LUNDS UNIVERSITET
Ekonomihögskolan

Regularization Methods and High Dimensional Data:

A Comparative Study Based on Frequentist and Bayesian Methods

Johan Sörstadius and Markus Gerholm
Lund University

Supervisor: Farrukh Javed
Bachelor Thesis in Statistics: 15 ECTS

2024

1 Abstract

As the amount of high dimensional data becomes increasingly accessible and common, the need for reliable methods to combat problems such as overfitting and multicollinearity increases. Models need to be able to manage large data sets where predictor variables often outnumber the amount of observations. In this study the frequentist and Bayesian framework is tested against each other based on three different simulated situations. One where the amount of predictor variables greatly outnumber the observations, one where the simulated data has a high correlation between variables and one where a situation is created where the coefficients to be estimated are known beforehand. This enables comparisons between true values and estimated values. Three different approaches are used from both of the statistical frameworks. The frequentist models consist of Ridge regression, least absolute shrinkage and selection operator (LASSO) regression as well as the combined model Elastic net regression. The Bayesian models consist of three regressions with different prior beliefs regarding the coefficients' probability distributions. The Normal distribution, the Cauchy distribution and the Horseshoe distribution were chosen in this thesis. To compare the different frameworks, different loss functions have been used such as predictability on new data, amount of explained variance and the amount of unnecessary predictor variables the model successfully regularizes. The results of the study show that the Bayesian Horseshoe model has the greatest overall performance regarding predictability, variable selection and parameter estimation. The LASSO regression performs better variable selection on highly correlated data than all of the other models. The frequentist models are also more easily computed if computational power or time is a limited resource, in the other cases the Horseshoe model is to prefer.

Contents

2 Introduction

2.1 Background

The widespread use of digital devices in today's digital era has led to a large increase in the amount of data generated. This growth is not only due to the expanding population but also due to the growing volume of multimedia. Often referred to as Big data, the data market is a billion - dollar industry who serves as a crucial component over multiple sectors (Manyika et al. 2010).

By analyzing historical data one can better understand the present and sometimes even predict the future. In order to forecast future events one method is to create predictive models. Predictive models is trained on past data and consist of one or multiple explanatory variables. As handling data is costly it is therefore important to choose the right data to analyze and select the most relevant explanatory variables. It is important to not include too many explanatory variables in the model, as this can lead to overfitting. Overfitting often occurs when the model is fitted too precisely to the training data, meaning it will include too many variables and become complex. Complex models often lead to worse predictions on new unseen data as the model is not generalized enough (Zhang et al. 2018).

2.2 Purpose

The aim of this thesis is to test already existing regularization and variable selection methods on different sets of high dimensional data. By testing different statistical methods on the same sets of data it is possible to make comparisons between the models. The methods that will be evaluated in this thesis are LASSO (Robert Tibshirani 1996), Ridge (E. Hoerl and W. Kennard 1970) and Elastic net (Zou and Trevor Hastie 2005) as well as Bayesian regression with three different prior distributions: Normal, Cauchy and Horseshoe. The performance of the different models will be evaluated based on loss functions as mean squared error (MSE), mean absolute error (MAE), R-squared (R^2) and the amount of non-zero coefficients the models produce. The results gained from this thesis can assist future decisions when choosing models in high dimensional settings. High dimensional data sets are often complicated and create several difficulties as mentioned in the background. This thesis aims to show which methods combat these difficulties in the most efficient way based on computational restraints, predictability, regularization and parameter estimation.

3 Methods

3.1 Regression

Regression is a statistical method that explores the relationship between variables, aiming to reveal patterns, predicting outcomes and unveiling underlying associations within data sets. One of the most basic and fundamental regression techniques is linear regression, modeling the relationship between a dependent variable and one or multiple independent variables. In simple linear regression, the relationship is mathematically expressed as (James et al. 2021):

$$Y = \beta_0 + \beta_1 x_1 + \varepsilon. \tag{1}$$

In this model, Y represents the dependent variable. The term β_0 is the model's intercept, essentially the value of Y when x_1 is zero. The slope of the line is given by β_1, which is the average increase or decrease in Y for every one-unit change in x_i (James et al, 2023). As a linear model can not capture the data perfectly, an error term ε is included, where

$$\varepsilon \sim \mathcal{N}(0, \sigma^2). \tag{2}$$

The values of the coefficients are determined by the least square estimator (LSE) which sets the value of the coefficients that minimizes the residual sum of squares (RSS), mathematically expressed as:

$$\beta_{\text{LSE}} = \text{argmin} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2. \tag{3}$$

