Penalized Regression Methods with Application to Generalized Linear Models, Generalized Additive Models, and Smoothing

Sri Utami Zuliana

A thesis submitted for the degree of

Doctor of Philosophy

Department of Mathematical Sciences

University of Essex

March 2017

Dedicated to

My father *Ayah Djahid* and my mother *Umi Rus*

My beloved husband *Mas Arief* and my beloved children *Faiq*, *Ica* and *Luki*

Acknowledgements

I would like to express my gratitude to Almighty Allah.

There are many people that helped, inspired and encourages me to progress and complete this thesis, to whom I am deeply thankful. First and foremost, I gratefully acknowledge the support of my supervisor, Dr. Aris Perperoglou. He has provided me with his remarkable insights, guidance and continuous encouragement. This work would not have been possible without his help and guiding. I am thankful to my supervisory board member Prof. Peter Higgins for his valuable suggestions and encouragement.

I am very thankful to Ministry of Religious Affair, Republic of Indonesia for financial support and UIN Sunan Kalijaga, Yogyakarta, Indonesia for full support. I am thankful to the administration of Department of Mathematical Sciences University of Essex for their support.

I would like to thanks to all my friends with whom I shared my happiness and sadness and who helped me during my study. I am thankful for my brother, my sisters in law and all my big family members for their support.

Abstract

Recently, penalized regression has been used for dealing problems which found in maximum likelihood estimation such as correlated parameters and a large number of predictors. The main issues in this regression is how to select the optimal model. In this thesis, Schall's algorithm is proposed as an automatic selection of weight of penalty.

The algorithm has two steps. First, the coefficient estimates are obtained with an arbitrary penalty weight. Second, an estimate of penalty weight λ can be calculated by $\hat{\lambda} = \frac{\hat{\sigma^2}}{\hat{\tau^2}}$, where $\hat{\sigma^2}$ is the variance of error and $\hat{\tau^2}$ is the variance of coefficient. The iteration is continued from step one until an estimate of penalty weight converge. The computational cost is minimized because the optimal weight of penalty could be obtained within a small number of iterations.

In this thesis, Schall's algorithm is investigated for ridge regression, lasso regression and two-dimensional histogram smoothing. The proposed algorithm are applied to real datasets and simulation dataset. In addition, a new algorithm for lasso regression is proposed. The performance of results of the algorithm was almost comparable in all applications. Schall's algorithm can be an efficient algorithm for selection of weight of penalty.

Declaration

The work in this thesis is based on research carried out at the Department of Mathematical Sciences, University of Essex, United Kingdom. I certify that this is all my own work, unless referenced in the text, no part of this thesis has been submitted elsewhere for any other degree or qualification.

Abreviations

AIC	Akaike information criterion
AICc	Akaike information criterion with a correction
B-splines	basis splines
BIC	bayesian information criterion
ED	effective dimensions
GAMs	generalized additive models
GCV	generalized cross-validation
GLMs	generalized linear models
i.i.d	independent and identically distributed random variables
IWLS	iterative weighted least square
Lasso	least absolute shrinkage and selection operator
MLE	maximum likelihood estimation
MSE	mean squared errors
OLS	ordinary least squared
P-GAMs	generalized additive models with penalized B-splines
P-splines	penalized B-splines
PRIDE	penalized regression with individual deviance effects

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Regression analysis is a method, which describes the relationships between a dependent variable and independent variables. The most simple method is a classical linear model. The model relates the dependent variable to a linear combination of independent variables. Classical linear models have the assumption of normally distributed errors.

The generalized linear models (GLMs) allows for non-normal error distributions. There are three components to any GLMs: the random components, the systematic component and the link function. The distribution in the random components may come from an exponential (Nelder and Wedderburn, 1972). Covariates $\boldsymbol{x}$ produce a linear predictor.

In addition, generalized additive models (GAMs) may have a linear or a non linear form via the use of smooth functions (Hastie and Tibshirani, 1986). GAMs will be exhibited by penalized-splines (P-splines).

Coefficients are estimated using maximum likelihood estimation (MLE). However, MLE has problems such as large variability or lack of interpretability i.e. a model is failed giving a useful prediction or representation of a phenomenon. A penalized regression gives more stable results, continuous, and computationally efficient (Cessie et al., 1992; Verweij and Van Houwelingen, 1994).

All of these models may include some penalty in the likelihood. This introduces the complexity of having to optimize the penalty weight. Other problem rises for penalized regression. It needs large grid of λ s to choose the optimal model. In this thesis, we are going to utilize Schall's algorithm for penalty optimisation. The performance of the Schall's algorithm will be investigated and compared to the commonly used methods such as Akaike information criterion (AIC), bayesian information criterion (BIC) and generalized cross-validation (GCV). The algorithm is applied to data from real and simulation data sets in GLMs, generalized additive models (GAMs), least absolute shrinkage and selection operator (Lasso), and two-dimensional histogram.

The remaining thesis consists of six chapters and is organized as follows: Chapter 2 presents how the Schall's algorithm is applied to ridge regression for generalized linear models. Chapter 3 still discusses how the Schall's algorithm is applied to ridge regression for generalized linear models but especially for Poisson regression and logistic regression. The algorithm is applied to a normal dependent variable. Chapter 4 is a discussion about how the Schall's algorithm is applied to ridge regression for generalized additive models. Chapter 5 is dedicated to discussing how the Schall's algorithm is applied to lasso regression. Chapter 6 is dedicated to discussing how the Schall's algorithm is applied to

a two-dimensional histogram. Finally, Chapter 7 presents the conclusion, and possible future works. The list of publications related to this thesis is presented in publications.

Chapter 2

The Weight of Penalty Optimization for Ridge Regression [1]

2.1 Introduction

Ridge regression (Hoerl and Kennard, 1970; Hoerl et al., 1975) is used in many applications to shrink estimates of coefficients towards zero. It was introduced originally within the family of linear models. It is implemented in generalized linear models (Cessie et al., 1992; Perperoglou, 2014) as well as within the context of high-dimensional data and machine learning.

On all these approaches, a penalty term is added to the likelihood, controlled by a weight λ. It is up to the researcher to decide what should the penalty weight be. A common

[1] This chapter is published in Zuliana, S. U., and Perperoglou, A. (2016). The Weight of Penalty Optimization for Ridge Regression. In Analysis of Large and Complex Data (pp. 231-239). Springer International Publishing.

method is used to optimize the penalty is to select a series of different λs, fit the model for each of the weights and choose a model that would maximize a criterion such as Akaike's Information Criterion (Akaike, 1974), the corrected version (AICc) (Hurvich and Tsai, 1989) or Bayesian Information criterion (BIC) (Schwarz et al., 1978). In other cases generalized cross validation may be used (GCV) (Golub et al., 1979). Examples of the latter approach can be found in Cessie et al. (1992) for logistic regression, or in simple linear regression one may use function `lm.ridge` available in package `MASS` (Venables and Ripley, 2002) within `R` (R Development Core Team, 2015) software. More recently, Goeman suggested leave-one-out cross validation (Goeman, 2010) which was implemented in package `penalized` (Goeman et al., 2012).

All of these approaches can be computationally expensive. In more complicated models where estimation time may be an issue, penalty optimization through a grid search of weights is counter-productive. Xue et al. (2007) suggested simple remedies to address the problem, within the framework of survival analysis, which where shown however to be inferior in simulation studies (Perperoglou, 2014). Recently, within the field of econometrics Kibria investigated penalty weights that are obtained by dividing the residual mean square estimate with the maximum, mean, median, etc of the coefficients (Kibria, 2003) and came up with suggestions in their follow up paper (Muniz and Kibria, 2009). More recently Månsson and Shukur (2011) investigated the performance of these estimators for Poisson regression. Cule and De Iorio (2013) introduced a four step algorithm to fit penalized models based on principal components of the eigenvectors of the regressors. This approach is implemented in package `ridge` (Cule, 2014), for linear and logistic regression.

Here we present an approach that is based on mixed models methodology. We view the penalty as a random effect added to the model and then we employ mixed model machinery to estimate optimal weight. Under that umbrella λ becomes a parameter to be estimated from the model with a repeating algorithm. Our approach is similar to the one suggested by Rigby and Stasinopoulos (2013). Their method is an automatic selection of the smoothing parameters when fitting a generalised additive model for location, scale and shape (GAMLSS) model. Whilst our method is an automatic selection of penalty weight when fitting a generalized linear models. They have implemented their method in package `gamlss` (Rigby and Stasinopoulos, 2005).

The chapter is organized as follows: In Section 2.2, we present the background theory on penalized regression methods in generalized linear models. We present the general framework and show how to optimize the penalty weight using a mixed models approach. The emphasis is on a special case of a GLM, a simple linear model. In Section 2.3, we use this simple case to illustrate the Bayesian viewpoint of our suggested algorithm and present simulation studies that evaluate the performance of the suggested algorithm and also compare it with other methods. It closes with a discussion (Section 2.4).

2.2 Ridge Regression in a Generalized Linear Models (GLM)

Consider the form of any generalized linear model as:

$$g(E(\boldsymbol{y})) = \boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta} \tag{2.1}$$

where $\boldsymbol{y}$ is a response variable coming from any of the exponential family distributions, g() is the link function and $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta}$ is the linear part of the model for $\mathbf{X}$, an $n \times p$ matrix of p covariates on n observations and $\boldsymbol{\beta}$ is the vector of unknown coefficients. Let $l(\boldsymbol{\beta})$ denotes the log-likelihood function of that general model and defines the penalized likelihood function as:

$$l^*(\boldsymbol{\beta}) = l(\boldsymbol{\beta}) - \frac{1}{2}\lambda \sum_{j=1}^{p} \beta_j^2 \tag{2.2}$$

To estimate the model an Iterative Weighted Least Squares (IWLS) algorithm can be used which takes the form:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{W}\mathbf{X} + \lambda \boldsymbol{I})^{-1}\mathbf{X}'\mathbf{W}\boldsymbol{z} \tag{2.3}$$

where $\mathbf{W}$ is a diagonal matrix with appropriate weights $w_1, w_2, \ldots, w_n$ in the diagonal , $\boldsymbol{z}$ is the intermediate variable given by $\boldsymbol{z} = \mathbf{W}^{-1}(\boldsymbol{y} - \hat{\boldsymbol{\mu}}) + \mathbf{X}\boldsymbol{\beta}$ and $\boldsymbol{I}$ is a $p \times p$ identity matrix.

The choice of penalty weight is crucial. In cases where λ tends to infinity coefficients become zero, while when λ approaches zero coefficients are allowed to vary freely.

2.2.1 Ridge regression from Bayesian perspective

Any penalized model may be seen as a mixed model. Let $p_{\beta^*}(\boldsymbol{x}^*, \boldsymbol{y}^*)$ be the joint density function of observed data $\boldsymbol{x}^*$ and unobserved data $\boldsymbol{y}^*$ when parameter $\boldsymbol{\beta}^*$ is known. We can then define the posterior probability $p(\boldsymbol{\beta}^*|y^*)$ as: the likelihood for $\boldsymbol{\beta}^*$ and $\boldsymbol{y}^*$ as:

$$L(\boldsymbol{\beta}^*; \boldsymbol{y}^*) = p_{\beta^*}(\boldsymbol{y}^*)p_{\beta^*}(\boldsymbol{x}^*|\boldsymbol{y}^*) \tag{2.4}$$

Lee and Nelder (1996) defined equation (2.4) as an *h-likelihood* while Green and Silverman as *penalized likelihood* (Green and Silverman, 1993). h-likelihood can also be seen mathematically as a Bayesian posterior distribution . The first part of the (2.4) corresponds to the likelihood of the simple model multiplied by the likelihood that corresponds random part, in this case, the ridge penalty. Hierarchical likelihood has many similarities to Bayesian methods.

Consider a simple linear model

$$\boldsymbol{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{2.5}$$

with $\mathbf{X}$ an $n \times p$ matrix of covariates and $\boldsymbol{\beta}$ a $p \times 1$ vector of coefficients. Then where $\boldsymbol{y} \sim \mathrm{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2\boldsymbol{I})$ and let $\boldsymbol{\beta} \sim \mathrm{N}(0, \tau^2\boldsymbol{I})$.

Then the likelihood can be written as:

$$L(\boldsymbol{\beta}|\boldsymbol{y}) \propto \exp\left(-\frac{1}{2\sigma^2}\left(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\right)'\left(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\right)\right)\exp\left(-\frac{1}{2\tau^2}\hat{\boldsymbol{\beta}}'\hat{\boldsymbol{\beta}}\right) \tag{2.6}$$

Taking the logarithm of (2.6) leads to:

$$\begin{aligned} -\log L(\boldsymbol{\beta}|\boldsymbol{y}) &= \frac{1}{2\sigma^2}\left(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\right)'\left(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\right) + \frac{1}{2\tau^2}\hat{\boldsymbol{\beta}}'\hat{\boldsymbol{\beta}} \\ &= \frac{1}{2\sigma^2}\left(\left(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\right)'\left(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\right) + \lambda\hat{\boldsymbol{\beta}}'\hat{\boldsymbol{\beta}}\right) \end{aligned}$$

with $\lambda = \frac{\sigma^2}{\tau^2}$.

Looking at model (2.5) from a mixed model perspective one needs to estimate, along with the coefficients, the variance of the random effects as well. Schall (1991) defined a two-step algorithm for fitting mixed models and estimating the variance of the random effect . In this study, the algorithm is used to estimate a penalty weight. It has the following steps:

1. For given $\hat{\sigma}^2, \hat{\lambda}$ estimate the coefficient $\hat{\boldsymbol{\beta}}$ by:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\tilde{\mathbf{W}}\mathbf{X} + \hat{\lambda}\boldsymbol{I})^{-1}\mathbf{X}'\tilde{\mathbf{W}}\tilde{\boldsymbol{z}}$$

2. Given estimates of coefficients $\hat{\boldsymbol{\beta}}$, variance estimators are obtained from

$$\hat{\sigma^2} = \frac{(\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{\beta}})'(\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{\beta}})}{n - \mathrm{ED}}$$

   and

$$\hat{\tau^2} = \frac{\hat{\boldsymbol{\beta}}'\hat{\boldsymbol{\beta}}}{\mathrm{ED}}$$

   where ED stands for effective dimensions and is the trace of the hat matrix of the mode (Hoaglin and Welsch, 1978). An estimate of the penalty weight $\hat{\lambda}$ can be then given by:

$$\hat{\lambda} = \frac{ED}{\hat{\boldsymbol{\beta}}'\hat{\boldsymbol{\beta}}}$$

3. Iterate until the estimated penalty weight $\hat{\lambda}$ convergence.

The algorithm can be initialized with any value for $\hat{\lambda}$ and usually converges within a small number of steps. For further applications see Perperoglou (2014) and Perperoglou

and Eilers (2010). An implementation of the method is also part of the `coxRidge` package in `R` (Perperoglou, 2013).

2.3 Simulation

A simulation study was designed to investigate the performance of different approaches to maximize penalty weight. The sample size of the full data was $n = 500$. The response variable $\boldsymbol{y}$ was simulated from

$$\boldsymbol{y} = \boldsymbol{\beta z} + 0.2\boldsymbol{\epsilon}$$

where $\boldsymbol{z}$ comes from a standard normal distribution ($\boldsymbol{z} \sim N(0, 1)$), and the true value of the coefficient is 1 ($\beta = 1$). The normal distribution is chosen because it is the most familiar distribution and ease of statistical flexibility. Some noise is added in the form of a random vector $\epsilon \sim N(0, 1)$ which is independent of $\boldsymbol{z}$.

In a second step, the simulated values of $\boldsymbol{z}$ where used to create a set of correlated regressors, given as:

$$\begin{aligned} \boldsymbol{x_1} &= \boldsymbol{z} + \boldsymbol{\epsilon_1} \\ \boldsymbol{x_2} &= \boldsymbol{z} + \boldsymbol{\epsilon_2} \\ \boldsymbol{x_3} &= \boldsymbol{x_1} + \boldsymbol{x_2} + 0.05\boldsymbol{\epsilon_3} \end{aligned}$$

where the errors $\boldsymbol{\epsilon_1}, \boldsymbol{\epsilon_2}, \boldsymbol{\epsilon_3}$ are once again random numbers generated from a normal distribution and assumed to be independent from $\boldsymbol{z}$. There are correlation between $\boldsymbol{x_1}$ and $\boldsymbol{x_1}$,

and also between $\boldsymbol{x_1}$ and $\boldsymbol{x_1}$. The data set was then split into a training (labelled d_1) and testing data set (labelled d_2), of size $n_1 = 400$ and $n_2 = 100$, respectively, and a linear model of the form $\boldsymbol{y} = \beta_1\boldsymbol{x_1} + \beta_2\boldsymbol{x_2} + \beta_3\boldsymbol{x_3}$ was fitted on the data set where $\boldsymbol{\beta} = (1, 1, 1)$. A simple linear regression model was fitted to the training data along with four more penalized approaches based on different methods of penalty weight optimization. These approaches were: leave-one-out cross validation using package `penalized`, penalized quasi likelihood optimization using package `gamlss`, generalized cross validation using package `MASS` and optimization via random effects models suggested here using Schall's algorithm.

Once a model has been fitted, the prediction error on the testing dataset was obtained based on the estimates of each approach as

$$p.error = \sum_{i \in d_2} (y_{i \in d_2} - \hat{\beta} X_{i \in d_2})^2$$

The whole process was repeated 1000 times.

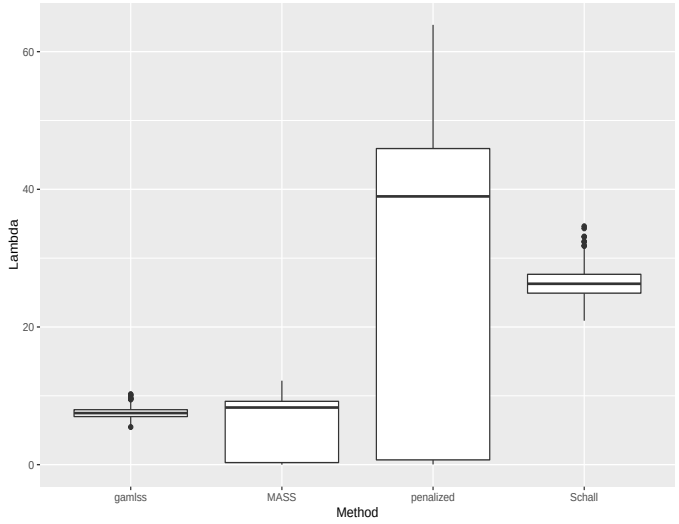

Figure 2.1: *Distribution of λs based on different methods of optimization*

Figure 2.1 illustrates the distribution of λs as they were obtained by the different methods. As it should be expected, the mixed models approach suggested here is almost identical to the penalized quasi likelihood optimization. On the other hand, leave-one-out cross validation produces on median λ which is high above all other approaches, while at the same time the spread of the distribution is much wider. On the other extreme of the spectrum, principal components optimization leads to very small weights and almost no penalization. Generalized cross validation also selects small penalty weights when compared with mixed models and leave-one-out cross validation.

