13]. Moreover, the method extends directly beyond the Cox proportional hazard model. In fact, we will focus on general relative risk survival models.

In section 4.2, we introduce the relative risk regression model together with its underlying likelihood geometry. In section 4.3 the sparse solution path of a relatively risk survival model is defined. By appealing to the theory of M-estimation, we derive a robust way of selecting a unique solution of the sparse survival regression model. Simulation studies compare the performance of the method to other sparse survival regression approaches, especially in the presence of correlated predictors. In section 4.5 we return to the motivating cancer survival studies and employ differential geometric Cox proportional hazards modelling to find a genetic signature for cancer survival in skin, colon, prostate and ovarian cancer.

4.2 Relative Risk Regression Models

In analyzing survival data, one of the most important tool is the hazard function, which is used to express the risk or hazard of failure at some time t. Formally, let T be the (absolutely) continuous random variable associated with the survival time and let $f(t)$ be the corresponding probability density function, the hazard function is defined as

$$\lambda(t) = \frac{f(t)}{1 - \int_0^t f(s)ds}, \tag{4.1}$$

and specifies the instantaneous rate at which failures occur for subjects that are surviving at time t. Suppose that the hazard function (4.1) can depend on a p-dimensional, possibly time-dependent, vector of covariates, denoted by $\mathbf{x}(t) = (x_1(t), \dots, x_p(t))^\top$. Relative risk regression models are based on the assumption that the vector $\mathbf{x}(t)$ influences the hazard function through the following relation

$$\lambda(t; \mathbf{x}) = \lambda_0(t)\psi(\mathbf{x}(t); \boldsymbol{\beta}), \tag{4.2}$$

where $\boldsymbol{\beta} \in \mathcal{B} \subseteq \mathbb{R}^p$ is a p-dimensional vector of unknown fixed parameters and $\lambda_0(t)$ is the baseline hazard function at time t, which is left unspecified. Finally, $\psi : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ is a fixed twice continuously differentiable function, called the *relative risk function*, and the parameter space $\mathcal{B}$ is such that $\psi(\mathbf{x}(t); \boldsymbol{\beta}) > 0$ for each $\boldsymbol{\beta} \in \mathcal{B}$. We also assume that the relative risk function is normalized, i.e. $\psi(\mathbf{0}; \boldsymbol{\beta}) = 1$. Model (4.2), originally proposed in [93], clearly extends the usual Cox regression model [26] which is obtained when $\psi(\mathbf{x}(t); \boldsymbol{\beta}) = \exp(\boldsymbol{\beta}^\top \mathbf{x}(t))$, but also allow us to work with applications in which the exponential form of the relative risk function is not the best choice. This issue was observed in [65] and further underlined in [24]. As a motiving example for the generalization (4.2), several authors have noted that the linear relative risk function $\psi(\mathbf{x}(t); \boldsymbol{\beta}) = 1 + \boldsymbol{\beta}^\top \mathbf{x}(t)$ provides a natural framework within which to assess departures from an additive relative risk model when two or more risk factors are studied in relation to the incidence of a disease (see for example [93], [75] and [73], among the other). Other possible choices for the relative risk functions are the logit relative risk function $\psi(\mathbf{x}(t); \boldsymbol{\beta}) = \log(1 + \exp(\boldsymbol{\beta}^\top \mathbf{x}(t)))$, proposed by [28], or the the excess relative risk model $\psi(\mathbf{x}(t); \boldsymbol{\beta}) = \prod_{m=1}^{p}(1 + x_m(t)\beta_m)$. For detailed theoretical treatments based on the counting process theory, the interested reader is referred to [1] or [52].

Suppose that n observations are available and let with t_i the ith observed failure time. Assume that we have k uncensored failure times and let by $\mathcal{D}$ the set of indices for which the corresponding failure time is observed. The remaining failure times are right censored. As explained in [27], if we denote by $\mathcal{R}(t)$ the risk set, i.e., the set of indices corresponding to the subjects who have not failed and are still under observation just prior to time t, under the assumption of independent censoring, inference about the $\boldsymbol{\beta}$ can be carried out by the following partial likelihood function

$$\mathcal{L}_p(\boldsymbol{\beta}) = \prod_{i \in \mathcal{D}} \frac{\psi(\mathbf{x}_i(t_i); \boldsymbol{\beta})}{\sum_{j \in \mathcal{R}(t_i)} \psi(\mathbf{x}_j(t_i); \boldsymbol{\beta})}. \tag{4.3}$$

When the exponential relative risk function is used in model (4.2) and we work with fixed covariates, (4.3) is clearly equal to the original partial likelihood introduced in [26] and discussed in great detail in [25]. The inferential aspects of the relative risk regression models (4.2) are studied in [74] where are extended the results given in [7] for the Cox regression model.

4.3 Sparse Relative Risk Regression

Aim of this section is to extend the dgLARS method [13] to the relative risk regression models described in section 4.2. The basic idea underlying the dgLARS method is to use the differential geometrical structure of a generalized linear model (GLM) [60] to generalize the LARS method originally proposed in [29]. This means that, our first step is relate the partial likelihood with the likelihood function of a specific GLM. As originally observed in [92] and studied in greater detail in [72], to solve this problem, in this chapter we shall use the identity that exists between the partial likelihood (4.3) and the likelihood function of a logistic regression model for matched case-control studies. The idea to use this identity to study the differential geometrical structure of a relative risk regression model is not new and was originally used in [63] to construct approximated confidence regions for the proportional hazards model. For a more complete description of the relationship between differential geometry and statistical models, the interested reader is refereed to [6] and [53].

4.3.1 Differential Geometrical Structure of the Relative Risk Regression Model

In order to define the generalized equiangularity condition for the relative risk regression model, it is useful to see the partial likelihood (4.3) as arising from a multinomial sample scheme. Consider an index $i \in tuning$ and let $\mathbf{Y}_i = (Y_{ih})_{h\in\mathcal{R}(t_i)}$ be a multinomial random variable with sample size equal to 1 and cell probabilities $\boldsymbol{\pi}_i = (\pi_{ih})_{h\in\mathcal{R}(t_i)} \in \Pi_i$, i.e. $p(\mathbf{y}; \boldsymbol{\pi}_i) = \prod_{h\in\mathcal{R}(t_i)} \pi_{ih}^{y_{ih}}$. Assuming that the random vectors $\mathbf{Y}_i$ are independent, the joint probability density function is an element of the model space

$$\mathcal{S} = \left\{ \prod_{i\in\mathcal{D}} \prod_{h\in\mathcal{R}(t_i)} \pi_{ih}^{y_{ih}} : (\boldsymbol{\pi}_i)_{i\in\mathcal{D}} \in \bigotimes_{i\in\mathcal{D}} \Pi_i \right\}. \qquad (4.4)$$

The set (4.4) will play the role of ambient space. We would like to underline that our differential geometric constructions are invariant to the chosen parameterization which means that the ambient space $\mathcal{S}$ can be equivalently defined by the canonical parameter vector and this will not change the results. In this chapter we prefer to use the mean value parameter vector to specify our differential geometrical description because this will make the relationship with the partial likelihood (4.3).

Consider the following model definition for the conditional expected value of the random variable Y_{ih}, i.e.

$$E_{\boldsymbol{\beta}}(Y_{ih}) = \pi_{ih}(\boldsymbol{\beta}) := \frac{\psi(\mathbf{x}_h(t_i); \boldsymbol{\beta})}{\sum_{j \in \mathcal{R}(t_i)} \psi(\mathbf{x}_j(t_i); \boldsymbol{\beta})}, \tag{4.5}$$

then our model space is the set

$$\mathcal{M} = \left\{ \prod_{i \in \mathcal{D}} \prod_{h \in \mathcal{R}(t_i)} \left(\frac{\psi(\mathbf{x}_h(t_i); \boldsymbol{\beta})}{\sum_{j \in \mathcal{R}(t_i)} \psi(\mathbf{x}_j(t_i); \boldsymbol{\beta})} \right)^{y_{ih}} : \boldsymbol{\beta} \in \mathcal{B} \right\}. \tag{4.6}$$

The partial likelihood (4.3) is formally equivalent to the likelihood function associated with the model space $\mathcal{M}$ if we assume that for each $i \in \mathcal{D}$, the observed y_{ih} is equal to one if h is equal to i and zero otherwise.

Let $\ell(\boldsymbol{\beta}) = \sum_{i \in \mathcal{D}} \sum_{h \in \mathcal{R}(t_i)} Y_{ih} \log \pi_{ih}(\boldsymbol{\beta})$ be the log-likelihood function associated to the model space $\mathcal{M}$ and let $\partial_m \ell(\boldsymbol{\beta}) = \partial \ell(\boldsymbol{\beta}) / \partial \beta_m$. The tangent space $T_{\boldsymbol{\beta}}\mathcal{M}$ of $\mathcal{M}$ at the model point $\prod_{i \in \mathcal{D}} \prod_{h \in \mathcal{R}(t_i)} \pi_{ih}(\boldsymbol{\beta})^{y_{ih}}$ is defined as that linear vector space spanned by the p elements of the score vector, formally

$$T_{\boldsymbol{\beta}}\mathcal{M} = \text{span}\{\partial_1 \ell(\boldsymbol{\beta}), \ldots, \partial_p \ell(\boldsymbol{\beta})\}.$$

Under the standard regularity conditions, it is easy to see that $T_{\boldsymbol{\beta}}\mathcal{M}$ is the linear vector space of the random variables $v_{\boldsymbol{\beta}} = \sum_{m=1}^{p} v_m \partial_m \ell(\boldsymbol{\beta}) \in T_{\boldsymbol{\beta}}\mathcal{M}$ with zero expected value and finite variance, i.e.,

$$E_{\boldsymbol{\beta}}(v_{\boldsymbol{\beta}}) = 0 \quad \text{and} \quad E_{\boldsymbol{\beta}}(v_{\boldsymbol{\beta}}^2) < \infty.$$

As a simple consequence of the chain rule we have that for any tangent vector belonging to the tangent space $T_{\boldsymbol{\beta}}\mathcal{M}$

$$v_{\boldsymbol{\beta}} = \sum_{m=1}^{p} v_m \partial_m \ell(\boldsymbol{\beta}) = \sum_{i\in\mathcal{D}} \sum_{h\in\mathcal{R}(t_i)} \left(\sum_{m=1}^{p} v_m \frac{\partial \pi_{ih}(\boldsymbol{\beta})}{\partial \beta_m} \right) \frac{\partial \ell(\boldsymbol{\beta})}{\partial \pi_{ih}} = \sum_{i\in\mathcal{D}} \sum_{h\in\mathcal{R}(t_i)} w_{ih} \frac{\partial \ell(\boldsymbol{\beta})}{\partial \pi_{ih}},$$

which shows that $T_{\boldsymbol{\beta}}\mathcal{M}$ is a linear sub vector space of the tangent space $T_{\boldsymbol{\beta}}\mathcal{S}$ spanned by the random variables $\partial_{ih}\ell(\boldsymbol{\beta}) = \partial\ell(\boldsymbol{\beta})/\partial\pi_{ih}$. To define the notion of angle between two given tangent vectors belonging to $T_{\boldsymbol{\beta}}\mathcal{M}$, say $v_{\boldsymbol{\beta}} = \sum_{m=1}^{p} v_m \partial_m \ell(\boldsymbol{\beta})$ and $w_{\boldsymbol{\beta}} = \sum_{n=1}^{p} w_n \partial_n \ell(\boldsymbol{\beta})$, we shall use the information metric [79, 17], i.e,

$$\langle v_{\boldsymbol{\beta}}; w_{\boldsymbol{\beta}} \rangle_{\boldsymbol{\beta}} = E_{\boldsymbol{\beta}}(v_{\boldsymbol{\beta}} \cdot w_{\boldsymbol{\beta}}) = \sum_{m,n=1}^{p} E_{\boldsymbol{\beta}} \left(\partial_m \ell(\boldsymbol{\beta}) \cdot \partial_n \ell(\boldsymbol{\beta}) \right) v_m w_n = \mathbf{v}^\top \mathcal{I}(\boldsymbol{\beta}) \mathbf{w}, \quad (4.7)$$

where $\mathbf{v} = (v_1, \ldots, v_p)$, $\mathbf{w} = (w_1, \ldots, w_p)$ and $\mathcal{I}(\boldsymbol{\beta})$ is the Fisher information matrix evaluated at $\boldsymbol{\beta}$. As observed in [63], the matrix $\mathcal{I}(\boldsymbol{\beta})$ used in (4.7) is not exactly equal to the Fisher information matrix of the relative risk regression model, however it has the appropriate asymptotic properties for the inference.

The tangent residual vector

$$r_{\boldsymbol{\beta}} = \sum_{i\in\mathcal{D}} \sum_{h\in\mathcal{R}(t_i)} r_{ih}(\boldsymbol{\beta}) \partial_{ih} \ell(\boldsymbol{\beta}), \quad (4.8)$$

where $r_{ih}(\boldsymbol{\beta}) = y_{ih} - \pi_{ih}(\boldsymbol{\beta})$, is an element of $T_{\boldsymbol{\beta}}\mathcal{S}$ and displays the difference between a model element in $\mathcal{S}$ and the data.

4.3.2 dgLARS Method for the Relative Risk Regression Model

The dgLARS method is a sequential method developed to estimate a sparse solution curve embedded in the parameter space $\mathcal{B}$. To explore the sparse structure of a relative risk regression model, we can use the following differential geometric characterization of the mth element of the score vector, i.e.

$$\partial_m \ell(\boldsymbol{\beta}) = \langle \partial_m \ell(\boldsymbol{\beta}); r_{\boldsymbol{\beta}} \rangle_{\boldsymbol{\beta}} = \cos(\rho_m(\boldsymbol{\beta})) \cdot \mathcal{I}_{mm}^{1/2}(\boldsymbol{\beta}) \cdot \|r_{\boldsymbol{\beta}}\|_{\boldsymbol{\beta}}, \quad (4.9)$$

where $I_{mm}(\boldsymbol{\beta})$ is the Fisher information for β_m and $\|r_{\boldsymbol{\beta}}\|^2_{\boldsymbol{\beta}}$ is equal to

$$\begin{aligned} E_{\boldsymbol{\beta}}\, r^2(\boldsymbol{\beta}) &= \sum_{i,j\in\mathcal{D}} \sum_{h\in\mathcal{R}(t_i)} \sum_{k\in\mathcal{R}(t_j)} E_{\boldsymbol{\beta}}(\partial_{ih}\ell\,(\boldsymbol{\beta}) \cdot \partial_{jk}\ell(\boldsymbol{\beta}))\; r_{ih}(\boldsymbol{\beta})\, r_{jk}(\boldsymbol{\beta}) \\ &= \sum_{i\in\mathcal{D}} \sum_{h,k\in\mathcal{R}(t_i)} E_{\boldsymbol{\beta}}(\partial_{ih}\ell\,(\boldsymbol{\beta}) \cdot \partial_{ik}\ell(\boldsymbol{\beta}))\; r_{ih}(\boldsymbol{\beta})\, r_{ik}(\boldsymbol{\beta}) \\ &= \sum_{i\in\mathcal{D}} \sum_{h,k\in\mathcal{R}(t_i)} \frac{r_{ih}(\boldsymbol{\beta})\, r_{ik}(\boldsymbol{\beta})}{\pi_{ih}(\boldsymbol{\beta})\, 1_{\{h=k\}} - \pi_{ih}(\boldsymbol{\beta})\, \pi_{ik}(\boldsymbol{\beta})}. \end{aligned}$$

The quantity $\rho_m(\boldsymbol{\beta})$ is a generalization of the Euclidean notion of angle between the mth column of the design matrix and the residual vector $\mathbf{r}(\boldsymbol{\beta}) = (r_{ih}(\boldsymbol{\beta}))_{i\in\mathcal{D},h\in\mathcal{R}(t_i)}$. Using (4.9) one can see that the signed Rao score test statistic is geometrically characterized as follows:

$$r^u_m(\boldsymbol{\beta}) = \mathcal{I}^{-1/2}_{mm}(\boldsymbol{\beta})\partial_m\ell(\boldsymbol{\beta}) = \cos(\rho_m(\boldsymbol{\beta})) \cdot \|r_{\boldsymbol{\beta}}\|_{\boldsymbol{\beta}}.$$

We shall say that two given predictors, say m and n, satisfy the generalized equiangularity condition at the point $\boldsymbol{\beta}$ when $|r^u_m(\boldsymbol{\beta})| = |r^u_n(\boldsymbol{\beta})|$. Inside the dgLARS theory, the generalized equiangularity condition is used to identify the predictors that are included in the active set. Formally, for a given value of the tuning parameter $\gamma \in \mathbb{R}^+$ the corresponding active set is denoted by $\hat{\mathcal{A}}(\gamma)$ and the dgLARS estimator, denoted by $\hat{\boldsymbol{\beta}}(\gamma)$, is such that the following conditions are satisfied:

$$\forall\, m \in \hat{\mathcal{A}}(\gamma) \quad \Rightarrow \quad \begin{cases} \left|r^u_m(\hat{\boldsymbol{\beta}}(\gamma))\right| = \gamma, & (4.10) \\ r^u_m(\hat{\boldsymbol{\beta}}(\gamma)) = s_m\gamma, & (4.11) \end{cases}$$

$$\forall\, m \notin \hat{\mathcal{A}}(\gamma) \quad \Rightarrow \quad |r^u_m(\hat{\boldsymbol{\beta}}(\gamma))| < \gamma, \tag{4.12}$$

where $s_m = \text{sign}(\hat{\beta}_m(\gamma))$.

Using the differential geometrical structure of a relative risk regression model explained in Section 4.3.1 and the previous conditions, the dgLARS method explores the sparse structure of a relative risk regression model. Formally, dgLARS computes a finite sequence of transition points, say $0 \leq \gamma^{(K)} \leq \ldots \leq \gamma^{(2)} \leq \gamma^{(1)}$, such that for each $\gamma^{(k)}$ one of the following two conditions can occur:

(i) either

$$\left| r^u_m(\hat{\boldsymbol{\beta}}(\gamma^{(k)})) \right| = \gamma^{(k)}, \tag{4.13}$$

and therefore $m \in \hat{\mathcal{A}}(\gamma^{(k)})$;

(ii) or

$$\text{sign}(r^u_m(\hat{\boldsymbol{\beta}}(\gamma^{(k)}))) \neq \text{sign}(\hat{\beta}_m(\gamma^{(k)})), \tag{4.14}$$

and therefore $m \notin \hat{\mathcal{A}}(\gamma^{(k)})$.

This means that a new predictor is included in the active set when the generalized equiangularity condition (4.13) is satisfied, or an active predictor is removed from the active set if the sign of the corresponding signed Rao score test statistic is not in agreement with the sign of the estimated coefficient, i.e., condition (4.14). In order to simplify our notation, we shall assume that $\hat{\mathcal{A}}(\gamma^{(k)}) = \{1, 2, \dots, k\}$. As for each $\gamma \in (\gamma^{(k+1)}; \gamma^{(k)}]$ the signs of the estimated coefficients do not change, condition (4.11) tells us that, for a fixed value of the tuning parameter γ, the dgLARS estimator can be defined as the Z-estimator implicitly defined by the following system of estimating equations:

$$\left\{ \begin{array}{lcl} r^u_1(\hat{\boldsymbol{\beta}}(\gamma)) - s_1\gamma & = & 0 \\ r^u_2(\hat{\boldsymbol{\beta}}(\gamma)) - s_2\gamma & = & 0 \\ \vdots & & \vdots \\ r^u_k(\hat{\boldsymbol{\beta}}(\gamma)) - s_k\gamma & = & 0. \end{array} \right. \tag{4.15}$$

To gain more insight about the differences between the dgLARS and the ℓ_1-penalized estimators as variable selection methods, it is essential to compare conditions (4.10) and (4.12) with the corresponding conditions that characterize the behaviour of the Lasso estimator as a variable selection method. These conditions are studied in [83] for a general convex loss function. The behaviour of the Lasso estimator is explained by three conditions involving the gradient vector as a direct consequence of the Karush-Kuhn-Tucker (KKT) conditions. The Proof of Theorem 2 in [83] shows that for a given value of the tuning parameter

γ, the Lasso estimator is defined by:

$$\forall\, m \in \hat{\mathcal{A}}(\gamma) \quad \Rightarrow \quad \begin{cases} |\partial_m \ell(\hat{\boldsymbol{\beta}}(\gamma))| = \gamma, & (4.16) \\ \partial_m \ell(\hat{\boldsymbol{\beta}}(\gamma)) = s_m\, \gamma, & (4.17) \end{cases}$$

$$\forall\, m \notin \hat{\mathcal{A}}(\gamma) \quad \Rightarrow \quad |\partial_m \ell(\hat{\boldsymbol{\beta}}(\gamma))| < \gamma. \tag{4.18}$$

Conditions (4.16) and (4.18) tell us that the behaviour of the Lasso estimator, as a variable selection method, depends on the behaviour of the elements of the gradient vector. Characterization (4.9) tell us that the Lasso estimator is an efficient variable selection method only when the Fisher information is constant with respect to the parameter vector $\boldsymbol{\beta}$. The dgLARS method overcomes this theoretical limitation taking into account the Fisher information. For more details, the reader is referred to [13] and [70]. The latter extend the dgLARS method to GLMs based on the exponential dispersion family by means of an improved predictor-corrector algorithm.