The goal is to set the coefficients so that the linear model is as close to the data as possible. While simple linear regression determines the value of Y based on one independent variable, multiple linear regression uses multiple independent variables and is mathematically expressed as (James et al. 2021):

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \varepsilon. \tag{4}$$

The difference is therefore that the linear model now includes more explanatory variables. In the same way, LSE determines the values of the coefficients, trying to minimize the residual square of sums expressed as:

$$\beta_{\text{LSE}} = \text{argmin} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} (\beta_j x_{ij}))^2. \tag{5}$$

Here, β_j represents the average change in Y for every x_j, where $p > 0$ while keeping all other coefficients constant. The model's intercept β_0 is then given when all x_j are set to 0.

3.2 Bias-variance tradeoff

When predicting new values the complexity of the model is an important factor when valuing accuracy. Low model complexity often implies underfitting which means that explanatory variables that explain relevant properties within the response variable are left out, resulting in poor predictions. On the other hand high model complexity often encourages overfitting which implies poor generalization to new data which also results in less accurate predictions. Figure 1 shows the tradeoff between model complexity and its resulting consequences. Including all variables within a data set equals zero bias as all available information is used. Selecting variables from all available variables introduces bias as not all information is utilized. The total error is in theory minimized when a combination of both is used (Fortmann-Roe 2012).

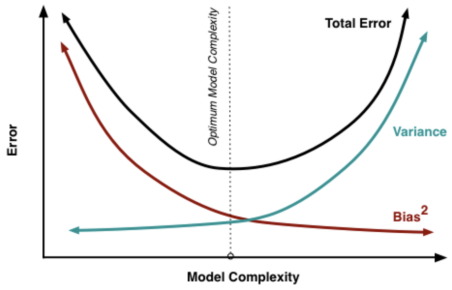

Figure 1: *(Image citation: Fortmann-Roe 2012). Bias-variance tradeoff based on model complexity*

There are several different techniques to introduce bias in order to lower variance when predicting values. In this thesis regularization methods have been used in order to perform variable selection or shrinkage. For high dimensional data this is crucial as variance increases as model complexity increases. Zero bias in this case implies a model based on regular multiple linear regression with a great risk of overfitting the model.

3.3 Ridge

Ridge regularization, first introduced by Hoerl and Kennard in 1970, is a type of regularization technique that is applied to linear regression. This combination is called Ridge regression which is a continuous process that is able to include all predictor variables for a model and aims to minimize the following objective function:

$$\beta_{\text{Ridge}} = \text{argmin} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} (\beta_j x_{ij}))^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\} \tag{6}$$

Which can also be written as:

$$\beta_{\text{ridge}} = \text{RSS} + \lambda \sum_{j=1}^{p} \beta_j^2. \tag{7}$$

Unlike classic linear regression, Ridge regression introduces some bias by adding a penalty term, denoted as "$\lambda \sum \beta_j^2$" from equation 6. The penalty term is a form of regularization which shrinks less influential coefficients towards zero. As with linear regression the model still tries to minimize RSS, but will shrink some coefficients with its penalty term, often referred to as the L2 penalty. Ridge will strive to shrink as much of the coefficients as possible without increasing the RSS too much. This will lead to a simpler model that does not worsen the fit of the line which could help manage problems like multicollinearity. This will help reduce models' overfitting, which occurs when models are too tailored to the training data. Avoiding overfitting is important as this will give a more general model that often performs better on new unseen data.

The amount of penalty that will be included is controlled by the value of λ. When λ is large, the penalty term will have a big impact, increasing the shrinkage of the coefficients. This means that when $\lambda = 0$ there will be no penalty term included, meaning the model will only perform ordinary LSE. However, Ridge does not set coefficients to exactly zero, meaning it will not perform variable selection. This can be a problem when dealing with high dimensional data, as Ridge won't eliminate coefficients that might seem unnecessary for the model (E. Hoerl and W. Kennard 1970).

