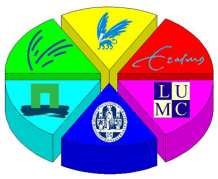
**Master Thesis**

# Penalized logistic regression: a quadratic difference penalty

Nynke C. Krol

30.08.2013




Universiteit Leiden
Mathematics, Specialization: Statistical Science

**Supervisors**
Dr. Jelle J. Goeman
Dr. Erik W. van Zwet
Prof. dr. Jacqueline J. Meulman

# Contents

# Abstract

We present a quadratic difference penalty on logistic regression as a solution for the high dimensional data problem and spatial correlation in the classification of genetic copy number data. The quadratic difference penalty is the $L_2$ norm of the first order difference penalty matrix times the coefficient vector, and thereby shrinks adjacent regression coefficients to each other. We propose an $L_2$ fused lasso, a logistic lasso with an extra quadratic difference penalty; and a smoothed logistic regression, a logistic regression with only a quadratic difference penalty. We construct algorithms for both penalized regressions. We explain the connection between our smoothed logistic regression and ridge regression. We demonstrate the challenges in fitting a lasso, and adapt the gradient ascent. The $L_2$ fused lasso and smoothed logistic regression are applied on genetic copy number data to classify the grade of bladder tumors.

# 1. Introduction

Genetic information is highly important in cancer research. The genome in cancer cells is mutated; cancer cells often contain an increased or decreased number of copies of a DNA segment (Stratton et al., 2009). A decreased number of copies may result in the complete absence of a DNA sequence from the genome. Copy numbers are altered at genomic regions across a wide range of cancer types (Beroukhim et al., 2010). Copy number data consists of the number of copies of small DNA sequences in a sample, and therefore contains information about possible alterations.

Copy number data is important in the understanding of the biology of cancer. Diagnosis and treatment of cancer can also be directly improved by copy number data analysis. Stuart & Sellers (2009) stress the importance of linking genetic alterations to therapy in cancers. Important findings from copy number data are i.a. that high gene copy numbers per cell show a trend towards poor prognosis in non–small-cell lung carcinomas (Hirsch et al., 2003), and that a specific copy number profile can be used to predict benefit from therapy in breast cancer (Vollebergh et al., 2011). Consequently, there is an important role in copy number data analysis to make valuable differentiations in cancers, e.g. separate patients with cancer in two groups that either will or will not respond to a specific therapy on basis of the copy number data of their tumor's cells. Furthermore, copy number data can be used to identify key genes in cancers (Beroukhim et al., 2010; Kan et al., 2010). And in the recent discussion about overdiagnosis in cancer due to extensive screening, it is important to distinguish between abnormal tissue which will lay dormant and tissue which will evolve into evasive cancer (Brewster et al., 2011; Fletcher & Fletcher, 2005).

Therefore, it is clinically relevant to develop classifiers that can distinguish between different genetic cancer groups, e.g. groups with a different prognosis, on the basis of copy number data.

In the next section, we will explore the nature of whole–genome copy number data, and how it is retrieved from microarrays. Then we will explain the challenges that arise when working with classifiers that use copy number data, which make traditional statistical techniques fail. Thereafter, we will explain why regularizing logistic regression by a difference penalty solves these problems. We will explain why we chose a quadratic difference penalty. Finally, we will propose two models which incorporate a quadratic difference penalty: 1) smoothed logistic regression, and 2) the $L_2$ fused lasso.

Aim of this thesis is to explore a lasso and a logistic regression with a quadratic difference penalty, i.e. the $L_2$ fused lasso and smoothed logistic regression, both in its model design and its practical implementation on copy number data.

## Copy Number Data from Microarrays.

As mentioned before, copy number data consists of the number of copies of small DNA sequences in a sample. Copy number data is usually measured via comparative genomic hybridization (CGH). Microarrays are used to retrieve whole–genome copy number data via array CGH (Pinkel et al., 1998; Pollack et al., 1999).

A microarray is a small chip with a large number of ordered DNA probes on it. DNA probes are a known sequence of nucleotide bases, and are often 25–50 bases long. For array CGH, a DNA sample of interest is fragmented into small segments and denatured, which means it is single–stranded. The microarray is washed with a solution of this DNA sample. The sample DNA connects with a complimentary DNA probe by base pairing: hybridization. Before or after hybridization, the DNA is labeled, e.g. with fluorophore. The more DNA segments connect to probes at a spot, the stronger the e.g. fluorescent signal from their labels is at that location. The signal intensity can be measured relative to an own differently labeled sample, or relative to a general reference genome. The signal intensity tells us how many copies of a probed DNA segment there are in our sample (Dziuda, 2010; Theisen, 2008).

The signal ratios from the microarray correspond to losses and gains of the probed DNA segments. A DNA segment with a normal amount of copies (2) has a ratio of 1, the gain of one copy is a ratio of $\frac{3}{2}$, the gain of two copies is a ratio of $\frac{4}{2}$, and so on. In line with that is the loss of a copy a ratio of $\frac{1}{2}$, and the complete loss of a chromosome fragment gives a ratio of 0. Copy number data usually consists of the base-2 logarithm of the ratio per probed DNA segment, so that a a normal amount of copies corresponds to a $\log_2$ ratio of 0, and the gain of a probed DNA segments is a $\log_2$ ratio of $\log_2(\frac{3}{2}) \approx .58$. In the analysis of tumors, the $\log_2$ intensity ratios often shrink towards zero, mainly because tumor biopsies generally consist of a mixture of normal and cancer cells (Hupé et al., 2004).

An example of copy number data of the genome of a tumor cell can be seen in Fig. 1.1. The $\log_2$ ratios for 2161 probed DNA segments from 22 chromosomes is plotted. As is usual in cancer cells, there are losses and gains of parts of the genome. For example, there is evidence that one copy of chromosome 4 is lost entirely in part of the sample.

Hereafter, we will for simplicity refer to probed DNA segments as genes.

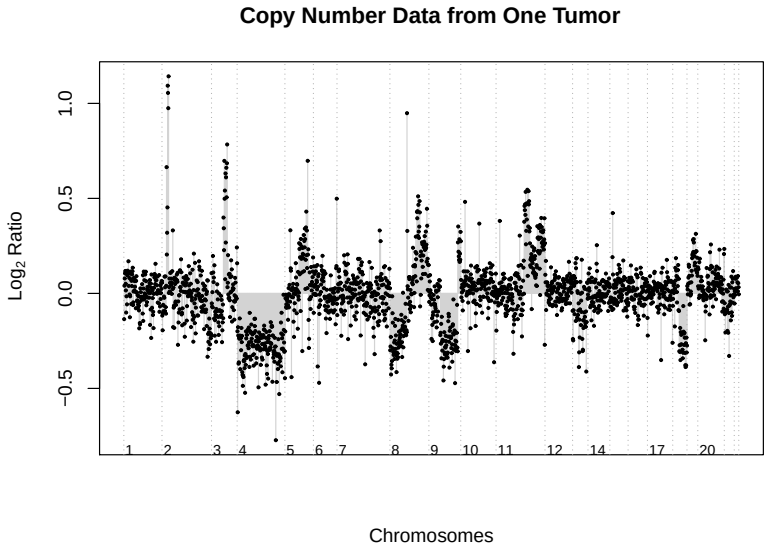



**Figure 1.1.:** *Example of a copy number profile from one bladder cancer tumor with grade 3. Gains and losses of 2161 probed DNA segments are represented by deviations of the $log_2$ ratio from zero.*

## Statistical Challenges

In the construction of a classification method for copy number data, there are two caveats. Firstly, copy number data is usually high dimensional data: thousands to tens of thousands measured genes per sample are not infrequent. Secondly, copy number data often has spatial correlation. Particularly important in the analysis of the DNA from cancer cells is that the copy number alterations in cancer cells often involve regions, as large as gains and losses of chromosome arms and entire chromosomes (Stratton et al., 2009; Beroukhim et al., 2010). This phenomenon can also be seen in Fig. 1.1. When regional copy number alterations are present, the copy number data will have spatial correlation.

How do we implement a method that can classify high dimensional copy number data with spatial correlation?

We need a method that:

- can classify;
- can be used on high dimensional data;
- can take spatial correlation into account.

*Classification.* A conventional method to classify samples in two distinct categories on basis of other information is logistic regression. In our case, logistic regression could be used to split the copy number profiles into two groups. Examples of the kind of groups have been previously mentioned, as groups that have a different

prognosis, benefit from a different treatment, tumors that progress to a lethal state, etc. In logistic regression, the probabilities of group membership are modeled. Logistic regression as a basic model permits additions and modifications to fit more complicated models.

*High Dimensional Data.* As regression coefficients can not be estimated when there are more variables measured than there are samples, in this case more genes measured than there are patients, a logistic regression can not be fitted on high dimensional data. This problem can be solved by a modification to the ordinary logistic regression, for example by penalized regression. Penalized regression shrinks the regression coefficients towards zero by putting a constraint on their size, thereby trading variance for bias. This constraint is most frequently an absolute value penalty, which constrains the $L_1$ norm of the regression coefficients as in lasso regression (Tibshirani, 1996), or a quadratic penalty, which constrains the $L_2$ norm of the regression coefficients as in ridge regression (Hoerl & Kennard, 1970). A combination of both, where both an $L_1$ and an $L_2$ penalty are imposed, is known as the elastic net (Zou & Hastie, 2005).

*Spatial Correlation.* Instead of, or in addition to, a penalty to solve the high dimensional data problem, a penalty that takes spatial correlation into account can be imposed on a regression. This is a so called difference, or fused penalty. The difference penalty minimizes the differences between adjacent regression coefficients, in our case the regression coefficients corresponding to neighboring genes. Most common are the absolute value difference penalty, that shrinks neighboring coefficients to exactly the same size; or the quadratic difference penalty, which shrinks the coefficients to one another in a curvature pattern. Both types of minimizing the differences between adjacent regression coefficients are known as smoothing, since they result in smooth coefficient patterns.

## Popular Solution: $L_1$ Fused Lasso

In summary, to classify cancer matters by copy number data, a logistic regression model with a difference penalty can be used. A popular choice in copy number analysis is the lasso with an additional absolute value difference penalty: the fused lasso (Tibshirani et al., 2005). To avoid confusion about the type of difference penalty that is applied, we will from here on refer to the lasso with an absolute value difference penalty as the $L_1$ fused lasso. The $L_1$ fused lasso has recently been implemented for regression into the `R` package genlasso (Tibshirani & Taylor, 2011) and for i.a. logistic regression into the `R` package penalized (Chatuverdi et al., 2013; Goeman, 2010).

The $L_1$ fused lasso as a classifier for copy number data matches not only with the regional copy number alterations, but also with the tendency that the neighboring genes often hold the exact same number of copies, when for example a whole chromosome arm is deleted. The sparsity in coefficients generated by the lasso penalty,

and the sparsity in coefficient change by the absolute value difference penalty is expected to fit copy number profiles.

## Our Solution: Quadratic Difference Penalty

In contrast to the absolute value difference penalty from the $L_1$ fused lasso, a difference penalty of a quadratic nature could also be imposed on a lasso: the $L_2$ fused lasso. The addition of a quadratic penalty to a lasso to incorporate the spatial structure of genetic data is most known for the implementation of prior biological knowledge on individual genes and gene functional groups (Li & Li, 2008; Slawski et al., 2010; Sun & Wang, 2012).

Using an $L_2$ fused lasso in copy number data analysis is not a conventional procedure. However, its wavy coefficient pattern and easier divergence from neighboring values could be advantageous in predicting clinically relevant outcomes, we expect that these characteristics make the $L_2$ fused lasso successful in selecting key genes and regions that are influential in the cancer process. We therefore chose to impose a quadratic difference penalty on logistic regression when classifying cancer matters by copy number data.

To explore the logistic regression with a quadratic difference penalty in greater depth, we not only applied the quadratic difference penalty to a logistic lasso but also to an ordinary logistic regression. We chose to explore the following $L_2$–type fused methods:

- *Smoothed Logistic Regression,* a logistic regression with only an $L_2$–type difference penalty. The smoothing as a penalized regression method has both the sparsity and simplicity of a one penalty problem, and the practical advantage of the inclusion of adjacent gene difference. It is a basic fused model with fast optimization properties.
- $L_2$ *fused lasso*, or structured elastic net (Li & Li, 2008; Slawski et al., 2010), which we applied as a logistic lasso with an additional $L_2$–type difference penalty. This method extends the most popular penalization method lasso with an $L_2$ fused penalty.

## Structure of this Thesis

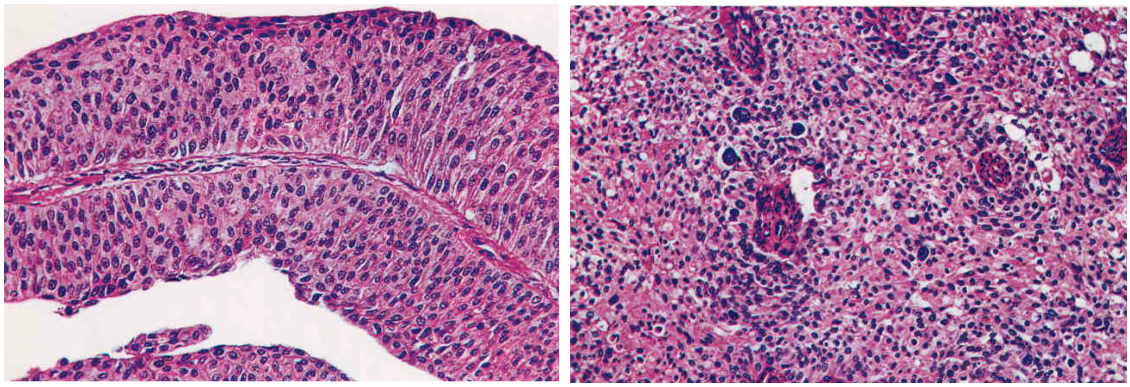
As the aim of this thesis is to explore a lasso and a logistic regression with a quadratic difference penalty, i.e. the $L_2$ fused lasso and smoothed logistic regression, both in its model design and as a practical classifier on copy number data, the following topics are discussed:

*Model Design.* Two of the chapters in this thesis are dedicated to model design, chapter 2 the smoothed logistic regression, and chapter 3 to the $L_2$ fused lasso. The

main focus in both chapters is the construction of algorithms that can fit our logistic regressions with a quadratic difference penalty, while also exploring the underlying theory of these models.