Method	Prediction error	% of $\hat{\beta}_3 < 0$
OLS	37.65	49.9
penalized	37.60	20.4
gamlss	80.71	0
MASS	37.60	19.8
Schall	37.58	0
mgcv	37.15	51.0

Table 2.1: *The average prediction error of different methods.*

Including a penalty term λ not only shrunks estimates towards zero, but in cases where collinearity is present, it reduces mean squared prediction error and corrects coefficient signs. Table 2.1 illustrates the average prediction error of all approaches. As expected the simple linear model has the largest prediction error. Although the differences among the models are small, using our proposed algorithm produces the smallest prediction error with correct coefficient sign. `Package mgcv` gave the smallest prediction error but has 51 % of wrong coefficient sign. When no penalization is applied, estimates obtained from

the ordinary least squares model have an opposite sign from the real one.Multicollinearity leads to estimates of coefficients with wrong signs (Greene, 2012). The wrong sign is examined only for β_3 because it has correlation with other two independent variables. Table 2.1 presents in the third column the percentage of cases where $\hat{\beta}_3$ coefficient was mistakenly estimated as negative. Three out of four methods estimate a correct sign for the coefficient. Figure 2.2 described the computation time with respect to the methods. Schall and MASS gave the smallest computation time.

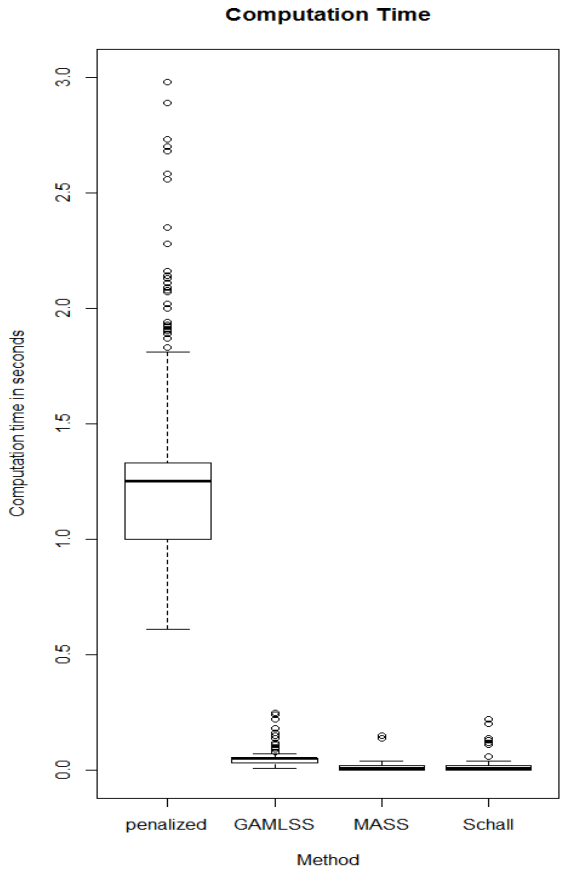

Figure 2.2: *Boxplot of computation time with respect to method for the simulated data with correlation between the independent variables*

A second simulation study was also applied to investigate the performance of the methods. This time, the regressors had the same distributional assumptions, however, correlation amongst them was 0. The data were simulated this way to investigate how each method performs when in fact penalization is not necessary. We simulated a single data set with a sample size of 500, and generated one outcome variable $\boldsymbol{y}$, and four covariates $\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \boldsymbol{x_4}$ which

$$\boldsymbol{x_1} \sim \mathcal{N}(0, 1)$$

$$\boldsymbol{x_2} \sim \mathcal{N}(0, 1)$$

$$\boldsymbol{x_3} \sim \mathcal{N}(0, 1)$$

The covariates $\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \boldsymbol{x_4}$ are independent of each other. The response $\boldsymbol{y}$ was generated from $\boldsymbol{y} = 0.7\boldsymbol{x_1} - 0.3\boldsymbol{x_2} + 0.2\boldsymbol{x_3} + 0.2\boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, 1)$. The data set was then split into a training (labelled d_1) and testing data set (labelled d_2), of size $n_1 = 400$ and $n_2 = 100$. A simple linear regression model was fitted to the training data along with four more penalized approaches based on different methods of penalty weight optimization. These approaches were: penalized quasi likelihood optimization using package `gamlss`, generalized cross validation using package `MASS`, integrated model selection via GCV using package `mgcv` and optimization via random effects models suggested here using Schall's algorithm.

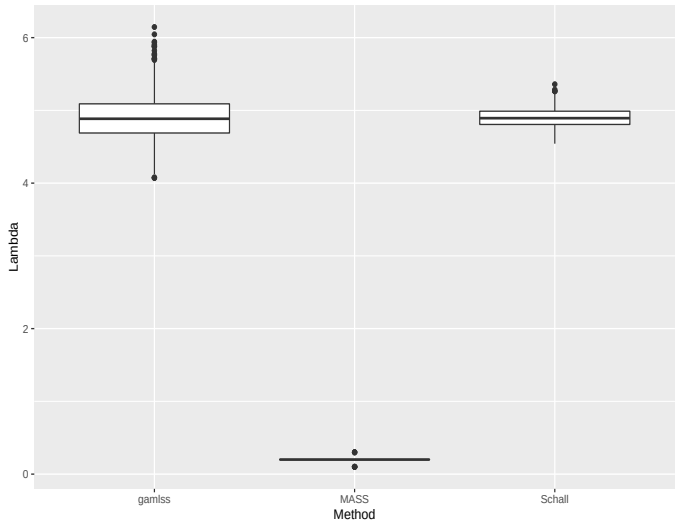

Figure 2.3: *Distribution of estimated λs based on different methods of optimization for the simulated data with no correlation between the independent variables*

Figure 2.3 illustrates the distribution of estimated λs. The graph reveals that both methods based on extended likelihoods (labelled as Schall and gamlss) overestimate the importance of the penalty. The median λ weight was 4.8 in both while in the one obtained by generalized cross validation, was 0.2. Figure 2.4 illustrates the distribution of computation time. The graph reveals that the computation time of ordinary least square is the shortest, following by the computation time of proposed algorithm, Schall's algorithm.

2.4 Summary

We have introduced a method for optimizing a penalty weight in ridge-type regression problems. The method is based on mixed models algorithms although in practice one does not need to regard the penalization as a random effect. We have shown the algorithm and illustrated application in two small simulation studies.

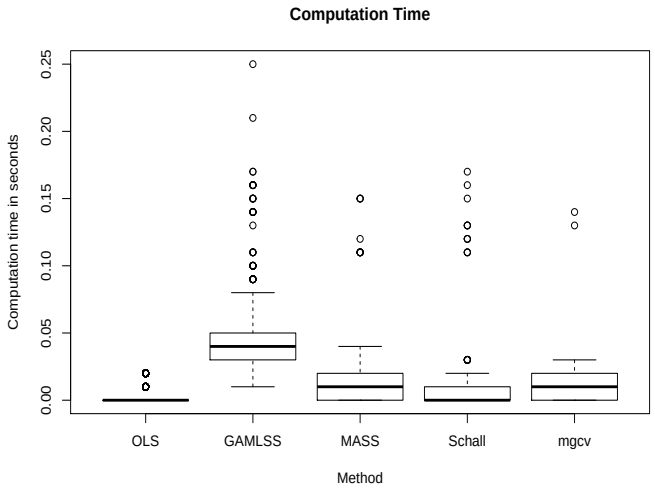

Figure 2.4: *Distribution of computation time based on different methods of optimization for the simulated data with no correlation between the independent variables*

The suggested method can work in any type of regression model, regardless of the distribution assumption of the response or the link function. In this work we have shown the advantages of our approach within the context of linear regression. Perperoglou has showed in other texts how the method can be used in survival analysis (Perperoglou, 2014). In future work we aim to show how the method performs when fitting Poisson or binary data.

We presented two simulation studies. As discussed earlier, some caution is needed when applying penalized methods in data that do not require that complexity from the model. Cross validation methods were able to perform quite well in the absence of collinearity and showed that λ has to be near zero, i.e. they ended up with no shrinkage of the coefficients. When mixed models methods were applied, some shrinkage was always present in the model. In any case, preliminary analysis of the data should reveal whether a penalty is needed or not.

It should be noted that using a mixed models approach as the one discussed here is similar to the approach within gamlss models. Both methods use a restricted maximum likelihood approach (REML) to estimate a variance of a random effect, and use that variance to obtain the penalty weight. The only difference is that Rigby and Stasinopoulos (2013) used their approach to optimize a roughness penalty when fitting regression splines for smoothing. An extension of either methods would be very useful in cases where a roughness penalty form smoothing models is needed in a model that also accounts for correlation, or in cases where penalties are applied into more than one dimensions. A similar idea has been explored in the Penalized Regression with Individual Deviance Effects models (PRIDE) (Perperoglou and Eilers, 2010). Unlike the other regression models, this model not only involve independent variables but also include individual deviance effects. Besides the model produces covariates estimates which give a general pattern of data, it gives information whether there is an invisible systematic pattern in data.

Chapter 3

Ridge Regression in Poisson Models and Logistic Models

3.1 Introduction

In the previous chapters, optimized ridge regression in generalized linear models was discussed, where the response variables have a normal distribution. In generalized linear models, the response variables belong to the exponential family of distributions. The most famous members of exponential families are normal, binomial, and Poisson distributions. Here we focus on generalized linear models with binomial and Poisson responses. The binomial distribution has applied in a lot of fields. It is used when there are two possible outcomes. It is applied for examining the presence of a characteristic. The Poisson distribution is often used to model rare events.

In Section 3.2, penalized Poisson regression will be presented followed by penalized logistic regression. In Section 3.3, we will illustrate the methods on practical applications.

3.2 Optimized Poisson Ridge Regression and Logistic Ridge Regression

Poisson regression analysis is commonly used for modelling data with a count independent variable. Suppose $\boldsymbol{Y} = (y_1, y_2, \ldots, y_n)$ is an i.i.d sample from a Poisson distribution with parameter $\boldsymbol{\mu}$, the likelihood is

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} \frac{\mu_i^{y_i} e^{-\mu_i}}{y_i!}$$

and the loglikehood is

$$l(y_i|\boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i \log \mu_i - \mu_i)$$

As it has been explained in the previous chapter, the penalized log-likelihood for this model is obtained from subtracting a ridge penalty term from the log-likelihood (equation 2.2). The log-likelihood for penalized log-linear Poisson model can be written as

$$l^*(\boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i \eta_i - \mu_i) - \frac{1}{2}\lambda \sum_{j=1}^{p} \beta_j^2$$

where the link function is $g(\mu_i) = \eta_i = \log \mu_i = \boldsymbol{x}_i^T \boldsymbol{\beta}$. The estimated coefficients are given by iterative weighted least square (IWLS):

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}' \tilde{\boldsymbol{W}} \mathbf{X} + \lambda \boldsymbol{I})^{-1} \mathbf{X}' \tilde{\boldsymbol{W}} \tilde{\boldsymbol{z}}$$

with the weights $\tilde{\boldsymbol{W}}$ is a diagonal matrix with elements a vector $\boldsymbol{\mu}$ on the diagonal and the intermediate variable $\tilde{\boldsymbol{z}} = (\boldsymbol{y} - \boldsymbol{\mu}) + \boldsymbol{\mu}\boldsymbol{\eta}$.

As for logistic regression is commonly used for data with binomial response variable. Suppose $\boldsymbol{Y^+} = (y_1^+, y_2^+, \ldots, y_n^+)$ is an iid sample from a binomial distribution with parameter n and $\boldsymbol{\mu^+}$, the likelihood is

$$L(\boldsymbol{\beta}) \quad = \quad \prod_{i-1}^{n} \mu_i^{y_i^+} (1 - \mu_i^+)^{1-y_i^+}$$

and the loglikehood is

$$l(\beta) = \sum_{i=1}^{n} \left(y_i^+ \eta_i^+ - \log(1 + e^{\eta_i^+}) \right)$$

where the link function is $g(\mu_i^+) = \eta_i^+ = \log\left(\frac{\mu_i^+}{1-\mu_i^+}\right)$.

Similar to Poisson ridge regression, the penalty is subtracted from the log likelihood:

$$l^*(\beta) = \sum_{i=1}^{n} \left(y_i^+ \eta_i^+ - \log(1 + e^{\eta_i^+}) \right) - \frac{1}{2} \lambda \sum_{j=1}^{p} \beta_j^2$$

where the weights $\mathbf{W}^+$, a diagonal matrix with the diagonal elements:

$$w^+ = n\mu^+(1-\mu^+)$$

and the intermediate variable $\tilde{z^+}$:

$$\tilde{z^+} = \eta^+ + \frac{y^+ - \mu^+}{\mu^+(1-\mu^+)}$$

As mentioned on previous chapter, the choice of penalty weight λ is important. The Poisson ridge regression and the logistic ridge regression will be optimized by the Schall's algorithm. The performance of algorithm will be compared to other model selection methods such as Akaike information criterion (AIC), Bayesian information criterion (BIC) and generalized cross validation (GCV).

3.3 Applications

The performance of the proposed method is investigated on real-life data and simulation. First, Schall's algorithm will be applied for a pattern of terrorism data in Afganistan between 1994-2008 (Piazza, 2012). Second, data with non-correlated covariates from Poisson distribution and binomial distribution will be generated. The simulation is designed in such a way to produce a data set where no penalty would be required. Finally, data set with correlated covariate from Poisson distribution and binomial distribution will be generated. The performance of the Schall's algorithm will be compared with AIC, BIC and GCV.

3.3.1 Example

A pattern of terrorism in Afganistan between 1994-2008 would be modelled. Data is published in Piazza (2012). This data set is obtained from 34 provinces. The aim of the analysis is for examining the relationship between terrorism in Afganistan and the opium trade, various economic development, infrastructure, geographic, security, and cultural factors. The response (y) is the total terrorism incidents, and has median 22 incidents.

The predictor variables are the average annual opium cultivation (opium in hectares), area (in hectares), mountainous (in %), literacy rate (literacy in %), access to drinking water (water in %), below minimum calories(calories in %), all-season roads (roads in %), under five mortality (mortality, out of 1000), Pashtun majority (majority, 1=Yes, 0=No), and the mean of foreign troops (troops in yearly). Median of the average annual opium cultivation, and foreign troops are 594 hectares and 4256 soldiers. Median of percentage mountainous, literacy rate, access to drinking water, below minimum calories, and all-season roads are 40.1 %, 17.5 %, 28 %, 28 %, and 43 %. There are sixteen provinces which is Pashtun majority (47 %). In this analysis, covariates are scaled to zero mean and unit standard deviation.

The first model applied is a simple Poisson regression. The results in Table 3.1 shows that all covariates are significant. However, these are highly correlated data. Opium has a correlation with area (0.48), mountainous has a correlation with all-season roads (-0.65) and acces to drinking water (-0.68), below minimum calories has a correlation with literacy rate (-0.46), and majority has a correlation with acces to drinking water (0.49).

	Coef	St. Err	Pr(> |z|)
Counts	0.3703	0.0125	< 0.001 ***
Opium	-0.1275	0.0079	< 0.001 ***
Area	-0.2716	0.0124	< 0.001 ***
Mountainous	0.2890	0.0178	< 0.001 ***
Literacy	-0.0886	0.0112	< 0.001 ***
Water	0.1264	0.0177	< 0.001 ***
Calories	0.2162	0.0114	< 0.001 ***
Roads	0.2503	0.0140	< 0.001 ***
Mortality	-0.0599	0.0123	< 0.001 ***
Majority	-0.2075	0.0138	< 0.001 ***
Troops	0.0992	0.0121	< 0.001 ***

Table 3.1: *The simple Poisson regression results show that all covariate has a significant P-values. In column 2 and 3, coefficients and standard errors of each variables are given. The last column shows that each variable has a significant P-values (less than* 0.001*). However, they are highly correlated data. Opium has a correlation with area (0.48), mountainous has a correlation with all-season roads (-0.65) and acces to drinking water (-0.68), below minimum calories has a correlation with literacy rate (-0.46), and majority has a correlation with acces to drinking water (0.49).*

In order to get the optimal model for this data, a Poisson penalized ridge regression is used. For model selection, Akaike criterion (AIC), Bayesian criterion (BIC), generalized cross validation (GCV) and Schall's algorithm are used.

The optimal coefficients from three criterions and Schall's algorithm can be seen in Figure 3.1. Schall's algorithm gives $\lambda = 95.78$. For one terrorism incident, opium, area, mountainous, literacy, water, calories, roads, mortality, majority, and troops contribute 0.0018, 0.0023, -0.0234, 0.0510, 0.0687, 0.0367, 0.0539, -0.0440, 0.0395, and 0.0258. The best coefficients from Schall's algorithm are located in the middle of three other criterions. Schall's algorithm has a simpler algorithm than other criterions because the weight of penalty λ is estimated and the iteration, which are done before convergence is usually less than five iterations.