3.4 LASSO

Least absolute shrinkage and selection operator, often referred to as LASSO, is a regularization technique first introduced by Robert Tibshirani in 1996. It is mathematically expressed as:

$$\beta_{\text{lasso}} = \arg\min \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} (\beta_j x_{ij}))^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{8}$$

which can also be written as:

$$\beta_{\text{lasso}} = \text{RSS} + \lambda \sum_{j=1}^{p} |\beta_j|. \tag{9}$$

As with Ridge regression, β_{lasso} is a vector of estimated coefficients, β_0 represents the intercepts, and $\lambda \sum |\beta_j|$ is a penalty term for the model otherwise called L1. LASSO also works as a regularization technique as it shrinks coefficients. The difference is that LASSO can set coefficients to exactly zero, meaning it can also be used for variable selection. Variable selection is an important technique to use when handling large data sets, as this can cope with overfitting by removing variables that are less important for the model. By incorporating

LASSO when creating prediction models, only the most influential coefficients will be included, and the rest set to zero. This leads to simpler models that are more generalized. As with Ridge regression, the value of λ will determine the amount of penalty that will be used. The value of λ can take any value from 0 to infinity, where the optimal value is found through cross-validation. When dealing with highly correlated data, LASSO tends to favor one of the correlated variables and removes the rest, which can lead to the loss of valuable information for the model. Another disadvantage LASSO has is that it can only select as many coefficients as there are observations, which becomes problematic when coping with high-dimensional data with more variables than observations (Robert Tibshirani 1996).

3.5 Elastic net

Elastic net is a regularization method that was first introduced in 2005 by Zou and Hastie (Zou and Trevor Hastie 2005). Elastic net works as a combination of LASSO and Ridge regression, incorporating both models' penalty terms and is mathematically expressed as:

$$\beta_{\text{net}} = \arg\min \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} (\alpha|\beta_j| + (1-\alpha)\beta_j^2) \right\}. \quad (10)$$

In this equation, the parameter α decides how much of the L1 and L2 penalty that should be included. When $\alpha = 1$, the elastic net model will be equivalent to LASSO as only the L1 penalty will be included. When $\alpha = 0$, this shifts, meaning only the L2 penalty will be included, and the model will behave like a Ridge regression. When α is between zero and one, Elastic net will combine both penalties, taking advantage of both the L1 and L2 penalty terms. The complementary approach of Elastic Net can address the limitation of one penalty term through the other. One major issue for Ridge regression is overfitting, which the penalty term from LASSO can solve by shrinking coefficients to zero, eliminating them from the model. Elastic net can also tackle the challenge with multicollinearity that LASSO struggles with as the L2 penalty from the Ridge component can group strongly correlated variables and assign them similar coefficients (Zou and Trevor Hastie 2005).

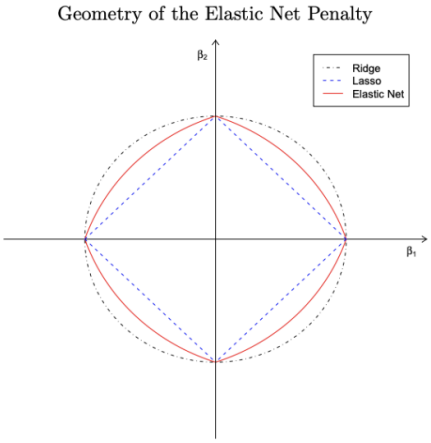

Figure 2: *(Image citation: Zou and Trevor Hastie 2005). Geometric shape of boundaries for coefficients for Ridge, LASSO and Elastic net*

Figure 2 illustrates the geometric constraints for LASSO, Ridge and Elastic net on regression coefficients. The geometric shapes are the boundaries where the coefficients must reside. As LASSO and Elastic net has corners it can set the coefficients to exactly zero, which Ridge can not due to its geometric shape of a circle.

3.6 Cross validation

Cross validation is a statistical technique that is used when developing predictive models. The objective is to create a model, trained on some specific data, that makes accurate predictions for future outcomes. Cross validation means that you split a data set into k-subgroups. The predictive model is trained on some specific subgroup and is then tested on another subgroup to evaluate its performance. This is repeated for different subgroups, meaning multiple predictive models are created and tested. The best model is then selected based on the lowest mean square error. This method is used for the frequentist regressions that have been presented, it is performed automatically with ten folds when fitting a general linear model in the R package *glmnet* (Friedman, Tibshirani, and Hastie 2010).

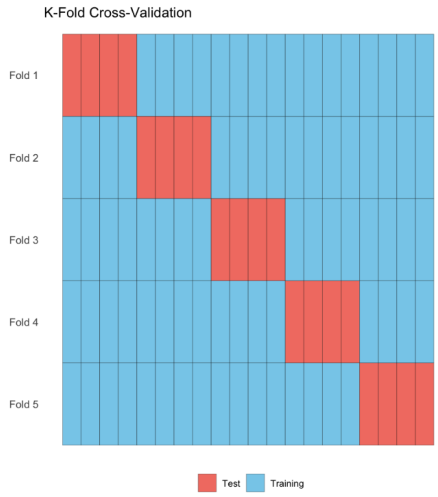

Figure 3: *Example of k-folded cross-validation, created in R*

Figure 3 illustrates an example of K-folded cross-validation. Each model, one for every stack, is trained on the k-1 subgroups (marked blue) and then evaluated on the remaining group (marked red). This systematic approach results in a robust model selection in order to ensure the model's effectiveness. Here k represents the amount of subgroups the data is split into.