*Data Analysis.* To test the capacities of the $L_2$ fused lasso and smoothed logistic regression as a practical classifier, we applied both methods on a copy number data classification problem. As an interest is the comparison of our methods to the $L_1$ fused lasso, we followed Chatuverdi et al. (2013) in classifying the grade of bladder tumors by copy number data from the tumor cells. The tumor grade is a measure of the aggressiveness of the cancer, and is usually classified by visual criteria. A microscopic image of bladder tumor samples can be seen in Fig. 1.2. We repeated the grade classification with an $L_2$ fused lasso and a smoothed logistic regression and, for completeness, with the basic penalized regression methods ridge and lasso (chapter 4).

*Discussion.* This thesis is concluded with a discussion about the proposed lasso and logistic regression with a quadratic difference penalty, its applications on the bladder cancer dataset, and directions for future research.



**(a)** *Grade 1* **(b)** *Grade 3*

**Figure 1.2.:** *Transitional cell tumor preparations, with a lower (left) and higher (right) grade. Transitional cell tumors typically occur in the urinary system, including the bladder.* Reprinted from *Histological typing of urinary bladder tumors,* by F. K. Mostofi, L. H. Sobin and H. Torloni, 1973, Geneva: World Health Organization.

# 2. Model Design: Smoothed Logistic Regression

This chapter explores the model design of the logistic regression with a quadratic difference penalty, *smoothed logistic regression*. The main focus while going through this model design is the construction of an algorithm to fit a smoothed logistic regression.

In this chapter, we will first outline the connection between smoothing via splines and penalized regression. Then, in pursuit of an efficient smoothed logistic regression algorithm, the form of a logistic regression is shown and the principle of maximum likelihood estimation to obtain regression coefficients is presented. The penalization of regression coefficients by a quadratic penalty is explained by ridge logistic regression, as well as an effective optimization by the Newton–Raphson method. The Newton–Raphson optimization is extended to maximum likelihood estimation in smoothed logistic regression. A failed attempt to speed up the smoothed logistic regression optimization is presented.

Then a successful attempt at an efficient smoothed logistic regression algorithm is described. The connection between smoothed regression and ridge regression is exploited; as a ridge algorithm is used as a smoothed logistic regression algorithm by algebraically rewriting the input and output submitted to an ordinary ridge algorithm. To extend the possibilities of quadratic penalized regressions, a small intermezzo about ridge regression with a quadratic difference penalty is presented. This chapter is concluded with a short summary.

## 2.1. The Basis of Smoothed Regression

To capture the underlying trend in data better than a regression model, a smooth curve can be fitted to the data, notably via splines. Regression splines are an extension of regression, but with a different basis. The basis of a straight line univariate regression, $y_i = \beta_0 + \beta_1 x_i + \epsilon$, are the functions 1 and $x$. The $n \times 2$ dimensional data matrix $\mathbf{X}$ that is used in the straight line univariate regression is of the form

$$\mathbf{X} = [\mathbf{1} \,|\, (x_1, ..., x_n)^{\mathrm{T}}],$$

where $n$ is the number of data points and $\mathbf{1}$ is a vector of length $n$ of all 1's. This basis of the functions 1 and $x$ can be adapted to improve the data fit, e.g. improve the local fit by constructing a regression line with a different slope per segment. The curve from a segmental spline can be (further) smoothed by a roughness penalty that constrains the influence of the knots at which the line segments connect. Popular choices for the roughness penalty of these penalized splines are a penalty based on the second derivative or a discrete difference penalty. More on penalized splines can be found in Ruppert et al. (2003).

Land & Friedman (1996) proposed fusion regression methods. Essentially, these fusion regression methods are penalized splines on a zero–order truncated power basis constructed in a way it corresponds to the first differences between regression coefficients of adjacent entries (in our case, of neighboring genes). How this basis is constructed will become clear in sec. 2.3. Land & Friedman (1996) note that their first–order variable fusion, a regression with a constraint on the $L_1$ norm of the first differences between adjacent regression coefficients, is a lasso on these same variables. Corresponding to that, our smoothed regression is ridge regression on these variables. Sec. 2.3 will demonstrate this.

When the basis is adapted, smoothed regression can be fitted as ridge regression. When the ordinary regression basis is used however, the first differences should be incorporated in the form of the penalty. The first difference penalty we use for smoothed regression is most well known from Eilers & Marx (1996) who add difference penalties to a regression on a B–spline basis. The form of the first difference penalty is shown in sec. 2.2.

## 2.2. Fitting with a Quadratic Difference Penalty

Now that the relationship between ridge regression and smoothed regression is formulated, it is illustrative to look at the form and fit of both methods. Since the focus of this thesis is logistic regression, we will start with ordinary logistic regression and expand to ridge and smoothed logistic regression. Finally, an algorithm to fit smoothed logistic regression on an ordinary regression basis is presented, and a failed trick to speed up this algorithm.

### 2.2.1. Logistic Regression

Logistic regression is a method to model a binary outcome variable. The logistic transformation of the vector of probability estimates $\mathbf{p}$ is modeled by a linear function of the $n \times p$ predictor matrix $\mathbf{X}$ as

$$\log \frac{\mathbf{p}}{1-\mathbf{p}} = \mathbf{X}^{\mathrm{T}}\boldsymbol{\beta},$$

where $n$ is the number of samples, $p$ is the number of predictors, and $\boldsymbol{\beta} = (\beta_1, \beta_2, ...\beta_p)^{\mathrm{T}}$ is the vector of regression coefficients. Solving for $\mathbf{p}$ gives

$$\mathbf{p} = \frac{\exp(\mathbf{X}^{\mathrm{T}}\boldsymbol{\beta})}{\{1 + \exp(\mathbf{X}^{\mathrm{T}}\boldsymbol{\beta})\}}.$$

Logistic regression models are usually fitted by maximum likelihood estimation, where given the data the set of coefficients $\boldsymbol{\beta}$ is found for which the probability of the observed data is greatest. The function to maximize is the log likelihood:

$$\ell(\boldsymbol{\beta}) = \mathbf{y}^{\mathrm{T}} \log \mathbf{p} + (1 - \mathbf{y})^{\mathrm{T}} \log(1 - \mathbf{p}),$$

where $\mathbf{y}$ is the binary output vector. Maximizing the log likelihood of logistic regression yields the maximum likelihood estimates $\hat{\boldsymbol{\beta}}$ of the logistic regression coefficients $\boldsymbol{\beta}$.

### 2.2.2. Ridge Logistic Regression

Hoerl & Kennard (1970) proposed the quadratic regularization method ridge regression, and le Cessie & van Houwelingen (1992) applied logistic ridge regression on a biomedical problem. Ridge regression shrinks the regression coefficients, by constraining the sums of squares of the coefficients. The ridge log likelihood is thus defined as

$$\ell_{ridge}(\boldsymbol{\beta}) = \ell(\boldsymbol{\beta}) - \frac{1}{2}\lambda\sum_{i=1}^{p}(\beta_i)^2,$$

where $\ell(\boldsymbol{\beta})$ is the unpenalized likelihood, and $\lambda$ determines the size of the ridge penalty. Despite the fact that the ridge log likelihood is not actually a likelihood, there will be referred to the penalized likelihoods of the regressions presented in this thesis as likelihoods. With coefficients $\boldsymbol{\beta} = (\beta_1, ...\beta_p)^{\mathrm{T}}$, the ridge log likelihood can be reformulated as

$$\ell_{ridge}(\boldsymbol{\beta}) = \ell(\boldsymbol{\beta}) - \frac{1}{2}\lambda \left\|\boldsymbol{\beta}\right\|_2^2,$$

where $||.||_2$ is the $L_2$ norm. The ridge estimates $\hat{\boldsymbol{\beta}}_{ridge}$ of the regression coefficients $\boldsymbol{\beta}$ follow from the constrained likelihood optimization:

$$\hat{\boldsymbol{\beta}}_{ridge} = \text{argmax } \ell_{ridge}(\boldsymbol{\beta}). \tag{2.1}$$

**The Newton–Raphson method.** The Newton–Raphson method is the usual routine for maximum likelihood optimization. Newton–Raphson approximates the target function by a second order Taylor series, and optimizes that approximation. A Newton–Raphson step is repeated until convergence at the optimum, resulting in an *iteratively reweighed least squares* (IRLS) algorithm. Because of the second order Taylor approximations, Newton–Raphson requires the target function to be twice differentiable. The matrix of second derivatives is called the hessian. Also, this hessian is required to be negative definite for maximum likelihood estimation by the Newton–Raphson method. If the hessian is positive definite, the IRLS algorithm converge to a minimum; If the hessian is indefinite, both positive and negative eigenvalues, the second order Taylor approximations converge to a saddle point.

**IRLS Ridge Algorithm.** Now assume that the hessian of the ridge log likelihood is indeed negative definite, and an IRLS algorithm can be used to solve Eq. 2.1. A Newton–Raphson step updates the regression coefficients as

$$\boldsymbol{\beta}_{new} = \boldsymbol{\beta}_{old} + \left( -\frac{\partial^2 \ell_{ridge}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^{\mathrm{T}}} \right)^{-1} \frac{\partial \ell_{ridge}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}, \tag{2.2}$$

where the vector of first derivatives of the ridge log likelihood is

$$\frac{\partial \ell_{ridge}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \mathbf{X}^{\mathrm{T}}(\mathbf{y}-\mathbf{p}) - \lambda \boldsymbol{\beta}, \tag{2.3}$$

and the hessian is

$$\mathbf{H} = \frac{\partial^2 \ell_{ridge}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^{\mathrm{T}}} = -\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} - \lambda \mathbf{I}, \tag{2.4}$$

where $\mathbf{W}$ is a diagonal matrix with entries $p_i(1-p_i)$ for $i = 1, \ldots, n$ and $\mathbf{I}$ is the $p \times p$ identity matrix. As can be seen, this hessian matrix is twice differentiable. The constraint that is subtracted from the diagonal of the $\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}$ matrix gives this matrix full rank. It is a positive definite matrix. The negative of this matrix, the hessian, is therefore negative definite. Hence an IRLS algorithm can be used to solve Eq. 2.1.

Filling in the vector of first derivatives (Eq. 2.3) and the hessian matrix (Eq. 2.4) in the Newton–Raphson updates (Eq. 2.2) gives

$$\begin{aligned}\boldsymbol{\beta}_{new} &= \boldsymbol{\beta}_{old} + (\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \lambda\mathbf{I})^{-1}(\mathbf{X}^{\mathrm{T}}(\mathbf{y}-\mathbf{p}) - \lambda\boldsymbol{\beta}_{old}) \qquad (2.5)\\ &= \{(\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \lambda\mathbf{I})^{-1}(\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \lambda\mathbf{I})\}\boldsymbol{\beta}_{old} + \\ &\{(\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \lambda\mathbf{I})^{-1}(\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \lambda\mathbf{I})\}(\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \lambda\mathbf{I})^{-1}(\mathbf{X}^{\mathrm{T}}(\mathbf{y}-\mathbf{p}) - \lambda\boldsymbol{\beta}_{old}) \\ &= \{(\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{W}\}\mathbf{X}\boldsymbol{\beta}_{old} + (\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \lambda\mathbf{I})^{-1}\lambda\boldsymbol{\beta}_{old} + \\ &\{(\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{W}\}\mathbf{W}^{-1}(\mathbf{y}-\mathbf{p}) - (\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \lambda\mathbf{I})^{-1}\lambda\boldsymbol{\beta}_{old} \\ &= (\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{W}\{\mathbf{X}\boldsymbol{\beta}_{old} + \mathbf{W}^{-1}(\mathbf{y}-\mathbf{p})\}. \qquad (2.6)\end{aligned}$$

By convention, the equation is algebraically rewritten from Eq. 2.5 to Eq. 2.6. One of the input vectors $\boldsymbol{\beta}_{old}$ is eliminated, the identity matrix $\mathbf{I} = \{(\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \lambda\mathbf{I})^{-1}(\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \lambda\mathbf{I})\}$ is added and the plus and minus $(\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \lambda\mathbf{I})^{-1}\lambda\boldsymbol{\beta}_{old}$ cross out. With this easy implementable equation of a Newton–Raphson step for the optimization of ridge regression (Eq. 2.6) an IRLS algorithm can be built that produces estimates of the regression coefficients $\boldsymbol{\beta}_{ridge}$.

### 2.2.3. Smoothed Logistic Regression

Now that the basic idea of optimizing a quadratic penalized regression is clear, we progress to the form of main interest: smoothed logistic regression.

The smoothed regression log likelihood is defined as

$$\ell_{smooth}(\boldsymbol{\beta}) = \ell(\boldsymbol{\beta}) - \frac{1}{2}\lambda_f\sum_{i=1}^{p-1}(\beta_{i+1} - \beta_i)^2, \qquad (2.7)$$

where $\ell(\boldsymbol{\beta})$ is the unpenalized likelihood, and $\lambda_f$ determines the size of the quadratic difference penalty. Tab. 2.1 presents an overview of the coefficient sums that are constrained in the penalized regressions. With coefficients $\boldsymbol{\beta} = (\beta_1, ...\beta_p)^{\mathrm{T}}$, the smoothed regression log likelihood can be reformulated as

$$\ell_{smooth}(\boldsymbol{\beta}) = \ell(\boldsymbol{\beta}) - \frac{1}{2}\lambda_f \left\|\mathbf{D}\boldsymbol{\beta}\right\|_2^2. \qquad (2.8)$$

**The Matrix D.** In Eq. 2.8, the matrix $\mathbf{D}$ is incorporated in the quadratic difference penalty. The simplest form of matrix $\mathbf{D}$ is a $p \times p$ first order difference penalty matrix, so that Eq. 2.7 and Eq. 2.8 exactly correspond:

$$\mathbf{D} = \begin{pmatrix} -1 & 1 & 0 & 0 & \cdots \\ 0 & -1 & 1 & 0 & \cdots \\ 0 & 0 & -1 & 1 & \cdots \\ 0 & 0 & 0 & -1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

| $q = 1$ | $q = 2$ |
|---|---|
| lasso<br>$-\lambda \sum_{i=1}^{p} \lvert\beta_i\rvert^q$ | ridge<br>$-\frac{1}{2}\lambda \sum_{i=1}^{p} \lvert\beta_i\rvert^q$ |
| | smoothing<br>$-\frac{1}{2}\lambda_f \sum_{i=1}^{p-1} \lvert\beta_{i+1} - \beta_i\rvert^q$ |

**Table 2.1.:** *In the penalized regressions, the penalty consists of the respective coefficient sums.*

This matrix $\mathbf{D}$ penalizes the differences between all adjacent regression coefficients. By adding the matrix $\mathbf{D}$, the penalty incorporates the differences between adjacent coefficients while the smoothed logistic regression is still regression on an ordinary basis.