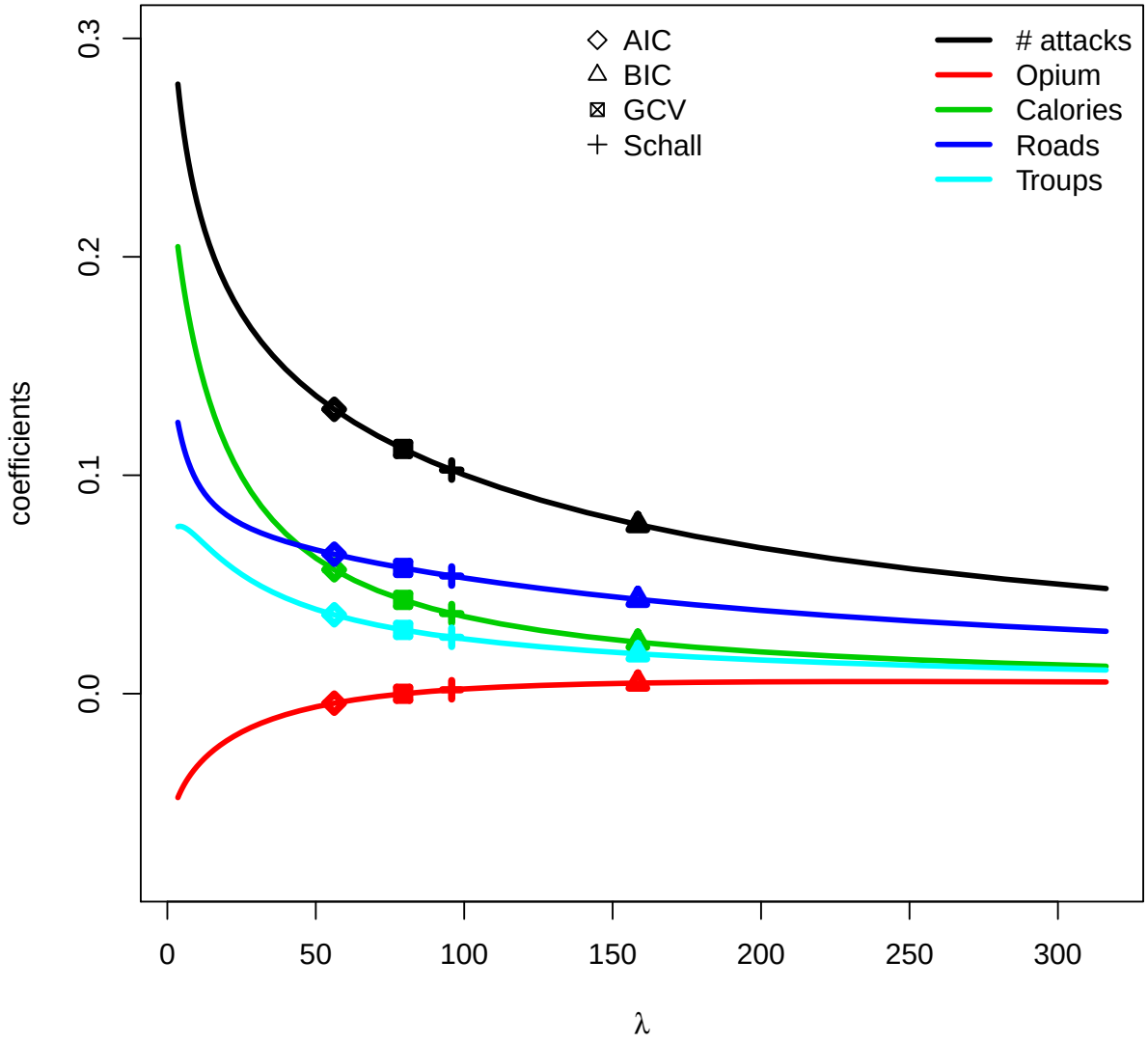

Figure 3.1: *The coefficients of five covariates are shrunk to zero. AIC, BIC, GCV and Schall's algorithm give different optimal fit. The optimal coefficients from Schall's algorithm (+) ($\lambda = 95.78$) are located in the middle of three other criterions.*

3.3.2 Datasets Simulations for non-correlated covariate

In this subsection, data sets with non-correlated covariate will be generated. Penalized regression will be applied on them, and some model selections will be used for choosing

the best model.

3.3.2.1 A non-correlated Poisson regression model

The data set with a sample size of 500 is generated. The data set has four covariates x_1, x_2, x_3, x_4 where each covariate has a standard normal distribution and are independent of each other. The response variable is random Poisson with a parameter equal to $\exp(\mathbf{X}\boldsymbol{\beta})$ where $\boldsymbol{\beta} = (1, -0.4, 0.7, 0.2)$. 1000 samples are generated and analyzed.

Poisson and Poisson ridge models are fitted the simulated data. According to Table 3.2, it can be seen that there are no differences between the different methods. Therefore, for non-correlated data, the simple Poisson regression analysis is enough.

3.3.2.2 A non-correlated logistic regression model

The data is generated data with four independent covariate x_1, x_2, x_3, x_4 from a standard normal distribution and are independent of each other. The response variable is a random binomial with a parameter equal to $\exp(\mathbf{X}\boldsymbol{\beta})/(1 + \exp(\mathbf{X}\boldsymbol{\beta}))$, where $\boldsymbol{\beta} = (1, -0.4, 0.7, 0.2)$. There are 1000 samples with different sample size: 400, 450, 475, 500, and 1000.

The data set is analyzed by logistic regression and logistic ridge regression. The results are displayed in Table 3.3. It can be seen that MLE is quite better than penalized regression (The value of MSE is the smallest). According to the theory, non-correlated covariate data doesn't need ridge regression.

	TRUE	Schall	AIC	BIC	GCV	mgcv	MLE
x_1	1	0.997	0.999	0.999	0.998	1.001	0.999
x_2	-0.4	-0.396	-0.397	-0.397	-0.397	-0.398	-0.397
x_3	0.7	0.698	0.699	0.699	0.699	0.701	0.699
x_4	0.2	0.199	0.199	0.199	0.199	0.200	0.199
MSE		2569.776	2569.776	2569.776	2569.776	2569.776	2569.267
mpb		0.004	0.003	0.003	0.004	0.002	0.003
computation time		0.012	1.369	1.369	1.369	0.013	0.003

Table 3.2: *The average of coefficients of Poisson regression using ridge regression for non-correlated data. Schall's algorithm, AIC, BIC, GCV and MLE doesn't give different MSE and mean percentage of bias (mpb) value. So the simple Poisson regression analysis is enough.*

3.3.3 Datasets Simulations for correlated covariates

In this subsection, the algorithm will be applied to the simulation. The data will be generated with the correlation coefficients 0.90, 0.95 and 0.99. The aim of this experiment is illustrating the performance of Schall's algorithm for estimating the penalty weight even for highly correlated designs.

3.3.3.1 A correlated Poisson regression model

The simulation is generated for a correlated data with four random covariates. Each covariate has a correlation with other covariates with the same value of correlation, i.e. 0.90, 0.95, and 0.99. The response variable is random Poisson with a parameter equal to $\exp(\mathbf{X}\boldsymbol{\beta})$ where $\boldsymbol{\beta} = (-0.309, 0.7503, 0.301, -0.501)$. Sample sizes are 20, 30, 50, and 80. 1000 samples are generated and analyzed with Poisson ridge regression. From the Table 3.4, it can be seen that MSE that resulted from Schall's algorithm are the smallest among other criterions.

n		OLS	Schall	AIC	GCV	BIC
400.0	λ	0.0	2.4	0.0	309.6	79.0
	MSE	0.063	0.059	0.063	1.119	0.248
450.0	λ	0.0	2.5	0.0	356.2	51.0
	MSE	0.056	0.055	0.056	1.133	0.198
475.0	λ	0.0	2.4	0.0	372.0	18.1
	MSE	0.059	0.053	0.059	1.119	0.114
500.0	λ	0.0	2.5	0.0	378.9	3.4
	MSE	0.049	0.048	0.049	1.115	0.065
1000.0	λ	0.0	2.4	0.0	756.8	0.0
	MSE	0.029	0.028	0.029	1.114	0.029

Table 3.3: *MSE from non-correlated logistic data for different sample sizes i.e. 400, 450, 475, 500 and 1000. The value of MSE for MLE is small. So the simple logistic regression analysis is enough.*

	OLS	Schall	AIC	GCV
rho=0.90				
20	1.539	0.770	1.361	0.959
30	0.838	0.572	0.878	0.905
50	0.465	0.364	0.549	0.765
80	0.287	0.284	0.313	0.584
rho=0.95				
20	3.009	1.086	2.061	0.997
30	1.758	0.835	1.452	0.972
50	1.031	0.600	1.052	0.924
80	0.598	0.450	0.684	0.811
rho=0.99				
20	18.079	1.617	7.705	1.042
30	9.960	1.598	5.506	1.126
50	5.426	1.276	3.785	1.096
80	3.290	1.069	2.521	1.113

Table 3.4: *MSE from correlated count data in different correlation coefficients (0.90, 0.95, and 0.99) and different sample sizes(20, 30, 50, and 80). MSE which is resulted from Schall's algorithm are the smallest ($\rho = 0.90$ and $\rho = 0.95$). For $\rho = 0.99$, MSE from Schall's algorithm and GCV give similar performance.*

3.3.3.2 A correlated logistic regression model

In this subsection, a correlated data will be generated with four random covariates that have a multivariate normal distribution with mean $\mu = 0$ and constant standard deviation $\sigma = 1$. The response variable with random binomial distribution with parameter equal

to $\exp(\mathbf{X}\boldsymbol{\beta})/(1+\exp(\mathbf{X}\boldsymbol{\beta}))$ where $\boldsymbol{\beta} = (-0.309, 0.7503, 0.301, -0.501)$. Each covariate has a correlation with other covariate. Every correlation has the same coefficient, i.e. 0.90, 0.95, and 0.99. Sample sizes are 20, 30, 50, 80, and 150. 1000 samples are generated and analyzed with logistic ridge regression.

n	rho	OLS	Schall	AIC	GCV	BIC	mgcv
20	0.9	31479.4386	0.6721	2.4014	0.7701	0.9866	31479.4386
30	0.9	6546.1637	0.5398	5.6852	0.7460	0.9804	6546.1637
50	0.9	0.4867	0.3036	0.9177	0.7281	0.9676	0.4867
80	0.9	0.2410	0.1947	0.7340	0.7093	0.9490	0.2410
150	0.9	0.1154	0.0995	0.2054	0.6800	0.8854	0.1154
20	0.95	19132.3881	0.7040	2.4466	0.7704	0.9878	19132.3881
30	0.95	5420.5156	0.5709	4.6707	0.7443	0.9820	5420.5156
50	0.95	0.5177	0.3284	0.9316	0.7262	0.9707	0.5177
80	0.95	0.2559	0.2080	0.7710	0.7049	0.9537	0.2559
150	0.95	0.1243	0.1079	0.2470	0.6741	0.9012	0.1243
20	0.99	15764.5653	0.7784	2.0702	0.7714	0.9899	15764.5653
30	0.99	5908.4778	0.6055	1.7492	0.7463	0.9851	5908.4778
50	0.99	0.5557	0.3644	0.9488	0.7228	0.9754	0.5557
80	0.99	0.3034	0.2439	0.8392	0.6989	0.9612	0.3034
150	0.99	0.1420	0.1242	0.3455	0.6650	0.9230	0.1420

Table 3.5: *MSE from correlated binomial data in different correlation coefficients (0.90, 0.95, and 0.99) and different sample sizes(20, 30, 50, 80, and 150). MSE which is resulted from Schall's algorithm are the smallest*

Logistic ridge regression has been applied for the data sets. The selection methods: AIC, GCV, BIC, and the proposed methods, Schall's algorithm are compared. The MSE values can be seen on Table 3.5. It can be seen that all MSE that resulted from Schall's algorithm is the smallest among other criterions.

3.4 Summary

Penalized Poisson ridge regressions give a better result for a correlated data such as terrorism data in Afganistan. The optimized fit using Schall's algorithm gives coefficients

with the right sign for correlated covariates. Mountainous has a different sign of coefficient (-), as we know before, it has a negative correlation with roads and water and the value of coefficient for roads and water are positive. Majority also has a different sign of coefficient (+), and it has a positive correlation with water.

In order to know the performance of algorithm, MSE was calculated. MSE for penalized regression using Schall's algorithm from correlated Poisson datasets and correlated logistic datasets are the lowest compare with MSE from other criterions.

Chapter 4

Generalized Additive Models

4.1 Introduction

The linearity assumption is violated for some applications. For example, `mcycle` data set consists of 133 observations with a series of measurement of head acceleration in a simulated motorcycle accident, used to test crash helmet. Based on Figure 4.1, it is obvious that a simple linear regression is not the best model for this dataset. We try to use a polynomial regression for this. In this case, the linear regression model sometimes produce incorrect values. Transformation or higher-degree polynomials can be used , but this needs a good deal of expertise and time.

Figure 4.2 shows the polynomial regression can fit to mcycle dataset for degree five, ten, fifteen and twenty. As the degree is higher, the fit is more sensitive. It can be seen there are unexpected wiggles. Under 15 ms, the datasets give constant acceleration but the polynomial regressions is oscillatory between the data points and it also can be seen

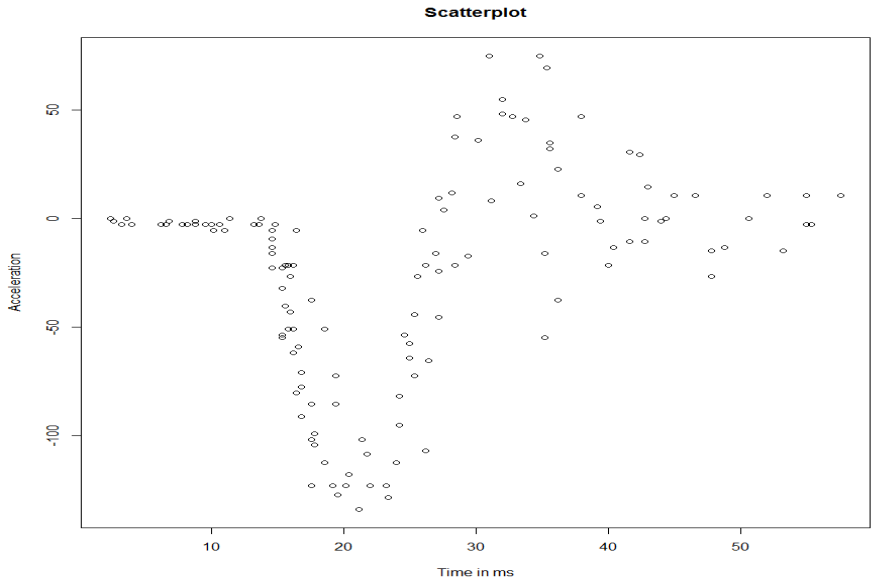

Figure 4.1: *Scatterplot of the motor-cycle impact data. It can be seen that a simple linear regression is not the best model.*

above 50 ms, for the polynomial regressions degree 20, the curve is not representing what happened on the data set.

Some methods have been developed for smoothing a scatterplot, for example: using a local weighting scheme (Cleveland, 1979), and the spline smoothing (Silverman, 1985; Craven and Wahba, 1978; De Boor, 1972). Generalized additive models (GAMs) give a solution for this kind of data (Hastie and Tibshirani, 1986). GAMs replace the linear combination with the respondent variable in GLMs with a sum of smooth functions of covariates. In this chapter, the definition of GAMs, penalized splines (P-splines), optimal smoothing, GAMs with P-splines and the application will be discussed.

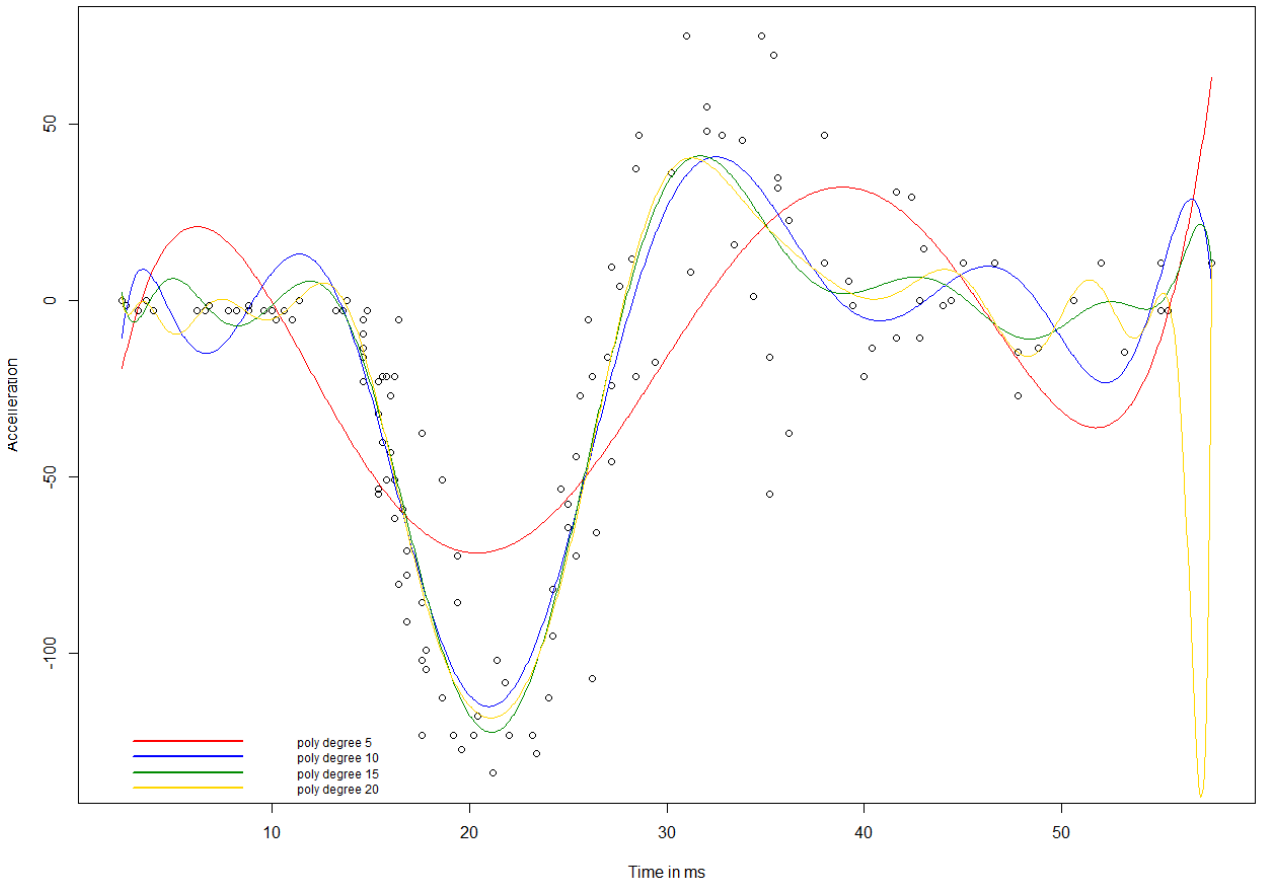

Figure 4.2: *Polynomial regressions are applied to* `mcycle`*. It can be seen that as the degree is higher, the fit is more sensitive. The polynomial regression degree 20 above 50 ms does not represent what happened on the data set.*

In Section 4.2, b-splines basis function will be explained. Next, in Section 4.3, penalized splines (P-splines) will be explained. After that, in Section 4.4, GAMs with P-splines (P-GAMs) will be discussed. In Section 4.5, the Schall algorithm for P-GAMs will be discussed. In Section 4.6, the algorithm will be applied to a datasets. Finally, Section 4.8 is the chapter summary.