3.7 Bayesian regression

The frequentist framework that was presented earlier differs from the Bayesian framework. In Ridge, Elastic net and LASSO an important assumption is that there exists a true value for all point estimates, in this case β_j. The estimation of the parameters will however differ depending on what data that is observed, this signifies the random aspect of the framework. The Bayesian perspective handles coefficient estimation in a slightly different way, instead of assuming a true value for the parameters one assumes that all parameters are random and follow a probability distribution. This distribution signifies the random aspect of the method and is based on the data that is observed, this data is considered fixed. This alternate way of estimating parameters creates the opportunity to update coefficients' distributions as more data is collected or as stronger prior beliefs about the coefficients distributions are gained (Muth, Oravecz, and J. Gabry 2018). The Bayesian framework is based on Bayes theorem which follows as:

$$P(\theta|y) = \frac{P(y|\theta) \times P(\theta)}{P(y)}. \tag{11}$$

The posterior distribution is the objective of Bayesian models and is represented on the left side of the equation. It signifies the distribution of coefficients, conditioned on the response variable. In simpler terms, it represents the distribution of β values based on the observed data. To calculate this, Bayes theorem can be utilized where the posterior distribution is equal to the probability of the observed data conditioned on the parameters multiplied by the distribution of the coefficients divided by the distribution of the observed data. The denominator is in practice a scaling constant that ensures that the posterior distribution when integrated equals one. Since the constant does not change the shape of the distribution it is often removed from the calculations which results in the posterior distribution being proportional to the right side of equation 11 rather than equal to as seen below:

$$P(\theta|y) \propto P(y|\theta) \times P(\theta). \tag{12}$$

The probability of the observed data conditioned on the coefficients is in turn proportional to the likelihood of the data being observed, given the coefficients are known:

$$P(y|\theta) \propto L(\theta) \tag{13}$$

The likelihood function can then be utilized after some minor algebraic reasoning (Gelman et al. 2021):

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \varepsilon \tag{14}$$

by rearranging the terms in the multiple regression we find that:

$$\varepsilon = y - \beta_0 - \beta_1 x_1 - \beta_2 x_2 - \ldots - \beta_p x_p \tag{15}$$

by assuming that the error term is normally distributed the likelihood function can then be expressed as:

$$L(\theta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \beta_0 - \beta_1 x_{i1} - \ldots - \beta_p x_{ip})^2}{2\sigma^2}\right). \tag{16}$$

To utilize the likelihood function the distribution of the coefficients $P(\theta)$ must be known, this is were the frequentist and Bayesian frameworks differ the most. Based on prior beliefs or knowledge the distribution of the β coefficients are assumed to be known. This is called the prior distribution and can follow any probability distribution for any value of β. This is then utilized in the likelihood function and in the prior distribution that is the distribution of the coefficients in order to obtain the posterior distribution. The prior distributions that have been used in this thesis give name to the different models, the Normal model, the Cauchy model and the Horseshoe model:

$$P(\theta_1) \sim \mathcal{N}(\mu = 0, \sigma = 1) \tag{17}$$

$$P(\theta_2) \sim \text{Cauchy}(x_0 = 0, \gamma = 0.5) \tag{18}$$

$$P(\theta_3) \sim \text{Horseshoe}(\tau = 1). \tag{19}$$

The posterior distribution is therefore different for all three models and produce different coefficient estimates, predictability and sparsity results for different data sets. The three different prior distributions are similar but have differing amounts of density both centered around zero and in the tails of the distribution which can be seen in figure 4 below. The Horseshoe distribution have more density around its tails encouraging values significantly different from zero to have larger values which can be a relevant property when encouraging sparsity in predictor variables (Carvalho, Polson, and Scott 2010).