There can be a desire to model more structure than is incorporated in the matrix $\mathbf{D}$. In copy number data for example, there is no interest in minimizing the difference between the end of one chromosome and the beginning of another. In that case, matrix $\mathbf{D}$ can be adapted to $\mathbf{D}_{chr}$, a matrix that consists of diagonal blocks of first order penalty matrices $\mathbf{C}_i$ for $i = 1, \dots m$, where $m$ is the number of chromosomes present in the data. And an unpenalized intercept can be added, represented by the empty first row and column of matrix $\mathbf{D}_{chr}$ so that:

$$\mathbf{D}_{chr} = \begin{pmatrix} 0 & 0 & 0 & 0 & \cdots \\ 0 & \mathbf{C}_1 & 0 & 0 & \cdots \\ 0 & 0 & \mathbf{C}_2 & 0 & \cdots \\ 0 & 0 & 0 & \mathbf{C}_3 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \text{ with } \mathbf{C}_1 = \begin{pmatrix} -1 & 1 & 0 & 0 & \cdots \\ 0 & -1 & 1 & 0 & \cdots \\ 0 & 0 & -1 & 1 & \cdots \\ 0 & 0 & 0 & -1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

When this algorithm is applied on other spatial data than genetic data, or when there is a desire to model another structure than the chromosomes, matrix $\mathbf{D}$ can be constructed out of other sized first order penalty matrices. Furthermore, relationships between not adjacent regression coefficients can be incorporated in the matrix $\mathbf{D}$, for example by a Laplacian matrix as the $\mathbf{D}$ matrix. A Laplacian matrix is a matrix representation of a graph, and thereby incorporates a structure of pathways.

**IRLS Smoothed Algorithm.** The smoothed logistic regression coefficients $\hat{\boldsymbol{\beta}}_{smooth}$ are estimated by optimizing the smoothed regression log likelihood

$$\hat{\boldsymbol{\beta}}_{smooth} = \text{argmax } \ell_{smooth}(\boldsymbol{\beta}).$$

Since the hessian is negative definite, which follows from its similarities to ridge regression, Newton–Raphson iterations can be used for optimization. Analogous to ridge regression (Eq. 2.5–2.6), the Newton–Raphson iterations become:

$$\begin{aligned}\boldsymbol{\beta}_{new} &= \boldsymbol{\beta}_{old} + (\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \lambda_f\mathbf{D})^{-1}(\mathbf{X}^{\mathrm{T}}(\mathbf{y}-\mathbf{p}) - \lambda_f\mathbf{D}\boldsymbol{\beta}_{old}) \\ &= (\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \lambda_f\mathbf{D})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{W}\{\mathbf{X}\boldsymbol{\beta}_{old} + \mathbf{W}^{-1}(\mathbf{y}-\mathbf{p})\}.\end{aligned}$$

### 2.2.4. Failed Attempt to Speed up the Newton–Raphson Smoothed Logistic Regression Algorithm

In the Newton–Raphson iterations in the IRLS logistic, ridge and smoothed logistic algorithm, a $p \times p$ matrix needs to be inverted. This is an expensive procedure, as large matrix inversions are very computational intensive. The computation time of the Newton–Raphson iterations can be substantially improved, if the problem is rewritten so than only an $n \times n$ matrix needs to be inverted. Indeed, the nature of high dimensional data is that $p \gg n$. This is straightforward in logistic and ridge regression, however not so in smoothed regression. We will present the reason by explaining a method to speed up a ridge regression IRLS algorithm.

Singular Value Decomposition (SVD) can be used to avoid the costly inversion of the hessian matrix in the Newton–Raphson iterations and instead invert an $n \times n$ matrix in ridge logistic regression. The singular value decomposition presented here is based on the work of Shen & Tan (2005) in high dimensional logistic ridge regression, but in simplified notation.

Singular values of matrix $\mathbf{X}$ are:

$$\mathbf{X} \approx \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathbf{T}}. \tag{2.9}$$

Let $\mathbf{R} = \mathbf{U}\boldsymbol{\Sigma}$ so that the SVD of $\mathbf{X}$ (Eq. 2.9) becomes $\mathbf{X} \approx \mathbf{R}\mathbf{V}^{\mathbf{T}}$. The matrix $\mathbf{R}$ is an $n \times n$ dimensional matrix. Substitute $\mathbf{R}\mathbf{V}^{\mathbf{T}}$ for $\mathbf{X}$ in the Newton–Raphson formula (Eq. 2.6 on page 13) so that

$$\boldsymbol{\beta}_{new} = (\mathbf{V}\mathbf{R}^{\mathrm{T}}\mathbf{W}\mathbf{R}\mathbf{V}^{\mathrm{T}} + \lambda\mathbf{I})^{-1}\mathbf{V}\mathbf{R}^{\mathrm{T}}\mathbf{W}\{\mathbf{X}\boldsymbol{\beta}_{old} + \mathbf{W}^{-1}(\mathbf{y}-\mathbf{p})\}. \tag{2.10}$$

Then assume $\boldsymbol{\beta} = \mathbf{V}\boldsymbol{\Theta}$. This is without unpenalized intercept. Multiply Eq. 2.10 by $\mathbf{V}^{\mathbf{T}}$ so that $\boldsymbol{\Theta}_{new}$ is computed instead of $\boldsymbol{\beta}_{new}$:

$$\boldsymbol{\Theta}_{new} = (\mathbf{R}^{\mathrm{T}}\mathbf{W}\mathbf{R} + \lambda\mathbf{I})^{-1}\mathbf{R}^{\mathrm{T}}\mathbf{W}\{\mathbf{X}\boldsymbol{\beta}_{old} + \mathbf{W}^{-1}(\mathbf{y}-\mathbf{p})\}. \tag{2.11}$$

After an IRLS algorithm of Eq. 2.11 converged, write as a final step the obtained coefficient estimates of $\boldsymbol{\Theta}$ back to $\boldsymbol{\beta}$:

$$\boldsymbol{\beta} = \mathbf{V}\boldsymbol{\Theta}.$$

And this is how an IRLS algorithm can be used to solve the ordinary ridge problem, with the inversion of an $n \times n$ matrix instead of the costly inversion of a $p \times p$ matrix.

Often, an (unpenalized) intercept is implemented in a ridge model. To solve this problem, remove the vector of 1's that is present in the first column of an $\mathbf{X}$ matrix that has an intercept and perform SVD on it:

$$\mathbf{X} = [\mathbf{1} \,|\, \mathbf{X}^{int}]$$

$$\mathbf{X}^{int} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathbf{T}}$$

Let $\mathbf{R}^* = [\mathbf{1} \,|\mathbf{U}\boldsymbol{\Sigma}]$, with $\mathbf{1}$ as a vector of length $n$ of all 1's. Use $\mathbf{R}^*$ instead of $\mathbf{R}$ in equation 2.11 to retrieve the $\boldsymbol{\Theta}_{new}$ when there is an intercept present. The $(n+1) \times (n+1)$ identity matrix $\mathbf{I}$ has an empty first entry when the intercept is not penalized. Then construct $\boldsymbol{\beta}$ out of the obtained $\boldsymbol{\Theta}$ as described below:

$$\boldsymbol{\beta} = \left(\boldsymbol{\Theta}_1, \mathbf{V}(\boldsymbol{\Theta}_2, ...\boldsymbol{\Theta}_p)^{\mathrm{T}}\right).$$

And a fast IRLS algorithm for ordinary ridge regression with intercept can be implemented.

Now, why can SVD not be so straightforwardly used for the smoothed logistic regression? Note how the $\lambda\mathbf{I}$ matrix in Eq. 2.10 is an $n \times n$ matrix. Adjacent entries do no longer correspond to e.g. neighboring genes but to "neighboring patients". Off–diagonal constrains as implemented by the $\mathbf{D}$ matrix can thus not be applied in this form. Therefore, SVD can not be so easily implemented in a fused regression optimization like smoothed logistic regression.

## 2.3. A Ridge Algorithm as a Smoothed Logistic Regression Algorithm

In the previous section, we used a non–adapted $\mathbf{X}$ matrix to fit a smoothed logistic regression. The differences between neighboring genes were incorporated in the penalty.

In this section, we adapt the basis of the regression, so that a ridge optimization can be used to perform smoothed regression. Big advantage is that a ridge algorithm can easily be fitted faster, as on the preceding page, since the inversion of an $n \times n$ matrix is a lot less computational intensive than the inversion of a $p \times p$ matrix in high dimensional data.

The ridge regression is fitted on a basis constructed in a way it corresponds to the first differences between regression coefficients of neighboring genes, by algebraically rewriting the input and output submitted to an ordinary ridge regression method. To change the ridge logistic regression to a smoothed logistic regression method, define

$$\boldsymbol{\Delta} = \mathbf{A}\boldsymbol{\beta},$$

where $\mathbf{A}$ is a $p \times p$ matrix:

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots \\ -1 & 1 & 0 & 0 & \cdots \\ 0 & -1 & 1 & 0 & \cdots \\ 0 & 0 & -1 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

Analogous to the matrix $\mathbf{D}$, as explained on page 13, the matrix $\mathbf{A}$ can also incorporate various structures. If the desire is to incorporate the chromosome structure, the matrix $\mathbf{A}$ can be adapted to $\mathbf{A}_{chr}$, a matrix that consists of diagonal matrices $\mathbf{K}_i$ for $i = 1, \ldots m$, where $m$ is the number chromosomes so that

$$\mathbf{A}_{chr} = \begin{pmatrix} \mathbf{K}_1 & 0 & 0 & 0 & \cdots \\ 0 & \mathbf{K}_2 & 0 & 0 & \cdots \\ 0 & 0 & \mathbf{K}_3 & 0 & \cdots \\ 0 & 0 & 0 & \mathbf{K}_4 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \text{ with } \mathbf{K}_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots \\ -1 & 1 & 0 & 0 & \cdots \\ 0 & -1 & 1 & 0 & \cdots \\ 0 & 0 & -1 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

When the ordinary matrix $\mathbf{A}$ is used, the coefficients $\boldsymbol{\Delta}$ now correspond to the first entry of the dataset and then to the differences between adjacent entries. In accordance with that, the $\boldsymbol{\lambda}_f$ is submitted to the ridge algorithm as a vector of length $p$ with a zero at the first entry, so that it penalizes the adjacent entry differences. When another structure is incorporated in the matrix $\mathbf{A}$, the vector $\boldsymbol{\lambda}_f$ is adapted accordingly. For example, when $\mathbf{A}_{chr}$ is used, the vector $\boldsymbol{\lambda}_f$ has a zero at each entry corresponding to the start of a chromosome. The smoothed regression has thus one or more unpenalized entries.

Ordinary regression is of the form

$$\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta},$$

where $\boldsymbol{\eta}$ is the vector of predicted values, $\mathbf{X}$ is a $n \times p$ predictor matrix, and $\boldsymbol{\beta}$ is the vector of regression coefficients.

Then since $\boldsymbol{\beta} = \mathbf{A}^{-1}\boldsymbol{\Delta}$,

$$\boldsymbol{\eta} = \mathbf{X}\mathbf{A}^{-1}\boldsymbol{\Delta}.$$

So instead of submitting $\mathbf{X}$ and retrieving $\boldsymbol{\beta}$, $\tilde{\mathbf{X}} = \mathbf{X}\mathbf{A}^{-1}$ is submitted and $\boldsymbol{\Delta}$ is retrieved from the model. However, in case an unpenalized intercept is added to the model by the ridge algorithm,

$$(\mathbf{A}^{-1})^* = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots \\ 0 & 1 & 0 & 0 & \cdots \\ 0 & 1 & 1 & 0 & \cdots \\ 0 & 1 & 1 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

is used to retrieve the coefficients $\boldsymbol{\beta}$ from the coefficients $\boldsymbol{\Delta}$; where an extra first row and column are added to $\mathbf{A}^{-1}$ that are empty except for the [1, 1] entry which represents the intercept. When no unpenalized intercept is added to the model, the matrix $\mathbf{A}^{-1}$ can be used to write the coefficients $\boldsymbol{\Delta}$ back to the coefficients $\boldsymbol{\beta}$.

By using $\tilde{\mathbf{X}}$ in the regression instead of $\mathbf{X}$, the basis of the regression is adapted. This adaption permits the use of an ordinary ridge algorithm for a smoothed regression, as a ridge regression on the $\tilde{\mathbf{X}}$ matrix is equivalent to a smoothed regression.