4.3.3 Example: Sparse Cox's Proportional Hazards Model

Let Z, C and $\mathbf{x}(t)$ respectively denote the survival time, the censoring time and their associated p-dimension vector of covariates which can depend on time t, respectively. Further denote by $T = \min\{Z, C\}$ the observed time and $Y = I\{Z \leq C\}$ the censoring indicator. For simplicity, we assume that Z and C are conditionally independent and the censoring mechanism is non-informative. The observed dataset of size n is denoted by $\{(\boldsymbol{x}(t_i), t_i, y_i)\,, i = 1, \ldots, n\}$.

The proportional hazards model is very popular in survival analysis partially due to its simplicity and its convenience in dealing with censoring. The proportional hazards model assumes that the hazard function is

$$\lambda(t; \mathbf{x}) = \lambda_0(t) \exp(\boldsymbol{\beta}^\top \mathbf{x}(t)), \tag{4.19}$$

where $\lambda_0(t)$ is the baseline hazard function is unspecified and needs to be estimated nonparametrically and $\boldsymbol{\beta}$ is a p-dimensional vector of unknown fixed parameters of interest.

In next section, Section 4.3.4, first and second derivatives of the log-likelihood, Fisher information and its derivative, and Rao score statistic obtained from the

Cox's partial likelihood are derived.

4.3.4 Derivation of GIC

In this section we develop the main equations needed to compute the Generalized Information Criterion (GIC) proposed in Konishi and Kitagawa [54]. In other to simplify our notation, we shall drop the dependence of the dgLARS estimator from the tuning parameter γ, so that we shall write $\hat{\boldsymbol{\beta}}$ instead of $\hat{\boldsymbol{\beta}}(\gamma)$.

Given the partial log-likelihood function

$$\ell_p(\boldsymbol{\beta}) = \sum_{i \in \mathcal{D}} \left[\boldsymbol{\beta}^\top \mathbf{x}_i(t_i) - \log \left(\sum_{j \in \mathcal{R}(t_i)} \exp(\boldsymbol{\beta}^\top \mathbf{x}_j(t_i)) \right) \right], \tag{4.20}$$

its first and second derivatives with respect to $\boldsymbol{\beta}$ are given by

$$\partial_m \ell_p(\boldsymbol{\beta}) = \frac{\partial \ell_p(\boldsymbol{\beta})}{\partial \beta_m} = \sum_{i \in \mathcal{D}} \left[X_{mi}(t_i) - \sum_{j \in \mathcal{R}(t_i)} \pi_{ij}(\boldsymbol{\beta}) X_{mj}(t_i) \right], \tag{4.21}$$

where

$$\pi_{ij} = \frac{\exp(\boldsymbol{\beta}^\top \mathbf{x}_j(t_i))}{\sum_{k \in \mathcal{R}(t_i)} \exp(\boldsymbol{\beta}^\top \mathbf{x}_k(t_i))}$$

and

$$\begin{aligned} \partial_{m,n} \ell_p(\boldsymbol{\beta}) &= \frac{\partial^2 \ell_p(\boldsymbol{\beta})}{\partial \beta_m \partial \beta_n} \\ &= - \sum_{i \in \mathcal{D}} \left[\sum_{j \in \mathcal{R}(t_i)} \pi_{ij}(\boldsymbol{\beta}) X_{mj}(t_i) X_{nj}(t_i) \right. \\ &\qquad \left. - \left(\sum_{j \in \mathcal{R}(t_i)} \pi_{ij}(\boldsymbol{\beta}) X_{mj}(t_i) \right) \left(\sum_{j \in \mathcal{R}(t_i)} \pi_{ij}(\boldsymbol{\beta}) X_{nj}(t_i) \right) \right]. \end{aligned} \tag{4.22}$$

Further, the $(m,n)^{\text{th}}$ entry of the Fisher information matrix for $\boldsymbol{\beta}$ is given by

$$\mathcal{I}_{mn}(\boldsymbol{\beta}) = \frac{1}{|\mathcal{D}|}\sum_{i\in\mathcal{D}}\Bigg[\sum_{j\in\mathcal{R}(t_i)}\pi_{ij}(\boldsymbol{\beta})X_{mj}(t_i)X_{nj}(t_i) \\ - \left(\sum_{j\in\mathcal{R}(t_i)}\pi_{ij}(\boldsymbol{\beta})X_{mj}(t_i)\right)\left(\sum_{j\in\mathcal{R}(t_i)}\pi_{ij}(\boldsymbol{\beta})X_{nj}(t_i)\right)\Bigg], \quad (4.23)$$

such that the Fisher information for $\boldsymbol{\beta}_m$ is

$$\mathcal{I}_{mm}(\boldsymbol{\beta}) = \frac{1}{|\mathcal{D}|}\sum_{i\in\mathcal{D}}\left[\sum_{j\in\mathcal{R}(t_i)}\pi_{ij}(\boldsymbol{\beta})X^2_{mj}(t_i) - \left(\sum_{j\in\mathcal{R}(t_i)}\pi_{ij}(\boldsymbol{\beta})X_{mj}(t_i)\right)^2\right], \quad (4.24)$$

with its derivative:

$$\begin{aligned}\partial_n\mathcal{I}_{mm}(\boldsymbol{\beta}) &= \frac{\partial\mathcal{I}_{mm}(\boldsymbol{\beta})}{\partial\beta_n} \\ &= \frac{1}{|\mathcal{D}|}\sum_{i\in\mathcal{D}}\left[\sum_{j\in\mathcal{R}(t_i)}\pi_{ij}(\boldsymbol{\beta})X^2_{mj}(t_i)\left(X_{nj}(t_i) - \sum_{j\in\mathcal{R}(t_i)}\pi_{ij}(\boldsymbol{\beta})X_{nj}(t_i)\right)\right] \\ &- \frac{2}{|\mathcal{D}|}\sum_{i\in\mathcal{D}}\Bigg[\left(\sum_{j\in\mathcal{R}(t_i)}\pi_{ij}(\boldsymbol{\beta})X_{mj}(t_i)\right) \\ &\quad\times\left(\sum_{j\in\mathcal{R}(t_i)}\pi_{ij}(\boldsymbol{\beta})X_{mj}(t_i)\left(X_{nj}(t_i) - \sum_{j\in\mathcal{R}(t_i)}\pi_{ij}(\boldsymbol{\beta})X_{nj}(t_i)\right)\right)\Bigg].\end{aligned} \quad (4.25)$$

We now give the details of the GIC derivation. As we have seen in Section 4.3.2, the dgLARS estimator can be defined as the Z-estimator implicitly defined by the following equations

$$\begin{aligned}
0 &= \phi_m(\hat{\boldsymbol{\beta}}, \gamma) \\
&= \partial_m \ell_p(\hat{\boldsymbol{\beta}}) - \gamma s_m \mathcal{I}_{mm}^{1/2}(\hat{\boldsymbol{\beta}}) \\
&= \sum_{i \in \mathcal{D}} \left(\partial_m \ell_{p,i}(\hat{\boldsymbol{\beta}}) - \gamma \frac{s_m \mathcal{I}_{mm}^{1/2}(\hat{\boldsymbol{\beta}})}{|\mathcal{D}|} \right) \\
&= \sum_{i \in \mathcal{D}} \phi_{m,i}(\hat{\boldsymbol{\beta}}, \gamma),
\end{aligned}$$

for any $m \in \hat{\mathcal{A}}(\gamma)$. Under this setting, the generalized information criterion [54] is defined as

$$GIC(\hat{\boldsymbol{\beta}}, \gamma) = -2\, \ell_p(\hat{\boldsymbol{\beta}}(\gamma)) + 2\, tr(R^{-1}(\hat{\boldsymbol{\beta}}, \gamma)\, Q(\hat{\boldsymbol{\beta}}, \gamma)), \tag{4.26}$$

where $\hat{\boldsymbol{\beta}}$ is the final estimate of $\boldsymbol{\beta}$ for a given γ and

$$\begin{aligned}
R_{m,n}(\hat{\boldsymbol{\beta}}, \gamma) &= -\frac{1}{|\mathcal{D}|} \partial_n \phi_m(\boldsymbol{\beta}, \gamma) \,\big|_{\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}} \\
&= -\frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} \left(\partial_{m,n} \ell_{p,i}(\hat{\boldsymbol{\beta}}) - \gamma \frac{s_m}{2|\mathcal{D}| \mathcal{I}_{mm}^{1/2}(\hat{\boldsymbol{\beta}})} \partial_n \mathcal{I}_{mm}(\hat{\boldsymbol{\beta}}) \right) \\
&= \frac{1}{|\mathcal{D}|} \left(\mathcal{I}_{mn}(\hat{\boldsymbol{\beta}}) + \gamma \frac{s_m}{2\, \mathcal{I}_{mm}^{1/2}(\hat{\boldsymbol{\beta}})} \partial_n \mathcal{I}_{mm}(\hat{\boldsymbol{\beta}}) \right),
\end{aligned}$$

and

$$\begin{aligned}
Q_{m,n}(\hat{\boldsymbol{\beta}}, \gamma) &= \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} \phi_{m,i}(\hat{\boldsymbol{\beta}}, \gamma) \cdot \partial_n \ell_{p,i}(\hat{\boldsymbol{\beta}}) \\
&= \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} \left(\partial_m \ell_{p,i}(\hat{\boldsymbol{\beta}}) - \gamma \frac{s_m \mathcal{I}_{mm}^{1/2}(\hat{\boldsymbol{\beta}})}{|\mathcal{D}|} \right) \cdot \partial_n \ell_{p,i}(\hat{\boldsymbol{\beta}}) \\
&= \frac{1}{|\mathcal{D}|} \left(\sum_{i \in \mathcal{D}} \partial_m \ell_{p,i}(\hat{\boldsymbol{\beta}}) \cdot \partial_n \ell_{p,i}(\hat{\boldsymbol{\beta}}) - \gamma \frac{s_m \mathcal{I}_{mm}^{1/2}(\hat{\boldsymbol{\beta}})}{|\mathcal{D}|}\, \partial_n \ell_p(\hat{\boldsymbol{\beta}}) \right),
\end{aligned}$$

where the partial log-partial likelihood $\ell_p(\hat{\boldsymbol{\beta}})$ and other components in $R_{m,n}(\hat{\boldsymbol{\beta}}, \gamma)$ and $Q_{m,n}(\hat{\boldsymbol{\beta}}, \gamma)$ are given in (4.20)-(4.25).

4.4 Simulation Study

In this section we want to demonstrate the performance of the differential geometric relative risk model. Given the fact that other methods have only been implemented for the cox proportional hazards model, our comparison will focus on this model although it is clear that it is easy to extend the method to other relative risk settings.

4.4.1 Comparison with Other Methods.

In this section we compare the dgCox model with three popular algorithms for sparse Cox regression: the coordinate descent method (`glmnet`) developed by [88], the predictor-corrector (`glmpath`) introduced by [68] and the gradient ascent algorithm (`penalized`) proposed by [38].

In our simulation study we generate survival times t_i, $i = 1, 2, \ldots, n$, following exponential distributions with subject-specific parameters $\lambda_i = \exp(\boldsymbol{\beta}^\top X_i)$. The explanatory variables $X_1, \ldots, X_p$ are sampled from a multivariate normal density $N(\mathbf{0}, \Sigma)$ where the entries of Σ are fixed to $corr(X_j, X_k) = \rho^{|j-k|}$ for $\rho \in (0.5, 0.7, 0.9)$. The censorship is randomly assigned to the survival times with probability $\pi \in (0.2, 0.4)$. We fix the sample size n to 50 and the number of predictors p to 100 to emulate a scenario in which $p > n$. From the 100 predictors used, we fix first 30 to 2 and the remaining 70 are set to zero.

For each one of the previous scenarios we generate 100 datasets and we calculate the receiver operating characteristic (ROC) curves for the four methods. In Figure 4.1 we show the averaged ROC curves, which are calculated using the 100 data sets. In scenarios (a) and (b), where $\rho = 0.5$, the four method methods exhibit a similar performance, having overlapping curves for both levels of censorship. A similar performance of the methods has been also observed for combinations of smaller values of ρ and π. In scenarios (c) and (d), where the value of ρ increases to 0.7, the glmnet, glmpath and penalized approaches still overlap, whereas the dgCox model appears to be consistently the best method. In scenarios (e) and (f) where the correlation among neighbouring predictors is high, say $\rho = 0.9$, the dgCox model is clearly the superior approach for both levels of censorship. For the same false positive rate, the true positive rate of the dgCox method is around 10% higher than the rate obtained by the `glmnet`, `glmpath` and `penalized` approaches.

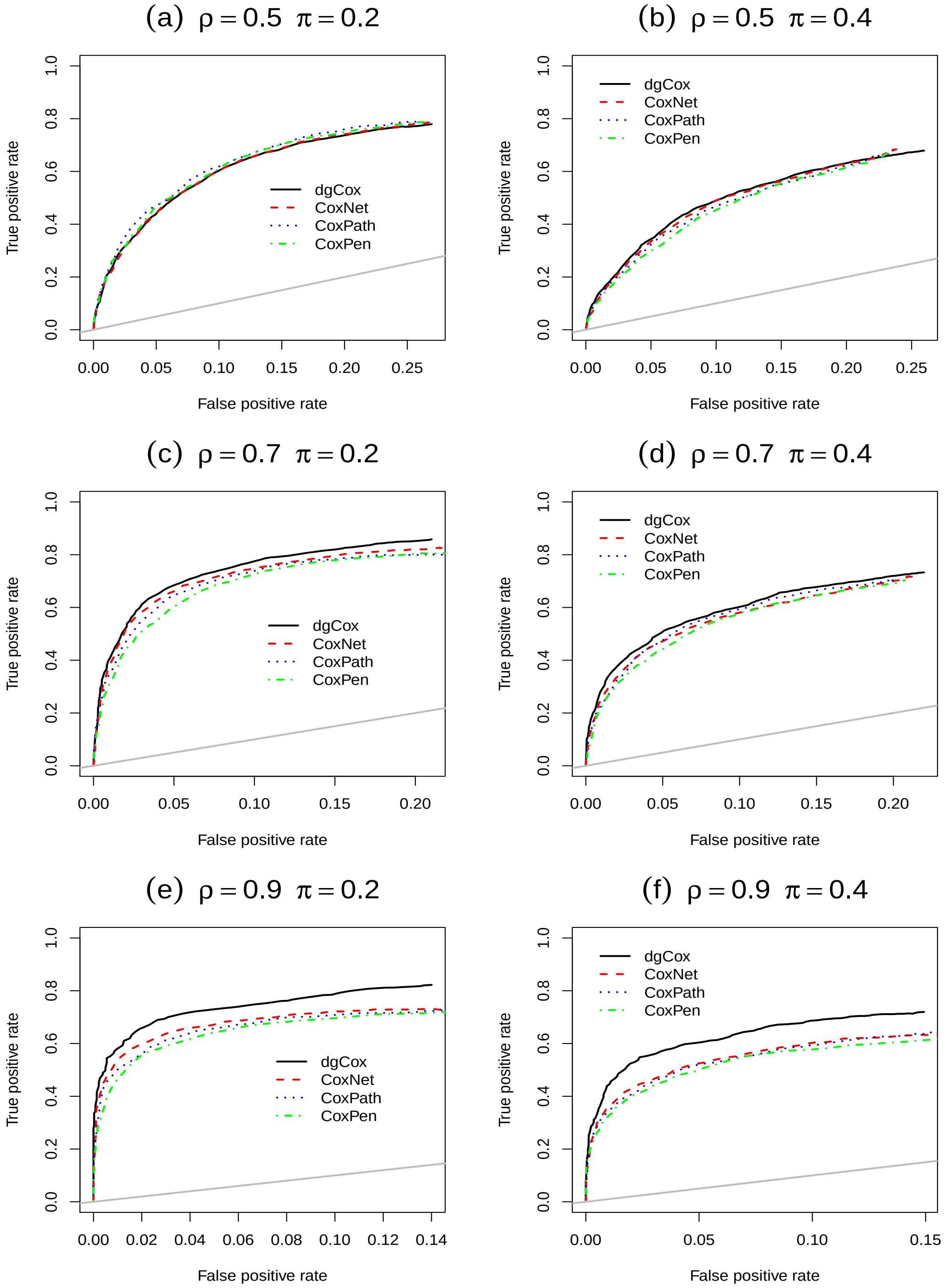


Figure 4.1: Results from the simulation study; for each scenario we show the averaged ROC curve (using 100 datasets) for the dgCox, the coordinate descent method (CoxNet, by glmnet), the predictor-corrector (CoxPath, by glmpath) and the gradient ascent algorithm (CoxPath, by penalized). The 45-degree diagonal is also included in the plots.

In summary, the performance of the four methods is similarly affected by the inclusion of different proportions of censored data. However, the dgCox models is much more efficient in cases in which the predictor variables show significant correlation levels.

4.4.2 Model Selection Comparisons

Under the proposed dgCox model the sparsity of the estimated regression coefficients is controlled by the tuning parameter γ. The simulation study in this section is intended to examine the finite sample performance of a number of model selection criteria. Though these methods take into account both goodness-of-fit and model complexity measures in selecting the tuning parameter, their performance differ in identifying the true model [33]. Hence, in order to effectively identify the true model it is crucial to choose among the model selection criteria. In general, information-based model selection criteria are defined by

$$IC(\gamma) = -2\ell(\hat{\beta}(\gamma)) + C \times comp$$

where $comp$ is a measure of model complexity and the factor C is determined by the type of model selection criteria in use. The minus 2 log-likelihood is commonly used as a measure of model goodness-of-fit. We consider the following model selection criteria:

AIC : Classical AIC with $C = 2$ and $comp = df$.

BIC : Classical BIC with $C = \log(n)$ and $comp = df$.

FAN13 : Another generalized information criterion proposed in Fan and Tang [33] with $C = \log(\log(n)) \times \log(p)$ and $comp = df$.

GIC$_{AIC}$: Generalized information criterion proposed in Konishi and Kitagawa [54] with $C = 2$ and $comp = edf$.

GIC$_{BIC}$: Generalized information criterion proposed in Konishi and Kitagawa [54] with $C = \log(n)$ and $comp = edf$.

GIC$_{FAN}$: A mixture of both with $C = \log(\log(n)) \times \log(p)$ and $comp = edf$.

As the criteria are based on the partial likelihood (rather than the full likelihood) which is a product across the non-censored observations, the effective number of observations in this model is the size of the set of non-censored observations, $n = |\mathcal{D}|$.

The simulation study in this section follows similar data generation mechanism as discussed in Section 4.4.1. We fix a censoring probability $\pi = 0.2$. The sample sizes are $\{50, 200\}$ and the number of predictors p are taken to be $\{50, 100, 1000\}$. This scenario covers $p \geq n$. The level of sparsity in the true model varies: these predictors we fix the coefficients of the first $d = \{2, 8, 32\}$ predictors to 2 and the remaining coefficients are set to zero. The same correlation structure Σ as in Section 4.4.1 is considered with $\rho = 0.9$. For each scenario we simulate 500 data sets and let the dgCox algorithm computes the entire path of the coefficient estimates.Then we use the AIC, BIC, GIC$_{AIC}$, GIC$_{BIC}$, FAN13 and GIC$_{FAN}$ criteria to select the tuning parameter.

We present the simulation results for all the scenarios in Tables 4.1 ($n = 50$) and 4.2 ($n = 200$). We report the median number of variables included in the final model (Size), the average false positive rate (FPR), the false discovery rate (FDR), the false negative rate (FNR) and F1-score (F1) to investigate the performance of the model selection criteria in identifying the true model. The results for $p = 100$ are not reported in Tables 4.1 and 4.2 for sake of brevity, and can be found in the supplementary materials.

The results, in Table 4.1, show that when $p \gg n$ all these model selection methods perform almost equally in terms of FPRs, and when n and p are equal FAN13 performs slightly better than others. However these methods differ in terms of FDRs especially when the true model is very sparse ($d = 2$), in all level of sparsity FAN13 outperforms others. These methods perform almost equally in terms of FNRs, though their performance decrease as the level of sparsity decreases from $d = 2$ predictors having non-zero coefficients to 32. In terms of FNRs, when the true model is very sparse AIC, GIC$_{AIC}$ and GIC$_{BIC}$ perform better than others, and for less sparse models GIC$_{AIC}$ and GIC$_{BIC}$ select the best model. As there is a clear trade-off between FNR and FPR. Therefore, it can be more informative to compare a summary measure, such as the F1-score. When the true model is very sparse FAN13 has the best performance in terms of F1-scores, and for less sparse models GIC$_{AIC}$ and GIC$_{BIC}$ have the better performance than others.