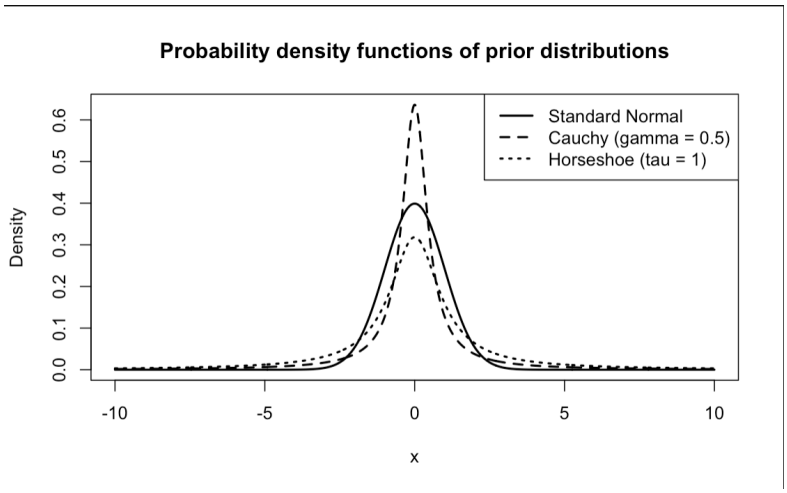

Figure 4: *Prior distributions probability density functions*

In order to obtain an estimate of the posterior distribution Markov Chain Monte Carlo (MCMC) sampling is performed. This is done since integrating over all coefficients create a very complex high dimensional problem as the amount of predictor variables increases. Using the MCMC method relatively high accuracy can be obtained when estimating the posterior distribution. Different algorithms are used when sampling to ensure efficiency, in the R packages *rstan* and *brms* the *No-U-Turn* Sampler is used which effectively explores the posterior probability distribution by tuning its parameters automatically (Bürkner 2017; Stan-Development-Team 2023).

The MCMC method starts with a chain that randomly moves through the parameter space in the posterior distribution. The next move that the chain chooses is based on different weights and probabilities that are out of the scope of this bachelors thesis. In this thesis four independent chains are used to gain an understanding of the probability distribution at hand, all chains start in different areas of the parameter space and converge at the same probability distribution if the sampling is done correctly. In order to gain accurate estimates the first half of the samplings are regarded as a warm-up phase and are disregarded for the parameters estimations. In this thesis 2000-3000 iterations are performed where the first 1000-1500 samples are discarded as warm-up samples while the second half of the iterations are considered part of the posterior distribution. To check if the correct posterior distribution has been estimated the package *rstan* calculates a type of loss function called $\hat{R}$ (Gelman et al. 2021).

$$\hat{R} = \sqrt{\frac{\text{Var}(\theta|y)}{W}} \tag{20}$$

The numerator is the variance of the estimated posterior distribution conditioned on the observed data, the denominator is the average of all the Markov chains' variances. $\hat{R}$ calculates the ratio of within chain variance and between chain variance. If the chains converge to the same posterior distribution the square root of the ratio equals to one. A common criteria is to increase the amount of iterations if the $\hat{R}$ value is above 1.1, in all tests where Bayesian models have been used the mean of $\hat{R}$ hat has not exceeded 1.1 which indicates that all chains have successfully converged to the same posterior distribution (Gelman et al. 2021).

In order to perform predictions on new data, N samples from the posterior distribution are taken in order to test many parameter values for the new predictor variables. The parameter values are then used in the regular regression equation in order to predict the new values for the response variable based on the new values for the predictor variables:

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{k}} = (\hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_N) \tag{21}$$

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \ldots + \hat{\beta}_p x_p + \hat{\varepsilon}. \tag{22}$$

This process generates a distribution of response variable values for each β_j since every β on its own is a vector of sampled values from its own distribution. In order to compare predictability with frequentist models that generate point estimates a simple mean is calculated for each distribution of response variables. This approach enables comparability since the same loss functions can be used on all models (Gelman et al. 2021).

Lastly samples from the computed posterior distributions can be plotted against the actual data that has been observed in order to check if the model fit the data well enough to make relevant predictions. If the variance between the different samples and the real data is low predictions and inference can be performed with higher certainty that the model capture the response variables variance effectively. This can be seen during the results section and was performed by using the *bayesplot* package in R (Gabry and Mahr 2022).