## Intermezzo: Fused Ridge

Now that we have come across the quadratic penalty in ridge logistic regression and the quadratic difference penalty in smoothed logistic regression, we will mention a logistic regression that combines the two penalties: the $L_2$ fused ridge. $L_2$ fused ridge regression has as an advantage over ordinary ridge regression in that is takes the neighboring gene effect into account, and the advantage over only the $L_2$ fused lasso that is is a model rich in coefficients which might suit specific data. The $L_2$ fused ridge likelihood is defined as

$$\ell_{ridge_f}(\boldsymbol{\beta}) = \ell(\boldsymbol{\beta}) - \frac{1}{2}\lambda \left\|\boldsymbol{\beta}\right\|_2^2 - \frac{1}{2}\lambda_f \left\|\mathbf{D}\boldsymbol{\beta}\right\|_2^2.$$

Analogous to ridge regression (Eq. 2.5–2.6), the Newton–Raphson iterations in an IRLS algorithm become:

$$\begin{aligned}\boldsymbol{\beta}_{new} &= \boldsymbol{\beta}_{old} + (\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \lambda\mathbf{I} + \lambda_f\mathbf{D})^{-1}(\mathbf{X}^{\mathrm{T}}(\mathbf{y}-\mathbf{p}) - \lambda_f\mathbf{D}\boldsymbol{\beta}_{old}) \\ &= (\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \lambda\mathbf{I} + \lambda_f\mathbf{D})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{W}\{\mathbf{X}\boldsymbol{\beta}_{old} + \mathbf{W}^{-1}(\mathbf{y}-\mathbf{p})\}.\end{aligned}$$

The $L_2$ fused ridge is beyond the scope of this thesis, and we therefore do not explore it any more in depth. Nevertheless, it is an interesting possibility; a logistic regression subject to both a quadratic and a quadratic difference penalty.

## 2.4. Smoothed Logistic Regression Summary

Several topics were addressed in this chapter regarding the model design of smoothed logistic regression and the construction of an algorithm to fit smoothed logistic regression. First of all, smoothed logistic regression and ridge regression are closely related as it is the same regression on a different basis. Smoothed logistic regression can be fitted on an ordinary basis by maximum likelihood estimation through Newton–Raphson iterations, similar to logistic and ridge logistic regression but with the matrix $\mathbf{D}$ to incorporate structure in the penalty.

Then we adapted the basis of the regression, so that the smoothed logistic regression became a ridge logistic regression problem. Hence, an ordinary ridge logistic regression algorithm could be used for smoothed logistic regression, by rewriting the input and output. This also permits Newton–Raphson iterations with the inversion of the smaller $n \times n$ matrix, for example through the SVD trick.

The analysis of copy number data with a smoothed logistic regression will be presented in chapter 4.

# 3. Model Design: $L_2$ Fused Lasso

This chapter explores the model design of the lasso logistic regression with a quadratic difference penalty, $L_2$ *fused lasso*. The main focus while going through this model design is the construction of an algorithm to fit an $L_2$ fused lasso.

In this chapter, we attempt to fit an $L_2$ fused lasso. Therefore, we show the $L_2$ fused lasso log likelihood to be optimized in maximum likelihood estimation. We describe the optimization challenges that arise when fitting an $L_2$ fused lasso model. Along the way, properties and theory underlying the lasso and the $L_2$ fused lasso become clear.

First, it is explaining why a Newton–Raphson IRLS will not succeed in optimizing the $L_2$ fused lasso likelihood. Then a basic gradient ascent algorithm is proposed. It is explained why this also fails. Some adaptions to the gradient ascent are proposed, that solve this convergence problem. We chose to implement the solution proposed by Goeman (2010) in `R` package penalized. Therefore, the lasso algorithm in package penalized is extended to fit an $L_2$ fused lasso. This chapter is concluded with a short summary.

## 3.1. Fitting an $L_2$ Fused Lasso

An extension of the commonly applied lasso (Tibshirani, 1996) is the fused lasso (Tibshirani et al., 2005), where an extra penalty is added that constrains the difference between the regression coefficients of adjacent covariates. In the introduction of this thesis, we outlined the choice for a lasso logistic regression with an extra quadratic difference penalty: the $L_2$ fused lasso or structured elastic net (Li & Li, 2008; Slawski et al., 2010). In this chapter, an algorithm is developed for an $L_2$ fused lasso logistic regression.

### 3.1.1. Form of the $L_2$ Fused Lasso Log Likelihood

The $L_2$ fused lasso log likelihood is defined as

$$\ell_{lasso_f}(\boldsymbol{\beta}) = \ell(\boldsymbol{\beta}) - \lambda\sum_{i=1}^{p}|\beta_i| - \frac{1}{2}\lambda_f\sum_{i=1}^{p-1}(\beta_{i+1} - \beta_i)^2, \tag{3.1}$$

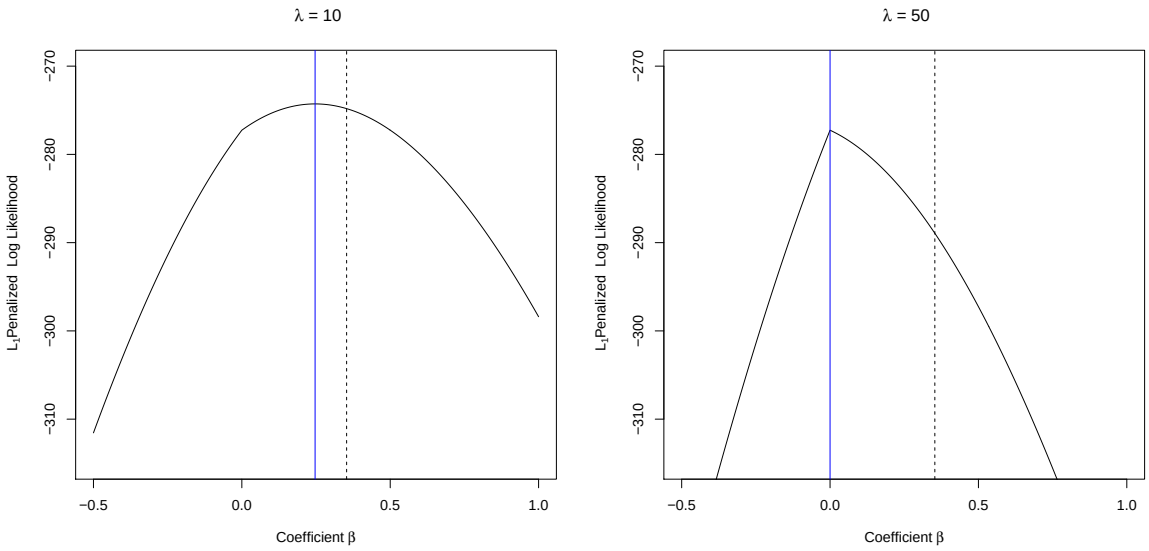



**Figure 3.1.:** *Lasso likelihood for one standardized coefficient with a $\lambda$ of 10 (left) and 50 (right). Dotted line indicates the value of the unpenalized coefficient estimate. Solid line indicates the lasso coefficient estimate. The influence of the smaller lasso penalty (left) on the coefficient estimate is limited, while the large lasso penalty (right) pushes the coefficient estimate to zero.*

where $\ell(\boldsymbol{\beta})$ is the unpenalized likelihood, $\lambda$ determines the size of the lasso penalty, and $\lambda_f$ determines the size of the quadratic difference penalty. With coefficients $\boldsymbol{\beta} = (\beta_1, ...\beta_p)^{\mathrm{T}}$, the $L_2$–fused lasso log likelihood can be reformulated as

$$\ell_{lasso_f}(\boldsymbol{\beta}) = \ell(\boldsymbol{\beta}) - \lambda \left\|\boldsymbol{\beta}\right\|_1 - \frac{1}{2}\lambda_f \left\|\mathbf{D}\boldsymbol{\beta}\right\|_2^2, \tag{3.2}$$

where $||.||_1$ is the $L_1$ norm, and $||.||_2$ is the $L_2$ norm. Recall from the paragraph on the matrix $\mathbf{D}$ on page 13 in chapter 2 that the simplest form of $\mathbf{D}$ is a $p \times p$ matrix so that Eq. 3.1 and Eq. 3.2 exactly correspond:

$$\mathbf{D} = \begin{pmatrix} -1 & 1 & 0 & 0 & \cdots \\ 0 & -1 & 1 & 0 & \cdots \\ 0 & 0 & -1 & 1 & \cdots \\ 0 & 0 & 0 & -1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

Recall also, that more structure can be implemented in the matrix $\mathbf{D}$.

The fused penalized regression coefficients $\hat{\boldsymbol{\beta}}$ are estimated by optimizing the fused penalized likelihood

$$\hat{\boldsymbol{\beta}}_{lasso_f} = \text{argmax } \ell_{lasso_f}(\boldsymbol{\beta}). \tag{3.3}$$

### 3.1.2. Why the Newton–Raphson Method Fails

Some challenges arise in solving Eq. 3.3. The shape of the lasso likelihood function is the sum of the parabola from the ordinary likelihood and the quadratic difference penalty, and the upturned V–shape from the absolute value function from the lasso penalty. How the lasso likelihood function changes by the effect of the penalty can be seen in Fig. 3.1.

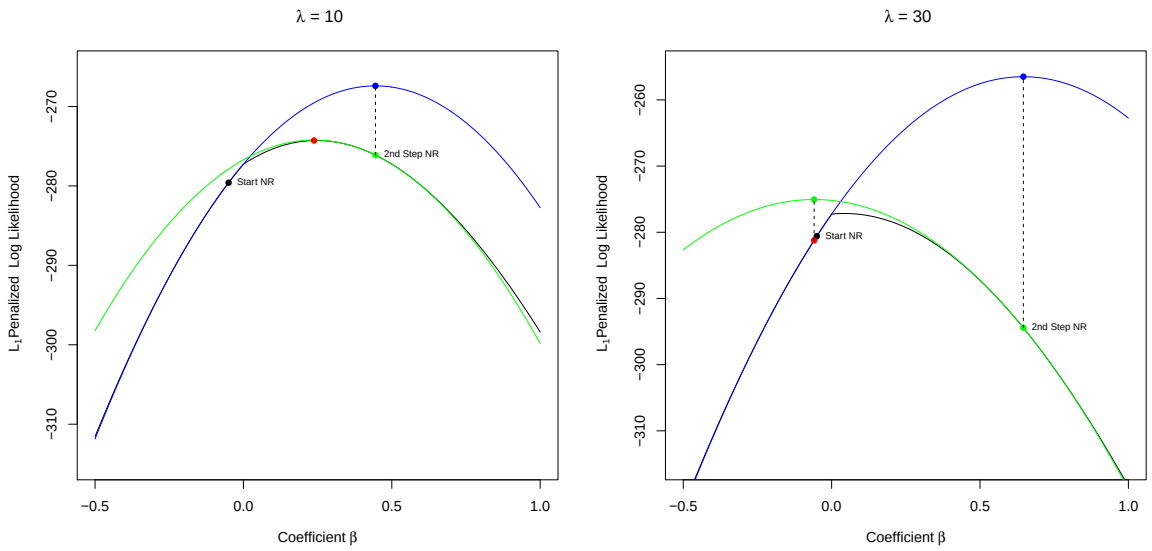



**Figure 3.2.:** *Newton–Raphson iterations on the lasso likelihood for one standardized coefficient at a lasso λ of 10 (left) and 30 (right). At the smaller lasso penalty (left), the algorithm converges. At the bigger lasso penalty (right), the algorithm fails to converge, even though the optimum is just outside zero.*

Although Newton–Raphson is a preferred optimization method in regression analysis, this method can not be used to solve Eq. 3.3. First of all, the function is not twice differentiable when the coefficient equals zero. Also, the Taylor series do not approximate the penalized likelihood function correctly around the sharp bend. This is because the second–order Taylor expansion approximates the function around a chosen point as if it is a quadratic function. As can be seen in Fig. 3.1, the likelihood function consists of different shapes: below and above zero. When the penalty is relatively small (left of Fig. 3.1), the kink at zero of the likelihood function will not be sharp. A quadratic approximation of the function on the side of the bend that has not the actual optimum will be near the actual optimum, and (one of) the next Newton–Raphson steps that start at the side of the optimum can correctly approximate the top. This is demonstrated at the left of Fig. 3.2. The problem is that when the penalty gets bigger, and with that the kink in the likelihood sharper (right of Fig. 3.2), approximating the top by quadratic approximations at either side of the bend will not work. Therefore, a Newton–Raphson optimization algorithm will fail

to converge when any of the estimates have an optimal penalized likelihood at zero.

### 3.1.3. Why a Basic Gradient Ascent Fails

A very robust optimization procedure is gradient ascent. A gradient ascent algorithm for the optimization of one coefficient just calculates the derivative at that point and takes a step in that direction. When the derivative is zero, it is the top. In this section, first a basic gradient ascent algorithm will be presented. After that, we will explain why it fails in a lasso maximum likelihood estimation.