4.2 B-spline basis functions

There are two properties on B-spline basis functions i.e., the domain is divided by knots and each basis function degree k, $B_{j,k}(x)$ (j-th basis function degree k), are zero on the entire interval except on a few adjacent subintervals ($k+1$ subintervals or $k+2$ knots). As a result, B-splines basis functions are strictly local.

Suppose a set of data $\{\boldsymbol{x}, \boldsymbol{y}\}$, where $\boldsymbol{x}$ is the independent variable and $\boldsymbol{y}$ is the dependent variable with n observations. The set $\boldsymbol{t} = \{t_1, t_2, \ldots, t_{k+(q+1)}\}$, called the knot vector which $t_j < t_{j+1}$, is defined to obtain a q parameter B-spline basis. A B-spline degree k (order $k+1$) can be presented as:

$$f(x) = \sum_{j=1}^{q} B_{j,k}(x) a_j$$

where q is a number of parameter B-spline basis and a_j are B-splines coefficients and can be viewed as the amplitudes of B-splines. Degree k must be $1 \leq k \leq q+1$. The shape of the basis functions is only dependent on the knot spacing. The positions of knots and the degree can modified to change the shape of a B-spline basis curve.

The q parameter B-spline basis functions degree k, $B_{j,k}(x)$, are most easily defined recursively referred to as the Cox-de Boor recursion formula (De Boor, 1972):

$$B_{j,k}(x) = \frac{x - t_j}{t_{j+k} - t_j} B_{j,k-1}(x) + \frac{t_{j+k+1} - x}{t_{j+k+1} - t_{j+1}} B_{j+1,k-1}(x)$$

where

$$B_{j,0}(x) = \begin{cases} 1 & , t_j \leq x < t_{j+1} \\ 0 & , \text{else} \end{cases}$$

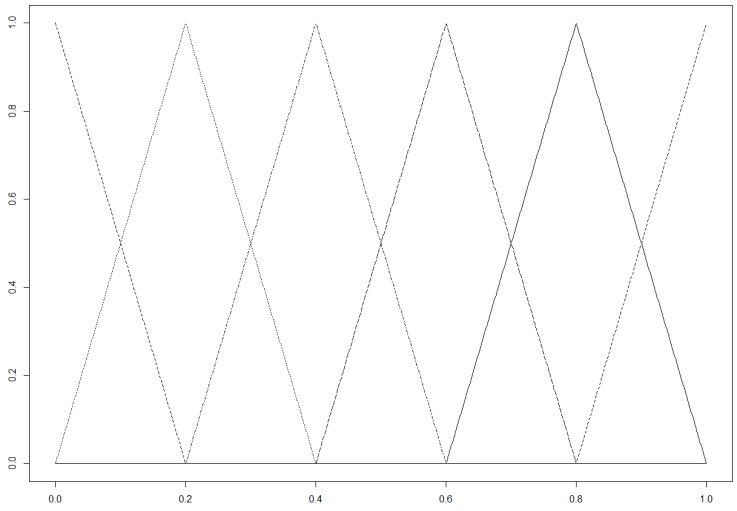

Figure 4.3: *Illustration of B-spline bases degree 1 with knot sequence* $t = \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$

For example, the B-splines bases with degree 1 with knot sequence $\boldsymbol{t} = \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ can be seen in Figure 4.3. One basis function consists of two linear pieces. It is defined on two subintervals (three knots); one piece from t_j to t_{j+1} and one piece from t_{j+1} to t_{j+2}. The knots are t_j, t_{j+1} and t_{j+2}. A basis function $B_{1,1}$ is a non-zero function on the interval $[t_1, t_3] = [0, 0.4]$. Outside the interval $[0, 0.4]$, a basis function $B_{1,1}$ is a zero function.

The B-splines bases with degree 2 with same knot sequence can be seen in Figure 4.4. One basis function is defined on three subintervals (four knots). A basis function $B_{1,2}$ is a non-zero function on the interval $[t_1, t_4] = [0, 0.6]$. Outside the interval $[0, 0.6]$, a basis function $B_{1,2}$ is a zero function.

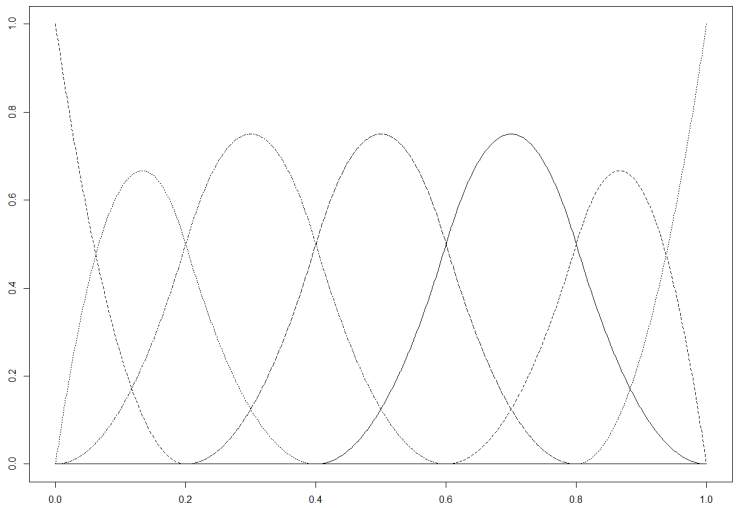

Figure 4.4: *Illustration of B-spline bases degree 2 with knot sequence* $\boldsymbol{t} = \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$

Taking into account the regression of n data points (x_i, y_i) on the set q B-splines $B_j(.)$. The fit of the data can be expressed by the sum of squared errors (SSE):

$$S = \sum_{j=1}^{n} \left(y_i - \sum_{i=1}^{q} B_j(x) a_j \right)^2$$

a_j can be estimated using an iterative method of scoring for GLM. The good fit of data is indicated by low S. The smoothness of the curve will depend on the number of B-splines and the value $\boldsymbol{a}$. If a_j for all as are nearly equal, next the function will be constant. If a_j vary wildly, then the function will be unstable.

4.3 Penalized splines (P-splines)

B-splines have stable numerical properties, but the user has to decide the number and the position of knots. The number of knots influence the fit, too many knots give an overfit model and too few knots give an underfit model. Penalized B-splines (P-splines) introduce

a penalty on roughness of a while using a B-spline with a large number of knots (Eilers and Marx, 1996). P-splines combine a B-spline and a difference penalty. The position of knots usually are defined as equally spaced knots.

$$S^* = \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{q} B_j(x) a_j \right)^2 + \lambda \sum_{j=k+1}^{q} (\Delta_d a_j)^2 \tag{4.1}$$

where Δ_d is the finite-order differences of the coefficients of adjacent B-splines and λ is a penalty weight. The first order differences can be written as:

$$\sum_{j=1}^{q-1} (\Delta a_j)^2 = \sum_{j=1}^{q-1} (a_{j+1} - a_j)^2 = |\mathbf{D}_1 \boldsymbol{a}|^2 = a_1^2 - 2a_1 a_2 + 2a_2^2 - 2a_2 a_3 + \ldots + a_q^2$$

This can be written in matrix form as:

$$(\mathbf{D}_1)'\mathbf{D}_1 = \begin{bmatrix} -1 & 0 & 0 & 0 \ldots 0 \\ 1 & -1 & 0 & 0 \ldots 0 \\ 0 & 1 & -1 & 0 \ldots 0 \\ \vdots & & & \\ 0 & 0 & 0 & 0 \ldots 1 \end{bmatrix} \begin{bmatrix} -1 & 1 & 0 & 0 \ldots 0 \\ 0 & -1 & 1 & 0 \ldots 0 \\ 0 & 0 & -1 & 1 \ldots 0 \\ \vdots & & & \\ 0 & 0 & 0 & 0 \ldots 1 \end{bmatrix}$$

The second order difference is:

$$\Delta_2 a_j = \Delta(\Delta a_j) = (a_j - a_{j-1}) - (a_{j-1} - a_{j-2}) = a_{j-2} - 2a_{j-1} + a_j.$$

So the second order difference operator Δ_2 can be represented in matrix as:

$$\underset{(q-2)\times q}{\mathbf{D}_2} = \begin{bmatrix} 1 & -2 & 1 & 0\ldots 0 & 0 & 0 \\ 0 & 1 & -2 & 1\ldots 0 & 0 & 0 \\ 0 & 0 & 1 & -2\ldots 0 & 0 & 0 \\ \vdots & & & & & \\ 0 & 0 & 0 & 0\ldots 1 & -2 & 1 \end{bmatrix}$$

The d-order difference operator $\Delta_d()$ ($\mathbf{D}_d$) can be called out with `R-code` by:

`D`=diff(diag(n),diff=d).

The coefficients is estimated from minimizing S^* in 4.1:

$$\hat{\boldsymbol{a}} = (\mathbf{B}'\mathbf{B} + \lambda\mathbf{P})^{-1}\mathbf{B}'\boldsymbol{y}$$

where $\mathbf{B}$ is a matrix consists of the elements $b_{\bullet j} = B_{jk}(\boldsymbol{x})$, the jth B-splines function and $\mathbf{P}$ is the sum of squares of differences, $\mathbf{P} = \mathbf{D}_d'\mathbf{D}_d, d = 0, 1, 2, \ldots$.

4.4 Univariate Smoothing with GAMs with P-splines(P-GAMs)

Hastie and Tibshirani (1986) introduced generalized additive models (GAMs) in order to cover nonlinear covariate effects. They proposed to change a linear form $\mathbf{X}\boldsymbol{\beta}$ in a GLM with a sum of smooth functions of the explanatory variables $\sum s_i(x_i)$. The GAMs have the

form:

$$g(E(\boldsymbol{y}|\mathbf{X})) = \sum_{i=1}^{p} s_i(\boldsymbol{x}_{ik}) \tag{4.2}$$

where p is the number of covariates, $\boldsymbol{x}_{ij}$ is k-th observation for i-th covariates. In this chapter, the univariate smoothing will be examined. Let a GAM model containing one smooth function of one covariate,

$$y_k = s(x_k) + \epsilon_i \tag{4.3}$$

where y_k is a response variable, x_k a covariate, s a smooth function and the ϵ_k are i.i.d $\mathcal{N}(0, \sigma^2)$ random variables.

The function $s()$ can be estimated by choosing a basis, defining the space of functions of which s (or a close approximation to it) is an element. Marx and Eilers (1998) proposed GAMs with P-splines (termed P-GAMs) which has $f_j = \mathbf{B_{jk}}\boldsymbol{a}_j$ as the jth GAM component where $\mathbf{B_{jk}}$ is the B-spline matrix (with q_j knots) of dimension $m \times q_j$ and $\boldsymbol{a}_j$ is the vector of coefficient associated with the B-spline bases. The smoothness can be achieved from the fit to the data which can be expressed by the sum of squared differences

$$S^{**} = \sum_{i=1}^{m} \left(y_i - \sum_{j=1}^{q} b_{ij} a_j \right)^2 .$$

4.5 Optimal Smoothing

In order to regularize the smoothness and avoid knot selection scheme, P-splines recommend using a large number of equally space knots (Eilers and Marx, 1996). The estimate

coefficient $\boldsymbol{a}$ is obtained from an iterative technique:

$$\hat{\boldsymbol{a}}_{t+1} = (\mathbf{B}'\hat{\mathbf{W}}_t\mathbf{B} + \lambda\mathbf{P})^{-1}\mathbf{B}'\hat{\mathbf{W}}_t\hat{\boldsymbol{z}}_t$$

until convergence, where $\hat{\mathbf{W}}_t$ and $\hat{\boldsymbol{z}}_t$ are the weight matrix and adjusted dependent vector used in GLM estimation. A difference penalty is applied for smoothing splines.

Besides the number and the degree of B-spline basis function, the smoothness of an estimated curve on generalized additive models using P-splines is influenced by the weight of penalty. The optimal weight of penalty can be obtained by some methods. Marx and Eilers (1998) proposed to use information criterion (IC) to get the optimal weight of penalty:

$$\text{IC} = \text{dev}(\boldsymbol{y}; \mathbf{a}, \lambda) + \delta\text{trace}(\hat{\mathbf{H}})$$

where $\hat{\mathbf{H}} = \mathbf{B}(\mathbf{B}'\hat{\mathbf{W}}\mathbf{B} + \lambda\mathbf{P})^{-1}\mathbf{B}'\hat{\mathbf{W}}$. The estimated effective dimension ED can be obtained from trace($\hat{\mathbf{H}}$). It is more efficiently computed using trace($\hat{\mathbf{H}}$) = trace$(\mathbf{B}'\hat{\mathbf{W}}\mathbf{B} + \lambda\mathbf{P})^{-1}$.

The Schall's algorithm will be proposed for selecting the optimal weight of penalty. It has the following steps:

1. For given $\hat{\sigma}^2, \hat{\lambda}$ estimate the coefficient $\hat{\mathbf{a}}$ by:

$$\hat{\mathbf{a}} = (\mathbf{B}'\hat{\mathbf{W}}\mathbf{B} + \lambda\mathbf{P})^{-1}\mathbf{B}'\hat{\mathbf{W}}\hat{\boldsymbol{z}}$$

2. Given estimates of coefficients $\hat{\mathbf{a}}$, variance estimators are obtained from

$$(\mathbf{B}'\hat{\mathbf{W}}\mathbf{B} + \lambda\mathbf{P})\hat{\mathbf{a}} = \mathbf{B}'\hat{\mathbf{W}}\hat{\boldsymbol{z}}$$

$$\hat{\sigma^2} = \frac{(\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{\beta}})'(\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{\beta}})}{n - \mathrm{ED}}$$

and

$$\hat{\tau^2} = \frac{\hat{a}'\mathbf{D}_d'\mathbf{D}_d\hat{\boldsymbol{a}}}{\mathrm{ED}}$$

where ED stands for effective dimensions and is the trace of the hat matrix of the mode (Hoaglin and Welsch, 1978). An estimate of the penalty weight $\hat{\lambda}$ can be then given by:

$$\hat{\lambda} = \frac{ED}{\hat{\boldsymbol{a}}'\mathbf{D}_d'\mathbf{D}_d\hat{\boldsymbol{a}}}$$

3. Iterate until the estimated penalty weight $\hat{\lambda}$ convergence.

4.6 Application

A data set consists of 133 observations with a series of measurement of head acceleration in a simulated motorcycle accident, used to test crash helmet. Time is measured in milliseconds after impact and head acceleration. In this section, mcycle data set will be used. It is used by Silverman (1985) for giving understanding about the spline smoothing. The data set is obtained from Schmidt et al. (1981).

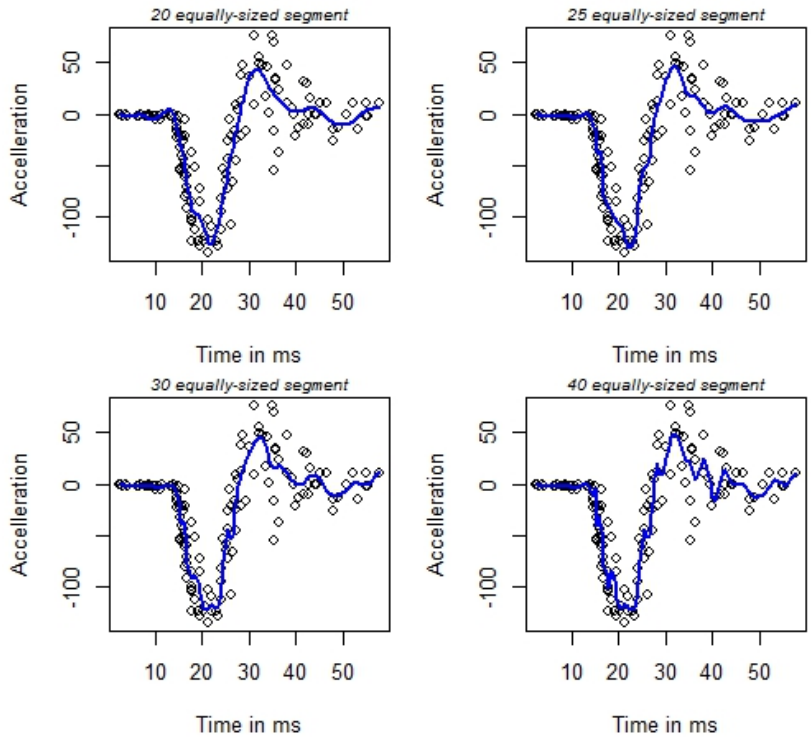

Figure 4.5: *B-spline regressions with different number of knots. Upper left, the fit is resulted from B-splines with 20 knots. Upper right, the fit is resulted from B-splines with 25 knots. Lower left, the fit is resulted from B-splines with 30 knots. Lower right, the fit is resulted from B-splines with 30 knots. As bigger the number of knots, the fit is more wavy*

An alternative method to solve this problem is B-spline basis regression. Regression fit for cubic spline with 20, 25, 30, and 40 equally-sized segment can be seen in Figure 4.5. It can be seen that for more knots, the curve is more wavy.

In order to solve the problem in choosing the number of knots, P-GAMs with P-spline is applied and the optimal smoothing used Schall's algorithm. The result with 40 knots can be seen in Figure4.6. The curve which is resulted from P-spline is more smooth than the curve which is resulted from B-spline.