3.8 Loss functions

Several different loss functions have been used in order to check the different models predictability and ability to select variables, the functions are expressed mathematically below before the actual values for the loss functions are presented.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|. \tag{23}$$

Equation 23 calculates the mean absolute error between new data from the testing data set and the predicted data from the model which gives less bias to outliers than the mean squared error that will be presented shortly. A small value indicates that the predictions are successful since the predicted value is close to the observed value.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2. \tag{24}$$

Mean squared error is similar to mean absolute error but creates a bias towards larger differences between observed values and predicted values since the difference is squared. MSE is therefore efficient when wanting to punish outliers more than smaller values since the loss function represents outliers with more weight. A small value indicates that the model can successfully predict new values relatively well.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}. \tag{25}$$

The ratio of the mean squared error and total variance in the response variable is also a way of measuring predictability. A value close to one indicates that the model fit is successful and that it predicts values with small variance.

$$\text{Non-zero coefficients} = \sum_{i=1}^{p} I(\beta_i \neq 0). \tag{26}$$

This calculates the amount of β coefficients that are non-zero where p equals the amount of predictor variables. For a model such as Ridge regression where no coefficients are set to exactly zero this equals to the amount of predictor variables in the data. For the Bayesian models the approach is slightly adjusted, since the model's coefficients are based on continuous probability distributions, coefficients that are smaller than 0.01 are set to zero when counting them in order to enable comparisons between the different frameworks.

4 Data

In this study the data was simulated using various R packages and functions. This implies total control over the experiment design and makes comparisons between models less complicated and also enables very specific situations to be tested. Three different sets of data were generated with different properties which all models then were trained and tested on. The packages used are referenced below all at once since citing them for each function would deem excessive (Gabry and Mahr 2022; Qiu and Joe 2023; Wickham, François, et al. 2023; DeBruine 2023; Stan-Development-Team 2023; Wickham, Vaughan, and Girlich 2023; Wickham 2016).

4.1 High dimensional data

The first data set was created by sampling 1000x3000 observations from a Normal distribution with expected value zero and standard deviation equal to one, this was created in matrix form. To create a dependency in the response variable a new vector was created where every value was equal to the sum of the first ten predictor variables, this was summed row wise plus a normally distributed error term with expected value zero and standard deviation one expressed as:

$$\begin{aligned} Y_1 &= X_{1,1} + X_{1,2} + X_{1,3} + \cdots + X_{1,10} + \epsilon \\ &\vdots \\ Y_n &= X_{n,1} + X_{n,2} + X_{n,3} + \cdots + X_{n,10} + \epsilon. \end{aligned}$$

This was done in order to replicate a realistic situation more accurately. The dependency created in the response variable allows for control in the experiment and ensures that the models regularization can be checked and compared. An optimal model would therefore set non-zero β coefficients to the first ten variables and assign all other 2990 variables exactly to zero or very close to zero. This would indicate that the dependency in the response variable is captured by said model. Apart from the variable selection aspect, predictability was also tested for each model. 70% of the data was sampled randomly and used to train the models while the remaining 30% of the data was used to test the models efficiency in regards to prediction. The predictability was then calculated through three different loss functions which was presented in the methods section.

4.2 Highly correlated data

The second set of data was generated to replicate a situation with high correlation. A 500x500 correlation matrix was created with 0.8 as assigned values except for the diagonal elements which represents the correlation between the same element which therefore have the value one assigned. This matrix was then used when generating 500x500 samples from a multivariate Normal distribution with expected value zero and the correlation matrix as standard deviation. This allowed for normally distributed data to be generated with pair wise

correlations of around 0.8 which indicates a high positive relationship between variables. This was created through the function *mvrnorm* which is included in the package *MASS* (Venables and Ripley 2002). The row wise sum of the first ten predictor variables were equal to the response variables of the data as in the high dimensional case. The data was then split into training data which accounted for 70% of the simulated data and the remaining 30% of the data was used for testing in order to evaluate predictability on new data for each of the models.

4.3 Data generated from known β coefficients

The third data set was created in a different way, firstly a matrix with 200x200 normally distributed samples was created with expected value zero and standard deviation one. Then a vector with 200 β coefficients was created where the first 100 values were equal to three and the remaining 100 values were equal to zero. The response variable was then generated by multiplying the β coefficients to each normally distributed predictor variable creating a weighted sum where the first half of the coefficients had value three and the other half of the sum was weighted by the value zero. A noise term which was normally distributed with expected value zero and standard deviation one was also added to each element in order to represent a more realistic situation, otherwise the response variable would only depend on the first half of the predictor variables. This was conducted in order to test the different models' accuracy when estimating β coefficients since the simulated situation is created to know the true values of the coefficients. Since all values for β are known, loss functions can be used to test how close the estimated coefficient is to the true value. This allows for testing the accuracy relatively easily for each of the six different models.