#### Basic Gradient Ascent Algorithm

Even though the gradient ascent is robust in its simplicity, when a basic gradient ascent algorithm is applied on the lasso problem, it is not successful. This is not only because the lasso log likelihood is not everywhere differentiable since the lasso penalty function is not differentiable when $\beta_i = 0$ for all $i$. To solve that, Goeman (2010) defines a directional derivative

$$\ell'_{lasso_f}(\boldsymbol{\beta}; \mathbf{v}) = \lim_{t \to 0} \frac{1}{t} \{\ell_{lasso_f}(\boldsymbol{\beta} + t\mathbf{v}) - \ell_{lasso_f}(\boldsymbol{\beta})\},$$

for every point $\boldsymbol{\beta}$ in every direction $\mathbf{v} \in \mathbb{R}^p$ with $||\mathbf{v}|| = 1$. The gradient can then be defined as the direction of steepest ascent. The algorithm follows the gradient in the direction $\mathbf{v}_{opt}$ which optimizes $\ell'_{lasso_f}(\boldsymbol{\beta}; \mathbf{v})$. The gradient is defined as

$$\mathbf{g}_i(\boldsymbol{\beta}) = \begin{cases} \ell'(\boldsymbol{\beta}; \mathbf{v}) \cdot \mathbf{v}_{opt} & \text{if } \ell'_{lasso_f}(\boldsymbol{\beta}; \mathbf{v}) \geq 0 \\ \mathbf{0} & \text{otherwise} \end{cases}.$$

Take the vector of unpenalized directional derivatives $\mathbf{h}$ as

$$\mathbf{h}(\boldsymbol{\beta}) = \frac{\partial \ell(\boldsymbol{\beta})}{\partial \beta} = \mathbf{X}^{\mathrm{T}}(\mathbf{y} - \mathbf{p}),$$

where $\mathbf{X}$ is a $n \times p$ predictor matrix, $\mathbf{y}$ is a binary output vector, and $\mathbf{p}$ is a vector of probability estimates constructed as

$$\mathbf{p} = \frac{\exp(\mathbf{X}^{\mathrm{T}}\boldsymbol{\beta})}{\{1 + \exp(\mathbf{X}^{\mathrm{T}}\boldsymbol{\beta})\}}.$$

The gradient vector $\mathbf{g}$ is calculated:

$$\mathbf{g}_i = \begin{cases} \mathbf{h}_i - \lambda \operatorname{sign}(\beta_i) - \lambda_f \mathbf{D}^{\mathrm{T}}\mathbf{D}\boldsymbol{\beta} & \text{if } \beta_i \neq 0 \\ \mathbf{h}_i - \lambda \operatorname{sign}(\mathbf{h}_i) - \lambda_f \mathbf{D}^{\mathrm{T}}\mathbf{D}\boldsymbol{\beta} & \text{if } \beta_i = 0 \text{ and } \left|\mathbf{h}_i - \lambda_f \mathbf{D}^{\mathrm{T}}\mathbf{D}\boldsymbol{\beta}\right| > \lambda \\ 0 & \text{otherwise,} \end{cases} \tag{3.4}$$

where

$$\operatorname{sign}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0. \end{cases}$$

Then we construct a gradient ascent algorithm is constructed that updates the coefficients $\boldsymbol{\beta}$ with step size $t$ until convergence:

$$\boldsymbol{\beta}_{new} = \boldsymbol{\beta}_{old} + t\,\mathbf{g}.$$

## Unadapted Gradient Ascent Fails

Above, a basic gradient ascent algorithm to fit an $L_2$ fused lasso is constructed. Now when it is applied on an $L_2$ fused lasso problem, it fails. That is unexpected. With taking steps in the direction of the top, should one not eventually get there? Below we will explain why not.

Lets first discuss the factor of dimensionality. In regression analysis on high dimensional data, there are a lot of predictors. All these predictors get a regression coefficient. Therefore, the space where we are optimizing in has a lot of dimensions. Fig. 3.3 presents the shape of the log likelihood in the directions of a coefficient $\beta_1$ and coefficient $\beta_2$. Consider a function with all kinds of ridges and edges in a $p$ dimensional space.

That feels as a difficult optimization problem, but still the $L_2$ fused lasso log likelihood has only one top. Now, one of the steps sets the coefficients $\beta_1$ and $\beta_2$ almost at the optimum, at the blue arrows. In both the direction of $\beta_1$ and the direction of $\beta_2$, the distance to the optimum is small. The blue lines represent the derivatives at the point of the blue arrow. The direction of the next step is clear. So let's take a step. In order not to overshoot the top at the right, the step size can only be very small. Recall:

$$\boldsymbol{\beta}_{new} = \boldsymbol{\beta}_{old} + t\,\mathbf{g},$$

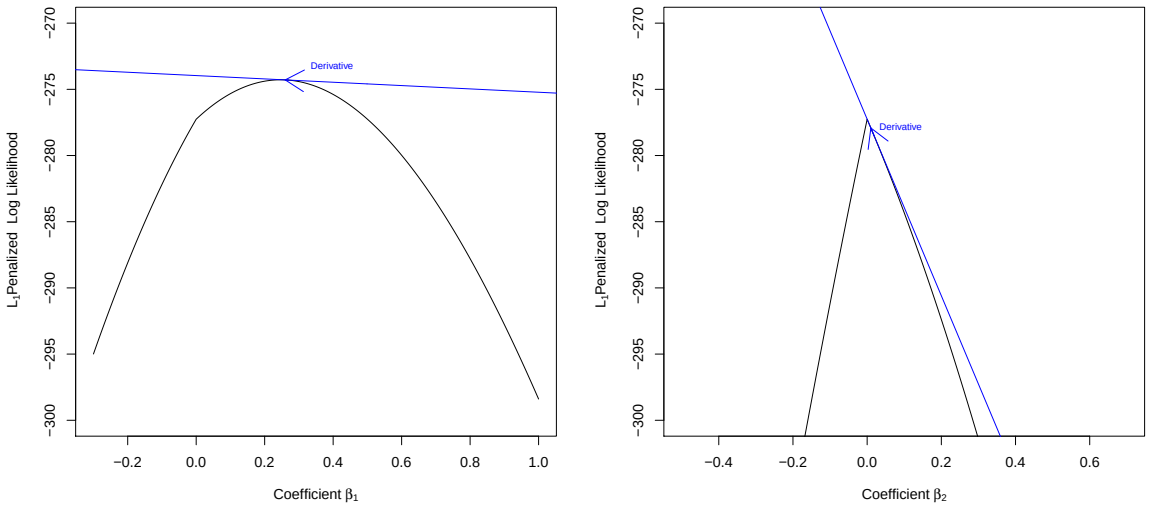



**Figure 3.3.:** *Example of the lasso log likelihood in the direction of a coefficient $\beta_1$ (left), and coefficient $\beta_2$ (right). The gradient near the top for $\beta_1$ is small, while the gradient for $\beta_2$ is large.*

where $t$ is the stepsize. The stepsize $t$ is the same for all dimensions. But the step that is actually taken is $t$ times the gradient. The gradient is very small for the direction of $\beta_1$, while big for the direction of $\beta_2$ in Fig. 3.3. The $t$ is thus going to be extremely little, or the step will overshoot the optimum for $\beta_2$. This $t$ is little, and then the gradient is also very small at coefficient $\beta_1$. Nothing is left for a step. This will go on for eternity with very small steps, the optimum is never reached, and the algorithm will not converge. The chances of the coefficient in the dimension in the right of Fig. 3.3 to hit exactly zero are too small.

Of course, there is no guarantee that the gradient ascent will fail. When the coefficient at the right hits zero, the stepsize increases and the model will converge. The changes are just not that high.

As a result, we propose some adaptions to a basic gradient ascent:

- Using the sign of the gradient instead of the gradient itself. Promising results. The algorithm got a lot more robust, and the step size was a lot better controlled. The difference between the actual size of the steps in the various dimensions is eliminated since the size of the gradient is no longer a factor. Nevertheless, this approach was not implemented in the final algorithm, since too much information about the direction of the optimum was thrown away.
- Setting the coefficient artificially to zero every time a step overshoots zero. This is not a very elegant solution, and the step is not in the optimal direction. It does solve the problem, as hitting exactly zero is the challenge.
- Taking the step size never bigger than the coefficient closest to zero. This is

what is done in R package penalized (Goeman, 2010). This solves the problem, but it doesn't compromise that much on the efficiency of the algorithm since it is still a step in the direction of the steepest ascent. Therefore, we chose to implement this adaption.

Since the gradient ascent algorithm in R package penalized was the most efficient adaption that we investigated, we chose to extend the package penalized to fit $L_2$ fused lasso problems.

## 3.2. Extending the R Package Penalized

The R package penalized not only uses the adapted version of gradient ascent. It uses both gradient ascent and Newton–Raphson, thereby using the robustness of gradient ascent and the fast and precise Newton–Raphson near the optimum. Below, the Goeman (2010) algorithm from package penalized with the extension to fit the $L_2$ fused lasso is explained.

The gradient is used as described in the previous section. The penalized algorithm updates the coefficients $\boldsymbol{\beta}$ with step size $t$ until convergence:

$$\boldsymbol{\beta}_{new} = \boldsymbol{\beta}_{old} + t\,\mathbf{g}, \text{ where } t \leq |\beta_i| \;\forall\, i,$$

so that a jump is never made further than a zero, to solve the convergence problem indicated before.

The gradient ascent algorithm from R package penalized also includes Newton–Raphson steps, to include the fast optimization properties of the Newton–Raphson when near the optimum. Let $t_{opt}$ be the optimum and $t_{edge}$ the borders of the sub domain, where a sub domain is a space that does not include any zero. Then

$$\beta^{i+1} = \begin{cases} \beta_{NR}^{i+1} & \text{if } \operatorname{sign}(\beta_{NR}) = \operatorname{sign}(\beta_{+}^{i}) \\ \beta_{G}^{i+1} & \text{otherwise} \end{cases}$$

where $\beta_+$ indicates the set of active variables, $\beta_G$ is a gradient ascent step, and $\beta_{NR}$ is a Newton–Raphson step for $L_2$ fused lasso regression:

$$\boldsymbol{\beta}_{new} = \boldsymbol{\beta}_{old} - \left(\frac{\partial^2 \ell_{lasso_f}(\boldsymbol{\beta})}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^{\mathrm{T}}}\right)^{-1} \frac{\partial \ell_{lasso_f}(\boldsymbol{\beta})}{\partial\boldsymbol{\beta}}, \text{ with}$$

$$\frac{\partial \ell_{lasso_f}(\boldsymbol{\beta})}{\partial\beta} = \mathbf{X}^{\mathrm{T}}(\mathbf{y}-\mathbf{p}) - \lambda \operatorname{sign}(\boldsymbol{\beta}) - \lambda_f \mathbf{D}\boldsymbol{\beta}, \text{ and}$$

$$\frac{\partial^2 \ell_{lasso_f}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^{\mathrm{T}}} = -\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} - \lambda_f \mathbf{D} \text{ if } \beta_i \neq 0 \; \forall \, i.$$

### Why the Newton–Raphson Iterations Remain Slower

Our $L_2$ fused algorithm is slower than the ordinary lasso algorithm. One step that is very expensive, is the large $p \times p$ matrix that is inverted in a Newton–Raphson step. There is a lot to gain by inverting the smaller $n \times n$ matrix, for example through singular value decomposition. This can be done for ordinary lasso regression, but is not possible for the $L_2$ fused lasso. In a Newton–Raphson step for the $L_2$ fused lasso, an off–diagonal penalty is added on neighboring genes. In the smaller $n \times n$ matrix however, the off–diagonal entries do not correspond to neighboring genes, and therefore a bigger matrix needs to be inverted in a Newton–Raphson step. It is the same reason that holds for the smoothed logistic regression, explained in more detail in chapter 3 on page 15.

## 3.3. $L_2$ Fused Lasso Summary

The $L_2$ fused lasso is not a simple model to fit, as a lasso is not a simple model to fit. Newton–Raphson is not successful in optimizing the penalized log likelihood, because the shape of the $L_2$ fused lasso log likelihood can not be approximated by quadratic functions.

Surprisingly, a basic gradient ascent algorithm also fails in fitting an $L_2$ fused lasso. This is because of the shape of the lasso log likelihood around zero and the dimensionality of the problem. Some adaptions to gradient ascent were investigated, and the adaption of Goeman (2010) was chosen to implement. Therefore, the algorithm in `R` package penalized which uses a combination of an adapted gradient ascent and Newton–Raphson, was extended to fit the $L_2$ fused lasso.

The analysis of copy number data with an $L_2$ fused lasso will be presented in chapter 4.

# 4. Data Analysis

In the previous chapters, we have explored the model design of smoothed logistic regression and the $L_2$ fused lasso. We have constructed algorithms that can fit these models. Here, we are interested in how these models perform as classifiers on copy number data.

In this chapter, the smoothed logistic regression and the $L_2$ fused lasso algorithms as constructed and discussed in chapter 2 and chapter 3 are fitted on genetic copy number data. Both models are fitted to classify the tumor grade in bladder cancer. To compare classification accuracy, the $L_1$ fused lasso, main focus, and the ordinary lasso and ridge are also fitted on the bladder cancer dataset.

## 4.1. Bladder Cancer Data

The bladder cancer data on which the penalized logistic regression models (i.e. ridge, lasso, $L_2$ fused lasso, $L_1$ fused lasso, and smoothed logistic regression) are fitted is a publicly available dataset with array CGH profiles of 57 bladder tumor samples from 53 affected patients (Stransky et al., 2006). Each profile gives the relative quantity of DNA for 2385 probes. Following Chatuverdi et al. (2013), we removed the probes corresponding to sex chromosomes, giving us a list of 2308 probes. We considered tumor classification with grade as the outcome (33 tumors of grade 3, 13 tumors of grade 1 and 11 tumors of grade 2). The probes with more than 20 per cent missing values were also removed and the remaining missing values were imputed using K nearest neighbors leaving us with 2161 probes and 57 samples. To remove any systematic trends from the microarray technology, the data is median normalized by subtracting the median of all genes on the array and then adding the median across all arrays.

Recall from the introduction, that the array CGH copy number profile consists of $\log_2$ ratios. The accompanying plot at that section (Fig. 1.1 on page 5), is copy number data from a grade 3 tumor in this dataset.

The tumors in this dataset were graded according to the 1973 WHO classification criteria (Mostofi et al., 1973). In this analysis, we classified between a grade 3 group and a lower grade group, that is the grades 1 and 2. To get a broad overview of the data, a graphical representation of the $\log_2$ of the mean signal ratio per chromosome for the two grade groups is visible in Fig. 4.1. Recall from the introduction of this

thesis, how the signal ratio reflects the number of copies of genes present in the DNA sample, and how copy number data usually contains the $\log_2$ signal ratios.

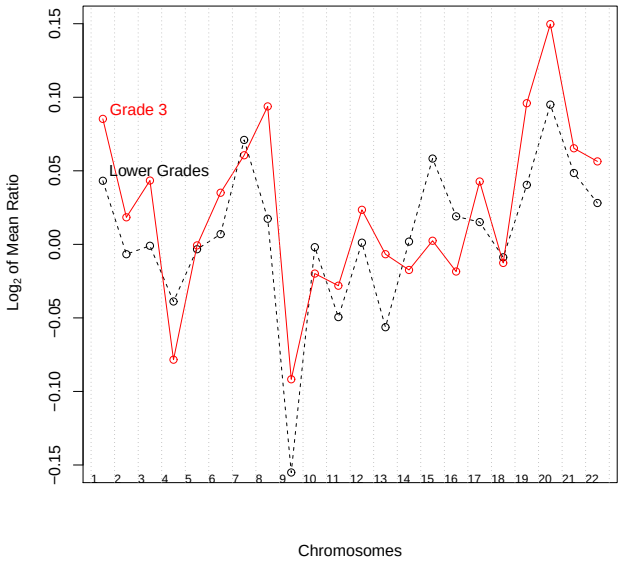



**Figure 4.1.:** *The* $\log_2$ *of the mean signal ratio per chromosome per grade. The solid line represents grade 3, the dotted line represents the lower grades. The disparities between the two lines indicate mean differences between the grade groups.*

## 4.2. The Optimal Models

In the previous chapters of this thesis, we have specified theoretical optimizations for the $L_2$ fused lasso and smoothed logistic regression. In fitting these models, and the ridge, lasso and $L_1$ fused lasso, some practical choices have to be made. First, all models are fitted by maximum likelihood optimization, which means that we are looking for an optimum in the penalized likelihood. How do we know that we are at such an optimum, and that our model has in fact converged? We answer that question in sec. 4.2.1. Then, we have to choose the size of the penalties. How we choose penalties so that our models describe the data best is explained in sec. 4.2.2.