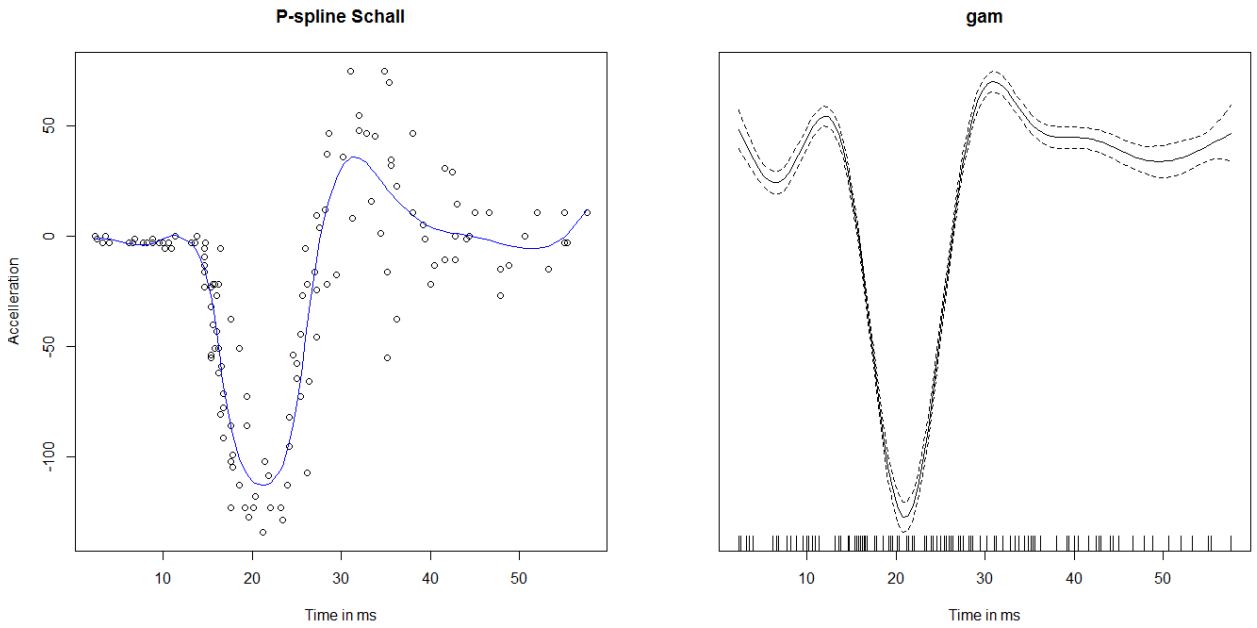

Figure 4.6: *The curve is resulted from P-spline with 40 knots. The optimal weight of penalty of P-spline regression is selected automatically using Schall's algorithm.*

4.7 Simulation

Schall's algorithm will be applied for smoothing a simulated data set. The data set consists of 200 observations. The predictor has a standard normal distribution ($\mathcal{N}(0,1)$) and the response variabel is $y = b_0 + b_1 * sin(2*x)^3 + e^2$ where $(b_0, b_1) = (5, 10)$ and e is an error with a normal distribution $\mathcal{N}(0, h(x))$. $h(x)$ is a function which performs heteroscedasticity function:

$$h(x) = 1 + 0.1x.$$

The plot of the data set can be seen on Figure 4.7.

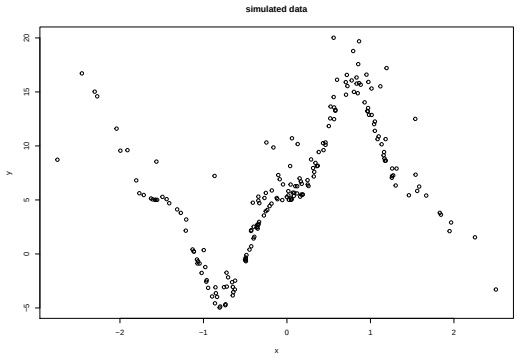

Figure 4.7: *Scatterplot of the simulated data. It can be seen that a simple linear regression is not the best model.*

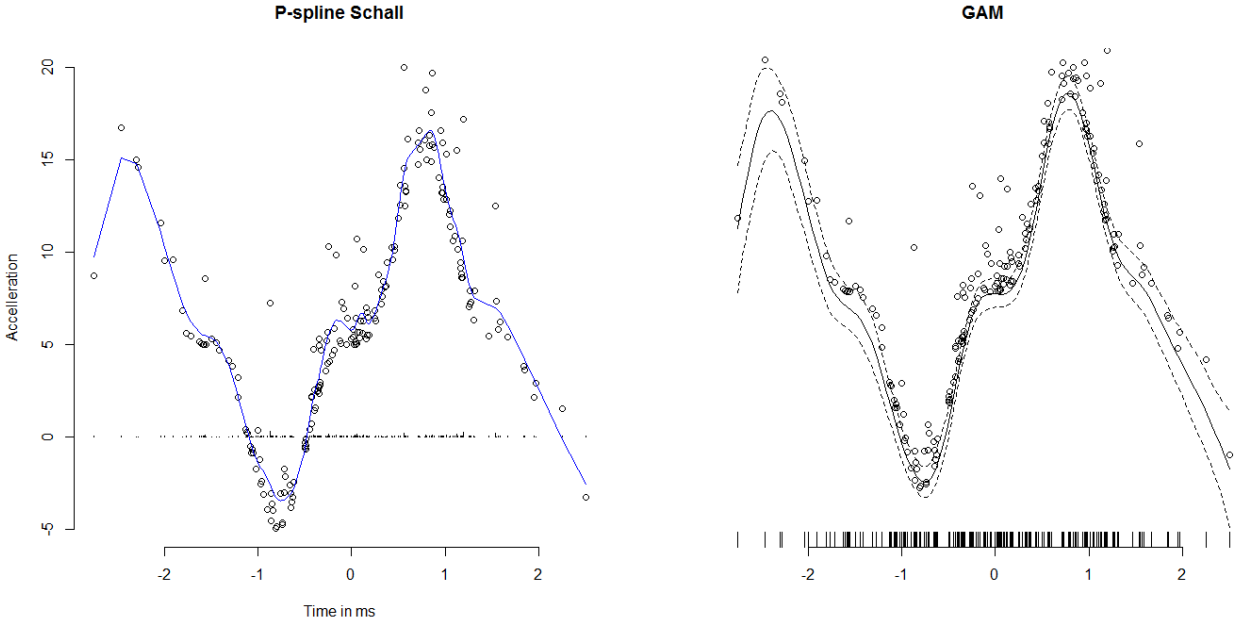

Figure 4.8: *Left: the smoothing result from optimized P-GAM.;Right: the smoothing result from tensor product (`package mgcv`).*

P-GAM using Schall's algorithm as automatic optimization will be applied and the result will be compared with smoothing using tensor product. The result is quite similar using the same number of knots (40 knots). Computation time of P-GAM using Schall's algorithm (0.20 second)is shorter than Computation time of tensor product smoothing (0.86 seconds).

4.8 Summary

Generalized Additive Models can be presented with generalized linear models. In order to get GLMs form, smooth functions in GAMs are replaced by P-splines. If in GLMs, there is linear combination of covariates then in GAMs with P-splines (P-GAMs), there is the linear combination of basis functions.

P-GAMs have some advantages such as GAM estimation is reduced to (generalized) linear regression with a manageable penalty; the system of equations is a low dimension and easy to solve; all the smooths are estimated simultaneously; the resulting GAM fit is compactly summarized by relatively few numbers of parameters that facilitate future prediction and standard errors, and regression diagnostics can be computed with relative ease (Marx and Eilers, 1998).

The weight of penalty choosing is the important steps in penalized regression. The Schall's algorithm is applied for choosing the optimal weight of penalty. This algorithm uses iterative to find the best model. It usually only needs a few number of iteration.

Chapter 5

Lasso Regression

5.1 Introduction

The lasso is a commonly used method for a variable selection. The method uses L_1 penalization to shrink estimates. It is often used on high dimensional data, not only to solve high dimensionality problem but also as a variable selection methods.

In this chapter we will propose a method for optimising a penalty weight. The novelty of our approach has to do with re-writing the lasso L_1 penalty as an L_2 type ridge penalty. Having done that, we will be able to optimise the weight using similar approaches to those of previous chapters. The procedure is a combination of a ridge regression approximation (Tibshirani, 1996), and a sum of absolute values approximation (Schnabel and Eilers, 2013). The algorithm will be applied to prostate data and simulation data.

The chapter is organized as follows: In section 5.2, the definition of the lasso regression will be explained. Later, section 5.3 will describe the commonly used computation for the

lasso regression. The proposed algorithm will be described in Section 5.4. Next, Section will apply the algorithm for some data set, real and simulation. Finally, Section 5.6 give a summary for this chapter.

5.2 Definition

The lasso (Least Absolute Shrinkage and Selection Operator) is a regularized regression method with an L_1-norm penalty. It was proposed by Tibshirani (1996). Where the ridge regression uses the sum of squared coefficients as a penalty, the lasso uses the sum of the absolute value of coefficients, such that:

$$\ell^*(\boldsymbol{\beta}) = \sum_{i=1}^{n} \ell_i(\boldsymbol{\beta}) - \lambda \sum_{j=1}^{p} |\beta_j|$$

The lasso coefficient estimates $\hat{\boldsymbol{\beta}}$ can be presented as

$$\hat{\boldsymbol{\beta}} = \text{argmax} \left\{ \sum_{1}^{n} \ell_i(\boldsymbol{\beta}) - \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{5.1}$$

or, alternatively, in matrix notation:

$$\hat{\boldsymbol{\beta}} = \text{argmax}\{\boldsymbol{\ell}(\boldsymbol{\beta}) - \lambda \|\boldsymbol{\beta}\|_1\}$$

The constraint shrinks coefficients and produces some coefficients that are exactly zero. That means that lasso also performs variable selection, not only shrinkage. As λ increases, the number of nonzero components of decreases.

5.3 Computation

Tibshirani (1996) used a quadratic programming for estimating lasso coefficients. Equation 5.1 is expressed as a least squares problem with 2^p inequality constraints, corresponding to the 2^p different possible signs for the β_js. Although 2^p may be very large, the problem can be solved with inequality constraints sequentially and trying to find a solution satisfying the Kuhn-Tucker conditions. The Kuhn-Tucker conditions for the lasso problem:

$$\mathbf{X}^T(y - \mathbf{X}\hat{\boldsymbol{\beta}}) = \lambda s, \tag{5.2}$$

where

$$s_i \in \begin{cases} \{\text{sign}(\hat{\boldsymbol{\beta}}_i)\} & \text{if } \hat{\boldsymbol{\beta}}_i \neq 0. \\ [-1, 1] & \text{if } \hat{\boldsymbol{\beta}}_i = 0. \end{cases} \text{, for } i = 1, \ldots, p. \tag{5.3}$$

$\hat{\boldsymbol{\beta}}$ is a solution in Equation 5.1 if and only if $\hat{\boldsymbol{\beta}}$ satiesfies Equation 5.2 and Equation 5.3 for some $\boldsymbol{s}$. Computation for lasso solutions with this procedure is however, expensive. The optimal weight of the penalty can be selected by generalized cross-validation. In the next section, we will propose an alternative algorithm.

The `R` package `penalized` implemented a method introduced by Goeman (2010). The algorithm improves the gradient-based algorithm by combining gradient ascent optimization and the Newton-Raphson algorithm in order to avoid the tendency to slow convergence. The package can be used for linear regression, logistic and Poisson regression as well as the Cox proportional hazards model. The optimal value of the tuning parameter λ is chosen by using cross-validation.

Next, the `R` package `glmnet` is developed by Friedman et al. (2009). The algorithm applies cyclical coordinate descent in a pathwise fashion. The idea of pathwise coordinate optimization is solving a sequence of single-parameter problems (β_j) with a fixed value penalty λ and holding the other parameters fixed at their current values. Equation 5.1 can be written as:

$$f(\tilde{\boldsymbol{\beta}}) = \frac{1}{2}\sum_{i=1}^{n}\left(y_i - \sum_{k\neq j} x_{ik}\tilde{\beta}_k - x_{ij}\tilde{\beta}_j\right)^2 + \lambda\sum_{k\neq j}|\tilde{\beta}_k| + \lambda|\tilde{\beta}_j|$$

where all the values of β_k for $k \neq j$ are held fixed at values $\tilde{\beta}_k(\lambda)$. The solution is:

$$\tilde{\beta}_j(\lambda) \leftarrow S\left(\tilde{\beta}_j(\lambda) + \sum_{i=1}^{n} x_{ij}\left(y_i - \tilde{y}_i\right), \lambda\right) \tag{5.4}$$

Here $S(t,\lambda) = \text{sign}(t)(t - |\lambda|)_+$. Iteration of 5.4 is repeated until convergence. The package can be implemented for linear regression, logistic and multinomial regression models, Poisson regression and the Cox model. The optimal model is chosen using cross validation.

5.4 The proposed algorithm

In his original paper, Tibshirani (1996), suggested computing the lasso estimate using an iterated ridge regression algorithm. He suggested writing the penalty $\sum|\beta_j|$ as $\sum\frac{\beta_j^2}{|\beta_j|}$. The lasso estimate $\tilde{\boldsymbol{\beta}}$ can be approximated by a ridge regression of the form $\boldsymbol{\beta}^* = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{W}^-)^{-1}\mathbf{X}^T y$ where $\mathbf{W}$ is a diagonal matrix with diagonal elements $|\tilde{\boldsymbol{\beta}}_j|$, $\mathbf{W}^-$ denotes the generalized inverse of $\mathbf{W}$. The number of effective parameters in the constrained fit $\tilde{\boldsymbol{\beta}}$ is

approximated by trace of the hat matrix.

$$p(\lambda) = \text{tr}\{\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{W}^-)^{-1}\mathbf{X}^T\}$$

Schnabel and Eilers (2013) approximated a sum of absolute values $S = \sum_j |\beta_j|$ as $\sum_j \frac{\beta_j^2}{\sqrt{\tilde{\beta}_j^2+\epsilon^2}}$, with ϵ a small number . This approximation is adapted from Schlossmacher (1973). Although the approach has been highlighted before, it was never really put into practice. There were no papers investigating the expression of L_1 penalization in an L_2 form. In this chapter, a ridge regression approximation (Tibshirani, 1996) and a sum of absolute values approximation (Schnabel and Eilers, 2013) will be combined. We got the form $\boldsymbol{\beta}^* = (\mathbf{X}'\mathbf{X} + \lambda\mathbf{W}^-)^{-1}\mathbf{X}'\boldsymbol{y}$ where $\mathbf{W}$ is a diagonal matrix with the diagonal elements $\sqrt{\tilde{\beta}_j^2 + \epsilon^2}$.

The result of the above lasso estimate, combination of a ridge regression approximation and a sum of absolute values approximation, still has ridge regression behaviour, the coefficients are not shrunk to be exactly zero. To force the coefficients to be zero, a thresholding scheme is applied to remove small $\hat{\boldsymbol{\beta}}$s. The thresholding scheme will kill covariates that are smaller than the standard deviation of all coefficients ($\hat{\beta}_j = 0$) (noises) and keeps some large covariate ($\hat{\beta}_j \neq 0$) (signals). The thresholding is defined as $\tilde{\beta}_j = \hat{\beta}_j^o \mathbf{I}(|\hat{\beta}_j^o| > \gamma)$ (Tibshirani, 1996).In this algorithm γ is chosen as $\gamma = \sum(\beta_j^0 - \hat{\beta}_j)^2/p$ and β_j^0 is coefficients for simple regression.

5.5 Application

5.5.1 Simulation

A set of 80 normally distributed variables $X_i \sim \mathcal{N}(0,1)$ with $i = 1,...,80$ on n=150 observations was simulated. Out of these features, only 20 of them were related to a normal y response, with coefficients simulated under a uniform distribution $\boldsymbol{\beta} \sim \mathcal{U}(-2.2, 2.2)$ .The final true model was of the form: $\boldsymbol{y} = \mathbf{X}\boldsymbol{\beta} + \sigma\boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}\mathcal{N}(0,2)$ is the Gaussian random noise added to the data and $\mathbf{X}$ is a matrix of size (150x20). Each dataset was repeated 1000 times, with four different models fitted within each step. The first approach was fitting the data with a lasso model optimized via the `penalized` package. Then, a second would fit the same lasso model optimised using `glmmnet` package. The last two models are based on the proposed algorithm of this chapter. We will use two different algorithms for optimising the penalty weight. One approach will optimise penalty by performing a grid search over different values of λs and choosing the one with the best GCV criterion. The other approach will be based on using Schall's algorithm for optimising the penalty. It should be noted that this approach starts with a λ value being 0, thus since there is no penalty at first step, none of the variables are dropped from the model.

Results are presented in the table below. The first column represents the fitting approach, second column present the average number of variables in the model and the third one present the average bias, measured as the sum of the absolute value of the estimate coefficients minus the true coefficients, divided by the total number of variables : Error= $\sum |\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}|/p$.

Approach	Variables in model	Error
penalized	31	0.078
glmnet	29	0.080
grid search	17	0.052
Schall	17	0.054

Table 5.1: *Number of variables in the model under different optimisation approach for 1000 repetitions of simulated data i.e.* `glmnet`, `penalized`*, proposed algorithm using grid search, and proposed algorithm using Schall's algorithm. The proposed algorithm using grid search, and Schall's algorithm give the smallest average bias.*

It is important to show that there are virtually no differences in results given by using Schalls approach or a grid search over λs, where results are very close together. Moreover, the new approach outperforms both penalized and glmnet approaches here.

5.5.2 Prostate Cancer Data

Four different approaches i.e. `glmnet`, `penalized`, proposed algorithm using grid search, and proposed algorithm using Schall's algorithm were used to optimise the penalty weight, in a model on the prostate cancer dataset. The data is obtained from men who were about to receive a radical prostatectomy Stamey et al. (1989). The relationship between the level of prostate specific antigen (the log of PSA) and a number of clinical measures will be examined. There are 97 observations and eight variables as predictors: lcavol:log cancer volume,lweight: log prostate weight, age in years, lbph: log of the amount of benign prostatic hyperplasia, svi: seminal vesicle invasion, lcp: log of capsular penetration, gleason a numeric vector and pgg45: percent of Gleason score 4 or 5. The data have been standardised to 0 mean and 1 standard deviation. The results of the different models are given in table (5.2).

The different approaches gave quite similar results. It has to be noted that optimisation via Schall's algorithm tends to penalise less than the other approaches in this instance.

covariate	glmnet	penalized	proposed+grid search	proposed+schall
lcavol	0.519	0.520	0.517	0.582
lweight	0.205	0.208	0.201	0.228
age	-0.060	-0.067	-0.050	-0.135
lbph	0.081	0.085	0.078	0.122
svi	0.213	0.215	0.211	0.270
lcp	0	0	0	-0.127
gleason	0.003	0.005	0	0
pgg45	0.058	0.058	0.055	0.130

Table 5.2: *Coefficient estimates under four different approaches i.e.* `glmnet`, `penalized`, *proposed algorithm using grid search, and proposed algorithm using Schall's algorithm. The proposed algorithm using Schall's algorithm penalized less than others.*

5.5.3 Microarray data set

The proposed algorithm is applied to gene expression data. The data contains information on 120 rates and theirs gene profiles. Here, 200 gene probes are used as predictors. Out of a total of 200 genes, just 20 have a non zero coefficient, when lasso penalisation was applied using leave-one-out cross validation from package penalized. Using glmnet, the number of genes that had a non-zero coefficient was 18. Our proposed algorithm once again penalized less then the other, leaving 23 genes in the equation. Results are presented in Table 5.3 and Figure 5.1.