Table 4.1: Results from the simulation studies when $n = 50$; for each scenario we report the median number of variables included in the final model (Size), the mean of the false positive rate (FPR), the false discovery rate (FDR), the false negative rate (FNR) and F1-score (F1). Standard errors are in parentheses. Bold values identify the best models for each scenario.

p	d	$Criterion$	$Size$	FPR	FDR	FNR	$F1$
50	2	AIC	5.000(0.163)	0.072(0.003)	0.475(0.013)	**0.001**(0.001)	0.642(0.011)
		BIC	2.000(0.052)	0.017(0.001)	0.198(0.010)	0.002(0.001)	0.868(0.007)
		FAN13	2.000(0.032)	**0.008**(0.001)	**0.118**(0.008)	0.003(0.002)	**0.924**(0.005)
		GIC$_{AIC}$	4.000(0.160)	0.072(0.003)	0.479(0.013)	**0.001**(0.001)	0.639(0.011)
		GIC$_{BIC}$	2.000(0.052)	0.072(0.003)	0.479(0.013)	**0.001**(0.001)	0.639(0.011)
		GIC$_{FAN}$	2.000(0.035)	0.010(0.001)	0.129(0.009)	0.002(0.001)	0.917(0.006)
	8	AIC	7.000(0.044)	0.002(0.000)	0.010(0.002)	0.157(0.005)	0.906(0.003)
		BIC	7.000(0.043)	**0.001**(0.000)	0.006(0.001)	0.172(0.005)	0.899(0.003)
		FAN13	7.000(0.043)	**0.001**(0.000)	**0.004**(0.001)	0.188(0.005)	0.890(0.003)
		GIC$_{AIC}$	7.000(0.045)	0.002(0.000)	0.010(0.002)	**0.155**(0.005)	**0.907**(0.003)
		GIC$_{BIC}$	7.000(0.044)	0.002(0.000)	0.010(0.002)	**0.155**(0.005)	**0.907**(0.003)
		GIC$_{FAN}$	7.000(0.043)	**0.001**(0.000)	0.006(0.001)	0.176(0.005)	0.896(0.003)
	32	AIC	16.00(0.103)	0.010(0.001)	0.010(0.001)	0.497(0.003)	0.664(0.003)
		BIC	15.00(0.111)	0.005(0.001)	0.005(0.001)	0.542(0.003)	0.624(0.003)
		FAN13	14.00(0.167)	**0.004**(0.001)	**0.004**(0.001)	0.628(0.005)	0.531(0.006)
		GIC$_{AIC}$	16.00(0.105)	0.011(0.001)	0.011(0.001)	**0.493**(0.003)	**0.667**(0.003)
		GIC$_{BIC}$	16.00(0.109)	0.011(0.001)	0.011(0.001)	**0.493**(0.003)	**0.667**(0.003)
		GIC$_{FAN}$	15.00(0.133)	0.005(0.001)	0.006(0.001)	0.552(0.004)	0.612(0.004)
1000	2	AIC	7.000(0.280)	0.007(0.000)	0.622(0.013)	**0.001**(0.001)	0.495(0.012)
		BIC	2.000(0.058)	0.001(0.000)	0.214(0.011)	0.002(0.001)	0.856(0.008)
		FAN13	2.000(0.020)	**0.000**(0.000)	**0.054**(0.006)	0.005(0.002)	**0.964**(0.004)
		GIC$_{AIC}$	8.000(0.325)	0.009(0.000)	0.671(0.012)	**0.001**(0.001)	0.445(0.012)
		GIC$_{BIC}$	2.000(0.088)	0.009(0.000)	0.671(0.012)	**0.001**(0.001)	0.445(0.012)
		GIC$_{FAN}$	2.000(0.022)	**0.000**(0.000)	0.062(0.006)	0.005(0.002)	0.959(0.004)
	8	AIC	6.000(0.060)	**0.000**(0.000)	0.057(0.004)	0.242(0.006)	0.833(0.004)
		BIC	6.000(0.056)	**0.000**(0.000)	0.046(0.003)	0.263(0.006)	0.823(0.004)
		FAN13	6.000(0.052)	**0.000**(0.000)	**0.028**(0.003)	0.330(0.006)	0.785(0.004)
		GIC$_{AIC}$	6.000(0.061)	**0.000**(0.000)	0.058(0.004)	**0.238**(0.006)	**0.834**(0.004)
		GIC$_{BIC}$	6.000(0.058)	**0.000**(0.000)	0.058(0.004)	**0.238**(0.006)	**0.834**(0.004)
		GIC$_{FAN}$	6.000(0.055)	**0.000**(0.000)	0.033(0.003)	0.313(0.006)	0.795(0.004)
	32	AIC	17.00(0.106)	**0.000**(0.000)	0.026(0.002)	0.493(0.003)	0.664(0.003)
		BIC	15.00(0.115)	**0.000**(0.000)	0.013(0.002)	0.529(0.003)	0.634(0.003)
		FAN13	4.000(0.127)	**0.000**(0.000)	**0.003**(0.000)	0.965(0.003)	0.062(0.005)
		GIC$_{AIC}$	17.00(0.107)	0.001(0.000)	0.029(0.003)	**0.488**(0.003)	**0.668**(0.003)
		GIC$_{BIC}$	17.00(0.115)	0.001(0.000)	0.029(0.003)	**0.488**(0.003)	**0.668**(0.003)
		GIC$_{FAN}$	6.000(0.205)	**0.000**(0.000)	0.005(0.001)	0.893(0.006)	0.167(0.009)

Table 4.2: Results from the simulation studies when $n = 200$; for each scenario we report the median number of variables included in the final model (Size), the mean of the false positive rate (FPR), the false discovery rate (FDR), the false negative rate (FNR) and F1-score (F1). Standard errors are in parentheses. Bold values identify the best models for each scenario.

p	d	$Criterion$	$Size$	FPR	FDR	FNR	$F1$
50	2	AIC	5.000(0.246)	0.108(0.005)	0.561(0.012)	**0.000**(0.000)	0.565(0.011)
		BIC	2.000(0.039)	0.011(0.001)	0.145(0.009)	**0.000**(0.000)	0.907(0.006)
		FAN13	2.000(0.029)	**0.007**(0.001)	**0.102**(0.008)	**0.000**(0.000)	**0.936**(0.005)
		GIC$_{AIC}$	5.000(0.198)	0.087(0.004)	0.528(0.012)	**0.000**(0.000)	0.598(0.011)
		GIC$_{BIC}$	2.000(0.038)	0.087(0.004)	0.528(0.012)	**0.000**(0.000)	0.598(0.011)
		GIC$_{FAN}$	2.000(0.030)	**0.007**(0.001)	**0.102**(0.008)	0.001(0.001)	**0.936**(0.005)
	8	AIC	8.000(0.028)	**0.000**(0.000)	**0.000**(0.000)	**0.072**(0.004)	**0.961**(0.002)
		BIC	8.000(0.028)	**0.000**(0.000)	**0.000**(0.000)	**0.072**(0.004)	0.960(0.002)
		FAN13	8.000(0.029)	**0.000**(0.000)	**0.000**(0.000)	**0.072**(0.004)	0.960(0.002)
		GIC$_{AIC}$	8.000(0.028)	**0.000**(0.000)	**0.000**(0.000)	**0.072**(0.004)	**0.961**(0.002)
		GIC$_{BIC}$	8.000(0.028)	**0.000**(0.000)	**0.000**(0.000)	**0.072**(0.004)	0.960(0.002)
		GIC$_{FAN}$	8.000(0.028)	**0.000**(0.000)	**0.000**(0.000)	**0.072**(0.004)	0.960(0.002)
	32	AIC	25.00(0.083)	**0.005**(0.001)	0.004(0.001)	**0.246**(0.003)	**0.858**(0.002)
		BIC	24.00(0.084)	**0.005**(0.001)	**0.003**(0.001)	0.245(0.003)	0.857(0.002)
		FAN13	24.00(0.084)	**0.005**(0.001)	**0.003**(0.000)	0.247(0.003)	0.856(0.002)
		GIC$_{AIC}$	25.00(0.083)	**0.005**(0.001)	0.004(0.001)	**0.246**(0.003)	**0.858**(0.002)
		GIC$_{BIC}$	25.00(0.083)	**0.005**(0.001)	0.004(0.001)	**0.246**(0.003)	**0.858**(0.002)
		GIC$_{FAN}$	25.00(0.083)	**0.005(**0.001)	0.004(0.001)	0.245(0.003)	0.857(0.002)
1000	2	AIC	9.000(0.439)	0.010(0.000)	0.603(0.013)	**0.000**(0.000)	0.446(0.012)
		BIC	2.000(0.041)	**0.000**(0.000)	0.124(0.009)	**0.000**(0.000)	0.920(0.006)
		FAN13	2.000(0.013)	**0.000**(0.000)	**0.023**(0.004)	**0.000**(0.000)	**0.986**(0.002)
		GIC$_{AIC}$	8.000(0.353)	**0.000**(0.000)	0.629(0.013)	**0.000**(0.000)	0.485(0.012)
		GIC$_{BIC}$	2.000(0.040)	0.008(0.000)	0.629(0.013)	**0.000**(0.000)	0.485(0.012)
		GIC$_{FAN}$	2.000(0.013)	**0.000**(0.000)	**0.023**(0.004)	**0.000**(0.000)	**0.986**(0.002)
	8	AIC	6.000(0.028)	**0.000**(0.000)	**0.000**(0.004)	**0.220**(0.003)	**0.874**(0.002)
		BIC	6.000(0.027)	**0.000**(0.000)	**0.000**(0.003)	0.222(0.003)	0.873(0.002)
		FAN13	6.000(0.025)	**0.000**(0.000)	**0.000**(0.003)	0.230(0.006)	0.868(0.002)
		GIC$_{AIC}$	6.000(0.028)	**0.000**(0.000)	**0.000**(0.004)	**0.220**(0.003)	**0.874**(0.002)
		GIC$_{BIC}$	6.000(0.027)	**0.000**(0.000)	**0.000**(0.004)	**0.220**(0.003)	**0.874**(0.002)
		GIC$_{FAN}$	6.000(0.026)	**0.000**(0.000)	**0.000**(0.003)	0.230(0.003)	0.868(0.002)
	32	AIC	23.00(0.106)	**0.000**(0.000)	0.004(0.001)	**0.300**(0.003)	**0.821**(0.002)
		BIC	22.00(0.115)	**0.000**(0.000)	0.004(0.001)	0.304(0.003)	0.818(0.002)
		FAN13	21.00(0.087)	**0.000**(0.000)	**0.003**(0.000)	0.352(0.003)	0.783(0.003)
		GIC$_{AIC}$	23.00(0.107)	**0.000**(0.000)	0.004(0.001)	**0.300**(0.003)	**0.821**(0.002)
		GIC$_{BIC}$	23.00(0.115)	**0.000**(0.000)	0.004(0.001)	**0.300**(0.003)	**0.821**(0.002)
		GIC$_{FAN}$	22.00(0.205)	**0.000**(0.000)	**0.003**(0.000)	0.340(0.003)	0.792(0.002)

On the other hand, as the sample size increases from 50 to 200, the performance of the six model selection criteria significantly improves especially when the true model is not too sparse.From Table 4.2, all the methods have the same performance in terms of FPRs and FDRs when the true model is less sparse, while for a very sparse true model FAN13 and GIC$_{FAN}$ perform better than others. In all scenarios these methods have the same performance in terms of FNRs.Moreover, the FAN13 and GIC$_{FAN}$ criteria select the best model for very sparse models in terms of F1-scores, whereas for less sparse models AIC, GIC$_{AIC}$ and GIC$_{BIC}$ have the same F1-scores and perform slightly better than others.

Summarizing, we found that in very sparse contexts, i.e., where not only $p \gg n$ but also the true number of effects is small ($d \ll p$), FAN13 performs well in terms of F1-score, which is a weighted average of the FPR and the FNR. In other settings GIC$_{AIC}$ and GIC$_{BIC}$ perform also well, slightly beating FAN13. Although in all scenarios GIC$_{AIC}$ and GIC$_{BIC}$ have the same performance in terms of FPRs, FDRs, FNRs and F1-scores, the GIC$_{BIC}$ selects the accurate size for the final model especially when the model is very sparse, and therefore it performs slightly better than GIC$_{AIC}$ overall.

4.5 Finding Genetic Signatures in Cancer Survival

In this section we test the predictive power of dgCox in four recent studies. In particular, we focus on the identification of genes involved in the regulation of colon cancer [59], prostate cancer [81], ovarian cancer [37] and skin cancer [49]. The set-up of the four studies was similar. In the patient cancer was detectedand treated. At the time of treatment a follow-up was started. In all cases, the expression of several genes were measured in the affected tissue together with the survival times of the patients, which is assumed to be censored if the patients were alive when they left the study. Although other socio-economical variables, such us age, sex, etc. are available, our analysis only focuses on the impact of the gene expression levels on the patients survival.

Table 4.3 contains a brief description of the four datasets used in this section. In the four scenarios p is larger than n. The dimensionality is especially high in the cases of the colon and skin cancer where several thousands of genes were

Table 4.3: Description of the four cancer experiments studied in this section. The four datasets are available at http://www.ncbi.nlm.nih.gov/.

Cancer	n	# uncenso.	p	# genes selec.	G.W. test	Reference
Colon	125	70	23698	62	0.0224	[59]
Prostate	61	24	162	33	0.0333	[81]
Ovarian	103	57	306	48	0.0039	[37]
Skin	54	47	30807	21	0.025	[49]

used in the studies. In the prostate and ovarian cancers the number of covariates is 162 and 306, which will also help us to study the performance of dgCox when the number of variables is just a few orders of magnitude larger than the number of observations.

In genomic studies it is a common hypothesis to assume that just a few number of genes affect the dependent variable of interest. To identify such genes in our survival data analysis context, we estimate a relative hazard risk model using the dgLARS algorithm described in Section 4.3. To this end, we randomly select a training sample that contains the 60% of the patients and we save the remaining data to test the models. We calculate the paths coefficients in the four scenarios and we select the optimal number of components by means of the GIC_BIC criterion derived in Section 4.4.2. The number of selected genes in each case is detailed in Table 4.3 ranging from 21 genes in the skin cancer data set to 62 in the colon dataset.

In order to illustrate the prediction performance of the dgLARS method we classify the test patients into a low-risk group and a high-risk groups by splitting the test sample into two subsets of equal size according to the individual predicted excess risk $\beta^\top \mathbf{X}$. To test the groups separation we use the non-parametric Peto & Peto modification of the Gehan-Wilcoxon test [71]. The p-values obtained in the four scenarios are shown in Table 4.3. In Figure 4.2 we show the Kaplan-Maier survival curves estimates for the two groups in together with the original training survival curve. The differences are significant in the four cases showing the predictive power of the survival function provided by the selected genes. This results demonstrates the power of dgLARS as a tool in medical analysis for massive gene screening studies.

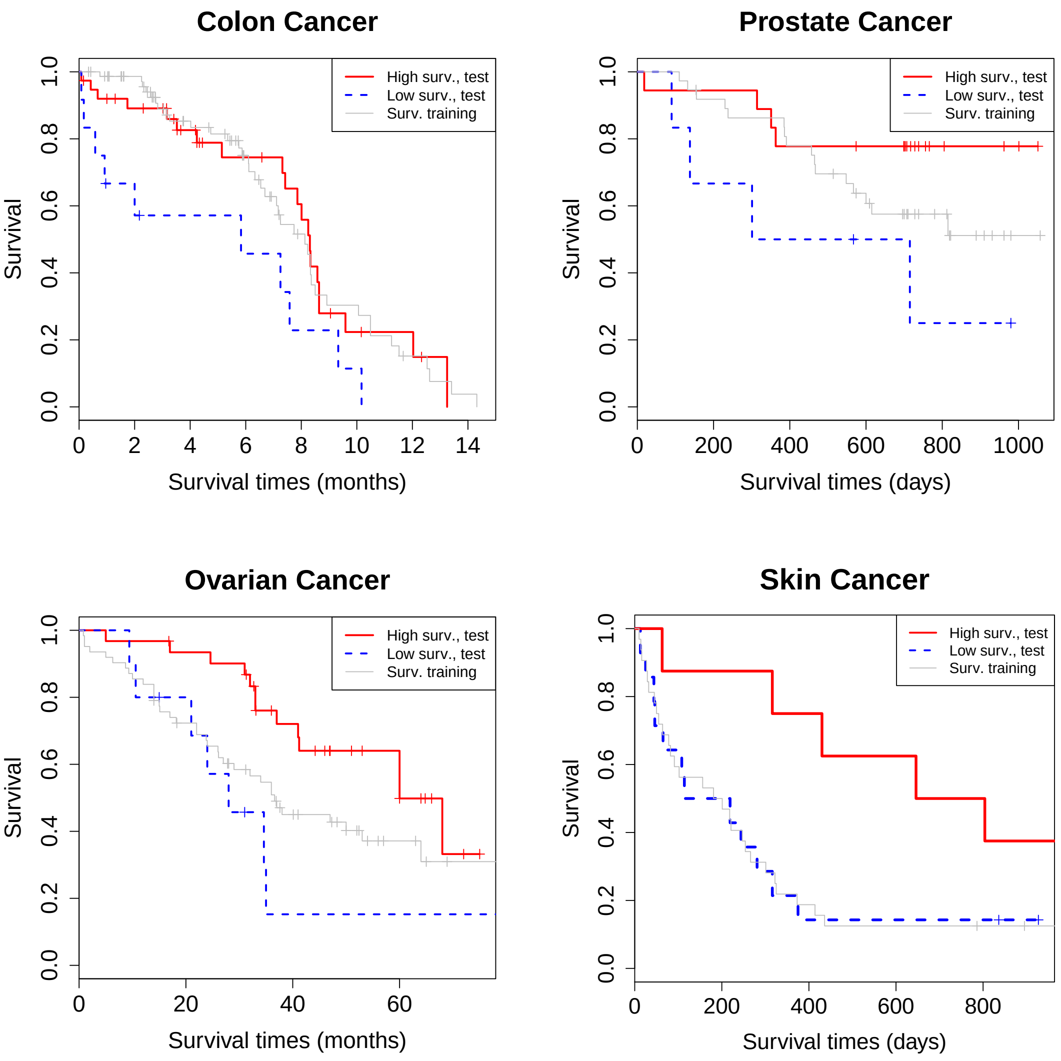
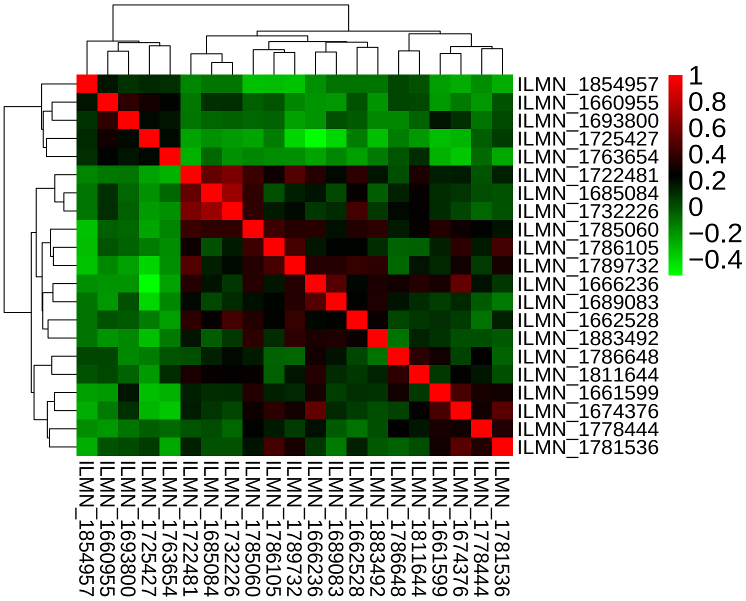


Figure 4.2: Illustration of the results obtained in the four datasets. The Kaplan-Meier survival curves estimates for training data are shown together with the curves associated to the two groups obtained in the test sample by means of the predicted excess risk $\beta^\top \mathbf{X}$. In the four cases, the two groups in the test sample show a significant separation according the Peto & Peto modification of the Gehan-Wilcoxon test.

4.5.1 Enrichment Analysis of the Found Genes Relevant for Skin Cancer

To gain some biological understanding of the process of cancer regulation we performed an enrichment analysis of the 21 genes that have been found to

Figure 4.3: Heatmap of the correlations of the 21 selected genes that have been found to be influential in the skin cancer. In terms of the genes correlations two main groups are apparent.



be relevant in the regulation of the skin cancer.

We used DAVID (https://david.ncifcrf.gov/) to identity and annotate the 21 genes, using the available Illumina IDs. Interestingly, the three unidentified genes (ILMN_1854957, ILMN_1693800 and ILMN_1660955) along with two genes ILMN_1763654 and ILMN_1725427 show a very low correlation among each other but a high and negative variance with the remaining 16 genes. This can be seen in the heatmap of the correlations of the 21 selected genes presented in Figure 4.3.