### 4.2.1. Error of Convergence

As discussed in previous chapters, the coefficients from the regression models are found by maximum likelihood optimization. But how do we know that we are at an optimum? To satisfy convergence criteria, the algorithm has to either satisfy *all* these three conditions:

- The Likelihood Condition. $\frac{2\cdot|\ell(\boldsymbol{\beta})-\ell(\boldsymbol{\beta}_{old})|}{2\cdot|\ell(\boldsymbol{\beta})-penalty|+.1} < \epsilon$, where $\ell(\boldsymbol{\beta})$ is the log likelihood of the coefficients in that iteration; $\ell(\boldsymbol{\beta}_{old})$ is the log likelihood of the coefficients in the previous iteration; *penalty* is the $\lambda$ value as discusses before times the respective coefficient sums depending on the type of penalization, and in a double penalization the sum of both $\lambda$ values times the respective coefficient sums (see Tab. 2.1); and $\epsilon$ is a prespecified convergence error.
- The Penalty Condition. $\frac{2\cdot|penalty-penalty_{old}|}{2\cdot|\ell(\boldsymbol{\beta})-penalty|+.1} < \epsilon$, where $penalty_{old}$ is the *penalty* of the previous iteration.
- The Active Set Condition. As the active set we specify the coefficients that are not zero. To meet convergence criteria, the active set in this iteration has to equal the active set in the previous iteration.

Or the algorithm has to satisfy this one condition:

- The Direction Condition. All values of the gradient vector are equal to zero, since this indicates it is the optimum.

As a result, a convergence error $\epsilon$ must be specified. We chose to accept a convergence error of $10^{-7}$ for all our models.

### 4.2.2. Optimal Penalties

To fit the models, optimal penalties need to be chosen. And as the size of the penalty is determined by the value of $\lambda$, optimal $\lambda$'s need to be specified. All optimal $\lambda$'s are found by maximizing the cross–validated likelihood at a range of values for the penalty or penalties. To compute the cross–validated likelihood, the predictions of all observations are computed by leaving out an observation and computing the class probability of that observation by a model on the rest of the observations (leave–one–out cross–validation). The log likelihoods for the individual predictions given their true value are computed, and then summed to retrieve the cross–validated likelihood. The cross–validated likelihood is therefore a measure of predictive ability of the model.

For ridge and lasso regression, this method is applied via a function in `R` package penalized. For the smoothed regression however, there is a small chance that a likelihood can not be calculated, and therefore the cross–validated likelihood can not be estimated. This is solved by taking the approximated cross–validated likelihood (Meijer & Goeman, 2013). The optimal penalties for the $L_1$ fused lasso are taken from Chatuverdi et al. (2013), from a cross–validation on models that stop after 100 iterations. The optimal penalties for the $L_2$ fused lasso are estimated via an own cross–validated likelihood algorithm, and estimated by a model with a convergence error of $10^{-3}$ due to time constrains in the cross–validation process.

## 4.3. Regression Coefficient Plots

The ridge, lasso, $L_2$ fused lasso, $L_1$ fused lasso, and smoothed logistic regression are fitted on the bladder cancer copy number data with the optimal $\lambda$'s as found by cross–validation as described above. An unpenalized intercept is included in all models. For the models subject to a difference penalty, the $L_2$ fused lasso, $L_1$ fused lasso, and smoothed logistic regression, the difference penalty is applied so that it penalizes the difference between neighboring genes *per chromosome*. All models are fitted to classify the data as the class grade 3 or as the class that contains the lower grades.

As expected, the difference in regression coefficients obtained from the different models is quite spectacular. As can be seen in Fig. 4.2a–e on the next page, the characteristics of the method are visible in the results, so all coefficients small in ridge, sparsity in coefficients for the lasso, regions as (in this case, small) curves in the $L_2$ fused lasso, regions as same value segments for the $L_1$ fused lasso and smooth curves for the smoothed regression.

## 4.4. Cross–validated Predictions

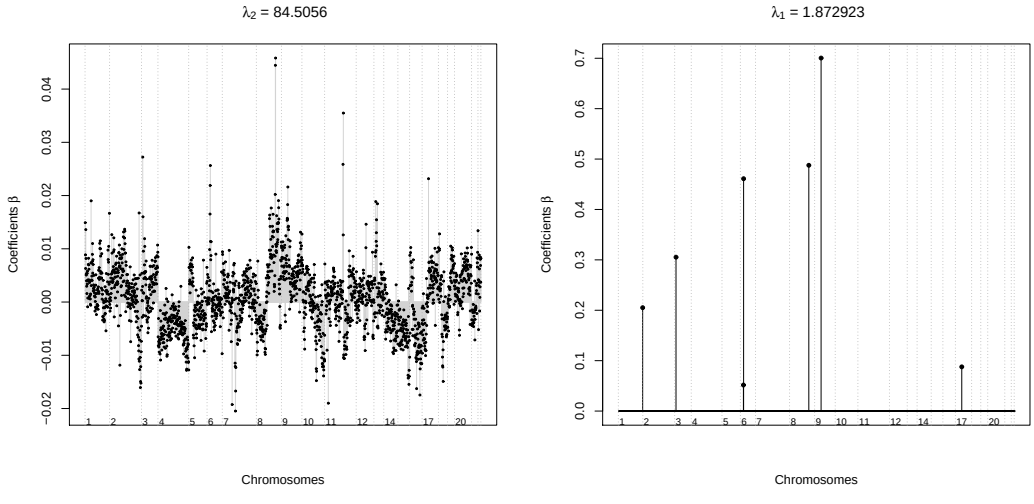
The predicted class probabilities that are used to compute the cross-validated likelihood, that are the predictions obtained by leave–one–out cross–validation, are also used to obtain information about the classification accuracy of the models. The cross–validated predicted class probabilities per tumor sample are presented in Appendix A for all models discussed here.

The cross–validated predicted class probabilities are rounded to their nearest class, and the predicted classes contrasted by the true class are presented in Tab. 4.1a–e on page 34. Between brackets are the not cross–validated classifications on the whole dataset. An overview of the total and cross–validated 0/1 loss per model is presented in Tab. 4.2, along with the cross–validated likelihood (CVL).

Pearsons' Chi–square test statistics are also presented in Tab. 4.2. The relationship between the original grade classification and the classification of the smoothed logistic regression model is significant, $\chi^2_{(1,\,N=50)} = 6.203,\ p < .05$.

Unfortunately, the smoothed logistic regression cross–validated predictions could only be computed for 50 out of the 57 tumor samples because the model would not converge for the other 7 cases. We will explain the reason in the Discussion of this thesis. We treat the obtained classification error equal to the classification errors of the other models.

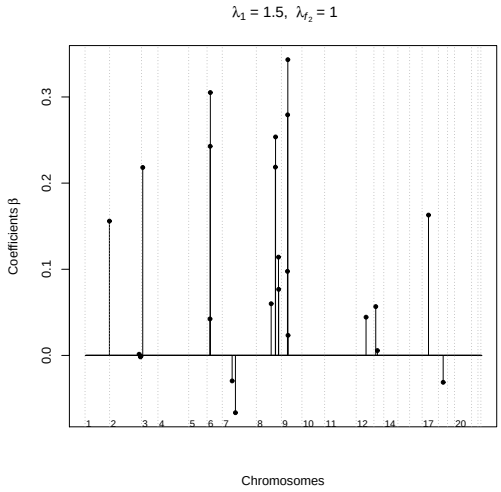
The receiver operating characteristic (ROC curve) for all models discussed here based on the cross–validated predictions can be seen in Fig. 4.2f on the facing page. As can be seen, all models appear similar in terms of classification accuracy.



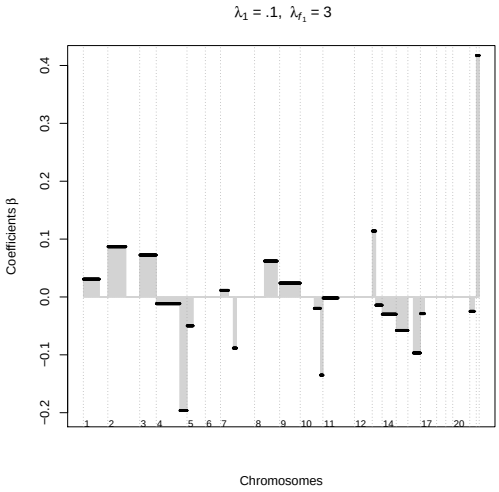

**(a)** *Ridge*



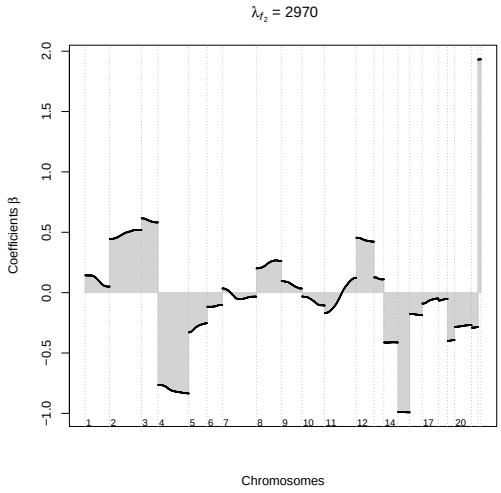

**(b)** *Lasso*



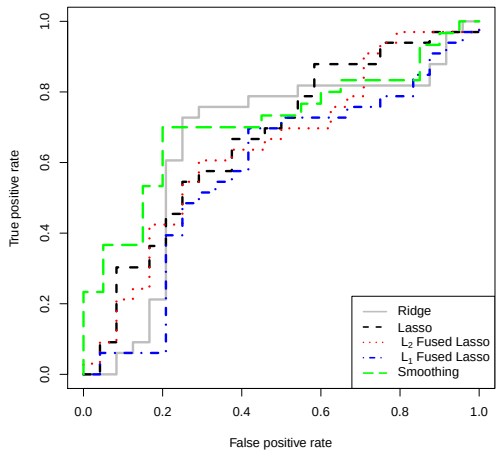

**(c)** $L_2$ *Fused Lasso*




**(d)** $L_1$ *Fused Lasso*




**(e)** *Smoothing*




**(f)** *ROC curves*

**Figure 4.2.:** *Plots of the regression coefficients (a–e). ROC curves to compare prediction accuracy (f).*

**(a)** *Ridge*

| Predicted Class | True Class: Lower Grades | True Class: Grade 3 |
|---|---|---|
| Lower Grades | 10 (17) | 6 (1) |
| Grade 3 | 14 (7) | 27 (32) |

**(b)** *Lasso*

| Predicted Class | True Class: Lower Grades | True Class: Grade 3 |
|---|---|---|
| Lower Grades | 7 (10) | 4 (1) |
| Grade 3 | 17 (14) | 29 (32) |

**(c)** $L_2$ *Fused Lasso*

| Predicted Class | True Class: Lower Grades | True Class: Grade 3 |
|---|---|---|
| Lower Grades | 7 (12) | 7 (0) |
| Grade 3 | 17 (12) | 26 (33) |

**(d)** $L_1$ *Fused Lasso*

| Predicted Class | True Class: Lower Grades | True Class: Grade 3 |
|---|---|---|
| Lower Grades | 16 (21) | 15 (2) |
| Grade 3 | 8 (3) | 18 (31) |

**(e)** *Smoothing*

| Predicted Class | True Class: Lower Grades | True Class: Grade 3 |
|---|---|---|
| Lower Grades | 14 (23) | 9 (2) |
| Grade 3 | 6 (1) | 21 (31) |

**Table 4.1.:** *True class contrasted by predicted class at optimal λ's. Predicted probabilities are rounded to nearest class. Between brackets are the not cross–validated predictions on the whole dataset. Only 50 values for the smoothed logistic regression method.*

| Technique | 0/1 Loss | | | $\chi^2$ test statistics | | |
|---|---|---|---|---|---|---|
| | Total | Cross–validated | CVL | $\chi^2$ | df | $p$ |
| Ridge | .14 | .35 | -38.30436 | 2.7215 | 1 | 0.10 |
| Lasso | .26 | .37 | -38.41869 | *n/a* | | |
| $L_2$ Fused Lasso | .21 | .42 | -38.18261 | 0.1423 | 1 | 0.71 |
| $L_1$ Fused Lasso | .09 | .40 | *n/a* | 1.7377 | 1 | 0.19 |
| Smoothing | .05 | .30 | *n/a* | 6.203 | 1 | 0.01 |

**Table 4.2.:** *Total and cross–validated 0/1 loss, cross–validated likelihood (CVL), and Pearsons' chi–squared ($\chi^2$) test statistics with Yates' continuity correction; all at optimal λ's. Cross–validated 0/1 loss and $\chi^2$ test statistics of the smoothed logistic regression is based on only 50 predictions. CVL's not available (n/a) for the $L_1$ fused lasso and smoothed logistic regression. Chi–square approximation is not available for the lasso, as it may be incorrect due to small cell values.*

# 5. Discussion

In this thesis, we discussed two models that can make classification based on genetic copy number data, since copy number data is very relevant i.a. in cancer research. We settled the problem that the data is high dimensional, which made traditional statistical techniques invalid methods, by imposing a penalty on logistic regression. Then we used the information that the copy number data is likely to have spatial correlation by using a penalty that minimizes the difference between adjacent coefficients. In comparison with the $L_1$ fused lasso, we chose a quadratic difference penalty. This led us to explore two logistic regressions with a quadratic difference penalty, namely:

- $L_2$ *fused lasso.* A logistic regression with two penalties, the lasso with an extra quadratic difference penalty.