5 Results

5.1 High dimensional data

Alpha	MAE	MSE	R2	Nonzerocoeff
0	2.44939753184898	9.49632699109864	0.176580070240599	3000
0.05	1.35158143729761	2.93368306863697	0.745622374990067	525
0.1	1.05860213983356	1.81112440693703	0.842958658295623	299
0.15	0.983460150006188	1.53684505595018	0.866741230666576	184
0.2	0.926189516156763	1.36237926386746	0.881869038543962	189
0.25	0.902855645840868	1.29519671915293	0.887694390419666	168
0.3	0.887185240192516	1.25045676905341	0.891573760475364	158
0.35	0.876453280266066	1.22313322018856	0.893942966454493	148
0.4	0.867872234428861	1.20092760305955	0.89586840012098	143
0.45	0.864490920728344	1.19009872714007	0.896807364444486	116
0.5	0.856336213448743	1.17342326376671	0.898253282312781	154
0.55	0.854619399914131	1.16619791590352	0.89887978721679	110
0.6	0.853981042701764	1.16252436507448	0.899198318262367	82
0.65	0.84685230852397	1.15030666252846	0.900257706779812	121
0.7	0.84865325149221	1.15026503235265	0.900261316503494	71
0.75	0.842632714672615	1.14292678054223	0.900897611230493	152
0.8	0.84381298268364	1.13930586774694	0.901211578068657	68
0.85	0.841862553715409	1.13502642229589	0.901582645816852	68
0.9	0.842125125193558	1.13536948853055	0.901552898781486	59
0.95	0.838836097063508	1.12813819155107	0.902179919529244	66
1	0.842016395611209	1.13483755921459	0.901599022003696	48

Figure 5: *Frequentist results, highlighted areas show from top to bottom the results from Ridge, Elastic net and LASSO*

Firstly the results from the high dimensional data set are to be presented. The frequentist linear regressions with different regularization methods begin this section and are followed by the Bayesian regressions. In figure 5, when the value for α equals zero the frequentist regression that is performed is the earlier presented Ridge regression. The table as well as the graph show that mean square error decreases as α increases, this also applies to mean absolute error. The value of α that minimizes both MSE and MAE is 0.95 which is a combination of both penalty terms in Ridge and LASSO regression which was presented as Elastic net regression previously. The Elastic net model is greatly dominated by its LASSO component but still outperforms the pure LASSO model where α equals one. An obvious consequence of the minimized MSE value is the maximized value for R^2 which also is located in the Elastic net model. The properties of the different models can be seen by the amount of non-zero coefficients that the models produce.

Ridge sets no variables to exactly zero by default which can be seen by its 3000 variables that all are above or below zero. However the model sets greater values for the first ten variables and values close to zero for the remainder of the variables. This indicates that the model has picked up the created dependency in the response variable successfully. The Elastic net model that performs the best sets 66 predictor variables to non-zero values and the remaining 2934 variables to exactly zero which is typical of the LASSO component of the model. Lastly

the LASSO model encourages the least amount of non-zero coefficients with a count of 48 variables and therefore successfully eliminates 2952 random noise terms. Both Elastic net and LASSO sets larger values for the first ten predictor variables which indicates that all three models capture the dependency structure that was simulated.

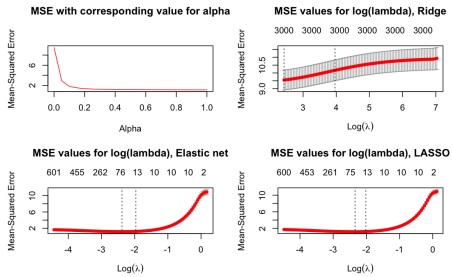

Figure 6: *Mean squared error for frequentist models*

Figure 6 shows the decreasing MSE for increasing α and for which $\log \lambda$ that minimizes the mean squared error for each model. To conclude the frequentist models' performance the results show that Ridge performs the worst based on predictability and variable selection, Elastic net predicts values on new data the best and LASSO outperforms the other two models based on regularizing variables that contain no relevant information.

Prior	MAE	MSE	R2	Nonzerocoefficients
Normal	2.385616	9.070317	0.2135191	2166
Cauchy	2.371013	9.107553	0.2102904	2183
Horseshoe	0.802268	1.066154	0.9075545	31

Figure 7: *Bayesian results*

Moving on to the Bayesian models with prior distributions Normal(0, 1), Cauchy(0, 0.5) and Horseshoe(0, 1) the results vary greatly between the Horseshoe model compared to the other two models. In figure 7 the table shows that the Normal model has the second smallest MSE (therefore second highest R^2) and the highest MAE of all three models. The model manages to set higher values for the first ten predictor variables, successfully discovering the simulated dependency. The model does however perform regularization worse than the other models, setting 834 coefficients to zero and keeping 2166 variables.