5.5.4 PRIDE models

PRIDE models have been used in a series of applications, either for logistic/Poisson regression, smoothing or survival analysis. The novelty of the models is that they include a deviance vector γ in the linear part of the model, that adds one parameter for each observation. This extra parameter will absorb any extra variation that is not captured by the models. Due to their flexibility, PRIDE models can be used for modelling overdispersion, data with extra variation or data where digit preference might be an issue. The deviance

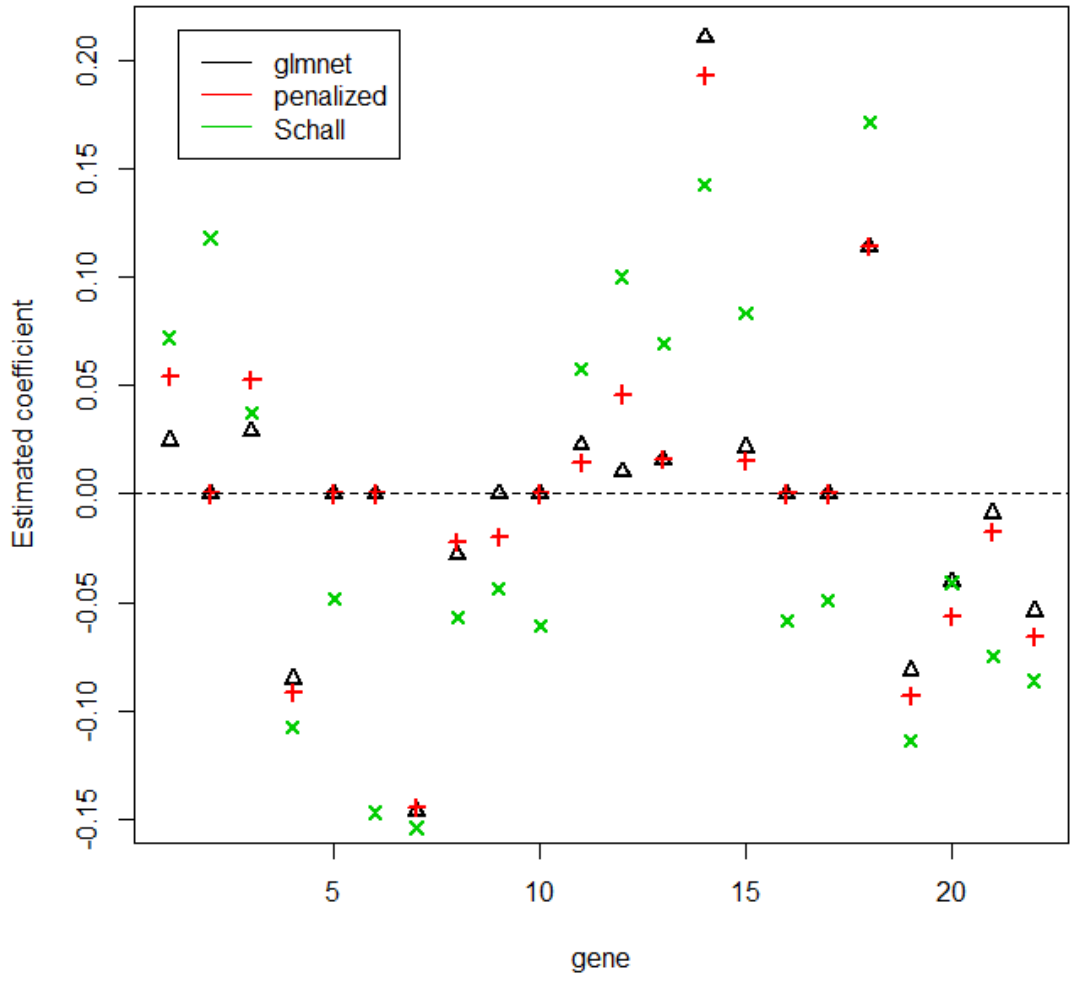

Figure 5.1: *Estimated coefficients under the different packages i.e.* `glmnet`, `penalized`, *proposed algorithm using grid search, and proposed algorithm using Schall's algorithm. The proposed algorithm using Schall's algorithm penalized less than others.*

gene number	glmnet	penalized	schall
6222	0.025	0.054	0.072
14046	0.000	0.000	0.118
14949	0.029	0.052	0.038
15863	-0.085	-0.092	-0.108
16984	0.000	0.000	-0.048
17599	0.000	0.000	-0.147
21092	-0.146	-0.145	-0.154
21550	-0.028	-0.022	-0.057
22140	0.000	-0.020	-0.044
22813	0.000	0.000	-0.060
24245	0.022	0.014	0.058
24565	0.010	0.046	0.100
24892	0.016	0.016	0.070
25141	0.210	0.193	0.143
25367	0.022	0.015	0.083
26672	0.000	0.000	-0.058
27354	0.000	0.000	-0.049
28680	0.114	0.114	0.171
28967	-0.081	-0.093	-0.114
29041	-0.040	-0.056	-0.041
29045	-0.009	-0.018	-0.075
30141	-0.054	-0.066	-0.086

Table 5.3: *Coefficient estimates under four different approaches i.e.* `glmnet`, `penalized`, *and proposed algorithm using Schall's algorithm. The proposed algorithm using Schall's algorithm penalized less than others.*

effects of the model are restrained by adding an L_2 penalty, so the parameters are identifiable. In their original paper, Perperoglou and Eilers (2010) illustrated how to fit the models using an efficient algorithm that does not require a grid search over several values of penalty weights. Here, we look into an application where the deviance effects can be controlled using an L_1 or even an L_0 penalty model.

For example, consider the data on the number of deaths of Greek males in 1960. Figure 5.2 presents a histogram of the raw data. It can be seen that every five years, from the age of 45 and onwards, there exists a spike of increased number of deaths. This phenomenon is known as age heaping in demography or digit preference in general. An L_2 type of penalty is (blue line) shows how the smoothed data should look like. Using the same idea as before, we also fitted an L_1 penalty (green line). The smooth line is almost identical, for

early ages, to the blue line and somewhat different for ages over 70. An L_0 penalty smooth (red line) was also fitted. In practice what that means is that when L_1 is selected, matrix $\mathbf{W}$ is a diagonal matrix with elements, $1/(\beta_j^2 + \epsilon)$, while when L_0 is selected $\mathbf{W}$ becomes a diagonal matrix with elements $1/\sqrt{\beta_j^2 + \epsilon}$.

The deviance effects can be plotted to gain insight on the patterns of extra variation in the data. The deviance effects are quite large (in absolute value) in ages multiple of five, showcasing the impact of digit preference. Additionally, large deviance effects (positive) are associated with ages plus or minus one year of the multiples of five. That illustrated the "popularity" of some numbers, and the "unpopularity" of some other. It also evident, that L_2 penalties tend to shrink all of the effects to smaller sizes, while L_1 penalty shrinks some effects closer to zero than others. L_0 penalization shrinks most of the deviance effects to absolute 0 and only leaves some in specific ages to absorb that digit preference.

5.6 Summary

In this chapter, the new algorithm for lasso is proposed. The algorithm is the combination of Tibshirani (1996) proposed, Schnabel and Eilers (2013)'s approximation for a sum of absolute values and a thresholding scheme which removes some small $\hat{\boldsymbol{\beta}}$s (a small $\hat{\boldsymbol{\beta}}$ become a zero coefficient) . Tibshirani (1996) proposed lasso regression and some algorithms for the regression. He used quadratic programming in his paper and also proposed the penalty $\sum |\beta_j|$ is written as $\sum \beta_j^2/|\beta_j|$. The lasso estimate $\tilde{\boldsymbol{\beta}}$ can be approximate by a ridge regression.

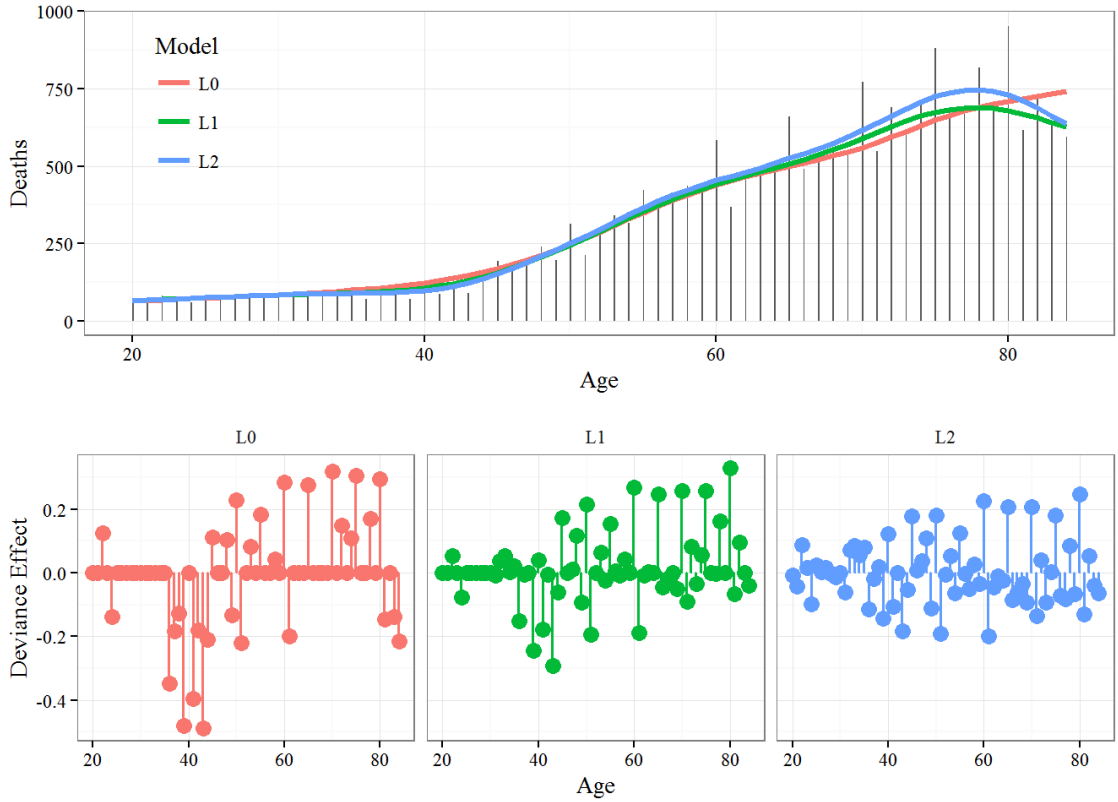

Figure 5.2: *Upper: A histogram of the number of deaths for Greek males in 1960. Three smoothers have been applied with PRIDE modelling, using L_2 (blue line), L_1 (green line) and L_0 (red line) penalization. Lower left: Plot of deviance effects under L_0 penalization, lower middle: deviance effects under L_1 penalization, lower right: deviance effects under L_2 penalization.*

Schnabel and Eilers (2013) adapted Schlossmacher (1973) for approximating a sum of absolute values $\sum_j |\beta_j|$ as a sum of weighted squares $\sum_j \beta_j^2/|\tilde{\beta}_j|$. They modified a sum of weighted squares to make it safer. A sum of weighted squares is written as $\sum_j \beta_j^2/\sqrt{\tilde{\beta}_j^2 + \epsilon^2}$, with ϵ is a small number. So $\mathbf{W}$ can be written as a diagonal matrix with the diagonal elements $\sqrt{\tilde{\beta}_j^2 + \epsilon^2}$.

The result of the above lasso estimate still has ridge regression behaviour, there are no ridge regression coefficients, which is exactly zero. A thresholding scheme is applied so that removes some small $\hat{\boldsymbol{\beta}}$s. If $|\hat{\beta}_j| \leq \gamma$ then the coefficients are zero where $\gamma = \sum(\beta_j^0 - \hat{\beta}_j)^2/p$ and β_j^0 is coefficients for simple regression.

The proposed algorithm is compared to other algorithms in R package `penalized` and `glmnet` for prostate data set and two simulations. Those three packages use cross-validation as model selection and the proposed algorithm use generalized cross-validation, but the results are quite similar.

In addition, Schnabel and Eilers (2013)'s approximation can be used in PRIDE models. PRIDE models involve the deviance effects in the models and control it using an L_2 penalty model. The deviance effects also can be controlled using an L_1 or an L_0 penalty model using Schnabel and Eilers (2013)'s approximation.

Chapter 6

Two Dimensional Smoothing via an Optimised Whittaker Smoother [1]

A large number of observations will produce a scatter-plot which is difficult to investigate due to a high concentration of points on a simple graph. We review the Whittaker smoother for enhancing scatter-plots and smoothing data in two dimensions. To optimise the behaviour of the smoother, an algorithm is introduced, which is simple and computationally efficient.

The Whittaker smoother are well-used to smooth and interpolate noisy data. The advantages of implementing the Whittaker smoother are having fast computation, providing continuous control of smoothness, automatic interpolation and ease of cross-validation Eilers (2003). The Whittaker smoother can be a valuable tool in producing better visualisations of big data or filter distorted images.

[1] This chapter has been published on Zuliana, S. U. and Perperoglou, A. "Two dimensional smoothing via an optimised whittaker smoother," Big Data Analytics, vol. 2, no. 1, p. 6, 2017.

Eilers and Goeman (2004) have applied the Whittaker smoother for visual enhancement of a scatterplot, using a smoothed histogram. However, the penalty weight λ is chosen by the user's taste. In this study, The optimisation process on two dimensional smoothing is proposed. The optimal penalty weight λ can be obtained automatically. The methods are illustrated using a simple dataset and simulations in two dimensions. Additionally, a noisy mammography is analysed. When smoothing scatterplots the Whittaker smoother is a valuable tool that produces enhanced images that are not distorted by the large number of points. The methods is also useful for sharpening patterns or removing noise in distorted images.

The article of this chapter has published on Zuliana, S. U. and Perperoglou, A. "Two dimensional smoothing via an optimised whittaker smoother," Big Data Analytics, vol. 2, no. 1, p. 6, 2017 and it is attached in the appendix. The contribution of study is the optimisation process on two dimensional smoothing. It can be done automatically without playing a grid of penalty weights λs, simple and low computational cost. It is started by any initial penalty weight λ. This study could be developed to more than two-dimensional smoothing.

On the published paper, the proposed approach has been compared with Whittaker smoothing without optimization and Kernel smoothing. In this section, the proposed approach will be compared with tensor product smoothing from `package mgcv`.

From Table 6.1, the optimized Whittaker smoothing needs only very short time for all image. For simulated image, the proposed algorithm needs 3.29 seconds and the tensor

	time in second	
	tensor product	optimized Whittaker
simulated histogram	87.8	3.29
simulated image	32.4	3.5
application	144.8	1.26

Table 6.1: *Computation time for smoothing simulated histogram, simulated image and the real image between optimized Whittaker and tensor product*

product needs 87.8 seconds to get the optimal smoothing. Also, the result is better for recognizing the true data (The true histogram can be seen on the appendix). Compare with the result from tensor product smoothing, the result from the proposed algorithm also better for reducing the noise, so the true signal can be seen. Besides,the tensor product need longer time than the proposed algorithm, it needs 87.8 second to get the optimal result.

The result from tensor product (`package mgcv`) is better than the proposed algorithm's result. However the proposed algorithm's computation time is significantly shorter than the tensor product's time. The Schall's algorithm only needs 3.5 seconds and 1.26 second to get the smoothing image and the tensor product needs 32.4 second and 144.8 seconds to get it.

The optimized Whittaker has a good result for bivariate smoothing. It can enhance the image and reduce the noise. The computation time is very short. It means the smoothing would be developed for more than two-dimensional smoothing.

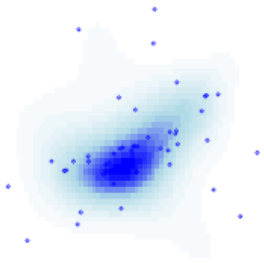

Figure 6.1: *Optimized Smoother Whittaker*

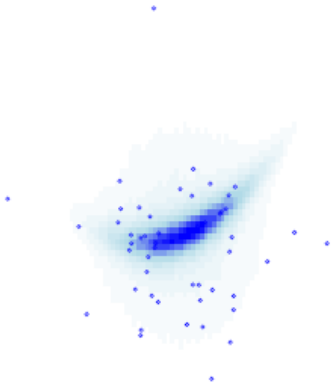

Figure 6.2: *Tensor Product*

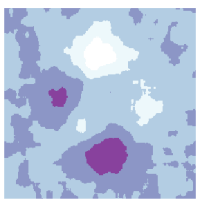

Figure 6.3: *Optimized Smoother Whittaker*

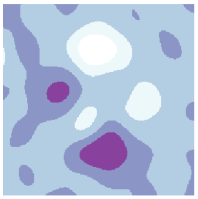

Figure 6.4: *Tensor Product*

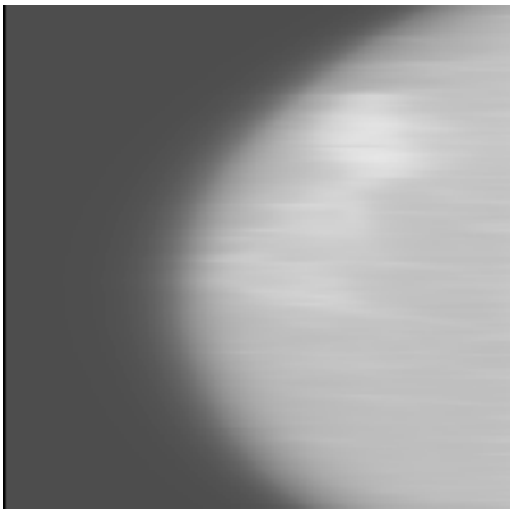

Figure 6.5: *Otimized Smoother Whittaker*

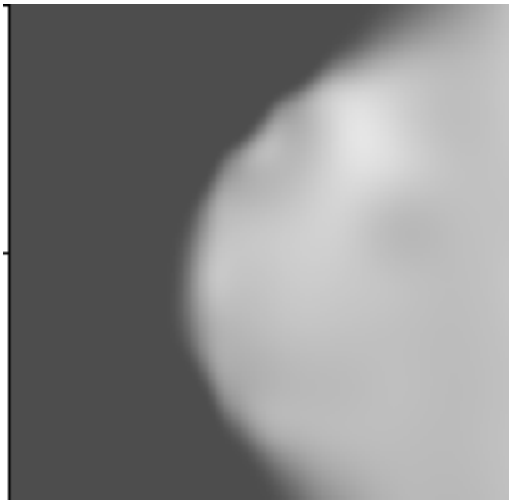

Figure 6.6: *Tensor Product*

Chapter 7

Conclusion and future work

Schall (1991) proposed an algorithm for estimating the variance of the random effect. In this thesis, we have adapted Schall's algorithm as automatic selection for an optimal weight of penalty in order to minimizing computational cost because the optimal penalized model can be obtained from the algorithm within a small number of iterations. Moreover, the proposed algorithm can be initialized with any value of penalty weight.