The IDs of the selected genes and a brief description provided by DAVID are detailed in Table 4.4. Within the 21 selected genes we found genes associated to transcription factors (ILMN_1689083), conjugating enzymes (ILMN_1789732) or DNA-damage-inducible transcripts (ILMN1_661599).

To provide further insight in the group of genes, we performed a Gene Ontology (GO) annotation based on three groups of GO terms: Molecular function, biological process and cellular component. In addition, we associated the 18 identified genes with the protein family. In Table 4.5 we account for the number of genes in each one of the categories of the GO terms, the percentage of the sam-

Table 4.4: Illumina IDs of the selected genes which have been found to be influential in the skin cancer. A short description of the genes provided by the application DAVID is included.

ILLUMINA ID	Short description
ILMN1689083	general TF IIH, polypeptide 2, 44kDa; general TF IIH, polypeptide 2C; general TF IIH, polypeptide 2B; general TF IIH, polypeptide 2D
ILMN_1674376	angiopoietin-like 4 Homo sapiens
ILMN_1786648	periaxin Homo sapiens
ILMN_1781536	fumarylacetoacetate hydrolase (fumarylacetoacetase) Homo sapiens
ILMN_1811644	family with sequence similarity 106, member A-like; family with sequence similarity 106, member A; family with sequence similarity 106, member B
ILMN_1763654	DENN/MADD domain containing 1B
ILMN_1685084	FK506 binding protein 7
ILMN_1785060	tetraspanin 14
ILMN_1666236	high mobility group AT-hook 2
ILMN_1661599	DNA-damage-inducible transcript 4
ILMN_1786105	pterin-4 alpha-carbinolamine dehydratase/dimerization cofactor of hepatocyte nuclear factor 1 alpha
ILMN_1732226	DEAH (Asp-Glu-Ala-Asp/His) box polypeptide 57
ILMN_1778444	FK506 binding protein 5
ILMN_1883492	hypothetical LOC728152
ILMN_1725427	beta-2-microglobulin
ILMN_1662528	KIAA0947
ILMN_1789732	ubiquitin-conjugating enzyme E2 variant 1; ubiquitin-conjugating enzyme E2 variant 1 pseudogene 2; transmembrane protein 189; TMEM189-UBE2V1 readthrough transcript
ILMN_1722481	COX15 homolog, cytochrome c oxidase assembly protein (yeast)
ILMN_1854957	Not identified
ILMN_1693800	Not identified
ILMN_1660955	Not identified

ple of 18 identified genes sample in each GO term (associated to the previous groups), and the percentage of genes in each category of each group. Regarding the molecular function, we observe that 8 of the 18 genes are associated with the catalytic activity. This result agrees with previous studies in skin cancer that state that the telomerase complex activity is dependent on its catalytic subunit [16]. Regarding the biological processes, 8 of the 18 genes are associated to cellular processes and 9 of them are related to metabolic processes, which are also known to affect this disease [64]. Organelles and other cells parts are the cellular components mainly represented in the selected group of genes. Finally, it is known that over-expression of Isomerase is related to different types of cancer, including skin cancer [15]. Interestingly, 3 genes are associated with this

Table 4.5: Annotation of the 21 selected genes by dgLARS in the Skin cancer dataset. The genes are grouped in terms of the gene ontology (GO) molecular function, biological process, cellular components, and the protein class (PC). For each category of the previous groups we compute the number of genes in the 21-genes sample, the % of the 21-genes sample in each GO term, and the % of genes in each category of each group.

Group	#genes	% sample	% group
Molecular Function			
Nucleic acid binding TF. activity (GO:0001071)	2	10.5%	10.5%
Binding (GO:0005488)	8	42.1%	42.1%
Receptor activity (GO:0004872)	1	5.3%	5.3%
Catalytic activity (GO:0003824)	8	42.1%	42.1%
Biological Process			
Reproduction (GO:0000003)	1	5.3%	3.7%
Response to stimulus (GO:0050896)	4	21.1%	14.8%
Immune system process (GO:0002376)	2	10.5%	7.4%
Cellular process (GO:0009987)	8	42.1%	29.6%
Metabolic process (GO:0008152)	9	47.4%	33.3%
Biological regulation (GO:0065007)	1	5.3%	3.7%
Biological adhesion (GO:0022610)	2	10.5%	7.4%
Cellular Component			
Membrane (GO:0016020)	2	10.5%	13.3%
Macromolecular complex (GO:0032991)	1	5.3%	6.7%
Cell part (GO:0044464)	6	31.6%	40.0%
Organelle (GO:0043226)	5	26.3%	33.3%
Extracellular region (GO:0005576)	1	5.3%	6.7%
Protein Class			
Chaperone (PC00072)	2	10.5%	11.8%
Hydrolase (PC00121)	1	5.3%	5.9%
Cell adhesion molecule (PC00069)	1	5.3%	5.9%
Lyase (PC00144)	1	5.3%	5.9%
Transcription factor (PC00218)	1	5.3%	5.9%
Nucleic acid binding (PC00171)	2	10.5%	11.8%
Receptor (PC00197)	1	5.3%	5.9%
Defense/immunity protein (PC00090)	1	5.3%	5.9%
Calcium-binding protein (PC00060)	2	10.5%	11.8%
Isomerase (PC00135)	3	15.8%	17.6%
Signaling molecule (PC00207)	2	10.5%	11.8%

protein.

4.6 Conclusions

In this chapter, we have introduced a general path-finding algorithm for high-dimensional relative risk regression models, not based on an arbitrary penalty, but on the underlying geometric structure of the partial likelihood. The advantage of this method is that the estimates are invariant to arbitrary changes in the measurement scales of the covariates. Unlike SCAD or L_1 sparse regression methods, no prior rescaling of the covariates is therefore needed. The method can be used for a large class of survival models and we have implementations for the Cox proportional hazards model and the excess relative risk model.

We have introduced and compared several model selection criteria for these sparse relative risk survival models through simulation studies. As our method involves shrinkage of the parameters, the issue of the underlying degrees of freedom of the sparse models is a complex one. We derive an estimator based on the Generalized Information Criterion [55], which performs particularly well when the true model is less sparse, whereas for a very sparse true model, the method by [33] performs well.

The method has been implemented in an efficient R package, that can deal with the high-dimensional and $n << p$ settings, such as, for example, a skin cancer study with $p = 30,807$ predictors and $n = 54$ observations. We consider four recent cancer survival studies, where we look for a genetic "survival signature". Due to the large number of predictors, the studies are unsuitable for traditional survival regression methods. Instead, the results we find go beyond univariate importance and by means of an enrichment study can be linked to potentially interesting biological explanations.

Chapter 5

A Software Tool for Estimating the Dispersion Parameter for High-dimensional GLMs

Contents

Abstract

Since the value of the dispersion parameter ϕ affects the value of the log-likelihood function, the value of various information criteria such as AIC and BIC can be affected, and so considerations about the selection of the optimal model are going to be significantly affected. In this chapter, we explain the improved estimator of the dispersion parameter, proposed in [70], for high-dimensional exponential dispersion generalized linear models, called General Refitted Cross-Validation (GRCV) estimator with an algorithm to improve the proposed estimator to obtain a more accurate estimator. Several dispersion parameter estimation methods and algorithms for computing the dgLARS solution curve, proposed in [13] and [70], are implemented in the new version of the R-package **dglars**. A numerical study is conducted to compare the proposed methods and algorithms. The proposed methods by means of the new functions of the package are applied to analyze a real dataset.

Keywords: *Dispersion parameter, dgLARS, Sparsity, High-dimensional GLMs,* **dglars**.

5.1 Introduction

Modern statistical methods developed to study high-dimensional data sets, namely data sets where the number of predictors, say p, is larger than the sample size n, are usually based on the idea to use a penalty function to estimate a solution curve embedded in the parameter space and then to find the point that represents the best compromise between sparsity and predictive behaviour of the model. Recent statistical literature has a great number of contributions devoted to this problem, such as the ℓ_1-penalty function [94], the SCAD method [31] and the Dantzig selector [20].

Differently from the methods cited above, [13] proposed a new approach based on the differential geometrical representation of a GLM. The derived method, that does not require an explicit penalty function, has been called differential geometric LARS (dgLARS) method because it is defined generalizing the geometrical ideas on which the least angle regression (LARS), proposed in [29], is based. [70] extended the dgLARS method to the high-dimensional GLMs based on the exponential dispersion models with arbitrary link functions. In the same paper the authors proposed a classical estimation of the dispersion parameter based on high-dimensional feature space and also a new estimation method showed that is more accurate than the classical estimator.

From a computational point of view, the dgLARS method consists essentially in the computation of the implicitly defined solution curve. In [13] this problem is satisfactorily solved by using a predictor-corrector (PC) algorithm, that however has the drawback of becoming intractable when working with thousands of predictors. From a computational point of view, using the PC algorithm lead to an increase in the run times needed for computing the solution curve. In this chapter we explain an improved version of the PC algorithm (IPC), proposed in [70], to decrease the effects stemming from this problem for computing the solution curve. The IPC algorithm allows the dgLARS method to be implemented using less steps, greatly reducing the computational burden because of reducing the number of points of the solution curve. In addition these two algorithms [12] proposed a much more efficient cyclic coordinate descend (CCD) algorithm to fit the dgLARS solution curve when we work with a high-dimensional data set. Although this algorithm is computationally fast, the solution curve (parameter estimation) is not accurate. We focus only on the PC and IPC algorithms, although all three algorithms are available in the new version of the R-package **dglars** [9]. The package is available on the Comprehensive R Archive Network (CRAN) at http://CRAN.R-project.org/package=dglars.

The remaining of this paper is organized as follows. In Section 5.2 we briefly review the differential geometrical theory underlying the dgLARS method and briefly explain the dispersion parameter estimation methods. In Section 5.3 is devoted to the description of some functions implemented in the **dglars** package that can be used to estimate the dispersion parameter, and also use the functions implemented in the package to compare run times between two different algorithms. In Section 5.4, by simulation studies we compare the behavior of the proposed estimation methods and also run times between the PC and IPC algorithms. In Section 5.5 we use the functions implemented in the **dglars** package to study two real data sets, and finally, in Section 5.6 we draw some conclusions.

5.2 Methodological Background

In this section we describe very briefly the dgLARS method and the dispersion parameter estimation methods. The interested reader is referred to [11] and [70]. The dgLARS method defines a continuous solution path for GLM, and the aim of the method is to define a continuous model path with highest likelihood with the fewest number of variables.

5.2.1 dgLARS Method

Let $\mathcal{Y}$ be a scalar random variable with probability density function belonging to the exponential family $p(y;\theta,\phi) = \exp\{(y\theta - b(\theta))/a(\phi) + c(y,\phi)\}$, where $\theta \in \Theta \subseteq \mathcal{R}$ is called canonical parameter, $\phi \in \Phi \subseteq \mathcal{R}^+$ is called dispersion parameter and $a(\cdot)$, $b(\cdot)$ and $c(\cdot,\cdot)$ are specific given functions. We shall assume that Θ is an open set. The expected value of $\mathcal{Y}$ is related to the canonical parameter by the mean value mapping, namely $\mathrm{E}(\mathcal{Y}) = \mu = \tau(\theta) = \partial b(\theta)/\partial\theta$, where $\tau : int(\Theta) \rightarrow \Omega$. Similarly, the variance of $\mathcal{Y}$ is related to its expected value by the identity $\mathrm{Var}(\mathcal{Y}) = a(\phi)\mathrm{V}(\mu)$, where $\mathrm{V}(\mu)$ is called variance function. Since μ is a reparameterization of the model, in the following of this paper we denoted by $p(y;\mu,\phi)$ the probability density function of $\mathcal{Y}$. Let $\mathcal{X}$ be the p-dimensional vector of random predictors. Under this setting a GLM is based on the assumption that the conditional expected value of $\mathcal{Y}$ give $\mathcal{X} = \mathbf{x}$ is specified by the link function $g(\cdot)$, namely $g(\mathrm{E}(\mathcal{Y}|\mathcal{X} = \mathbf{x})) = \beta_0 + \sum_{m=1}^{p} x_m \beta_m$. For the notation purposes it is more convenient to denote $g^{-1}(\beta_0 + \sum_{m=1}^{p} x_m \beta_m) = \mu(\mathbf{x}^\top \boldsymbol{\beta}) = \mu(\eta)$, where $x_0 = 1$. When we work with n independent and identically distributed copies of the the pair $(\mathcal{Y}, \mathcal{X})$, the marginal distribution of the n-dimensional random vector $\mathcal{Y} = (\mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_n)^\top$ is an element of the set

$$\mathcal{S} = \left\{ p(\mathbf{y}; \boldsymbol{\mu}, \phi) = \prod_{i=1}^{n} p(y_i; \mu_i, \phi) : \boldsymbol{\mu} \in \Omega^n, \phi \in \mathcal{R}^+ \right\},$$

which is a minimal and regular exponential family of order n then it can be treated as a differential manifold in which $\boldsymbol{\mu}$ is a coordinate system [5]. For a rigorous definition of a differential manifold the reader is referred to [90]. The tangent space of $\mathcal{S}$ at the point $p(\mathbf{y}; \boldsymbol{\mu})$ is defined as the linear vector space spanned by the n score functions $\partial_i \ell(\boldsymbol{\mu}, \phi; \mathcal{Y}) = \partial \log p(\mathcal{Y}; \boldsymbol{\mu}, \phi)/\partial \mu_i$, namely

$$T_{p(\boldsymbol{\mu})}\mathcal{S} = \mathrm{span}\{\partial_1 \ell(\boldsymbol{\mu}, \phi; \mathcal{Y}), \partial_2 \ell(\boldsymbol{\mu}, \phi; \mathcal{Y}), \ldots, \partial_n \ell(\boldsymbol{\mu}, \phi; \mathcal{Y})\}.$$

In order to study the geometrical structure of a GLM, we shall assume that $\boldsymbol{\beta} \to \{g^{-1}(\mathbf{x}_1^\top \boldsymbol{\beta}), \ldots, g^{-1}(\mathbf{x}_n^\top \boldsymbol{\beta})\}^\top = \boldsymbol{\mu}(\boldsymbol{\beta})$ is an embedding, this means that the set $\mathcal{M} = \{p_{\mathbf{Y}}(\mathbf{y}; \boldsymbol{\mu}(\boldsymbol{\beta}), \phi) \in \mathcal{S} : \boldsymbol{\beta} \in \mathcal{R}^{p+1}, \phi \in \mathcal{R}^+\}$ is a $p+1$-dimensional submanifold of $\mathcal{S}$, which inherits the dualistic structure from its ambient space, then, as a simple consequence of theorem 3.5 in [6], $\mathcal{M}$ is a dually flat space only when we work with the canonical link function. The tangent space of $\mathcal{M}$ at the point $p(\mathbf{y}; \boldsymbol{\mu}(\boldsymbol{\beta}), \phi)$, denoted by $T_{p(\boldsymbol{\mu}(\boldsymbol{\beta}))}\mathcal{M}$, is the linear vector space spanned by the $p+1$ score functions $\partial_m \ell(\boldsymbol{\beta}, \phi; \mathcal{Y}) = \partial \log p(\mathcal{Y}; \boldsymbol{\mu}(\boldsymbol{\beta}), \phi)/\partial \beta_m$.

The dgLARS estimator is based on a differential geometric characterization of the Rao score test statistic, which is obtained considering the inner product between the bases of the tangent space $T_{p(\boldsymbol{\mu}(\boldsymbol{\beta}))}\mathcal{M}$ and the tangent residual vector $\mathbf{r}(\boldsymbol{\beta}, \phi, \mathbf{y}; \mathbf{Y}) = \sum_{i=1}^{n} r_{\boldsymbol{\beta},i} \, \partial_i \ell(\boldsymbol{\beta}, \phi; \mathbf{Y})$, where $r_{\boldsymbol{\beta},i} = y_i - \mu_i(\boldsymbol{\beta})$. The dgLARS method is a sequential method developed to estimate a sparse solution curve embedded in the in the parameter space $\mathcal{B}$. To explore the sparse structure of a GLM, we can use the following differential geometric characterization of the

m^{th} element of the score vector, i.e.,

$$\begin{aligned}\partial_m \ell(\boldsymbol{\beta}; \mathbf{Y}) &= \langle \partial_m \ell(\boldsymbol{\beta}; \mathbf{Y}); \mathbf{r}(\boldsymbol{\beta}, \mathbf{y}; \mathbf{Y}) \rangle_{p(\boldsymbol{\mu}(\boldsymbol{\beta}))} \\ &= \cos(\rho_m(\boldsymbol{\beta})) \cdot ||\mathbf{r}(\boldsymbol{\beta}, \mathbf{y}; \mathbf{Y})||_{p(\boldsymbol{\mu}(\boldsymbol{\beta}))} \cdot \mathcal{I}_{mm}^{1/2}(\boldsymbol{\beta}),\end{aligned} \tag{5.1}$$

where $\mathcal{I}_{mm}(\boldsymbol{\beta})$ is the Fisher information for β_m, and $\rho_m(\boldsymbol{\beta})$ is a generalization of the Euclidean notion of angle between the m^{th} column of the design matrix and the residual vector $\mathbf{r}_{\boldsymbol{\beta}} = (r_{\boldsymbol{\beta},i})_{i=\{1,2,\ldots,n\}}$. The dispersion parameter can be deleted out of the equation [70].

Importantly, Equation (5.1) shows that the gradient of the log-likelihood function does not generalize the equiangularity condition proposed in [29] to define the LARS algorithm, since the latter does not consider the variation related to $\mathcal{I}_{mm}^{1/2}(\boldsymbol{\beta})$, which in the case of a GLM is typically not constant. One can see that the signed Rao score test statistic can be geometrically characterized as follows:

$$\begin{aligned}r_m(\boldsymbol{\beta}) &= \mathcal{I}_{mm}^{-1/2}(\boldsymbol{\beta}) \cdot \partial_m \ell(\boldsymbol{\beta}; \mathbf{Y}) \\ &= \cos(\rho_m(\boldsymbol{\beta})) \cdot \|\mathbf{r}(\boldsymbol{\beta}, \mathbf{y}; \mathbf{Y})\|_{p(\boldsymbol{\mu}(\boldsymbol{\beta}))}.\end{aligned} \tag{5.2}$$

This equation shows that, for generalized linear models, we can define dgLARS with respect to the Rao score test statistics, rather than the angles. From Equation (5.2) we shall say that two given predictors, say m and n, satisfy the generalized equiangularity condition at the point $\boldsymbol{\beta}$ when $|r_m(\boldsymbol{\beta})| = |r_n(\boldsymbol{\beta})|$. Inside the dgLARS theory, the generalized equiangularity condition is used to identify the predictors that are included in the active set. Formally, for a given value of the Rao score test statistic $\gamma \in \mathcal{R}^+$ the corresponding active set is denoted by $\hat{\mathcal{A}}(\gamma)$ and the dgLARS estimator, denoted by $\hat{\boldsymbol{\beta}}(\gamma)$, is such that the following conditions are satisfied:

$$\forall\, m \in \hat{\mathcal{A}}(\gamma) \quad \Rightarrow \quad r_m(\hat{\boldsymbol{\beta}}(\gamma)) = s_m \gamma, \tag{5.3}$$

$$\forall\, m \in \hat{\mathcal{A}}^c(\gamma) \quad \Rightarrow \quad \left| r_m(\hat{\boldsymbol{\beta}}(\gamma)) \right| < \gamma, \tag{5.4}$$

where $s_m = \text{sign}(\hat{\beta}_m(\gamma))$ and $\hat{\mathcal{A}}^c(\gamma)$ is the complement of the active set.

[29] show that the LASSO solution curve can be obtained by a simple modification of the LARS method. Let $\hat{\boldsymbol{\beta}}(\gamma)$ be the solution of a GLM penalized

using the ℓ_1-penalty function, then it is easy to show that the sign of any non-zero coefficient has to agree with the sign of the score function, namely $sign(\partial_m \ell(\hat{\boldsymbol{\beta}}(\gamma); \mathbf{y})) = sign(\hat{\beta}_m(\gamma))$. When this condition is violated the corresponding predictor is removed from the active set. The dgLARS method can be easily modified to compute a differential geometric extension of the LASSO solution curve, called the dgLASSO solution curve. For more details about the dgLASSO method see [11]. The dgLASSO estimator of a GLM is given by Equations (5.3) and (5.4), such that also the following restrictions on the signs of the non-zero coefficients are satisfied

$$\text{sign}(r_m(\hat{\boldsymbol{\beta}}(\gamma))) = \text{sign}(\hat{\beta}_m(\gamma)), \ \ \forall m \in \mathcal{A}(\gamma),$$

where $\gamma \in [0, \gamma_{\max}]$ is a fixed value.

The dgLASSO method computes the dgLASSO solution curve in the same way as the dgLARS method, but it removes a predictor from the active set when the sign of the corresponding estimate is not in agreement with the sign of the Rao score test statistic. Although for the LASSO and SCAD estimators the problem of how to estimate the dispersion parameter ϕ is still an open question and theoretical results are not available, for the dgLARS and dgLASSO estimators [70] tried to present a dispersion parameter estimation method. For this, in Section 5.2.3, we will give a briefly description on the dispersion parameter estimation methods.