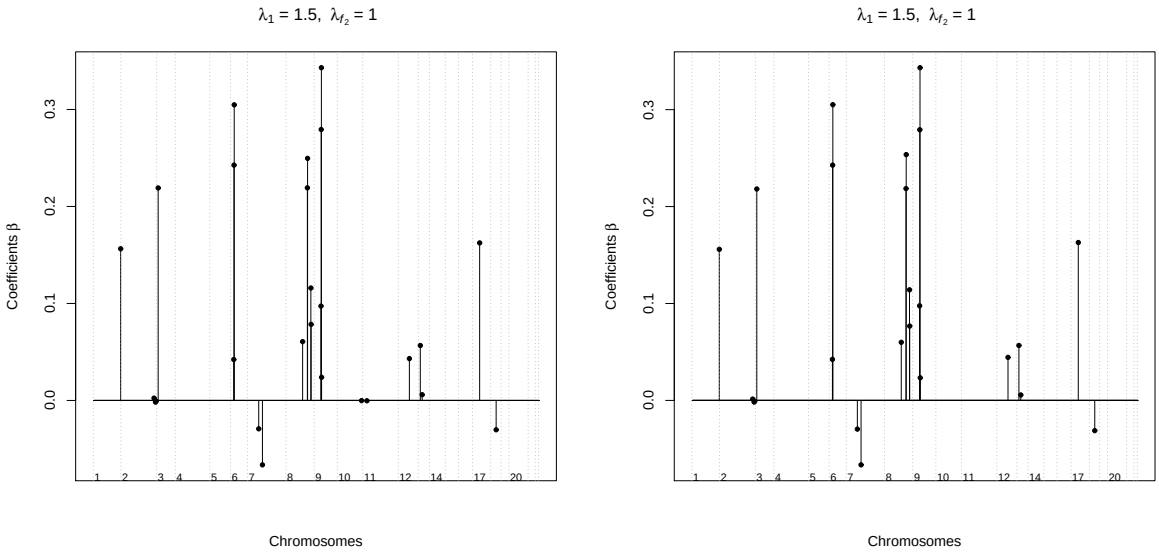
- *Smoothed logistic regression.* A logistic regression with only a quadratic difference penalty.

Both models were fitted to classify the tumor grade in bladder cancer based on genetic copy number data. Their performance was compared to other penalized regression models, that is the $L_1$ fused lasso, ridge, and lasso. Based on leave–one–out cross–validated predictions, all models have a similar prediction accuracy with correct classifications between 58 and 70 per cent; smoothed logistic regression performed best with 70 per cent correctly classified. The relationship between the original grade classification and the classification of the smoothed logistic regression model is significant, $\chi^2_{(1,\, N=50)} = 6.203,\ p < .05$.

In this section, we will first highlight some limitations of this thesis specifically and in general. This thesis is concluded with ideas and recommendations on the lasso and logistic regression with a quadratic difference penalty.

## Limitations

**$L_2$ Fused Lasso: Large convergence error.** The $L_2$ fused lasso algorithm as described in this thesis is slow. To speed up the cross–validation process when we searched for the optimal penalty values, we chose to compute the cross–validated likelihoods of the $L_2$ fused lasso models with a larger error than preferred. We used a convergence error of $10^{-3}$ compared to the convergence error of $10^{-7}$ which was used for the other models. Increasing the convergence error does not improve the classification accuracy of the model. What the algorithm with a larger convergence




**(a)** *Convergence error equals* $10^{-3}$ **(b)** *Convergence error equals* $10^{-7}$

**Figure 5.1.:** *Coefficient plots for the $L_2$ fused lasso on the bladder cancer dataset for a smaller (left) and larger (right) convergence error, at optimal penalty values computed for the model subject to the larger convergence error of* $10^{-3}$.

error does is stopping earlier, further from the actual optimum of the penalized log likelihood. The coefficients obtained from a model with a larger convergence error might therefore not exactly correspond to the coefficients obtained by an algorithm with a smaller convergence error.

The original assumption was that we could search for optimal penalties with an $L_2$ fused lasso algorithm subject to a bigger convergence error, and search around the so found optimum again with an $L_2$ fused lasso algorithm with a smaller convergence error. It was assumed that the optimal penalty values from the model with a larger convergence error would roughly correspond to the optimal penalty values for a model with a smaller convergence error.

**(a)** *Convergence error equals* $10^{-3}$

| Predicted Class | | True Class | |
|---|---|---|---|
| | | Lower Grades | Grade 3 |
| | Lower Grades | 7 | 7 |
| | Grade 3 | 17 | 26 |

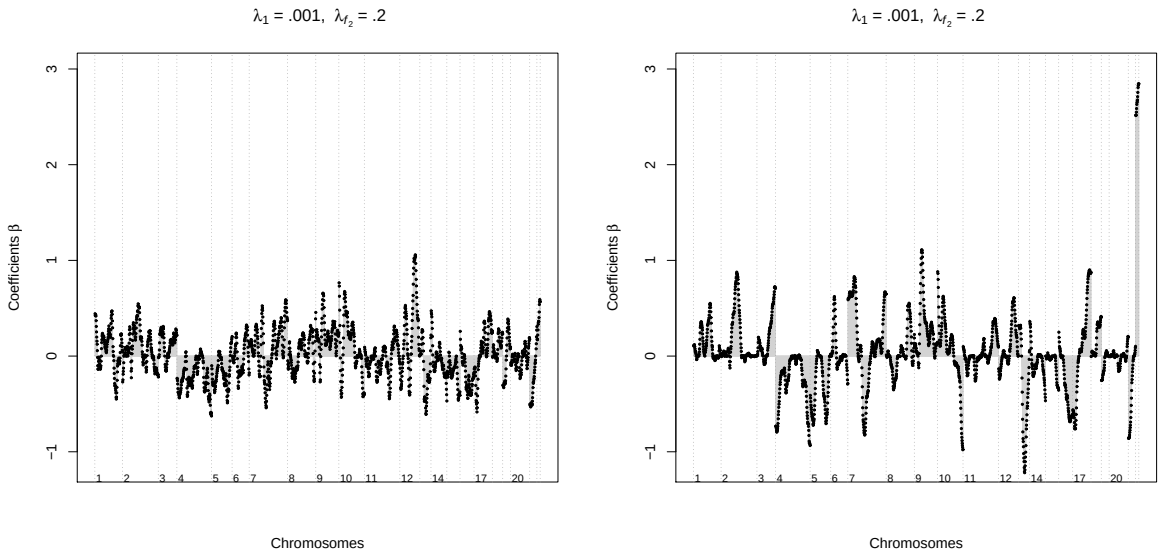
**(b)** *Convergence error equals* $10^{-7}$

| Predicted Class | | True Class | |
|---|---|---|---|
| | | Lower Grades | Grade 3 |
| | Lower Grades | 7 | 7 |
| | Grade 3 | 17 | 26 |

**Table 5.1.:** *$L_2$ fused lasso classification on the bladder cancer dataset compared for smaller (left) and bigger (right) convergence error at optimal penalty values.*

For the $L_2$ fused lasso model with optimal penalty values, this appears to be the case. As can be seen in Fig. 5.1, the coefficient plots look very alike. Also, as can be seen in Tab. 5.1, both models make the same classifications.

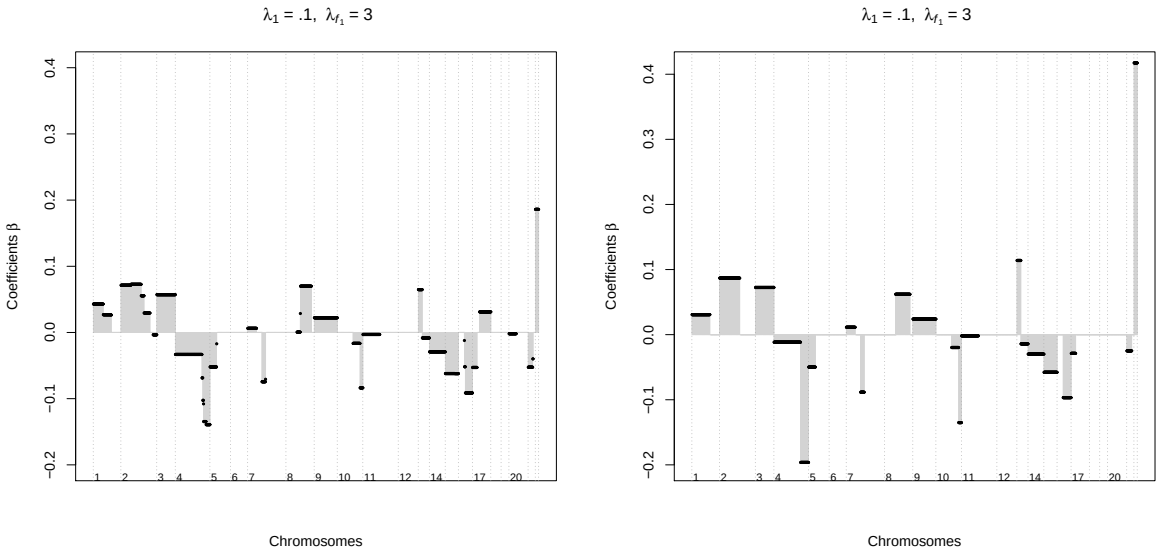
However, this similarity between the model returned by the $L_2$ fused lasso algorithm subject to a larger and smaller convergence error, does not hold for all penalty values. For example, for the $\lambda$ values in Fig. 5.2 considerable different coefficients are returned when the algorithm is terminated at a convergence error of $10^{-3}$, or subsequently run till convergence at an error of $10^{-7}$. Therefore, the optimal penalty values found by cross–validation might not be the theoretical optimal penalty values of the $L_2$ fused lasso.




**(a)** *Convergence error equals* $10^{-3}$ **(b)** *Convergence error equals* $10^{-7}$

**Figure 5.2.:** *Coefficient plots for the $L_2$ fused lasso on the bladder cancer dataset for a smaller (left) and larger (right) convergence error, at non–optimal penalty values.*

**$L_1$ Fused Lasso: Maximum Number of Iterations.** The $L_1$ fused lasso algorithm is computationally intensive, with the double absolute value penalty. The grid cross–validation based on the prediction accuracy of the $L_1$ fused lasso is therefore an expensive procedure. As it was noticed by Nimisha Chatuverdi, first author of Chatuverdi et al. (2013), that the coefficients of the $L_1$ fused algorithm with a maximum of iterations of 100 gave similar prediction accuracy as the fully converged $L_1$ fused lasso, Chatuverdi et al. (2013) used 100 iterations whether the model had converged or not to compute the optimal penalty values and to compute the classification accuracy.




**(a)** *Coefficients after 100 iterations* **(b)** *Coefficients at convergence*

**Figure 5.3.:** *Coefficient plots for $L_1$ fused lasso on the bladder cancer dataset, at a maximum of 100 iterations (left), and at convergence (right). The model with a convergence error of $10^{-7}$ converged at 998 iterations.*

As can be seen in Fig. 5.3, the $L_1$ fused lasso after 100 iterations is less rigid in having all coefficients the same size. It is therefore a smoother method than a converged $L_1$ fused lasso with a convergence error of $10^{-7}$, and might predict as well (or even better than) a converged $L_1$ fused lasso. As it was stated that the prediction accuracy was similar to the fully converged model, it is suggested that the model is close to convergence even though no value can be given to how close that is. In this thesis, we have used the optimal penalty values from Chatuverdi et al. (2013) which turned out to be based on the model with a maximum of 100 iterations as if they would be the optimal penalties of the converged $L_1$ fused lasso model. The $L_2$ fused lasso method suggested that that is not preferable.

Taking 100 iterations is in practice similar to our solution of increasing the convergence error of the $L_2$ fused lasso. In theory, especially in a failing algorithm, the coefficient point after 100 iterations can be anywhere in the predictor space. The distance to the optimum is unknown. Therefore, we chose to increase the convergence error in the $L_2$ fused lasso optimizations.

**Smoothed Logistic Regression: Not Fully Cross–Validated.** Cross–validated class predictions by the smoothed logistic regression method were not available for all tumor samples. For 7 out of the 57 models that left one observation out, maximum likelihood estimation was not possible. Influenced by the sparsity in the predictor space, the cases were perfectly linearly separable by a hyperplane. The smoothed

logistic regression is more prone to this, because it has one free parameter per chromosome and then also the unpenalized intercept.

**Grade Classification in Bladder Cancer.** In tumorgenesis, it is a far from standard procedure to use copy number data in clinical practice. There is more focus on identifying key genes. The copy number profile is sometimes perceived as too rough, as copy number data can not detect mutations, inversions, infections, or relocation of segments of DNA. It solely counts the copies. In our dataset, there are some grade 3 copy number profiles without any visible alterations to the DNA, which confirms the statement that copy number data is missing important information. There are on the other hand also some grade 1 classifications with a very altered copy number profile in our dataset.

It is questionable if the grade classification in bladder cancer as by the 1973 WHO classification criteria (Mostofi et al., 1973) should be completely replicated. The classification has since been updated, a notable adaption is that the 2004 classification criteria also regards genetic instability as a factor (Montironi & Lopez-Beltran, 2005).

## Concluding

We first propose some solutions to the limitations presented above. First of all, it is advised to increase the number of tumor samples in the bladder cancer dataset. This would not only solve the maximum likelihood estimation problem in the smoothed logistic regression, it would also permit cross–validation through training and test sets. This in its turn would have limited the amount of models to be fitted compared to leave–one–out cross–validation, thereby decreasing computation time.

Furthermore, a suggestion for future research is to improve the speed of the $L_2$ fused algorithm so that its optimal penalty values as indicated by the cross–validated likelihood can be retrieved for an $L_2$ fused model with a smaller convergence error, so that the $L_2$ fused lasso can be compared to other classification methods at optimal penalty values. The same holds for the $L_1$ fused lasso.