First of all, Schall's algorithm has been applied to ridge regression for linear models, generalized linear models and generalized additive models. For linear models, the performance of proposed algorithm has been compared to other approaches that have been previously provides for selecting an optimal model such as leave-one-out cross validation, principal components, and generalized cross validation which can be found in `R` package i.e. `penalized`, `gamlss`, `ridge`, and `MASS`. The performances of Schall's algorithm and other approaches are measured with prediction error. Prediction error of Schall's algorithm is the smallest value and the coefficient estimates was not mistakenly estimated as an opposite

sign from the real one.

Furthermore, for the implementation of Schall's algorithm on Poisson and logistic regressions, a real data set and simulated data sets have been considered. The simulated datasets have correlated parameter with different correlation coefficients and different sample sizes, and the results are compared with model criterion, i.e. Akaike information criterion (AIC), bayesian information criterion (BIC), and generalized cross validation (GCV). The results demonstrated that the proposed automatic selection for an optimal weight of penalty method outperform as compared to the model criterion. In addition, Schall's algorithm is also applied to generalized additive models with P-splines (P-GAMs) and the results show a smoother curve compare to polynomial regression and B-spline regression.

Next, Schall's algorithm is applied to lasso regression. For the implementation of Schall's algorithm, we need to calculate effective dimension, therefore a new algorithm for lasso regression has been proposed and the new algorithm is Tibshirani (1996)'s suggestion with improvement using Schnabel and Eilers (2013)'s approximation . The new algorithm of lasso and Schall's algorithm as an automatic model selection is applied `prostate` data, `eyedata`, simulated data with five zero parameters, and simulated data with twenty zero parameters.

Finally, Schall's algorithm is applied to two dimensional smoothing. The optimised Whittaker smoother has been proposed. In this work an attempt has been made to focus on application of an automatic model selection for Eilers and Goeman (2004)'s work. The algorithm is applied to simulated noisy data, simulated image, and real image. The results

of smoothing can enhance the signal and reduce the noise.

The ideas suggested in this thesis can be very useful in any framework of penalized regression. We showcased applications in linear regression, GLMs and GAMs, as well as smoothing in two dimensions. We can quite easily expand our suggestions in more than two dimensions, where tensor products are needed and penalization and optimisation can be a computationally expensive task. Further work can also illustrate properties of what we have presented in a series of examples. More simulations can highlight the behaviour of our approach in different datasets.

A benefit can be seen also in the use of Schall's algorithm in conjunction with PRIDE models. We did present one example on a simple smoothing case, but the applications expand also to more complicated settings. One might consider smoothing with PRIDE in more than one dimensions but using an automated penalty optimisation. It would be very interesting to see the behaviour of L_1 and L_0 penalties both in many dimensions but also in regression problems with several covariates.

Bibliography

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723.

Cessie, S. L., Houwelingen, J. C. V., and Society, R. S. (1992). Ridge estimators in logistic regression. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 41:191–201.

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American statistical association*, 74(368):829–836.

Craven, P. and Wahba, G. (1978). Smoothing noisy data with spline functions. *Numerische Mathematik*, 31(4):377–403.

Cule, E. (2014). ridge: Ridge regression with automatic selection of the penalty parameter. *R package version*, pages 1–21.

Cule, E. and De Iorio, M. (2013). Ridge regression in prediction problems: automatic choice of the ridge parameter. *Genetic epidemiology*, 37(7):704–714.

De Boor, C. (1972). On calculating with B-splines. *Journal of Approximation Theory*, 6:50–62.

Eilers, P. H. (2003). A perfect smoother. *Analytical chemistry*, 75(14):3631–3636.

Eilers, P. H. and Goeman, J. J. (2004). Enhancing scatterplots with smoothed densities. *Bioinformatics*, 20(5):623–628.

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

Friedman, J., Hastie, T., and Tibshirani, R. (2009). glmnet: Lasso and elastic-net regularized generalized linear models. *R package version*, 1.

Goeman, J., Meijer, R., and Chaturvedi, N. (2012). penalized: L1 (lasso and fused lasso) and l2 (ridge) penalized estimation in glms and in the cox model. *URL http://cran. r-project. org/web/packages/penalized/index. html*.

Goeman, J. J. (2010). L1 penalized estimation in the cox proportional hazards model. *Biometrical journal*, 52(1):70–84.

Golub, G. H., Heath, M., and Wahba, G. (1979). Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223.

Green, P. J. and Silverman, B. W. (1993). *Nonparametric regression and generalized linear models: a roughness penalty approach*. CRC Press.

Greene, W. H. (2012). *Econometric analysis*. Pearson Education.

Hastie, T. and Tibshirani, R. (1986). Generalized additive models. *Statistical Science*, 1:297–310.

Hoaglin, D. C. and Welsch, R. E. (1978). The hat matrix in regression and anova. *The American Statistician*, 32(1):17–22.

Hoerl, A. E., Kannard, R. W., and Baldwin, K. F. (1975). Ridge regression: some simulations. *Communications in Statistics-Theory and Methods*, 4(2):105–123.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Hurvich, C. M. and Tsai, C.-L. (1989). Regression and time series model selection in small samples. *Biometrika*, 76(2):297–307.

Kibria, B. M. G. (2003). Performance of Some New Ridge Regression Estimators. *Communications in Statistics - Simulation and Computation*, 32(2):419–435.

Lee, Y. and Nelder, J. A. (1996). Hierarchical generalized linear models. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 619–678.

Månsson, K. and Shukur, G. (2011). A poisson ridge regression estimator. *Economic Modelling*, 28(4):1475–1481.

Marx, B. D. and Eilers, P. H. (1998). Direct generalized additive modeling with penalized likelihood. *Computational Statistics & Data Analysis*, 28(2):193–209.

Muniz, G. and Kibria, B. G. (2009). On some ridge regression estimators: An empirical comparisons. *Communications in Statistics-Simulation and Computation®*, 38(3):621–630.

Nelder, J. A. and Wedderburn, R. W. M. (1972). Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3):370–384.

Perperoglou, A. (2013). Package coxridge.

Perperoglou, A. (2014). Cox models with dynamic ridge penalties on time-varying effects of the covariates. *Statistics in Medicine*, 33(1):170–180.

Perperoglou, A. and Eilers, P. H. (2010). Penalized regression with individual deviance effects. *Computational Statistics*, 25(2):341–361.

Piazza, J. A. (2012). The opium trade and patterns of terrorism in the provinces of afghanistan: An empirical analysis. *Terrorism and Political Violence*, 24(2):213–234.

R Development Core Team (2015). R: A language and environment for statistical computing. (2015).

Rigby, R. A. and Stasinopoulos, D. M. (2005). Generalized additive models for location, scale and shape. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 54(3):507–554.

Rigby, R. A. and Stasinopoulos, D. M. (2013). Automatic smoothing parameter selection in GAMLSS with an application to centile estimation. *Statistical methods in medical research*, 23(4):318–332.

Schall, R. (1991). Estimation in generalized linear models with random effects. *Biometrika*, 78(4):719–727.

Schlossmacher, E. (1973). An iterative technique for absolute deviations curve fitting. *Journal of the American Statistical Association*, 68(344):857–859.

Schmidt, G., Mattern, R., and Schüler, F. (1981). Biomechanical investigation to determine physical and traumatological differentiation criteria for the maximum load capacity of head and vertebral column with and without protective helmet under effects of impact. *Final report phase III, Project*, 65.

Schnabel, S. K. and Eilers, P. H. (2013). Simultaneous estimation of quantile curves using quantile sheets. *AStA Advances in Statistical Analysis*, 97(1):77–87.

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

Silverman, B. W. (1985). Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–52.

Stamey, T. A., Kabalin, J. N., McNeal, J. E., Johnstone, I. M., Freiha, F., Redwine, E. A., and Yang, N. (1989). Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate. II. Radical prostatectomy treated patients. *The Journal of urology*, 141(5):1076–1083.

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

Venables, W. and Ripley, B. (2002). *Modern applied statistics with S*. Springer, New York, NY, USA.

Verweij, P. J. M. and Van Houwelingen, H. C. (1994). Penalized likelihood in cox regression. *Statistics in Medicine*, 13(23-24):2427–2436.

Xue, X., Kim, M. Y., and Shore, R. E. (2007). Cox regression analysis in presence of collinearity: an application to assessment of health risks associated with occupational radiation exposure. *Lifetime data analysis*, 13(3):333–50.

Publications

1. Zuliana, S. U., and Perperoglou, A. (2016). The Weight of Penalty Optimization for Ridge Regression. In Analysis of Large and Complex Data (pp. 231-239). Springer International Publishing.

2. Zuliana, S. U., and Perperoglou, A.. Two Dimensional Smoothing via an Optimised Whittaker Smoother. Big Data Analytycs, vol. 2, no. 1, p. 6, 2017.

Appendix A

Zuliana and Perperoglou *Big Data Analytics* (2017) 2:6
DOI 10.1186/s41044-017-0021-9

Big Data Analytics

SOFTWARE Open Access

Two dimensional smoothing via an optimised Whittaker smoother

Sri Utami Zuliana[1*] and Aris Perperoglou[1,2]

*Correspondence:
sutami@essex.ac.uk
[1]Department of Mathematical Sciences, University of Essex, Wivenhoe Park, CO4 3SQ Colchester, UK
Full list of author information is available at the end of the article

Abstract

Background: In many applications where moderate to large datasets are used, plotting relationships between pairs of variables can be problematic. A large number of observations will produce a scatter-plot which is difficult to investigate due to a high concentration of points on a simple graph.
In this article we review the Whittaker smoother for enhancing scatter-plots and smoothing data in two dimensions. To optimise the behaviour of the smoother an algorithm is introduced, which is easy to programme and computationally efficient.

Results: The methods are illustrated using a simple dataset and simulations in two dimensions. Additionally, a noisy mammography is analysed. When smoothing scatterplots the Whittaker smoother is a valuable tool that produces enhanced images that are not distorted by the large number of points. The methods is also useful for sharpening patterns or removing noise in distorted images.

Conclusion: The Whittaker smoother can be a valuable tool in producing better visualisations of big data or filter distorted images. The suggested optimisation method is easy to programme and can be applied with low computational cost.

Keywords: Histogram smoothing, Data visualisation, H-likelihood

Background

The histogram -in all its simplicity- is one of the most powerful tools of data visualization. Plotting the values of a variable x against a variable y will reveal whether there are is some sort of correlation between the variables or not, whether the relationship is linear or more complicated, whether there are interesting subgroups in the data or whether outliers are present. A problem might rise however, when trying to plot many points onto one simple graph. As the number of observations becomes larger and larger many scatter-plots end up being to busy for the eye to understand. Often, in moderate to large datasets, a collection of many observations on one plane will end up revealing a cloud of points where all structure remains obscured by the superposition of one point onto another. Depending on what is the medium where such a graph will be illustrated, it becomes a waste of ink or space.

To address this problem, some researchers have suggested smoothing data to obtain a heat plot image, rather than the original scatter plot. A heat plot will use colour, or shades of black, to represent areas of great concentration of points. A common way is via the use of Kernel smoothers [1], employed in `R` with the function `smoothScatter`

which is part of the base distribution [2]. More recently, Eilers and Goeman [3] illustrated a way of smoothing scatter-plots in two directions using penalized b-splines or p-splines. This approach has been implemented in package `gamlss.util` via command `scattersmooth` [4].

In this work we are focusing on the paper by Eilers and Goeman [3] where a scatter-plot is enhanced using smoothed densities. We will start off with the same approach, where penalized splines are applied on the x and y directions, respectively. However, we will also go a step further and so how the optimal smoothed scatter-plot can be obtained by estimating the amount of penalty needed for each graph. We view penalized splines as random effects whose variance depends on the penalty weight. This is not a completely new approach but has only been applied to one dimension before (see [5–8]). We will revise the algorithm and extend it to apply to two dimensional smoothing.

The paper will start by illustrating a simple spline, the Whittaker smoother [9] and how this is applied in smoothing in one direction. In the next section we will introduce a simple dataset on which we will show how to obtain an optimised smoother where the penalty weight is estimated. We will then extend the method into two dimensions and show how to optimise smoothing penalties. The paper is ended with a discussion.

Implementation

The Whittaker smoother

Consider a simple scatter-plot in which the logarithm of the ratio of received light from two laser sources (given as y) is plotted against the distance travelled before the light is reflected back to its source, or range x. These particular data are produced using the Light Detection and Ranging (LIDAR) technique. The data have been used in [10] (Chapter 3) and can be downloaded from http://matt-wand.utsacademics.info/webspr/data.html.

We would like to obtain a smooth function of y given by a vector α. That means that for each observation in vector y, written as y_i with $i = 1, 2, ..., m$ an estimate α_i is obtained. Adding one parameter α_i for each observation y_i has the benefit of allowing the smoother to be very flexible and follow any kind of pattern the data might have. The drawback of course is that the number of parameters is as big as the number of observations which can lead to over-fitting. To control for over-fitting, a roughness penalty is imposed based on the differences of the parameters.

Let D_d be a matrix that forms differences of order d. For example, a first order difference is denoted as $\Delta\alpha_i = \alpha_i - \alpha_{i-1}$, while a second order difference would be $\Delta^2\alpha_i = \Delta(\Delta\alpha_i) = \alpha_i - \alpha_{i-1} - (\alpha_{j-1} - \alpha_{j-2})$, with corresponding D_1 and D_2 matrices given by:

$$D_1 = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}; D_2 = \begin{bmatrix} -1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix}$$

The penalized Whittaker smoother is computed by minimising the following penalised least-squares function:

$$S = ||y - \alpha||^2 + \lambda||D_d\alpha||^2 \quad (1)$$

Then, to get an explicit solution for α one needs to minimise S in (1). That would lead to penalized normal equations given as:

$$\hat{\alpha} = (I + \lambda D'D)^{-1} y \tag{2}$$

where I is an identity matrix of dimension $m \times m$. The smoothed vector $\hat{\alpha}$ depends, of course, by the choice of the penalty weight. When λ tends to zero, hardly any penalization is imposed on the estimates giving a non-smoothed curve, close to the actual values. On the other extreme, as λ tends to infinity the penalty weight dominates and it results in a straight line. Optimal values of λ should provide a smooth curve that reveals the true nature of the data whilst removing roughness and randomness. Figure 1 illustrates the raw data along with three smooth curves based on different penalty weights. For small values of λ the data are undersmoothed, while as λ increases the methods provides a smoother curve.

Penalty optimization

A common way to choose the optimal weight is to perform a search for an optimal criterion over a fine grid of λ values. The user has to define a number of distinct possible values of λ, fit a model for each one of those and then decide which one is preferred based on some sort of a loss function or a criterion. Common choices include cross-validation or Akaikes-type criteria (including Akaike Information Criterion (AIC), Akaike Information Criterion correction (AICc), Bayesian Information Criterion (BIC) etc, see [11–13]).

One popular approach is the use of Generalized Cross Validation (GCV) [14]. Define H the hat matrix as $H = (I + \lambda D'D)^{-1}$ and let ed= trace(H) be the effective dimensions, given as the sum of the diagonal elements of H. Then

$$GCV(\lambda) = \frac{\sum_{i=1}^{m}(y_i - \hat{\alpha_i})^2}{(m - tr(H))^2} \tag{3}$$

Here, we use an algorithm for penalty optimisation that treats the penalty weight as a parameter to be estimated from the model. A penalized likelihood can be seen through a Bayesian model framework [15], or a random effects framework [10], or an extended

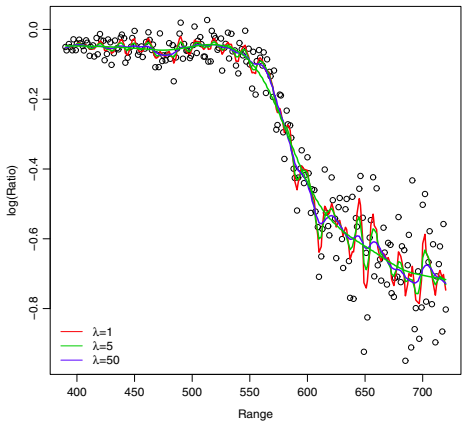

Fig. 1 Whittaker smoother on the LIDAR data using three different penalty weights

likelihood of a random effect parameter [16]. These different viewpoints allow for the use of an algorithm that was first suggested by [17] to estimate the variance of the random effect in a random effects model. Variations of the algorithm have also been published in [6, 7].

In the Whittaker smoother model, define $e = y - \hat{\alpha}$ and let

$$\hat{\sigma^2} = \frac{e'e}{m - ed} \tag{4}$$

where m and ed as before, and let

$$\hat{\sigma_\alpha^2} = \frac{\hat{\alpha}'D'D\hat{\alpha}}{ed - 2} \tag{5}$$

More details can be found in [18] (Chapter 9).

The algorithm that chooses an optimal weight then has the following steps:

1. For given $\hat{\sigma^2}, \hat{\sigma_\alpha^2}$ find $\hat{\lambda} = \frac{\hat{\sigma^2}}{\hat{\sigma_\alpha^2}}$.
2. Estimate vectors by: $\hat{\alpha} = (I + \hat{\lambda}D'D)^{-1}y$
3. Given α update $\hat{\lambda} = \frac{\hat{\sigma^2}}{\hat{\sigma_\alpha^2}}$.
4. Iterate until convergence.

The algorithm usually converges within a few steps. In rare cases convergence is sensitive to starting values of λ but we have found that this is rarely happening when both $\hat{\sigma^2}, \hat{\sigma_\alpha^2}$ are 1.

Smoothing a two dimensional histogram

Consider a two dimensional domain $x - y$ that is being cut into rectangles and the number of observations that lie within each rectangle been counted. For such an $x - y$ plain a matrix $R_{m\times n}$ is formed that contains counts. To smooth a two-dimensional histogram based on R, one has to smooth first the columns $R_{\bullet n}$, that form a vector y, using the same algorithm defined before for one dimensional smoothing. That would produce a new matrix $G_{m\times n}$. Then, using exactly the same procedure it is easy to smooth the columns of $G'_{\bullet m}$, which are the rows of $G_{m\times n}$. The new smoothed matrix will be the transposed of the desired outcome. This is the algorithm that was defined in [3]. There are two different penalty weights in the algorithm, λ_1 that penalises the smooth over columns of $R_{m\times n}$ and λ_2 which is used for the penalty in rows of $G_{m\times n}$. In the original paper, as well as in the function `scatterSmooth` the penalties are not optimised, instead they are taken with the default values: $\lambda_1, \lambda_2 = 1$. Since this is a two step algorithm, it would be rather straightforward to optimise λ-s into both direction. In the first step, the algorithm for one dimensional smoothing can be applied to get the optimal for the columns of $R_{m\times n}$ and in the second step, the same algorithm will be applied to optimize λ_2. That would result in an overall better image of the data.