5.2.2 Estimation of the dgLARS Solution Path

From a computational point of view, the problem of how to estimate the dgLARS solution curve can be formalized in the following way. Formally, the dgLARS method computes a finite sequence of transition points, say $0 \leq \gamma^{(p)} \leq \ldots \leq \gamma^{(2)} \leq \gamma^{(1)}$, such that for each $\gamma^{(k)}$, where $2 \leq k \leq p$, the following condition can occur:

$\exists m \in \hat{\mathcal{A}}^c(\gamma^{(k-1)})$ such that

$$\left| r_m(\hat{\boldsymbol{\beta}}(\gamma^{(k)})) \right| = \gamma^{(k)} \tag{5.5}$$

then $\hat{\mathcal{A}}(\gamma^{(k)}) = \hat{\mathcal{A}}(\gamma^{(k-1)}) \cup \{m\}$, which means that a new predictor is included in the active set when the generalized equiangularity condition is satisfied. When we want to estimate the dgLASSO solution curve, it is necessary to add the following condition

$$\begin{aligned} &\exists m \in \hat{\mathcal{A}}(\gamma^{(k-1)}) \ \text{ such that} \\ &\qquad\qquad\qquad\qquad \operatorname{sign}(r_m(\hat{\boldsymbol{\beta}}(\gamma^{(k)}))) \neq \operatorname{sign}(\hat{\beta}_m(\gamma^{(k)})) \\ &\text{then } \hat{\mathcal{A}}(\gamma^{(k)}) = \hat{\mathcal{A}}(\gamma^{(k-1)}) \setminus \{m\}, \end{aligned} \tag{5.6}$$

which means that an active predictor is removed from the active set if the sign of the corresponding signed Rao score test statistic is not in agreement with the sign of the estimated coefficient. In order to simplify our notation, in the following of this section we shall assume that $\hat{\mathcal{A}}(\gamma) = \{0, 1, 2, \ldots, k\}$, where the index 0 stands for the intercept. Observing that for each $\gamma \in (\gamma^{(k+1)}; \gamma^{(k)}]$ the signs of the estimated coefficients do not change, condition (5.3) tells us that, for a fixed value of the tuning parameter γ, the dgLARS estimator can be defined as the Z-estimator implicitly defined by the following system of estimating equations:

$$\begin{cases} r_0(\hat{\boldsymbol{\beta}}(\gamma)) & = & 0 \\ r_1(\hat{\boldsymbol{\beta}}(\gamma)) - s_1\gamma & = & 0 \\ r_2(\hat{\boldsymbol{\beta}}(\gamma)) - s_2\gamma & = & 0 \\ \vdots & & \vdots \\ r_k(\hat{\boldsymbol{\beta}}(\gamma)) - s_k\gamma & = & 0. \end{cases} \tag{5.7}$$

where $s_i = \operatorname{sign}(\hat{\beta}_i(\gamma))$.

[13] proposed to use a predictor-corrector (PC) method to compute the dgLARS/dgLASSO solution curve. From a computational point of view, using the PC algorithm lead to an increase in the run times needed for computing the solution curve. In the following of the section, we briefly review the improved version of the PC algorithm to decrease the effects stemming from this problem for computing the solution curve. For more details the interested reader is referred to [70]. We define $\tilde{\boldsymbol{\varphi}}_{\mathcal{A}}(\gamma) = \boldsymbol{\varphi}_{\mathcal{A}}(\gamma) - \mathbf{v}_{\mathcal{A}}\gamma$, where $\boldsymbol{\varphi}_{\mathcal{A}}(\gamma) = (\partial_0 \ell(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma); \mathbf{y}), r_1(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)), \cdots, r_k(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)))^\top$ and $\mathbf{v}_{\mathcal{A}} = (0, v_1, \ldots, v_k)^\top$. By differentiating $\tilde{\boldsymbol{\varphi}}_{\mathcal{A}}(\gamma)$ with respect to γ, we can locally approximate the solu-

tion curve at $\gamma - \Delta\gamma$ by the following expression

$$\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma - \Delta\gamma) \approx \tilde{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma - \Delta\gamma) = \hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma) - \Delta\gamma \cdot \left(\frac{\partial \boldsymbol{\varphi}_{\mathcal{A}}(\gamma)}{\partial \hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)} \right)^{-1} \mathbf{v}_{\mathcal{A}} \,, \tag{5.8}$$

where $\Delta\gamma \in [0; \gamma - \gamma^{(k+1)}]$ and $\partial \boldsymbol{\varphi}_{\mathcal{A}}(\gamma) / \partial \hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)$ is the Jacobian matrix of the vector function $\boldsymbol{\varphi}_{\mathcal{A}}(\gamma)$ evaluated at the point $\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)$. An efficient implementation of the improved PC method requires a suitable method to compute the smallest step size $\Delta\gamma$ that changes the active set of the non-zero coefficients. For each $m^c \in \mathcal{A}^c(\gamma)$ we have a value for $\Delta\gamma^{m^c}$ as follows

$$\Delta\gamma^{m^c} = \begin{cases} \Delta\gamma_1 & \text{if} \quad 0 \leq \Delta\gamma_1 \leq \gamma; \\ \Delta\gamma_2 & \text{if} \quad o.w. \end{cases}$$

with

$$\Delta\gamma_1 = \frac{\gamma - r_{m^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))}{1 - \dfrac{d r_{m^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))}{d\gamma}} \qquad \text{and} \qquad \Delta\gamma_2 = \frac{\gamma + r_{m^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))}{1 + \dfrac{d r_{m^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))}{d\gamma}} \,,$$

We consider the smallest value of the set of $\Delta\gamma^{m^c}$s as a optimal value for the step size, namely

$$\Delta\gamma^{opt} = \min \left\{ \Delta\gamma^{m^c} \mid m^c \in \mathcal{A}^c(\gamma) \right\}. \tag{5.9}$$

Equation (5.8) with the step size given in Equation (5.9) are used for the predictor step of the PC algorithm. In the corrector step, $\tilde{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma - \Delta\gamma)$ is used as starting point for the Newton-Raphson algorithm that is used to solve (5.7).

When we want to estimate the dgLASSO solution curve it is necessary to adjust the step size given by Equation (5.9) in order to consider Equation (5.6). From Equation (5.8), it is easy to see that the first sign change will, approximately, occur at

$$\Delta\gamma^{opt\_out} = \min_{m \in \mathcal{A}(k)} \{ \beta_m(\gamma^{(k)}) / d_m(\gamma^{(k)}) \}, \tag{5.10}$$

where $\mathbf{d}_m(\gamma^{(k)}) = (\partial \boldsymbol{\varphi}_{\mathcal{A}}(\gamma^{(k)}) / \partial \hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma^{(k)}))^{-1} \mathbf{v}_{\mathcal{A}}$. The predictor step of the improved PC algorithm developed to estimate the dgLASSO solution curve is

Table 5.1: Pseudo-code of the improved PC algorithm to compute the solution curve defined by the dgLARS method for a model with the intercept.

```
Step  Algorithm
1     First compute $\hat{\boldsymbol{\beta}}_0 = (\hat{\beta}_0, 0, \ldots, 0)$
2     $\mathcal{A} \leftarrow \arg\max_{m^c \in \mathcal{A}^c(\gamma)}\{|r_{m^c}(\hat{\boldsymbol{\beta}}_{\mathcal{P}})|\}$ and $\gamma \leftarrow |r_1(\hat{\boldsymbol{\beta}}_0)|$
3     Repeat
4       Use (5.9) to compute $\triangle\gamma^{opt}$ and set $\triangle\gamma \leftarrow \triangle\gamma^{opt}$ and $\gamma \leftarrow \gamma - \triangle\gamma^{opt}$
5       Use (5.8) to compute $\tilde{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)$   (predictor step)
6       Use $\tilde{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)$ as starting point to solve system (5.7)  (corrector step)
7       For all $ma^c \in \mathcal{A}^c(\gamma)$ compute $r_{m^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))$
8       If $\exists \mathcal{N} \subset \mathcal{A}^c(\gamma)$ such that $\left|r_{m^{c*}}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))\right| > \gamma$ for all $m^{c*} \in \mathcal{N}$, then
9         use (5.11) to compute $\gamma_{rf}^{(m)}$ and set $\gamma_{rf} \leftarrow \max_m\{\gamma_{rf}^{(m)}\}$
10        first set $\triangle\gamma \leftarrow \triangle\gamma^{opt} - (\gamma_{rf} - \gamma)$ and then $\gamma \leftarrow \gamma_{rf}$, and go to step 5
11      If $\exists m^c \in \mathcal{A}^c(\gamma)$ such that $\left|r_{m^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))\right| = \left|r_m(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma))\right|$ for all $m \in \mathcal{A}(\gamma)$,
12        then update $\mathcal{A}(\gamma)$ and $\mathcal{A}^c(\gamma)$
13    Until convergence criterion rule is met
```

based on Equation (5.8) with step size $\Delta\gamma = \min\{\Delta\gamma, \Delta\gamma^{opt\_out}\}$.

Since the optimal step size is based on a local approximation, we also include an exclusion step for removing incorrectly included variables in the model. Determining how to implement this exclusion step is the main difference between the PC and IPC algorithms. When an incorrect variable is included in the model after the corrector step, we have that there exists a non-active variable such that the absolute value of the corresponding Rao score test statistic is greater than γ. To adjust the step size in the case of incorrectly including certain variables in the active set, the PC algorithm reduces the optimal step size from the previous step, $\triangle\gamma^{opt}$, using a contractor factor cf, which is a fixed value, i.e., $\gamma_{cf} = \gamma_{new} + \triangle\gamma^{opt} - (\Delta\gamma^{opt} \cdot cf)$. While the IPC algorithm applies the regula-falsi (rf) method which always converges. The regula-falsi method draws a secant from $h(\gamma_{new})$ to $h(\gamma_{old})$, and estimates the root as where it crosses the γ-axis, so that in our case $h(\gamma) = r_{m^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)) - s_{m^c} \cdot \gamma$ where $s_{m^c} = \text{sign}\{r_{m^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma_{new}))\}$ and $m^c \in \mathcal{A}^c(\gamma)$. From (5.3), we have that $h(\gamma) = r_m(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)) - s_m\gamma = 0$ for all $m \in \mathcal{A}(\gamma)$. Indeed, after the corrector step, when there is a non-active variable such that the absolute value of the corresponding Rao score test statistic is greater than γ, we want to find a exact point, γ_{rf}, which is very close or even equal to the true point, called transition point, that changes the active set, so that at the end, it reduces the number of the points of the solution curve. It is easy to verify that the root γ_{rf} is given by

$$\gamma_{rf} = \frac{\gamma_{new}\, r_{m^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma_{old})) - \gamma_{old}\, r_{m^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma_{new}))}{r_{m^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma_{old})) - r_{m^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma_{new})) + s_{m^c} \cdot (\gamma_{new} - \gamma_{old})}, \quad \forall m^c \in \mathcal{A}^c(\gamma_{new}), \tag{5.11}$$

where $s_{m^c} = \text{sign}\{r_{m^c}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma_{new}))\}$. Then, we first set $\triangle\gamma = \triangle\gamma^{opt} - (\gamma_{rf} - \gamma_{new})$ and then $\gamma = \gamma_{rf}$, to be able to go to the predictor step.

If at γ_{new} there exists a set $\mathcal{N}(\gamma_{new}) \subset \mathcal{A}^c(\gamma_{new})$ such that $|r_{m^{c*}}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma_{new}))| > \gamma_{new}$ for all $m^{c*} \in \mathcal{N}(\gamma_{new})$, the equation (5.11) gives a vector with an element of $\gamma_{rf}^{(m)}$, so that we consider $\gamma_{rf} = \max_m\{\gamma_{rf}^{(m)}\}$, and if $\max_m\{\gamma_{rf}^{(m)}\}$ is greater than γ_{old}, then we consider $\gamma_{rf} = \gamma_{old}$. When the Newton-Raphson algorithm does not converge, the step size is reduced by the contractor factor cf, and then the predictor and corrector steps are repeated.

In total, the main difference of the PC and IPC algorithms is the different techniques used in these algorithms for adjusting the step size to find the true transition points. In Table 5.1 we report the pseudo-code of the improved PC algorithm for a model with the intercept. In Section 5.3.3 and 5.4.1 we examine the performance of the IPC algorithm and compare it with the original PC algorithm by using the functions in the **dglars** package.

5.2.3 Estimations of the Dispersion Parameter

Since the value of the dispersion parameter ϕ affects the value of the log-likelihood function, the value of various information criteria such as AIC and BIC can be affected, and so considerations about the selection of the optimal model are going to be significantly affected. There are three commonly used estimates of the dispersion parameter: deviance, maximum likelihood (ml) and Pearson methods, see [60]. For high-dimensional generalized linear models,

[70] used a generalized version of the Pearson estimator $\hat{\phi}_P(\gamma)$ as follows:

$$\hat{\phi}_P(\gamma) = \frac{1}{n - k(\gamma)} \sum_{i=1}^{n} \frac{(y_i - g^{-1}(\mathbf{x}_i^\top \hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)))^2}{V(g^{-1}(\mathbf{x}_i^\top \hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)))}, \tag{5.12}$$

where $k(\gamma) = |\mathcal{A}(\gamma)| = \#\{j : \hat{\beta}_j(\gamma) \neq 0\}$ such that $\hat{\beta}_j(\gamma)$ is the element of the extended dgLARS estimator $\hat{\boldsymbol{\beta}}_{\mathcal{A}}(\gamma)$. In the same paper the authors proposed a four-stage refitted procedure for estimating the dispersion parameter in high-dimensional GLMs via a data splitting technique, called General Refitted Cross-Validation (GRCV) estimator, to attenuate the influence of irrelevant variables with high spurious correlations. In the rest of this section, we briefly explain the GRCV estimator and its iterative algorithm.

The idea of the GRCV method is as follows; We split the data $(\mathbf{y}_n, \mathbf{X}_{n\times p})$ randomly into two halves $(\mathbf{y}^{(1)}_{n_1}, \mathbf{X}^{(1)}_{n_1\times p})$ and $(\mathbf{y}^{(2)}_{n_2}, \mathbf{X}^{(2)}_{n_2\times p})$, where $n_1 + n_2 = n$. Without loss of generality, for notational simplicity, we assume that the sample size n is even, and $n_1 = n_2 = n/2$. In the first stage, our high dimensional variable selection method, extended dgLARS, is applied to these two data sets separately to estimate whole solution path, which yields $\hat{\boldsymbol{\beta}}_{\mathcal{A}_1}(\gamma)$ selected by $(\mathbf{y}^{(1)}, \mathbf{X}^{(1)})$ and $\hat{\boldsymbol{\beta}}_{\mathcal{A}_2}(\gamma)$ selected by $(\mathbf{y}^{(2)}, \mathbf{X}^{(2)})$, where $|\mathcal{A}_1| \leq \min(\frac{n}{2}-1, p)$ and $|\mathcal{A}_2| \leq \min(\frac{n}{2}-1, p)$.

In the second stage, we do model selection on each data set to determine two small subsets of selected variables $\hat{\mathcal{A}}_1$ and $\hat{\mathcal{A}}_2$, where $\hat{\mathcal{A}}_1 \subseteq \mathcal{A}_1$ and $\hat{\mathcal{A}}_2 \subseteq \mathcal{A}_2$. For this we estimate ϕ by (5.12) on the two data sets separately, $\hat{\phi}^{(1)}_P(\gamma)$ and $\hat{\phi}^{(2)}_P(\gamma)$, to obtain the log-likelihood functions $\ell(\hat{\boldsymbol{\beta}}_{\mathcal{A}_1}(\gamma), \hat{\phi}; \mathbf{y}^{(1)})$ and $\ell(\hat{\boldsymbol{\beta}}_{\mathcal{A}_2}(\gamma), \hat{\phi}; \mathbf{y}^{(2)})$, respectively.

In the third stage, the MLE method is applied to each subset of the data with the variables selected by another subset of the data, namely $(\mathbf{y}^{(2)}, \mathbf{X}^{(2)}_{\hat{\mathcal{A}}_1})$ and $(\mathbf{y}^{(1)}, \mathbf{X}^{(1)}_{\hat{\mathcal{A}}_2})$, to re-estimate the coefficient $\boldsymbol{\beta}$. Since the MLE may not always exist in GLMs, in this stage we propose to use the dgLARS method to estimate the coefficients based on the selected variables, $\hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}_1}(\gamma_0)$ and $\hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}_2}(\gamma_0)$, where γ_0 is close to zero, because the dgLARS estimate $\hat{\boldsymbol{\beta}}_{\mathcal{A}}(0)$ is equal to the MLE of $\boldsymbol{\beta}_{\mathcal{A}}$.

The refitting in the third stage is fundamental to reduce the influence of the spurious variables in the second stage of variable selection. Finally, in the fourth stage, we estimate ϕ by averaging the two following estimators on the two data sets $(\mathbf{y}^{(2)}, \mathbf{X}^{(2)}_{\hat{\mathcal{A}}_1})$ and $(\mathbf{y}^{(1)}, \mathbf{X}^{(1)}_{\hat{\mathcal{A}}_2})$;

$$\hat{\phi}_1(\hat{\mathcal{A}}_2) = \frac{1}{\frac{n}{2} - |\hat{\mathcal{A}}_2|} \sum_{i=1}^{\frac{n}{2}} \frac{\left(y_i^{(1)} - g^{-1}\left((\mathbf{x}_{i,\hat{\mathcal{A}}_2}^{(1)\top}\, \hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}_2}(0) \right) \right)^2}{V\left(g^{-1}\left(\mathbf{x}_{i,\hat{\mathcal{A}}_2}^{(1)\top}\, \hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}_2}(0) \right) \right)},$$

and

$$\hat{\phi}_2(\hat{\mathcal{A}}_1) = \frac{1}{\frac{n}{2} - |\hat{\mathcal{A}}_1|} \sum_{i=1}^{\frac{n}{2}} \frac{\left(y_i^{(2)} - g^{-1}\left(\mathbf{x}_{i,\hat{\mathcal{A}}_1}^{(2)\top}\, \hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}_1}(0) \right) \right)^2}{V\left(g^{-1}\left(\mathbf{x}_{i,\hat{\mathcal{A}}_1}^{(2)\top}\, \hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}_1}(0) \right) \right)},$$

where $\mathbf{x}_{i,\hat{\mathcal{A}}_j}^{(l)}$ is the i^{th} row of the l^{th} subset of the data $\mathbf{X}_{\hat{\mathcal{A}}_j}^{(l)}$, $|\hat{\mathcal{A}}_j| = \#\{k : (\hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}_j}(\gamma))_k \neq 0\}$, $\hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}_j}(\gamma)$ is the extended dgLARS estimator at γ, so that $\gamma \in [0, \gamma_{max}]$, and $\hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}_j}(0)$ is the ML estimate of $\boldsymbol{\beta}_{\hat{\mathcal{A}}_j}$. The average of these two estimators is the GRCV estimator:

$$\hat{\phi}_{GRCV}(\hat{\mathcal{A}}_1, \hat{\mathcal{A}}_2) = \frac{\hat{\phi}_1(\hat{\mathcal{A}}_2) + \hat{\phi}_2(\hat{\mathcal{A}}_1)}{2}. \tag{5.13}$$

An extension of the GRCV technique to get a more accurate estimate is using a repeated data splitting procedure; since there are many ways to split the data randomly, many GRCV estimators can be obtained. To reduce the influence of the randomness in the data splitting we may take the average of the resulting estimators. For a review of the GRCV method, the reader is referred to [70].

In the following of this section we present the iterative algorithm proposed by [70] to show how the GRCV estimator can be improved to have numerically more stable and accurate behavior. This algorithm yields a new estimate for ϕ, called the MGRCV estimate.

As mentioned above, inside the second stage of the GRCV estimator the value of the model selection criterion (AIC, BIC or k-fold CV) should be calculated. Since the AIC and BIC criteria depend on the dispersion parameter, the dispersion parameter has to be estimated and for this reason the generalized Pearson estimator $\hat{\phi}_P(\gamma)$, given in (5.12), is used inside the extended dgLARS method during the calculation of the solution path.