For the fused lasso models, we suggest to explore the possibility of increasing the fused lambda by not taking the value which maximizes the cross–validated likelihood, but instead take a larger value within one standard deviation to capture a cleaner trend by denoising, as suggested by an $L_1$ fused lasso in the `R` package genlasso (Tibshirani & Taylor, 2011).

The proposed models could be extended to other types of regression, for example survival analysis. Also the models could be extended to other high dimensional data that is ordered, as the quadratic difference penalty is not the most obvious choice in copy number data. After all, the $L_1$ fused lasso is theoretically a more promising

model in the analysis of copy number data, since a copy number profile generally has equal gains and losses over large regions.

Finally we would suggest to make the smoothing regions smaller, per chromosome arm or per known cancer related region. As a remaining remark on fitting specific patterns: constraining a regression with a structured matrix opens doors to a whole variety of models than can capture trends in data, which leads to a wide amount of interesting possibilities to further explore.

# A. All Predicted Class Probabilities

| № | True Class | Ridge | Lasso | $L_2$ Fused Lasso | $L_1$ Fused Lasso | Smoothing |
|---|---|---|---|---|---|---|
| 1 | 1 | .57 | .57 | .55 | .83 | 1 |
| 2 | 1 | .59 | .58 | .58 | .95 | 1 |
| 3 | 0 | .49 | .46 | .44 | .46 | 1 |
| 4 | 0 | .80 | .79 | .77 | .97 | n/a |
| 5 | 1 | .46 | .52 | .50 | .17 | .07 |
| 6 | 0 | .43 | .46 | .45 | .21 | 0 |
| 7 | 1 | .70 | .69 | .71 | .43 | .99 |
| 8 | 0 | .52 | .53 | .51 | .30 | .57 |
| 9 | 0 | .51 | .51 | .50 | .26 | .01 |
| 10 | 1 | .67 | .53 | .59 | .96 | 1 |
| 11 | 1 | .55 | .52 | .51 | .70 | 1 |
| 12 | 0 | .50 | .53 | .53 | .27 | .13 |
| 13 | 0 | .52 | .53 | .53 | .31 | .03 |
| 14 | 1 | .52 | .56 | .54 | .45 | .84 |
| 15 | 1 | .62 | .52 | .52 | .62 | 1 |
| 16 | 1 | .44 | .55 | .54 | 0 | 0 |
| 17 | 1 | .53 | .48 | .50 | .45 | 1 |
| 18 | 1 | .51 | .52 | .49 | .12 | .21 |
| 19 | 1 | .63 | .83 | .84 | .92 | 1 |
| 20 | 1 | .59 | .80 | .81 | .59 | 0 |
| 21 | 1 | .54 | .72 | .69 | .54 | .85 |
| 22 | 1 | .56 | .55 | .48 | .47 | .92 |
| 23 | 0 | .49 | .53 | .52 | .31 | .16 |
| 24 | 0 | .47 | .47 | .46 | .08 | 0 |
| 25 | 0 | .53 | .58 | .56 | .49 | 1 |
| 26 | 1 | .44 | .36 | .40 | .06 | .96 |
| 27 | 1 | .80 | .87 | .90 | 1 | 1 |
| 28 | 1 | .55 | .60 | .59 | .45 | n/a |
| 29 | 1 | .57 | .54 | .53 | .08 | n/a |
| 30 | 1 | .68 | .58 | .66 | .75 | 1 |
| 31 | 1 | .45 | .53 | .51 | .24 | .96 |
| 32 | 0 | .51 | .55 | .54 | .37 | .61 |
| 33 | 0 | .44 | .48 | .47 | .06 | 0 |
| 34 | 1 | .48 | .49 | .49 | .34 | .01 |

| № | True Class | Ridge | Lasso | $L_2$ Fused Lasso | $L_1$ Fused Lasso | Smoothing |
|---|---|---|---|---|---|---|
| 35 | 0 | .66 | .56 | .56 | 1 | 1 |
| 36 | 1 | .53 | .54 | .54 | .13 | 0 |
| 37 | 1 | .75 | .53 | .52 | 1 | 1 |
| 38 | 0 | .56 | .52 | .52 | .71 | .01 |
| 39 | 0 | .53 | .54 | .54 | .40 | n/a |
| 40 | 0 | .49 | .49 | .49 | .30 | .31 |
| 41 | 1 | .57 | .60 | .63 | .62 | n/a |
| 42 | 0 | .81 | .59 | .63 | 1 | n/a |
| 43 | 1 | .62 | .60 | .59 | .84 | 1 |
| 44 | 1 | .60 | .59 | .60 | .85 | 1 |
| 45 | 0 | .49 | .48 | .48 | .02 | 0 |
| 46 | 0 | .52 | .50 | .51 | .60 | .43 |
| 47 | 0 | .49 | .50 | .50 | .13 | 0 |
| 48 | 0 | .51 | .48 | .46 | .58 | n/a |
| 49 | 1 | .56 | .51 | .49 | .03 | 0 |
| 50 | 0 | .70 | .88 | .87 | .98 | .05 |
| 51 | 0 | .48 | .54 | .53 | .24 | .24 |
| 52 | 1 | .67 | .63 | .62 | .97 | 0 |
| 53 | 1 | .48 | .49 | .49 | .30 | .04 |
| 54 | 0 | .70 | .59 | .60 | .99 | .97 |
| 55 | 1 | .61 | .55 | .55 | .94 | 1 |
| 56 | 1 | .60 | .55 | .55 | .85 | 1 |
| 57 | 1 | .67 | .60 | .59 | .94 | 1 |

**Table A.1:** *True class versus predicted class probabilities. True class value 1 corresponds to grade 3, true class value 0 corresponds to the lower grades. Seven probabilities could not be predicted for the smoothing method, and are therefore not available (n/a).*

# Bibliography


Beroukhim, R., Mermel, C. H., Porter, D., Wei, G., Raychaudhuri, S., Donovan, J., Barretina, J., Boehm, J. S., Dobson, J., Urashima, M., Mc Henry, K. T., Pinchback, R. M., Ligon, A. H., Cho, Y.-J., Haery, L., Greulich, H., Reich, M., Winckler, W., Lawrence, M. S., Weir, B. A., Tanaka, K. E., Chiang, D. Y., Bass, A. J., Loo, A., Hoffman, C., Prensner, J., Liefeld, T., Gao, Q., Yecies, D., Signoretti, S., Maher, E., Kaye, F. J., Sasaki, H., Tepper, J. E., Fletcher, J. A., Tabernero, J., Baselga, J., Tsao, M.-S., Demichelis, F., Rubin, M. A., Janne, P. A., Daly, M. J., Nucera, C., Levine, R. L., Ebert, B. L., Gabriel, S., Rustgi, A. K., Antonescu, C. R., Ladanyi, M., Letai, A., Garraway, L. A., Loda, M., Beer, D. G., True, L. D., Okamoto, A., Pomeroy, S. L., Singer, S., Golub, T. R., Lander, E. S., Getz, G., Sellers, W. R., & Meyerson, M. (2010). The landscape of somatic copy-number alteration across human cancers. *Nature*, 463, 899–905.

Brewster, A. M., Thompson, P., Sahin, A. A., Do, K., Edgerton, M., Murray, J. L., Tsavachidis, S., Zhou, R., Liu, Y., Zhang, L., Mills, G., & Bondy, M. (2011). Copy number imbalances between screen- and symptom-detected breast cancers and impact on disease-free survival. *Cancer Prevention Research (Philadelphia, Pa.)*, 4, 1609–1616.

Chatuverdi, N., de Menezes, R. X., & Goeman, J. J. (2013). A fused lasso algorithm for Cox and generalized linear models with application to copy number profiles. *Manuscript submitted for publication.*

Dziuda, D. M. (2010). *Data mining for genomics and proteomics: Analysis of gene and protein expression data.* Wiley series on methods and applications in data mining. Wiley-Interscience.

Eilers, P. H. C. & Marx, B. D. (1996). Flexible smoothing with B-splines and penalties. *Statistical Science*, 11, 89–102.

Fletcher, R. W. & Fletcher, S. W. (2005). *Clinical epidemiology: The essentials.* Lippincott Williams & Wilkins, fourth edition.

Goeman, J. J. (2010). $L_1$ penalized estimation in the Cox proportional hazards model. *Biometrical Journal*, 52, 70–84.

Hirsch, F. R., Varella-Garcia, M., Bunn, P. A., Jr., Di Maria, M. V., Veve, R., Bremnes, R. M., Barón, A. E., Zeng, C., & Franklin, W. A. (2003). Epidermal growth factor receptor in non–small-cell lung carcinomas: Correlation between gene copy number and protein expression and impact on prognosis. *Journal of Clinical Oncology*, 21, 3798–3807.

Hoerl, A. E. & Kennard, R. W. (1970). Ridge regression: Biased estimates for nonorthogonal problems. *Technometrics*, 12, 55–67.

Hupé, P., Stransky, N., Thiery, J.-P., Radvanyi, F., & Barillot, E. (2004). Analysis of array CGH data: From signal ratio to gain and loss of DNA regions. *Bioinformatics*, 20, 3413–3422.

Kan, Z., Jaiswal, B. S., Stinson, J., Janakiraman, V., Bhatt, D., Stern, H. M., Yue, P., Haverty, P. M., Bourgon, R., Zheng, J., Moorhead, M., Chaudhuri, S., Tomsho, L. P., Peters, B. A., Pujara, K., Cordes, S., Davis, D. P., Carlton, V. E. H., Yuan, W., Li, L., Wang, W., Eigenbrot, C., Kaminker, J. S., Eberhard, D. A., Waring, P., Schuster, S. C., Modrusan, Z., Zhang, Z., Stokoe, D., de Sauvage, F. J., Faham, M., & Seshagiri, S. (2010). Diverse somatic mutation patterns and pathway alterations in human cancers. *Nature*, 466, 869–873.

Land, S. & Friedman, J. H. (1996). *Variable fusion: A new method of adaptive signal regression.* Technical Report 114, Department of Statistics, Stanford University.

le Cessie, S. & van Houwelingen, J. C. (1992). Ridge estimators in logistic regression. *Applied Statistics*, 41, 191–201.

Li, C. & Li, H. (2008). Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*, 24, 1175–1182.

Meijer, R. J. & Goeman, J. J. (2013). Efficient approximate k-fold and leave-one-out cross-validation for ridge regression. *Biometrical Journal*, 55, 141–155.

Montironi, R. & Lopez-Beltran, A. (2005). The 2004 WHO classification of bladder tumors: A summary and commentary. *International Journal of Surgical Pathology*, 13, 143–153.

Mostofi, F. K., Sobin, L. H., & Torloni, H. (1973). *Histological typing of urinary bladder tumors.* Number 10 in International histological classification of tumours. Geneva: World Health Organization.

Pinkel, D., Segraves, R., Sudar, D., Clark, S., Poole, I., Kowbel, D., Collins, C., Kuo, W. L., Chen, C., Zhai, Y., Dairkee, S. H., Ljung, B. M., Gray, J. W., & Albertson, D. G. (1998). High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays. *Nature Genetics*, 20, 207–211.

Pollack, J. R., Perou, C. M., Alizadeh, A. A., Eisen, M. B., Pergamenschikov, A., Williams, C. F., Jeffrey, S. S., Botstein, D., & Brown, P. O. (1999). Genome-wide analysis of DNA copy-number changes using cDNA microarrays. *Nature Genetics*, 23, 41–46.

Ruppert, D., Wand, M. P., & Carroll, R. J. (2003). *Semiparametric regression.* Cambridge series in statistical and probabilistic mathematics. Cambridge University Press.

Shen, L. & Tan, E. C. (2005). PLS and SVD based penalized logistic regression for cancer classification using microarray data. In *Proceedings of 3rd Asia-Pacific Bioinformatics Conference* (pp. 219–228). Singapore.

Slawski, M., zu Castell, W., & Tutz, G. (2010). Feature selection guided by structural information. *The Annals of Applied Statistics*, 4, 1056–1080.

Stransky, N., Vallot, C., Reyal, F., Bernard-Pierrot, I., de Medina, S. G., Segraves, R., de Rycke, Y., Elvin, P., Cassidy, A., Spraggon, C., Graham, A., Southgate, J., Asselain, B., Allory, Y., Abbou, C. C., Albertson, D. G., Thiery, J. P., Chopin, D. K., Pinkel, D., & Radvanyi, F. (2006). Regional copy number-independent deregulation of transcription in cancer. *Nature Genetics*, 38, 1386–1396.

Stratton, M. R., Campbell, P. J., & Futreal, P. A. (2009). The cancer genome. *Nature*, 458, 719–724.

Stuart, D. & Sellers, W. R. (2009). Linking somatic genetic alterations in cancer to therapeutics. *Current Opinion in Cell Biology*, 21, 304–310.

Sun, H. & Wang, S. (2012). Penalized logistic regression for high-dimensional DNA methylation data with case-control studies. *Bioinformatics*, 28, 1368–1375.

Theisen, A. (2008). Microarray-based comparative genomic hybridization (aCGH). *Nature Education*, 1.

Tibshirani, R. (1996). Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society: Series B*, 58, 267–288.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., & Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B*, 67, 91–108.

Tibshirani, R. J. & Taylor, J. (2011). The solution path of the generalized lasso. *Annals of Statistics*, 39, 1335–1371.

Vollebergh, M. A., Lips, E. H., Nederlof, P. M., Wessels, L. F. A., Schmidt, M. K., van Beers, E. H., Cornelissen, S., Holtkamp, M., Froklage, F. E., de Vries, E. G. E., Schrama, J. G., Wesseling, J., van de Vijver, M. J., van Tinteren, H., de Bruin, M., Hauptmann, M., Rodenhuis, S., & Linn, S. C. (2011). An aCGH classifier derived from BRCA1-mutated breast cancer and benefit of high-dose platinum-based chemotherapy in HER2-negative breast cancer patients. *Annals of Oncology*, 22, 1561–1570.

Zou, H. & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, 67, 301–320.