Results

The LIDAR data

For the LIDAR data, the GCV criterion was first used as a reference. A fine grid of values was defined, ranging from a very small penalty weight 0.001 that would allow the estimates to vary freely, to a large penalty of 100000 that would essentially make the estimates close to zero. The optimal value was determined to be for a high value of $\lambda = 7943$. Using

the algorithm to optimise the penalty weight the estimated value was $\hat{\lambda} = 5758$. Although the two values look different, in fact the smooth line they produce is not distinguishable, as seen from Fig. 2, where one smooth lies on top of the other.

Simulated histogram

To illustrate the methods, a simple simulation dataset was created. Let $x \sim N(0,1)$ and $y = 0.7 * x + 0.4x^2 + 0.3e$ where e is Gaussian noise. A total of 10000 observations were created and plotted in the upper left scatter-plot in Fig. 3. The relationship between the two variables is obscured by random Gaussian noise (showing in upper right graph). The latter scatter-plot was then smoothed using first a Whittaker smoother with optimised penalties. The algorithm estimated a penalty close to zero along the columns $\lambda_1 < 0.001$ and a second penalty $\lambda_2 = 4.3$ along the rows. The image produced be the smoother is shown in the lower left scatter-plot. The heatmap shows areas of great concentration of points, towards the centre of the graph, and also clearly reveals the signal behind the noise. A few randomly selected points are plotted around the heatmap. In the lower right graph, the Kernel smoother (using `smoothScatter` in `R`) also reveals the true signal, however, it is more sensitive to the noise and provides a heatmap with some features of the noise still in it.

Simulated image

The Whittaker smoother can also be used of any 2-dimensional smoothing. To illustrate, consider the image in Fig. 4 (upper left) in which some Gaussian noise was added to mask the patterns (upper right). The addition of Gaussian noise masks completely the previously clear patterns. The application of a Whittaker smoother without a penalty optimisation uses a default line for both weights, thus here: $\lambda_1 = \lambda_2 = 1$. However, in this case there is a need for bigger penalties that will control the smooth in both directions. As seen in the Fig. 3 (lower right graph) the smoother does remove some noise and hints on some of the patterns but it does not reveal the true image. Instead, when the weights are optimised (here $\lambda_1 = 23.8$ and $\lambda_2 = 33.4$) the pattern is clearly revealed.

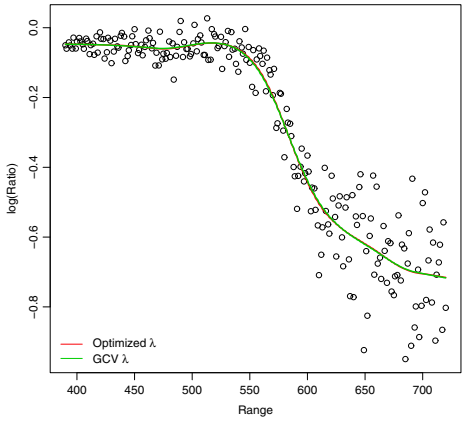

Fig. 2 Whittaker smoother on the LIDAR data, using cross-validation and optimisation

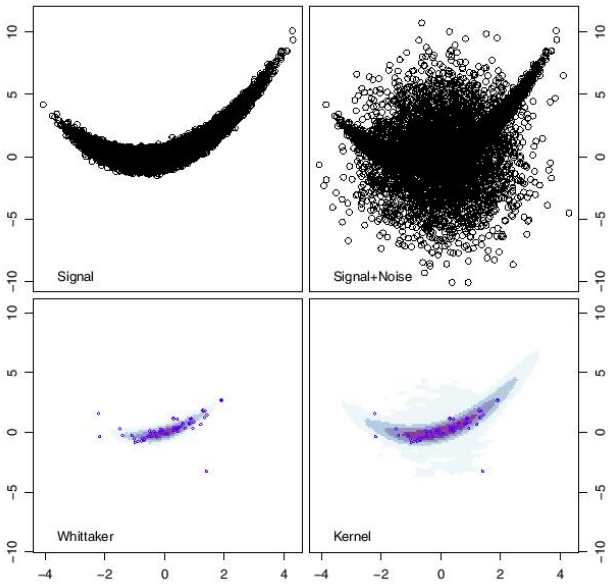

Fig. 3 Smoothing a two dimensional histogram: simulated histogram. [*Upper left graph*:] The true relation between x and y, [*Upper right graph*:] obscured by noise, [*Lower left graph*:] smoothed by Whittaker smoother with optimised penalties and [*Lower right graph*:] smoothed by Kernel smoother

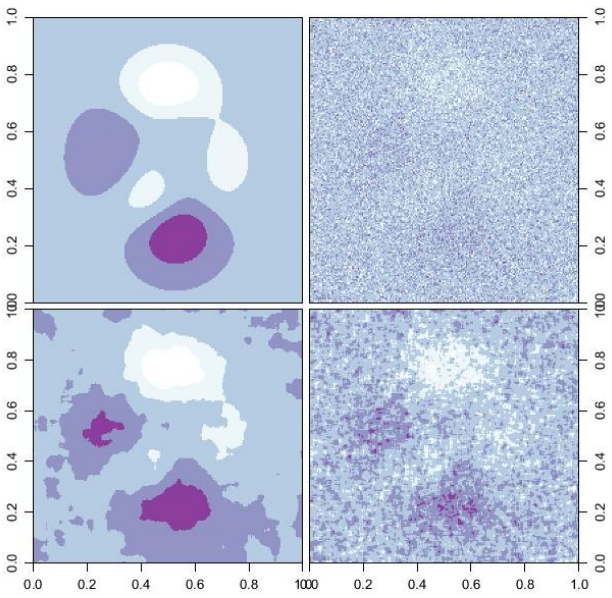

Fig. 4 Smoothing a two dimensional histogram: simulated image. [*Upper left graph*:] A simulated image [*Upper right graph*:] obscured by Gaussian noise, [*Lower left graph*:] smoothed by Whittaker smoother with optimised penalties and [*Lower right graph*:] smoothed by Whittaker smoother without optimisation

Filtering a noisy mammography

Smoothing can also be used to filter out noise from a distorted image. As an example we consider the case of a mammography. In the upper left part of Fig. 5 a mammography is displayed. In the upper part of the breast, a white shade (marked with a cross) shows signs of what might be a tumour. The original image can be found online at: http://img.medscape.com/news/2014/dt_140703_mammography_breast_cancer_800x600.jpg. To make the problem more challenging Gaussian noise has been added to the image, in a way that distorts the definition of the tumour. In Fig. 5, the original image has been slightly distorted, as it can be seen on the upper right part of the graph. To filter noise out, a kernel smoother has been used that resulted in the image shown in the lower left part of the figure. The smoother was created using function: `image.smooth` from library `fields` [19]. The smoother has removed a lot of noise and the image looks sharper, though not as sharp as the original. The Whittaker smoother was applied, with an automated selection of penalty weights. In the lower right part of the figure the Whittaker method produces a better image, has removed more noise than the kernel smoother and defined the tumour more clearly.

The merit of the method can also be seen when the image is more noisy. Figure 6 presents the same mammograph, where the addition of noise now completely distorts the image (upper right). The kernel smoother fails to reveal the original features of the image. On the contrary, using a Whittaker smoother, the features of the image are restored (lower

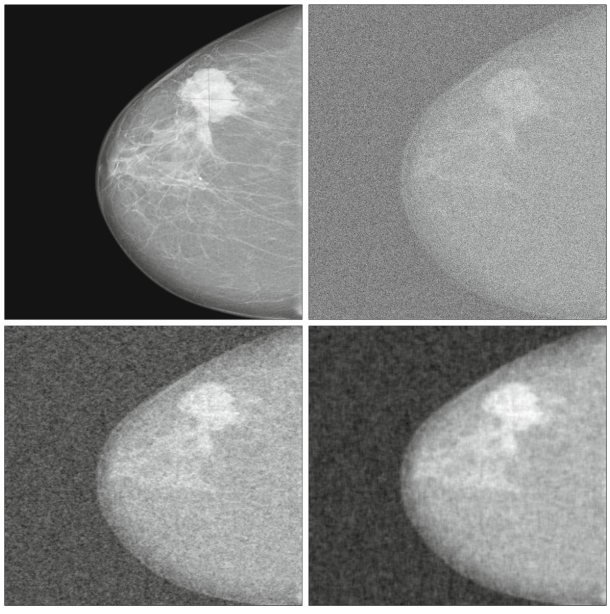

Fig. 5 Smoothing a two dimensional histogram: Added some noise to mammography. [*Upper left graph*:] A breast mammography. An *area* that seems like a tumour has been marked with a *cross*. [*Upper right graph*:] obscured by Gaussian noise, [*Lower left graph*:] smoothed by kernel smoother [*Lower right graph*:] smoothed by Whittaker smoother with optimisation

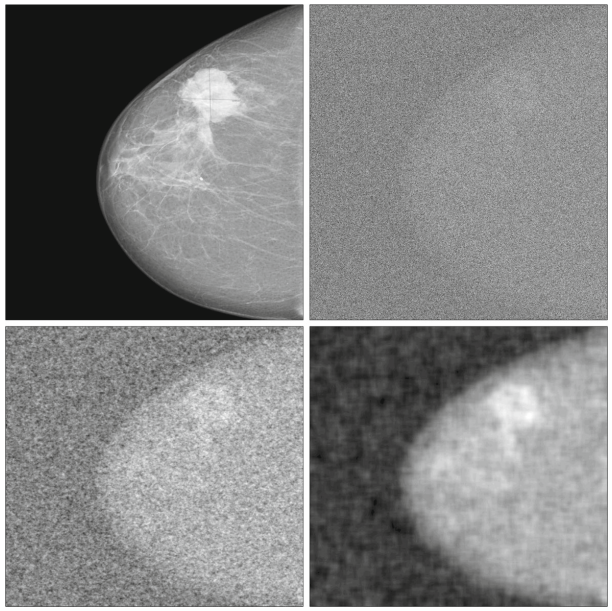

Fig. 6 Smoothing a two dimensional histogram: Mammography completely distorted by noise. [*Upper left graph*:] A breast mammography. An *area* that seems like a tumour has been marked with a *cross*. [*Upper right graph*:] obscured by Gaussian noise, [*Lower left graph*:] smoothed by Whittaker smoother with optimised penalties and [*Lower right graph*:] smoothed by Whittaker smoother without optimisation

right). Although there is still noise left, it is now more clear that there is a finding in the mammography.

Conclusions

A simple - yet powerful addition to a Whittaker smoother was presented. The addition is based on an efficient algorithm that will lead to an optimised penalty weight. Thus, the degree of smoothing that is needed can be objectively decided by the procedure rather than subjectively by the user. The methods can be applied to one or two dimensional smoothing.

The methods presented here are intended as a tool for the applied user who would like to have an effective and computationally efficient way to smooth scatter-plots or images. The approach was illustrated and compared to a Kernel smoother or a simple Whittaker smoother. When compared with the Kernel smoother the optimised Whittaker approach produced an image with less noise and closer to the true relationship between the variables. We see a two-fold advantage here; first the optimised smoother can be used as a simple data visualisation device. It will produce a plot that is visually more compelling whilst on the same hand communicating significant information on the data. As such the differences with the Kernel smoother are minimal. Another advantage however, is that the optimised smoother can be used to gain a better insight and understanding at the data, since it removes more noise than a Kernel smoother when needed. As such, the Whittaker smooth can be used as a more in-depth explanatory method for making sense out of data.

The benefits of optimising penalty weights were also illustrated further in a second example of smoothing a simulated image. Of course, an experienced researcher will probably have been able to identify the need of a larger penalty in Fig. 4 (lower right) and experiment with larger values for the penalties. That would probably led to a better image but leads to a subjective fit that depends on the used. On the other hand, one could also optimise penalty weights by minimising some sort of loss function or criterion, as illustrated in "Background" section, but this would be a computational expensive method to follow, especially in two dimensions.

When working with real mammography images, the method was able to outperform kernel smoothers. In further investigation of the same problem, Gaussian filters have been used, to blur the image and obtain better results. When specifying a Gaussian blur, the user has to specify the variance of the Gaussian distribution. With some trial and error approach, we where able to filter the noise out to a satisfactory level, but we could not outperform the Whittaker smoother (data not shown). Additionally, the filter did require tuning from the user and was not based on an automated procedure.

A merit of our approach is that it can work even in cases where smoothing is not required. When the image is not noisy, the algorithm with converge to extremely small values for the penalty weights, thus removing the effect of the penalty altogether. The more noisy the image the bigger the penalty weights will be. These are situations where the method has great advantages over other approaches.

The algorithm presented in this paper was coded in R in just a few lines of code. It is very easy however to implement it in another programming language like Matlab or Java. The appendix contains the R programme.

Availability and requirements

Operating System: Windows 7
Language: R

Appendix: R code

```
smooth2D = function(Hraw, lambda=1) {
### Hraw: A plane given as an m x m matrix
### lambda: penalty weight
if (length(lambda) == 1)
    lambda = c(lambda, lambda)
    m <- nrow(Hraw)
    n <- ncol(Hraw)
    E1 <- diag(m)
    E2 <- diag(n)
    Dx <- diff(E1)
    Dy <- diff(E2)
dz <- 5
    while(dz>1e-5){
        Qx <- E1 + lambda[1] * t(Dx)
        sQx <- solve(Qx)
        z1 <- sQx
```

```
        HQx <- sQx
        edx <- sum(diag(HQx))
        s2 <- sum(t(Hraw-z1)
        su2 <- sum(t(z1)
        dz <- abs(lambda[1]-s2/su2)
        lambda[1]<- s2/su2}
dz <- 5
    while(dz>1e-5){
        Qy <- E2 + lambda[2] * t(Dy)
        sQy <- solve(Qy)
        z2 <- sQy
        HQy <- sQy
        edy <- sum(diag(HQy))
        s2 <- sum(t(Hraw-z2)
        su2 <- sum(t(z2)
        dz <- abs(lambda[2]-s2/su2)
        lambda[2]<- s2/su2}
out <- list(H=t(z2), Hx = HQx, Hy=HQy, Dx=Dx, Dy=Dy,
     Hraw=Hraw, lambda=lambda)
     out}
```

Abbreviations
AIC: Akaike information criterion; AICc: Akaike information criterion correction; BIC: Bayesian information criterion; GCV: Generalized cross validation

Acknowledgements
This research has been supported by the Institute of Analytics and Data Science (IADS) funded by EPSRC. We thank the referees and the editor for valuable comments which improved the paper considerably. The first author would like acknowledge the support of UIN Sunan Kalijaga, Yogyakarta, Indonesia which sent the first author to do doctoral study.

Funding
This paper is a part of PhD research Project of the first author sponsored by The Ministry of Religious Affairs of the Republic of Indonesia. Aris Perperoglou is an academic expert in the Institute of Analytics and Data Science partly sponsored by EPSRC.

Availability of data and materials
Lidar dataset can be accessed from R-package `SemiPar`.

Authors' contributions
SUZ and AP developed the formulas for estimation and conducted the simulations using R. AP conceived the original idea. Both authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Consent for publication
Not applicable.

Ethics Approval and Consent to Participate
Not applicable.

Author details
[1]Department of Mathematical Sciences, University of Essex, Wivenhoe Park, CO4 3SQ Colchester, UK. [2]Institute of Analytics and Data Science, University of Essex,Wivenhoe Park, CO4 3SQ Colchester, UK.

Received: 22 June 2016 Accepted: 22 February 2017
Published online: 13 March 2017

References
1. Wand M. Fast computation of multivariate kernel estimators. J Comput Graph Stat. 1994;4(4):433–45.
2. R Development Core Team. R: a language and environment for statistical computing. 2014. http://www.R-project.org. Accessed 2 Mar 2017.

3. Eilers PH, Goeman JJ. Enhancing scatterplots with smoothed densities. Bioinformatics. 2004;20(5):623–8.
4. Stasinopoulos M, Rigby B, Eilers P. Gamlss.util: GAMLSS Utilities. 2015. http://CRAN.R-project.org/package=gamlss.util. Accessed 2 Mar 2017.
5. Pawitan Y. In all likelihood: statistical modelling and inference using likelihood. Oxford: Oxford Science Publications; 2001.
6. Perperoglou A, Eilers PH. Penalized regression with individual deviance effects. Comput Stat. 2010;25(2):341–61.
7. Perperoglou A. Cox models with dynamic ridge penalties on time-varying effects of the covariates. Stat Med. 2014;33(1):170–80.
8. Chountasis S, Katsikis VN, Pappas D, Perperoglou A. The whittaker smoother and the moore-penrose inverse in signal reconstruction. Appl Math Sci. 2012;6(25):1205–19.
9. Whittaker ET. On a new method of graduation. Proc Edinb Math Soc. 1923;41:63–75.
10. Ruppert D, Wand MP, Carroll RJ. Semiparametric Regression. Cambridge: Cambridge University Press; 2003.
11. Akaike H. Information theory and an extension of the maximum likelihood principle. In: Selected Papers of Hirotugu Akaike. New York: Springer; 1998. p. 199–213.
12. Burnham KP, Anderson DR. Model selection and multimodel inference: a practical information-theoretic approach. New York: Springer; 2002.
13. Bhat H, Kumar N. On the derivation of the bayesian information criterion. Merced: School of Natural Sciences, University of California; 2010.
14. Craven P, Wahba G. Smoothing noisy data with spline functions. Numer Math. 1978;31(4):377–403.
15. Lambert P, Eilers PH. Bayesian proportional hazards model with time-varying regression coefficients: a penalized poisson regression approach. Stat Med. 2005;24(24):3977–89.
16. Lee Y, Nelder JA. Hierarhical generalized linear models: a synthesis of generalized linear models, random effects models and structured dispersions. Biometrika. 2001;88(4):987–1006.
17. Schall R. Estimation in generalized linear models with random effects. Biometrika. 1991;78(4):719–27.
18. Lee Y, Nelder JA, Pawitan Y. Generalized linear Models with random effects: unified analysis via h-likelihood. Florida: CRC Press; 2006.
19. Douglas Nychka, Reinhard Furrer, John Paige, Stephan Sain. fields: Tools for spatial data Boulder, CO, USA R package version 8.4-1. 2015. doi:10.5065/D6W957CT; http://www.image.ucar.edu/fields. Accessed 2 Mar 2017.