To decrease the influence of the classical Pearson estimate on the GRCV estimate $\hat{\phi}_{GRCV}$ and improve its accuracy, we propose an algorithm which repeats

Table 5.2: Pseudo code for the iterative algorithm to stabilize the GRCV estimator with T iterations.

```
Step  Algorithm
1     $pearson \leftarrow 1$
2     $grcv.vec \leftarrow 0$
3     $i \leftarrow 1$
4     while $i \leq T$
5       split the data into two random groups: $D_1$ and $D_2$
6       apply the extended dgLARS to $D_1$ and $D_2$ separately to obtain whole
        solution paths $\hat{\boldsymbol{\beta}}_{\mathcal{A}_1}(\gamma)$ and $\hat{\boldsymbol{\beta}}_{\mathcal{A}_2}(\gamma)$          (first stage)
7       if $pearson = 1$ then
8         use (5.12) to compute $\hat{\phi}_P^{(1)}(\gamma)$ and $\hat{\phi}_P^{(2)}(\gamma)$ for $D_1$ and $D_2$
9         use $\hat{\phi}_P^{(1)}(\gamma)$ and $\hat{\phi}_P^{(2)}(\gamma)$ to do model selection* on $D_1$ and $D_2$, respectiv-
          ely, to obtain $\hat{\mathcal{A}}_1$ and $\hat{\mathcal{A}}_2$          (second stage)
10        $pearson \leftarrow 0$
11      else
12        use $\hat{\phi}_{GRCV}(\hat{\mathcal{A}}_1, \hat{\mathcal{A}}_2)$ for model selection* on each $D_1$ and $D_2$ to obtain
          $\hat{\mathcal{A}}_1$ and $\hat{\mathcal{A}}_2$          (second stage)
13      end if
14      apply again extended dgLARS to $D_1$ and $D_2$ separately to obtain $\hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}_1}(0)$
        and $\hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}_2}(0)$          (third stage)
15      use (5.13) to compute $\hat{\phi}_{GRCV}(\hat{\mathcal{A}}_1, \hat{\mathcal{A}}_2)$          (fourth stage)
16      $grcv.vec[\, i \,] \leftarrow \hat{\phi}_{GRCV}(\hat{\mathcal{A}}_1, \hat{\mathcal{A}}_2)$
17      $i \leftarrow i + 1$
18    end while
19    $\hat{\phi}_{MGRCV} \leftarrow$ median( $grcv.vec$ )
20    use $\hat{\phi}_{MGRCV}$ to do model selection
```

* The AIC or BIC criteria.

the process of finding the GRCV estimate iteratively, such that for the $(k+1)^{\text{th}}$ iteration the k^{th} GRCV estimate ($\hat{\phi}^{(k)}_{GRCV}$) is used to compute the new $(k+1)^{\text{th}}$ GRCV estimate ($\hat{\phi}^{(k+1)}_{GRCV}$), and so on. Therefore, by using this algorithm, the GRCV estimator uses the Pearson-type estimate inside its process only for the first time, and after that the algorithm applies the obtained GRCV estimates inside the extended dgLARS algorithm instead of the generalized Pearson estimate.

Since the estimate contains some random variation due to the random CV splits, D_1 and D_2, the algorithm will not numerically converge, one in practice simply needs to define a maximal number of iterations T (which should not be too large). Therefore, the median of the T GRCV estimates, called MGRCV estimate, is used as the final GRCV estimate $\hat{\phi}_{MGRCV} = \text{median}\{\hat{\phi}^{(1)}_{GRCV}, \ldots, \hat{\phi}^{(T)}_{GRCV}\}$. The MGRCV estimate $\hat{\phi}_{MGRCV}$ is more stable and accurate than the first estimate $\hat{\phi}^{(1)}_{GRCV}$. Finally, the overall model selection is performed using $\hat{\phi}_{MGRCV}$. Table 5.2 shows how this algorithm works. It should be mentioned that, in this table, $\hat{\phi}^{(1)}_{P}(\gamma)$ and $\hat{\phi}^{(2)}_{P}(\gamma)$ are vectors of the estimates calculated during the solution path, while $\hat{\phi}_{GRCV}(\hat{\mathcal{A}}_1, \hat{\mathcal{A}}_2)$ is a fixed number.

5.3 The `dglars` package

The **dglars** package [9] is an R [77] package containing a collection of tools related to the dgLARS method, for more details see [11]. The package is available on the Comprehensive R Archive Network (CRAN) at http://CRAN.R-project.org/.

The new version of the **dglars** package (version 2.0.0) supports the `gaussian`, `binomial`, `poisson`, `Gamma` and `inverse.gaussian` families with the most commonly used link functions. The main function of this package, `dglars()`,

```
dglars(formula, family = gaussian, g, unpenalized, b_wght,
       data, subset, contrast = NULL, control = list())
```

is a wrapper function implemented to handle the formula interface usually used in R to create the $n \times p$-dimensional design matrix X and the n-dimensional response vector y. This function is used to compute the dgLARS/dgLASSO solution curve. As in the **glm** package, the user can specify family and link function using the argument `family`, see Section 5.3.2. This can be a character string naming a family function or the result of a call to a family function. In the new version of the package, the model can be specified combining family and link functions as described in Table 5.3. By default the `gaussian` family with `identity` link function is used.

Table 5.3: Some families and their link functions that can be used in the **dglars** package.

Family	Link function
gaussian	`"identity","log","inverse"`
binomial	`"logit","probit","cauchit","cloglog","log"`
poisson	`"log","identity","sqrt"`
Gamma	`"inverse","log","identity"`
inverse.gaussian	`"1/mu^2","inverse","log","identity"`

The argument `control` is a named list of control parameters with the following elements

```
control = list(algorithm = "pc", method = "dgLASSO",
         g0 = NULL, nNR = 200, nv = NULL, eps = 1.0e-05,
         np = NULL, dg_max = 0, NReps = 1.0e-06, cf = 0.5,
         ncrct = 50, nccd = 1.0e+05)
```

Using the control parameter `algorithm` it is possible to select the algorithm used to fit the dgLARS solution curve, i.e., setting `algorithm = "pc"` the default PC algorithm is used, whereas the IPC and CCD algorithms are used when `algorithm = "ipc"` and `algorithm = "ccd"` are selected, respectively. In order to reduce the computational time needed to compute the dgLARS/dgLASSO solution curve, the three algorithms are written in Fortran 90. The argument `method` is used to choose between the dgLASSO solution curve (`method = "dgLASSO"`) and the dgLARS solution curve (`method = "dgLARS"`). The `g0` control parameter is used to define the smallest value of the tuning parameter, by default this parameter is set to `1.0e-06` when $p > n$ and to 0.05 otherwise. For more details about the other control parameters and arguments see [11, 9].

In the following of this section we describe the `phihat()` and `phihat.fit()` functions which are now available in the new version of the package.

5.3.1 Description of the `phihat()` and `phihat.fit()` functions

Since the gaussian, Gamma and inverse Gaussian error distributions have an additional dispersion parameter, this package implements the functions `phihat()` and `phihat.fit()`, to estimate the dispersion parameter for high-dimensional exponential dispersion GLMs by means of four methods:

- `phihat()`, estimates of the dispersion parameter ϕ by means of the deviance, maximum likelihood estimation, generalized Pearson and GRCV methods.

- `phihat.fit()`, estimate of the dispersion parameter ϕ by means of the GRCV method.

The use of the function is the following:

```
phihat(object, type = c("pearson", "deviance", "mle",
       "grcv"), g = NULL, ordering = "AIC", n_rep = 5,
       n_iter = 5)
```

```
phihat.fit(X, y, type = c("grcv"), ordering = "AIC",
           n_rep = 5, n_iter = 5, control = list())
```

with *arguments*

`object` fitted `dglars` object.

`type` a description of the used estimator.

`g` vector of values of the tuning parameter γ. This argument are used only when `type` is equal to "`grcv`".

`ordering` a description of the model selection tool used in the second stage of the GRCV estimator to select one of the "AIC", "BIC" or "CV" criterion. Default is `ordering = "AIC"`. This argument is used only when `type` is equal to "`grcv`".

`n_rep` a non negative integer used to specify the number of repeatations only for the GRCV estimator (`type = "grcv"`). To get a more accurate estimator the user can use a repeated data splitting procedure. Default is `n_rep = 5`.

n_iter a non negative integer used to specify the maximum number of iterations for the iterative GRCV algorithm (only when type = "grcv"). If n_iter is greater than or equal to 2 then the algorithm gives the median of these n_iter GRCV estimates, called MGRCV estimate $\hat{\phi}_{MGRCV} = median(\hat{\phi}^{(1)}_{GRCV}, \ldots, \hat{\phi}^{(n\_iter)}_{GRCV})$. Default is n_iter = 5.

control a list of control parameters available only for type = "grcv", and supplies any of the control parameters explained in the function dglars() .

X design matrix of dimension $n \times p$.

y response vector.

When there is a fitted 'dglars' object the function phihat() can be used to estimate the dispersion parameter ϕ by any of the four methods, while the user can use phihat.fit() with the design matrix X and the response vector y to estimate the parameter ϕ only by the GRCV estimator.

phihat() returns a vector with the estimates of the dispersion parameter. When type = "grcv" all elements of the vector are the same, because the GRCV estimator does not depend on the tuning parameter γ while the other three estimators do. For more details see [70] and [9].

The optional argument g is used to specified the values of the tuning parameter γ; if not specified (default), the estimates of the dispersion parameter are computed for the sequence of models storage in the argument object (see the example in Section 5.3.2).

When gaussian, Gamma or inverse Gaussian is used, the function dglars() returns the vector of the estimates of the dispersion parameter ϕ; by default, the generalized Pearson statistic is used as estimator but the user can use the function phihat() to specify other estimators. For the binomial and Poisson family, the dispersion parameter is assumed known and equal to one.

The function phihat() is called by the logLik(), AIC() and coef() methods for 'dglars' objects:

```
logLik(object, phi = c("pearson", "deviance", "mle",
         "grcv"), ...)

AIC(object, phi = c("pearson", "deviance", "mle", "grcv"),
         k = 2, complexity = c("df", "gdf"), ...)

coef(object, type = c("pearson", "deviance", "mle",
         "grcv"), ...)
```

when the argument `phi` (or `type` in `coef()`) is set to any of the four estimation methods, i.e., "`pearson`", "`deviance`", "`mle`" or "`grcv`". In the **dglars** package, the `summary()` method:

```
summary(object, type = c("AIC", "BIC"),
        digits = max(3, getOption("digits") - 3), ...)
```

uses the generalized Pearson estimator to define the BIC or AIC values, but the user can use "`dots`" to pass to the method `AIC()` the additional arguments needed to compute a more general measure of goodness-of-fit, e.g., "`phi`", "`k`" or "`complexity`". For the description of these arguments and methods see [9].

5.3.2 An example of use for a simulated Gamma model

To gain more insight about the use of the `phihat()` function and the differences among the estimation methods, we have simulated a data set from a Gamma regression model with the `log` link function where sample size is equal to 20 and $p = 100$. We assume that only the first two predictors influence the response variable. First we load the **dglars** package in the R session by the code

```
R> library("dglars")
```

The corresponding R code is given by:

```
R> set.seed(112358)
R> n <- 100
R> p <- 5
R> s <- 2
R> X <- matrix(abs(rnorm(n * p)), n, p)
R> bs <- rep(2, s)
```

```
R> Xs <- X[, 1:s]
R> eta <- drop(1 + (Xs %*% bs))
R> mu <- Gamma("log")$linkinv(eta)
R> shape <- 0.5
R> phi <- 1 / shape
R> y <- rgamma(n, shape = shape, scale = mu * phi)
R> fit <- dglars(y ~ X, Gamma("log"),
+                 control = list(algorithm = "ipc",
+                 method = "dgLARS"))
```

The `fit` object is a fitted object of S3 class '`dglars`'. For this object, we apply the dgLARS method with the IPC algorithm. Using the `summary()` method the user can obtain more information about the estimated sequence of models for the '`dglars`' object. For example, the following R code shows the output printed by the `summary()` method with the BIC criterion and the GRCV estimate for the dispersion parameter.

```
R> summary(fit, type = "BIC", phi = "grcv")

Call:  dglars(formula = y ~ X, family = Gamma("log"),
              control = list(algorithm = "ipc",
              method = "dgLARS"))

     Sequence          g      %Dev  df   BIC   Rank
                12.50763  0.00000   2  1245  10
         + X2
                10.45156  0.08625   3  1223   9
                10.44988  0.08631   3  1223   8
         + X1
                2.476473  0.55899   4  1079   7
                2.452213  0.55969   4  1079   6
         + X4
                1.048410  0.60506   5  1069   2
                1.041003  0.60520   5  1069   1 <-
         + X5
                0.719319  0.61228   6  1071   4
```

```
               0.711903  0.61241   6  1071   3
         + X3
               0.000001  0.62372   7  1072   5
Details:
          BIC values computed using k = 4.605 and
          complexity = ‘df’, dispersion parameter
          estimated by ‘grcv’

=================================================

Summary of the Selected Model

   Formula: y ~ X1 + X2 + X4
    Family: ‘Gamma’
      Link: ‘log’

Coefficients:
        Estimate
   Int.   1.6560
   X1     1.5282
   X2     1.4554
   X4     0.3665

Dispersion parameter: 1.996 (estimated by ‘grcv’ method)
---

                   g: 1.041
       Null deviance: 627.4
   Residual deviance: 247.7
                 BIC:  1069

Algorithm ‘ipc’ ( method = ‘dgLARS’ )
```

From this output we can see that the dgLARS method first finds the true predictors (`X1` and `X2`) and then includes the other false predictors. The ranking of the estimated models obtained by the number of estimated non-zero coefficients as a measure of goodness of fit (`complexity = "df"`) is also shown and the corresponding best model is identified by an arrow on the right. The formula of the identified best model, the corresponding estimated coefficients and the estimate of the dispersion parameter ϕ are shown in the second section of the output. These values are obtained at the optimal value of the tuning parameter γ which can be calculated by the BIC or AIC criteria. For example, from the previous output we can see that the values of the BIC criterion, GRCV estimate and optimal tuning parameter are 1069, 1.996 and 1.041, respectively.

Since the deviance, MLE and generalized Pearson estimators depend on the tuning parameter γ, the values of these estimates can change during the solution path. But the GRCV estimator is fixed by changing the tuning parameter. These estimates can be extracted using the `phihat()` and `phihat.fit()` functions. For example, with the following R code we can see the sequence of the values of the tuning parameter with the estimated values of the dispersion parameter by means of the generalized Pearson and GRCV methods. For the GRCV method we apply the BIC criterion and the 10 times of iterations inside the algorithm.

```
R> set.seed(11235)
R> g <- fit$g
R> grcv <- phihat(fit, type = "grcv", ordering = "BIC",
+                 n_iter = 10)
R> pearson <- phihat(fit, type = "pearson")
R> deviance <- phihat(fit, type = "deviance")
R> mle <- phihat(fit, type = "mle")
R> path <- cbind(g, pearson, deviance, mle, grcv)
R> print(path, digits = 3)
```

```
            g  pearson  deviance   mle  grcv
 [1,]  1.25e+01    31.13      6.34  3.83     2
 [2,]  1.05e+01    22.24      5.85  3.59     2
 [3,]  1.04e+01    22.23      5.85  3.59     2
 [4,]  2.48e+00     2.82      2.85  2.06     2
 [5,]  2.45e+00     2.81      2.85  2.06     2
 [6,]  1.05e+00     1.96      2.58  1.89     2
 [7,]  1.04e+00     1.95      2.58  1.88     2
 [8,]  7.19e-01     1.84      2.56  1.86     2
 [9,]  7.12e-01     1.84      2.56  1.86     2
[10,]  1.00e-06     1.73      2.51  1.81     2
```

By the following R code, we can specify the values of the tuning parameter γ to compute the estimate of the dispersion parameter;

```
R> set.seed(11235)
R> new_g <- seq(range(fit$g)[2], range(fit$g)[1],
+               by = -1.0)
R> grcv <- phihat(fit, type = "grcv",
+                 ordering = "BIC", n_iter = 10, g=new_g)
R> pearson <- phihat(fit, type = "pearson", g=new_g)
R> deviance <- phihat(fit, type = "deviance", g=new_g)
R> mle <- phihat(fit, type = "mle", g=new_g)
R> path <- cbind(new_g, pearson, deviance, mle, grcv)
R> print(path, digits = 3)

       new_g  pearson  deviance   mle  grcv
 [1,] 12.508    31.13      6.34  3.83     2
 [2,] 11.508    26.39      6.12  3.71     2
 [3,] 10.508    22.43      5.86  3.59     2
 [4,]  9.508    14.12      5.35  3.34     2
 [5,]  8.508     8.90      4.83  3.09     2
 [6,]  7.508     5.88      4.37  2.87     2
 [7,]  6.508     4.16      3.96  2.66     2
 [8,]  5.508     3.21      3.60  2.47     2
 [9,]  4.508     2.75      3.30  2.31     2
[10,]  3.508     2.63      3.04  2.17     2
[11,]  2.508     2.81      2.86  2.06     2
[12,]  1.508     2.11      2.65  1.93     2
[13,]  0.508     1.77      2.55  1.84     2
```

5.3.3 Comparing PC and IPC Algorithms

The **dglars** package implements three different algorithms to compute the dgLARS solution curve, i.e., the PC, IPC and CCD algorithms. Although these algorithms compute the same solution curve, the results can be looked different. Here, however, we focus only on the PC and IPC algorithms to compare their performance by means of a simple simulation; for extensive simulation study see Section 5.4.1. For information about comparing the CCD and PC algorithms see [11].

To gain more insight we consider the following R code to simulate an inverse Gaussian model with the canonical link function (`link = "1/mu^2"`) and sample size equal to 100 and 5 predictors. First we load the **statmod** package to use the function `rinvgauss()` for generating the random numbers for the inverse Gaussian distribution by the code

```
R> library("statmod")
```

The corresponding R code is given by:

```
R> set.seed(112358)
R> n <- 200
R> p <- 10
R> X <- matrix(abs(rnorm(n * p)), n, p)
R> b <- 1:2
R> eta <- drop(b[1] + (X[, 1] * b[2]))
R> mu <- inverse.gaussian()$linkinv(eta)
R> phi <-  0.5
R> y <- rinvgauss(n, mean = mu, disp = phi)
```

Only the first predictor affects the response variable y. By the following code we estimate the dgLASSO solution curve using the PC and the improved PC algorithm, respectively;

```
R> fit_pc <- dglars(y ~ X, inverse.gaussian("1/mu^2"),
+                  control = list(algorithm = "pc",
+                  method = "dgLASSO"))
R> fit_ipc <- dglars(y ~ X, inverse.gaussian("1/mu^2"),
+                  control = list(algorithm = "ipc",
+                  method = "dgLASSO"))
```

By printing the 'dglars' object fit_pc for our simulated data set, we can see that by using the PC algorithm the dgLASSO solution curve has the 34 transition points;

```
R> fit_pc

Call:  dglars(formula = y ~ X,
              family = inverse.gaussian("1/mu^2"),
              control = list(algorithm = "pc",
              method = "dgLASSO"))

     Sequence        g      Dev     %Dev  n. non zero
               1.30330    90.33  0.00000            1
          +X1
               0.99185    87.57  0.03052            2
               0.83912    86.60  0.04131            2
               0.76325    86.19  0.04580            2
               0.72543    86.01  0.04784            2
               0.70654    85.92  0.04881            2
               0.68766    85.84  0.04974            2
          +X6
               0.59029    85.16  0.05717            3
               0.59014    85.16  0.05718            3
          +X9
               0.55107    84.80  0.06125            4
               0.53169    84.62  0.06317            4
               0.52204    84.54  0.06409            4
               0.51240    84.46  0.06500            4
          +X4
               0.43666    83.70  0.07339            5
               0.39888    83.37  0.07708            5
               0.36111    83.06  0.08044            5
          +X3
               0.33442    82.83  0.08306            6
               0.32116    82.71  0.08429            6
          0.31455    82.66  0.08489            6
          0.30794    82.61  0.08547            6
     +X2
          0.30603    82.59  0.08570            7
     +X5
          0.19255    81.42  0.09858            8
          0.13938    81.06  0.10260            8
          0.11402    80.93  0.10406            8
          0.10167    80.87  0.10467            8
          0.09559    80.85  0.10494            8
          0.09257    80.84  0.10507            8
          0.08956    80.83  0.10519            8
     +X10
          0.05807    80.72  0.10640            9
          0.05803    80.72  0.10640            9
     +X8
          0.05191    80.70  0.10659           10
          0.04886    80.69  0.10668           10
          0.04582    80.68  0.10676           10
     +X7
          0.00010    80.62  0.10747           11

Algorithm ‘pc’ ( method = ‘dgLASSO’ ) with exit = 0
```

The number of the iterations to compute the solution points by the PC algorithm and the values of the tuning parameter can be obtained by the following code:

```
R> fit_pc$np

[1] 34

R> fit_pc$g

 [1] 1.3032970549 0.9918525125 0.8391243838 0.7632531467
 [5] 0.7254261662 0.7065384830 0.6876634402 0.5902862409
 [9] 0.5901385647 0.5510746529 0.5316927115 0.5220378797
[13] 0.5124007653 0.4366644718 0.3988758027 0.3611128245
[17] 0.3344235555 0.3211580697 0.3145454019 0.3079428460
[21] 0.3060269170 0.1925485474 0.1393809155 0.1140160987
[25] 0.1016708751 0.0955858509 0.0925656090 0.0895565832
[29] 0.0580652597 0.0580312821 0.0519057820 0.0488601606
[33] 0.0458231185 0.0001000001
```

By printing `fit_ipc`, we can see that the IPC algorithm reduces the number of the iterations during computing the solution curve such that leads to potentially computational saving;

```
R> fit_ipc

Call:  dglars(formula = y ~ X,
              family = inverse.gaussian("1/mu^2"),
              control = list(algorithm = "ipc",
              method = "dgLASSO"))

     Sequence          g     Dev      %Dev  n. non zero
               1.303297  90.33  0.00000            1
          + X1
               0.687731  85.84  0.04974            2
               0.687668  85.84  0.04974            2
          + X6
               0.590286  85.16  0.05717            3
               0.590139  85.16  0.05718            3
          + X9
               0.512409  84.46  0.06500            4
          + X4
               0.361118  83.06  0.08044            5
          + X3
               0.307947  82.61  0.08547            6
          + X2
               0.306027  82.59  0.08570            7
          + X5
```

```
               0.090291  80.83  0.10516            8
               0.089560  80.83  0.10519            8
          + X10
               0.058065  80.72  0.10640            9
               0.058031  80.72  0.10640            9
          + X8
               0.045826  80.68  0.10676           10
          + X7
               0.000001  80.62  0.10747           11

Algorithm 'ipc' ( method = 'dgLASSO' ) with exit = 0
```

By the following code we can see that when we use the IPC algorithm, the number of iterations is less than half of the number of iterations when we use the PC algorithm, so that it leads to a decrease in the run times needed for computing the solution curve;

```
R> fit_ipc$np

[1] 15

R> fit_ipc$g

[1] 1.303297e+00 6.877312e-01 6.876676e-01 5.902864e-01
[5] 5.901386e-01 5.124086e-01 3.611181e-01 3.079471e-01
[9] 3.060270e-01 9.029056e-02 8.956011e-02 5.806524e-02
[13] 5.803128e-02 4.582614e-02 1.000056e-06
```

From a computational point of view, the main consequence of using the technique used in the improved PC algorithm to adjust the step size and find the true transition points is a decrease in the run times. In next section, we investigate the performance of the improved PC algorithm by a simulation study.

5.4 Simulation Studies

In this section we present a comprehensive simulation study to investigate the performance of the improved PC algorithm implemented in the **dglars** package.

5.4.1 Comparison of Run Times

In this section we compare the improved PC algorithm (IPC) with the original PC algorithm proposed in [13]. Although these two algorithms compute the same solution curve, the results can be looked different. As mentioned before, the main problem of the PC algorithm is related to the number of the points of the solution curve (q), so that the number of arithmetic operations needed to compute the solution curve causes an increase in the run times.

In order to better understand the effects of the number of the points of the solution curve (q) on the run times of the two algorithms, we use a simple simulation study based on a Gamma model with a non-canonical link function (*log*) with sample size equal to $n = (50, 100, 200)$ and $p = (10, 100, 500)$. The study is based on three different configurations of the covariance structure of the p predictors, such that $X_1, X_2, \cdots, X_n$ sampled from an $N(\mathbf{0}, \Sigma)$ distribution, where the diagonal elements of Σ are 1 and the off-diagonal elements follow $corr(X_i; X_j) = \rho^{|i-j|}$, where $i \neq j$ and $\rho = (0, 0.9)$. To simulate the response vector we use a model with intercept and choose

$$\boldsymbol{\beta} = (1, \underbrace{2, 2, 2}_{3}, \underbrace{0, \cdots, 0}_{p-3}).$$

In Table 5.4 we report the average CPU times in seconds and the mean number of the points of the solution curve (q) coming from 100 simulation runs. All timings reported were carried out on a personal computer with Intel Core $i5$ $520M$ dual-core processor. This table shows that the IPC algorithm has a lower average CPU time than the PC algorithm. Moreover, the mean number of the points of the solution curve (q) in the IPC algorithm is always less than q in the PC algorithm. Since the IPC algorithm reduces the number of the points of the solution curve (q), it is obvious that its speed is more than the PC algorithm. Thus, we clearly see that the proposed IPC algorithm is always faster than the PC algorithm. The difference between the two algorithms is greater when $\rho = 0$

Table 5.4: Average CPU times ($time$) in seconds to compute the solution curve using the IPC and PC algorithms based on the Gamma regression model, and the mean number of the points of the solution curve (q). Bold values identify the best algorithms for each scenario.

			n					
			50		100		200	
ρ	p		PC	IPC	PC	IPC	PC	IPC
0	10	time	0.340	**0.202**	0.358	**0.204**	0.430	**0.237**
		q	35.48	**21.88**	35.07	**20.78**	35.38	**20.18**
	100	time	23.23	**17.73**	87.5	**70.48**	179.1	**136.0**
		q	118.6	**87.61**	196.9	**148.0**	192.9	**142.4**
	500	time	145.8	**116.4**	583.8	**497.3**	1409	**1252**
		q	120.2	**90.70**	198.7	**156.6**	321.2	**271.2**
0.9	10	time	0.359	**0.245**	0.374	**0.260**	0.430	**0.285**
		q	32.78	**22.12**	33.05	**21.82**	32.54	**21.37**
	100	time	20.75	**16.55**	59.33	**50.07**	170.0	**146.1**
		q	109.4	**83.11**	158.3	**125.9**	172.2	**137.6**
	500	time	136.1	**112.4**	485.6	**432.6**	1986	**1803**
		q	111.7	**87.30**	181.6	**150.0**	298.8	**257.0**

(no correlation among the predictors).

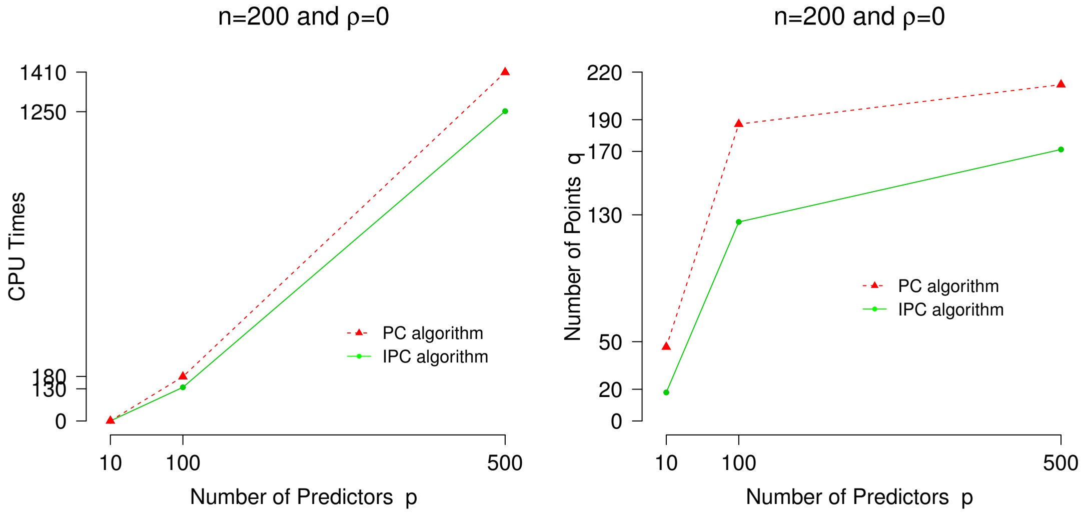
Moreover, in Figure 5.1 we show the average CPU times and the mean number of the points of the solution curve (q) for the considered algorithms from the simulation study based on the Gamma regression model when $n = 200$ and $\rho = 0$. Both timing and q are showed as a function of the number of predictors $p = (10, 100, 500)$. The difference between the two algorithms can be clearly seen in these figures.

5.5 Application to Real Data

In this section we analyze a real dataset by using the functions available in the **dglars** package. In Section 5.5.1 we consider the branchmark *Diabetes* data available in the **dglars** package.

5.5.1 Diabetes Dataset

In this section we use the functions available in the **dglars** package to study the sparse structure of a inverse Gaussian regression model applied to the dia-



(a)

(b)

Figure 5.1: (a) CPU times, (b) mean number of the points of the solution curve, q, for the IPC and PC algorithms from the simulation study based on the Gamma regression model with $n = 200$ and $\rho = 0$, which are showed as a function of p.

betes data used in [29] and [46], among others. The response y is a quantitative measure of disease progression for patients with diabetes one year later. The data includes 10 baseline measurements (design matrix `x`) for each patient, such as "*age*", "*sex*" (gender, which is binary), "*bmi*" (body mass index), "*map*" (mean arterial blood pressure), and six blood serum measurements: "*ldl*" (high-density lipoprotein), "*hdl*" (low-density lipoprotein), "*ltg*" (lamotrigine), "*glu*" (glucose), "*tc*" (triglyceride) and "*tch*" (total cholesterol), in addition to 45 interactions and 9 quadratic terms, for a total of 64 variables (design matrix `x2`) for each patient, so that this data has $n = 442$ observations on $p = 64$ variables. For high-dimensional purpose we add a thousand noise variables to the original data to also have a high-dimensional dataset with $p = 1064$ (design matrix `x3`). These low- and high-dimensional diabetes data (`diabetesH`) can be found in the **dglars** package. The aim of the study is to identify which of the covariates are important factors in disease progression. For that we need to estimate the dispersion parameter to be able to do model selection.

To study the considered dataset, we first load the data in the R session

```
R> data("diabetesH", package = "dglars")
R> attach(diabetesH)
```

The `diabetesH` data-frame has 442 rows and 4 columns as follows:

`x` : a matrix with 10 columns,

`x2` : a matrix with 64 columns,

`x3` : a matrix with 1064 columns,

`y` : a numeric vector with a length of 10.

First, we estimate the optimal value of the tuning parameter by the 10-fold cross-validation method by using the `cvdglars()` function, i.e.,

```
R> cv_diabetes <- cvdglars(y ~ x, inverse.gaussian("log"),
+               data = diabetesH, control =
+               list(algorithm = "ipc", method = "dgLARS"))
R> cv_diabetes

Call:  cvdglars(formula = y ~ x,
                family = inverse.gaussian("log"),
                data = diabetesH, control = list(
                algorithm = "ipc", method = "dgLARS"))

Coefficients:
        Estimate
   Int.   4.9539
    sex  -2.0273
    bmi   2.8447
    map   2.1969
     tc  -0.3811
    hdl  -2.4124
    ltg   3.8501

Dispersion parameter: 0.001141
```

```
Details:
   number of non zero estimates: 7
      cross-validation deviance: 0.06296
                              g: 0.01533
                        n. fold: 10

Algorithm 'ipc' ( method = 'dgLARS' )
```

This output shows that the dgLARS method selects a inverse Gaussian regression model with eight covariates that can be seen by the following code:

```
R> cv_diabetes$formula
y ~ sex + bmi + map + tc + hdl + ltg
```

Moreover, the optimal tuning parameter is 0.01533 and the dispersion parameter estimate by the generalized Pearson method is 0.001141.

We then call our functions to fit the dgLARS method and estimate the dispersion parameter by the GRCV method to do the model selection by the BIC criterion.

```
R> diabetes_dglars <- dglars(y ~ x,
+    inverse.gaussian("log"), data = diabetesH,
+    control = list(algorithm = "ipc", method = "dgLARS"))

R> summary(diabetes_dglars, type = "BIC", phi = "grcv")

Call:  dglars.fit(X = x, y = y,
       family = inverse.gaussian("log"),
       control =list(method = "dgLARS", algorithm = "ipc"))

     Sequence          g     %Dev  df   BIC   Rank
                0.505974  0.00000   2  5089  18
        + bmi
                0.481473  0.02290   3  5084  17
                0.481262  0.02309   3  5084  16
        + ltg
```

```
                0.250152  0.26744   4  4952  15
                0.233248  0.27846   4  4945  13
                0.233174  0.27851   4  4945  12
        + map
                0.222313  0.28613   5  4946  14
        + hdl
                0.100212  0.36560   6  4895  11
                0.099904  0.36572   6  4895  10
        + sex
                0.030320  0.41322   7  4865   2
                0.030263  0.41324   7  4865   1 <-
        + tc
                0.014883  0.41892   8  4866   3
        + glu
                0.005757  0.42063   9  4871   4
        + tch
                0.002389  0.42122  10  4877   6
                0.002384  0.42122  10  4877   5
        + ldl
                0.001704  0.42199  11  4882   8
                0.001691  0.42200  11  4882   7
        + age
                0.000001  0.42272  12  4887   9

Details:
          BIC values computed using k = 6.091 and
          complexity = ‘df’, dispersion parameter
          estimated by ‘grcv’

===============================================================

Summary of the Selected Model

   Formula: y ~ sex + bmi + map + hdl + ltg
    Family: ‘inverse.gaussian’
```

```
      Link: 'log'

Coefficients:
       Estimate
   Int.   4.9495
   sex   -1.6834
   bmi    2.7786
   map    1.9536
   hdl   -2.2917
   ltg    3.5420

Dispersion parameter: 0.001140 (estimated by 'grcv' method)
---

                 g: 0.03026
      Null deviance:    1.0361
  Residual deviance:    0.6079
                BIC: 4864.8971

Algorithm 'ipc' ( method = 'dgLARS' )
```

The fitted model, the estimate of the coefficients, the GRCV estimate of the dispersion parameter (0.001140) and the optimal value of the tuning parameter (0.03026) can be found in this output. We can estimate the dispersion parameter by the GRCV method without a fitted '`dglrs`' object and only with the design matrix (`x`) and the response variable (`y`) by the following code:

```
R> phihat.fit(x,y, type = c("grcv"))

[1] 0.001139591
```

The outputs of the two model selection tools (BIC and CV) shows that five predictors ("*sex*", "*bmi*", "*map*", "*hdl*" and "*ltg*") are selected by the BIC criterion while the CV method, in addition to the five predictors selected by the BIC, selects the predictor "*tc*" as another important variable.

To finish this section, we compare the run times of the original PC and improved PC algorithms with the following R code:

```
R> system.time(diabetes_dglars_pc <- dglars(y ~ x,
+   inverse.gaussian("log"), data = diabetesH,
+   control = list(algorithm = "pc", method = "dgLARS")))

    user  system elapsed
   0.440   0.000   0.437

R> diabetes_dglars_pc$np

[1] 21

R> system.time(diabetes_dglars_ipc <- dglars(y ~ x,
+   inverse.gaussian("log"), data = diabetesH,
+   control = list(algorithm = "ipc", method = "dgLARS")))

    user  system elapsed
   0.364   0.000   0.363

R> diabetes_dglars_ipc$np

[1] 18
```

Since the number of points of the dgLARS solution curve in the improved PC algorithm is less than the number of points in the original PC algorithm, the total run time for computing the solution path by the IPC algorithm is less than the run time by the PC algorithm.

5.6 Conclusions

We briefly reviewed the differential geometrical theory underlying the dgLARS method and briefly explained the dispersion parameter estimation methods. We described some functions implemented in the new version of the **dglars** package that can be used to estimate the dispersion parameter, and we also used these functions to compare run times between two different PC and

IPC algorithms. In simulations and the actual datasets we have shown that the improved PC algorithm is faster than the original PC algorithm, and now the dgLARS method can be used for a variety of distributions with different types of the canonical and non-canonical link functions. A new version of **dglars** [9] with new functions is available on CRAN.

Academic Summaries

Summary

A large class of modelling and prediction problems involve outcomes that belong to an exponential family distribution. Generalized linear models (GLMs) are a standard way of dealing with such situations. GLMs can be extended to deal with high-dimensional feature spaces. Penalized inference approaches, such as the ℓ_1 or SCAD, or extensions of least angle regression, such as dgLARS, have been proposed to deal with GLMs with high-dimensional feature spaces. Although the theory underlying these methods is in principle generic, the implementation has remained restricted to dispersion free models, such as the Poisson and logistic regression models in which the dispersion parameter is equal to one.

The aim of Chapter 2 is to extend the differential geometric least angle regression method for high-dimensional GLMs to arbitrary exponential dispersion family distributions with arbitrary link functions. This entails, first, extending the improved predictor-corrector (IPC) algorithm to arbitrary distributions and link functions, and second, proposing a classical estimator of the dispersion parameter. Furthermore, improvements to the computational algorithm lead to an important speed-up of the PC algorithm. In Chapter 3, we develop a new method to make high-dimensional inference on the dispersion parameter of the exponential family. Moreover, we propose an iterative algorithm to improve the accuracy of the new proposed method. Simulation studies provide supporting evidence concerning the proposed efficient algorithm for estimating dispersion

parameter. The resulting methods have been implemented in the R-package **`dglars`**.

Many clinical and epidemiological studies rely on survival modelling to detect clinically relevant factors that affect various event histories. With the introduction of high-throughput technologies in the clinical and even large-scale epidemiological studies, the need for inference tools that are able to deal with fat data-structures, i.e., relatively small number of observations compared to the number of features, is becoming more prominent. Chapter 4 introduces a principled sparse inference methodology for proportional hazards modelling, based on differential geometrical analyses of the high-dimensional likelihood surface.

Since the value of the dispersion parameter ϕ affects the value of the log-likelihood function, the value of various information criteria such as AIC and BIC can be affected, and so considerations about the selection of the optimal model are going to be significantly affected. In Chapter 5, we explain the improved estimator of the dispersion parameter, proposed in [70], for high-dimensional exponential dispersion generalized linear models, called General Refitted Cross-Validation (GRCV) estimator with an algorithm to improve the proposed estimator to obtain a more accurate estimator. Several dispersion parameter estimation methods and algorithms for computing the dgLARS solution curve, proposed in [13] and [70], are implemented in the new version of the R-package **dglars** [14].

Samenvatting

Een grote groep van modellerings- en voorspellingsproblemen betreffen uitkomsten die behoren tot een exponentiële familieverdeling. Gegeneralizeerde lineaire modellen (GLMs) zijn een standaard manier om dergelijke situaties te behandelen. Zelfs in hoogdimensionale kenmerkruimten kunnen GLMs uitgebreid worden om dergelijke situaties aan te pakken. Penalized inferentie benaderingen, zoals de ℓ_1 of SCAD, of extensies van de minste hoekregressie, zoals dgLARS, zijn voorgesteld voor GLMs met hoogdimensionale kenmerkenruimtes. Hoewel de theorie die aan deze methodes ten grondslag ligt in principe generiek is, blijft de implementatie beperkt tot dispersievrije modellen, zoals de Poisson- en logistieke regressiemodellen waarin de dispersieparameter gelijk is aan één.

Het doel van Hoofdstuk 2 is het uitbreiden van de differentiaal geometrische minimale hoekregressie methode voor hoge-dimensionale GLMs naar willekeurige exponentiële dispersie familie verdelingen met willekeurige verbindingsfuncties. Dit houdt in dat, eerst, het verbeterde predictor-corrector (IPC) algoritme wordt uitgebreid naar willekeurige verdelingen en verbindingsfuncties, en ten tweede, een klassieke schatter van de dispersieparameter wordt voorgesteld. Voorts leiden verbeteringen van het berekeningsalgoritme tot een belangrijke versnelling van het PC-algoritme. In Hoofdstuk 3 ontwikkelen wij een nieuwe methode om de hoogdimensionale inferentie van de dispersieparameter van de exponentiële familie mogelijk te maken. Bovendien stellen we een iteratief algoritme voor om de nauwkeurigheid van de nieuwe voorgestelde methode te verbeteren. Simulatiestudies bieden ondersteunend bewijs over het voorgestelde efficiënte algoritme voor het beoordelen van de dispersieparameter. De resulterende methoden zijn geïmplementeerd in het R-pakket **`dglars`**.

Veel klinische en epidemiologische studies zijn gebaseerd op overlevingsmodellering om klinisch relevante factoren te detecteren die verschillende gebeurtenisgeschiedenissen beïnvloeden. Met de introductie van high-throughput technologieën in de klinische en zelfs grootschalige epidemiologische studies, is de behoefte aan inferentie instrumenten die in staat zijn om te gaan met vette datastructuren, dat wil zeggen een relatief klein aantal waarnemingen in vergelijking met het aantal kenmerken, prominenter geworden.

Hoofdstuk 4 introduceert een principiële sparse inferentie methodologie voor proportionele gevaren modellering, gebaseerd op differentiële geometrische analyses van het hoogdimensionale waarschijnlijkheid oppervlak.

Aangezien de waarde van de dispersieparameter ϕ de waarde van de log-waarschijnlijkheidsfunctie beïnvloedt, kan de waarde van verschillende informatiecriteria zoals AIC en BIC worden beïnvloed en tevens overwegingen over de selectie van het optimale model. In Hoofdstuk 5, leggen we de verbeterde schatter van de dispersieparameter voor, die in [70] voorgesteld wordt, voor hoge-dimensionale exponentiële dispersie generalizeerde lineaire modellen. Hij draagt de naam General Refitted Cross Validation (GRCV) schatter en we stellen een algorithme voor om deze schatter te implementeren. Verschillende dispersieparameterschattingmethoden en algoritmen voor het berekenen van de dgLARS-oplossingscurve, voorgesteld in [13] en [70], worden geïmplementeerd in de nieuwe versie van het R-pakket **dglars** [14].

References

[1] O. O. Aalen, Ø. Borgan, and H. K. Gjessing. *Survival and Event Hostory Analysis: A Process Point of View*. Springer, 2008.

[2] K. Aho, D. Derryberry, and T. Peterson. “Model selection for ecologists: the worldviews of AIC and BIC”. In: *Ecology* 95.3 (2014), pp. 631–636.

[3] H. Akaike. “A New Look at the Statistical Model Identification”. In: *IEEE Transactions on Automatic Control* 19 (1974), pp. 716–723.

[4] E. Allgower and K. Georg. *Introduction to Numerical Continuation Methods*. New York: Society for Industrial and Applied Mathematics, 2003.

[5] S. -I. Amari and H. Nagaoka. *Differential-geometrical Methods in Statistics*. New York: Springer, 1985.

[6] S. -I. Amari and H. Nagaoka. *Methods of Information Geometry*. Vol. 191. Translations of Mathematical Monographs. Providence, R. I.: American Mathematical Society, 2000.

[7] P. K. Andersen and R. D. Gill. “Cox’s regression model for counting processes: a large sample study”. In: *The Annals of Statistics* 10.4 (1982), pp. 1100–1120.

[8] S. Arlot and A. Celisse. “A survey of cross-validation procedures for model selection”. In: *Statistics Surveys* 4 (2010), pp. 40–79.

[9] L. Augugliaro. *dglars: Differential Geometric LARS (dgLARS) Method*. http://CRAN.R-project.org/package=dglars. R package version 1.0.5. 2014b.

[10] L. Augugliaro, A. M. Mineo, and E. C. Wit. “A Differential Geometric Approach to Generalized Linear Models with Grouped Predictors”. In: *Biometrika* 103.3 (2016), pp. 563–577.

[11] L. Augugliaro, A. M. Mineo, and E. C. Wit. “dglars: An R Package to Estimate Sparse Generalized Linear Models”. In: *Journal of Statistical Software* 59.8 (2014a), pp. 1–40.

[12] L. Augugliaro, A. M. Mineo, and E. C. Wit. “Differential Geometric LARS via Cyclic Coordinate Descent Method”. In: *International Conference on Computational Statistics (COMPSTAT 2012)* (2012). Limassol, Cyprus, pp. 67–79.

[13] L. Augugliaro, A. M. Mineo, and E. C. Wit. “Differential Geometric Least Angle Regression: A Differential Geometric Approach to Sparse Generalized Linear Models”. In: *Journal of the Royal Statistical Society: Series B* 75.3 (2013), pp. 471–498.

[14] L. Augugliaro and H. Pazira. *dglars: Differential Geometric Least Angle Regression*. http://CRAN.R-project.org/package=dglars. R package version 2.0.0. 2017.

[15] L. Bao et al. “Prevalent overexpression of prolyl isomerase Pin1 in human cancers”. In: *The American journal of pathology* 164.5 (2004), pp. 1727–1737.

[16] L. Boldrini et al. “Telomerase activity and hTERT mRNA expression in glial tumors”. In: *International journal of oncology* 28.6 (2006), pp. 1555–1560.

[17] J. Burbea and R. C. Rao. “Entropy Differential Metric, Distance and Divergence Measures in Probability Spaces - A Unified Approach”. In: *Journal of Multivariate Analysis* 12 (1982), pp. 575–596.

[18] K. P. Burnham and D. R. Anderson. *Model selection and multimodel inference: a practical information-theoretic approach*. 2^{nd}. New York: Springer, 2002.

[19] J. Cai, J. Fan, and R. Li. “Variable selection for multivariate failure time data”. In: *Biometrika* 92.2 (2005), pp. 303–316.

[20] E. J. Candes and T. Tao. “The Dantzig selector: Statistical estimation when p is much larger than n”. In: *Annals of Statistics* 35 (2007), pp. 2313–2351.

[21] M. P. do Carmo. *Riemannian Geometry*. Boston: Birkhäuser, 1992.

[22] Y. Chen, P. Du, and Y. Wang. “Variable selection in linear models”. In: *Wiley Interdisciplinary Reviews: Computational Statistics* 6 (2014), pp. 1–9.

[23] G. M. Cordeiro and P. McCullagh. “Bias correction in generalized linear models”. In: *Journal of the Royal Statistical Society: Series B* 53.3 (1991), pp. 629–643.

[24] D. R. Cox. "Discussion of paper by D. Oakse entitled "Survival Times: Aspects of Partial Likelihood"". In: *International Statistical Review* 49.3 (1981), p. 258.

[25] D. R. Cox. "Partial Likelihood". In: *Biometrika* 62.2 (1975), pp. 269–276.

[26] D. R. Cox. "Regression Models and Life-Tables". In: *Journal of the Royal Statistical Society. Series B* 34.2 (1972), pp. 187–220.

[27] D. R. Cox and D. Oakes. *Analysis of Survival Data*. Monographs on Statistics and Applied Probability. London: Chapman and Hall, 1984.

[28] B. Efron. "The Efficiency of Cox's Likelihood Function for Censored Data". In: *Journal of the American Statistical Association* 72.359 (1977), pp. 557–565.

[29] B. Efron et al. "Least Angle Regression". In: *The Annals of Statistics* 32.2 (2004), pp. 407–499.

[30] J. Fan, S. Guo, and N. Hao. "Variance estimation using refitted cross-validation in ultrahigh dimensional regression". In: *Journal of the Royal Statistical Society: Series B* 74.1 (2012), pp. 37–65.

[31] J. Fan and R. Li. "Variable Selection via Nonconcave Penalized Likelihood and Its Oracle Properties". In: *Journal of the American Statistical Association* 96.456 (2001), pp. 1348–1360.

[32] J. Fan and J. Lv. "Sure independence screening for ultrahigh dimensional feature space". In: *Journal of the Royal Statistical Society: Series B* 70.5 (2008), pp. 849–911.

[33] Y. Fan and C. Y. Tang. "Tuning parameter selection in high dimensional penalized likelihood". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 75.3 (2013), pp. 531–552.

[34] C. P. Farrington. "On assessing goodness of fit of generalized linear model to sparse data". In: *Journal of the Royal Statistical Society: Series B* 58.2 (1996), pp. 349–360.

[35] J. Friedman, T. Hastie, and R.Tibshirani. *glmnet: Lasso and Elastic-Net Regularized Generalized Linear Models*. http://CRAN.R-project.org/package=glmnet. R package version 1.1-5. 2010b.

[36] J. Friedman, T. Hastie, and R.Tibshirani. "Regularization Paths for Generalized Linear Models via Coordinate Descent". In: *Journal of Statistical Software* 33.1 (2010a), pp. 1–22.

[37] J. P. Gillet et al. "Multidrug Resistance–Linked Gene Signature Predicts Overall Survival of Patients with Primary Ovarian Serous Carcinoma". In: *Clinical Cancer Research* 18.11 (June 1, 2012), pp. 3197–3206. ISSN: 1557-3265. DOI: 10.1158/1078-0432.ccr-12-0056. URL: http://dx.doi.org/10.1158/1078-0432.ccr-12-0056.

[38] J. J. Goeman. "L1 Penalized Estimation in the Cox Proportional Hazards Model". In: *Biometrical Journal* 52.1 (2010), pp. 70–84. ISSN: 1521-4036. DOI: 10.1002/bimj.200900028. URL: http://dx.doi.org/10.1002/bimj.200900028.

[39] J. J. Goeman. "L1 penalized estimation in the Cox proportional hazards model". In: *Biometrical journal* 52.1 (2010), pp. 70–84.

[40] J. Gui and H. Li. "Penalized Cox regression analysis in the high-dimensional and low-sample size settings, with applications to microarray gene expression data". In: *Bioinformatics* 21.13 (2005), pp. 3001–3008.

[41] T. Hastie and B. Efron. *lars: Least Angle Regression, Lasso and Forward Stagewise*. http://CRAN.R-project.org/package=lars. R package version 1.2. 2013.

[42] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer, 2009.

[43] T. Hastie et al. "Forward stagewise regression and the monotone lasso". In: *Electronic Journal of Statistics* 1 (2007), pp. 1–29. URL: http://www.ams.org/mathscinet-getitem?mr=2312144.

[44] T. Hesterberg et al. "Least angle and L_1 penalized regression: A review". In: *Statistics Surveys* 2 (2008), pp. 61–93. URL: http://projecteuclid.org/euclid.ssu/1211317636.

[45] A. E. Hoerl and R. Kennard. "Ridge regression: Biased estimation for nonorthogonal problems". In: *Technometrics* 12 (1970), pp. 55–67.

[46] H. Ishwaran, U. B. Kogalur, and J. S. Rao. "spikeslab: Prediction and variable selection using spike and slab regression". In: *The R Journal* 2.2 (2010), pp. 68–73.

[47] H. Ishwaran, U. B. Kogalur, and J. S. Rao. *spikeslab: Prediction and variable selection using spike and slab regression*. http://CRAN.R-project.org/package=spikeslab. R package version 1.1.2. 2010b.

[48] G. James and P. Radchenko. "A generalized dantzig selector with shrinkage tuning". In: *Biometrika* 96 (2009), pp. 323–337.

[49] G. Jonsson et al. "Gene expression profiling-based identification of molecular subtypes in stage IV melanomas with different clinical outcome." In: *Clin Cancer Res* 16.13 (2010), pp. 3356–67.

[50] B. Jorgensen. "Exponential dispersion models". In: *Journal of the Royal Statistical Society, Series B* 49 (1987), pp. 127–162.

[51] B. Jorgensen. *The Theory of Dispersion Models*. London: Chapman & Hall, 1997.

[52] J. D. Kalbfleisch and R. L. Prentice. *The Statistical Analysis of Failure Time Data*. Wiley series in probability and statistics. Hoboken, New Jersey: John Wiley & Sons, Inc, 2002.

[53] R. E. Kass and P. W. Vos. *Geometrical Foundations of Asymptotic Inference*. New York: John Wiley & Sons, 1997.

[54] S. Konishi and G. Kitagawa. "Generalised information criteria in model selection". In: *Biometrika* 83.4 (1996), pp. 875–890.

[55] S. Konishi and G. Kitagawa. *Information criteria and statistical modeling*. Springer, 2008.

[56] S. Kullback and R. A. Leibler. "On Information and Sufficiency". In: *The Annals of Mathematical Statistics* 22 (1951), pp. 79–86.

[57] K. C. Li. "Asymptotic Optimality for C_p , C_L , Cross-Validation and Generalized Cross-Validation: Discrete Index Set". In: *Annals of Statistics* 15 (1987), pp. 958–975.

[58] R. C. Littell, W. W. Stroup, and R. J. Feund. *SAS for Linear Models*. 4^{th}. Cary, North Carolina: Sas Institute Inc., 2002.

[59] A. Loboda et al. "EMT is the dominant program in human colon cancer". In: *BMC Med Genomics* 20 (2011), pp. 4–9.

[60] P. McCullagh and J. A. Nelder. *Generalized Liner Models*. 2^{nd}. London: Chapman & Hall, 1989.

[61] A. D. R. McQuarrie and C. L. Tsai. *Regression and Time Series Model Selection*. 1^{st}. Singapore: World Scientific Publishing Co. Pte. Ltd., 1998.

[62] Ruoyan Meng. "Estimation of Dispersion Parameters in GLMs with and without Random Effects". MA thesis. Institute of Mathematical Statistics, 2004.

[63] S. H. Moolgavkar and D. J. Venzon. "Confidence regions in curved exponential families: application to matched case-control and survival studies with general relative risk function". In: *The Annals of Statistics* 15.1 (1987), pp. 346–359.

[64] G. Nagel et al. “Metabolic risk factors and skin cancer in the Metabolic Syndrome and Cancer Project (Me-Can)”. In: *British Journal of Dermatology* 167.1 (2012), pp. 59–67.

[65] D. Oakes. “Survival Times: Aspects of Partial Likelihood”. In: *Internat. Statist. Rev.* 49.3 (1981), pp. 235–252.

[66] H. H. Panjer. *Operational Risk: Modeling Analytics*. New York: John Wiley & Sons, 2006.

[67] M. Y. Park and T. Hastie. *glmpath: L_1 Regularization Path for Generalized Linear Models and Cox Proportional Hazards Model*. http://CRAN.R-project.org/package=glmpath. R package version 0.94. 2007b.

[68] M. Y. Park and T. Hastie. “L_1-Regularization Path Algorithm for Generalized Linear Models”. In: *Journal of the Royal Statistical Society, Series B* 69.4 (2007a), pp. 659–677.

[69] H. Pazira, L. Augugliaro, and E.C. Wit. “A Software Tool for Estimating the Dispersion Parameter for High-dimensional GLMs”. In: *Submitted to Journal of Statistical Software* (2017).

[70] H. Pazira, L. Augugliaro, and E.C. Wit. “Extended differential geometric LARS for high-dimensional GLMs with general dispersion parameter”. In: *Statistics and Computing* (2017). DOI: 10.1007/s11222-017-9761-7. URL: http://dx.doi.org/10.1007/s11222-017-9761-7.

[71] R. Peto and J. Peto. “Asymptotically Efficient Rank Invariant Test Procedures”. English. In: *Journal of the Royal Statistical Society. Series A (General)* 135.2 (1972), ISSN: 00359238. URL: http://www.jstor.org/stable/2344317.

[72] R. L. Prentice and N. E. Breslow. “Retrospective Studies and Failure Time Models”. In: *Biometrika* 65.1 (1978), pp. 153–158.

[73] R. L. Prentice and M. W. Mason. “On the application of linear relative risk regression models”. In: *Biometrics* 42.1 (1996), pp. 109–120.

[74] R. L. Prentice and S. G. Self. “Asymptotic distribution theory for Cox-type regression models with general relative risk form”. In: *The Annals of Statistics* 11.3 (1983), pp. 804–813.

[75] R. L. Prentice, Y. Yoshimoto, and M. Mason. “Relationship of cigarette smoking and radiation exposure to cancer mortality in Hiroshima and Nagasaki”. In: *Journal of National Cancer Institute* 70.4 (1983), pp. 611–622.

[76] W. H. Press et al. *Numerical Recipes in Fortran 77: The Art of Scientific Computing*. 2^{nd}. England: Cambridge University Press, 1992.

[77] R Development Core Team. *R: A Language and Environment for Statistical Computing*. ISBN 3900051-07-0. R Foundation for Statistical Computing. Vienna, Austria, 2012. URL: http://www.R-project.org.

[78] C. R. Rao. "Information and the accuracy attainable in the estimation of statistical parameters". In: *Bull. Calc. Math. Soc.* 37 (1945), pp. 81–91.

[79] C. R. Rao. "On the distance between two populations". In: *Sankhy* 9 (1949), pp. 246–248.

[80] R. C. A. Rippe, J. J. Meulman, and P. H. C. Eilers. "Visualization of genomic changes by segmented smoothing using an L 0 penalty". In: *PloS one* 7.6 (2012), e38230.

[81] R. W. Ross et al. "A whole-blood RNA transcript-based prognostic model in men with castration-resistant prostate cancer: a prospective study." In: *Lancet Oncol* 13.11 (2012), pp. 1105–13.

[82] S. Rosset and J. Zhu. "Discussion of "Least Angle Regression" by Efron". In: *Annals of Statistics* 32 (2004), pp. 469–475. URL: http://www.ams.org/mathscinet-getitem?mr=2060166.

[83] S. Rosset and J. Zhu. "Piecewise linear regularizated solution paths". In: *The Annals of Statistics* 35.3 (2007), pp. 1012–1030.

[84] G. Schwarz. "Estimating the Dimension of a Model". In: *Annals of Statistics* 6.2 (1978), pp. 461–464.

[85] J. Shao. "An Asymptotic Theory for Linear Model Selection". In: *Statistica Sinica* 7 (1997), pp. 221–264.

[86] R. Shibata. "An Optimal Selection of Regression Variables". In: *Biometrika* 68 (1981), pp. 45–54.

[87] R. Shibata. "Approximation Efficiency of a Selection Procedure for the Number of Regression Variables". In: *Biometrika* 71 (1984), pp. 43–49.

[88] N. Simon et al. "Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent". In: *Journal of Statistical Software* 39.5 (2011), pp. 1–13.

[89] I. Sohn et al. "Gradient lasso for Cox proportional hazards model". In: *Bioinformatics* 25.14 (2009), pp. 1775–1781.

[90] M. Spivak. *A Comprehensive Introduction to Differential Geometry*. 2^{nd}. Boston: Publish or Perish, 1979.

[91] M. Stone. "Asymptotics for and against cross-validation". In: *Biometrika* 64 (1977), pp. 29–35.

[92] D. C. Thomas. "Addendum to the paper by Liddell, McDonald, Thomas and Cunliffe". In: *Journal of the Royal Statistical Society. Series A* 140.4 (1977), pp. 483–485.

[93] D. C. Thomas. "General Relative-Risk Models for Survival Time and Matched Case-Control Analysis". In: *Biometrics* 37.4 (1981), pp. 673–686.

[94] R. Tibshirani. "Regression Shrinkage and Selection Via the Lasso". In: *Journal of the Royal Statistical Society, Series B* 58.1 (1996), pp. 267–288.

[95] R. Tibshirani. "The lasso method for variable selection in the Cox model". In: *Statistics in medicine* 16 (1997), pp. 385–395.

[96] J. Ultricht and G. Tutz. *Combining Quadratic Penalization and Variable Selection via Forward Boosting*. Tech. rep. Technical Reports No. 99. Munich University: Department of Statistics, 2011.

[97] P. W. Vos. "A geometric approach to detecting influential cases". In: *Annals of Statistics* 19 (1991), pp. 1570–1581.

[98] B. -C. Wei. *Exponential Family Nonlinear Models*. Singapore: Springer, 1998.

[99] E. T. Whittaker and G. Robinson. *The Calculus of Observations: An Introduction to Numerical Analysis*. 4^{th}. New York: Dover Publications, 1967.

[100] E.C. Wit, L. Augugliaro, and H. Pazira. "Sparse Relative Risk Regression Models". In: *Submitted to Biostatistics* (2017).

[101] S. N. Wood. *Generalized Additive Models: An Introduction with R*. Boca Raton: Chapman & Hall/CRC, 2006.

[102] C. H. Zhang. "Nearly Unbiased Variable Selection Under Minimax Concave Penalty". In: *Annals of Statistics* 38.2 (2010), pp. 894–942.

[103] H. Zou and T. Hastie. *elasticnet: Elastic-Net for Sparse Estimation and Sparse PCA*. http://CRAN.R-project.org/package=elasticnet. R package version 1.1. 2005b.

[104] H. Zou and T. Hastie. "Regularization and Variable Selection via the Elastic Net". In: *Journal of the Royal Statistical Society, Series B* 67.2 (2005a), pp. 301–320.

[105] H. Zou and T. Hastie. “Regularization and variable selection via the elastic net”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.2 (2005), pp. 301–320.

Acknowledgements

First and foremost, I want to thank my supervisor, Prof. Ernst Wit, for his guidance, advice and encouragement throughout my Ph.D journey. Without his constant support, this Ph.D would not have been achievable. I am deeply grateful to Prof. Luigi Augugliaro, who acts like a co-supervisor. He has been helpful in guiding me during my Ph.D.

I would like to show my appreciation to assessment committee, Prof. Edwin Van den Heuvel, Prof. Christine Gräfin zu Eulenbur, and Prof. Angelo Mineo for providing valuable comments.

Special acknowledgments go to all my teachers from various universities in the world. I am very grateful to Prof. Ernst wit, Prof. Wim Krijnen, Prof. Fentaw Abegaz, Prof. Parviz Nasiri, Prof. Ali Shadrokh, Prof. Masoud Yarmohammadi, Prof. Farhad Yaghmaie and Dr. Einolah Deiri.

I am indeed very thankful to all colleagues from the Statistics & Probability group of the Johann Bernoulli Institute (JBI), particularly to Mahdi Mahmoudi, Mahdi Shafiee, Vladimír, Francisco, Reza, Pariya, Balafas and Sourab.

I also would like to thank those who supported me and my wife at initial stage of PhD in the Netherlands. These are Dick (Deceased) and Anneke. We enjoyed talking to you both. God bless Dick.

I am very grateful to my friend Mohammad Shabanian for enormous support. His sense of humor is another inspiring thing. I also would like to thank Dr. Saemeh Dehghan, the MRI center Badr for funding me during my Ph.D.

Now, time to thank family. In my case these are parents Mahin and Mousa, brother Behrouz and sister Behnoush. They have always supported me throughout my career. Especially my mother was always ready to provide any support she can. My deep appreciation goes to my wife Saemeh for standing beside me and belief in me throughout my Ph.D. She has been my inspiration and motivation for continuing to improve my knowledge and move my career forward. Thank you for everything especially for your endless love and encouragement! Without your help I wouldn't have had the opportunity to continue my study!

And last but definitely not least, I would like to dedicate this book to my son Nikan, who is six months old at the time of writing. You always makes me smile. You are the best child a dad could hope for: happy, loving, and fun to be with. It's wonderful watching you grow! I hope that one day you can read this book and understand why I spent so much time in front of my computer. I look forward to discussing this book with you. Love you!

Hassan, Groningen, 2017
pazira.b@gmail.com