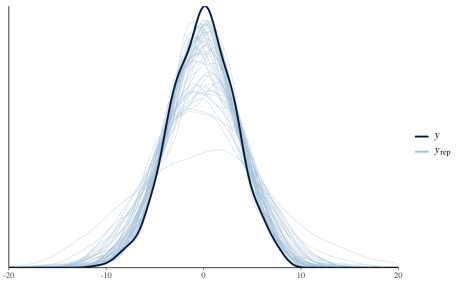

Figure 8: *Posterior distribution of Normal model*

The posterior distribution in figure 8 shows that the sampled data from the model resembles the observed data, the variance however is relatively high since there are several samples from the 50 draws which does not fit the observed values very well. This is enforced by the high MSE value indicating poor predictive performance. As stated earlier, Bayesian models are unable to set coefficients to precisely zero, to combat this a small amount of bias was introduced in order to create comparability between the frequentist and Bayesian frameworks by including a threshold of 0.01 which equals coefficients to zero when counting the amount of coefficients. This arbitrary threshold should be kept in mind when interpreting these results. The Cauchy model has a slightly smaller MAE value and slightly higher MSE value and performs regularization in the least effective way setting 817 variables to zero and keeping 2183 coefficients in the model. The model manages to capture the dependency and sets larger values for the first ten predictor variables.

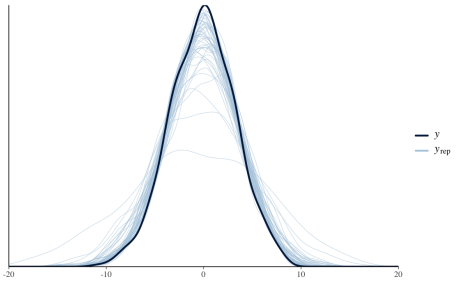

Figure 9: *Posterior distribution of Cauchy model*

The posterior distribution from the model with a Cauchy distributed prior distribution in figure 9 shows similar patterns as the posterior distribution from the Normal model, high variance compared to the observed data which explains the high MSE value. The Horseshoe model outperforms the other two Bayesian models greatly by minimizing both MAE and MSE and therefore maximizing R^2. The model successfully sets larger values for the ten first variables and sets 2969 variables to zero while keeping only 31 variables above the chosen

threshold.

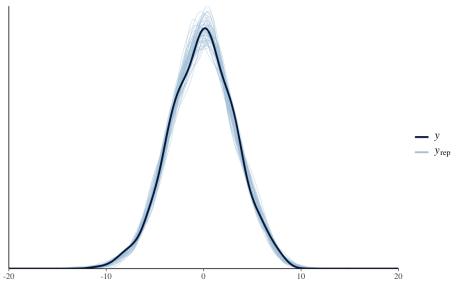

Figure 10: *Posterior distribution of Horseshoe model*

The posterior distribution shown in figure 10 illustrates a smaller variance compared to the other two Bayesian models indicating a more successful fit. This in turn explains the significantly smaller mean square error and mean absolute error. The model also manages to outperform the Elastic net model by gaining a smaller prediction error and successfully selecting fewer variables from the large number of irrelevant variables than the LASSO model.

5.2 Highly correlated data

Alpha	MAE	MSE	R2	Non_zero_coeff
0	1.36763075331895	3.12154423998471	0.957571313731482	500
0.05	1.07463824709535	1.86940400073909	0.974590667388119	131
0.1	1.02332079822906	1.60749531463308	0.978150585370844	84
0.15	1.00129592751498	1.49705682134996	0.979651688614378	68
0.2	0.98985907615618	1.44620571091865	0.98034286760946	60
0.25	0.978425446196481	1.40499735185717	0.980902980298514	54
0.3	0.969514687675131	1.37606105385116	0.981296288550931	51
0.35	0.962922561755407	1.35406455510281	0.981595269591291	50
0.4	0.958008166281024	1.33491954229934	0.981855492634565	48
0.45	0.953722859811179	1.32131984595119	0.982040342569515	48
0.5	0.95026502572785	1.30998792967676	0.982194368360426	45
0.55	0.947856768014014	1.3024990814333	0.982296158361851	44
0.6	0.945782306071094	1.29578256898996	0.982387450612534	44
0.65	0.94397885882271	1.29070380573995	0.982456482231503	44
0.7	0.942592185285635	1.28657796585493	0.982512561515542	44
0.75	0.941145869538359	1.28222787100997	0.982571688920193	42
0.8	0.939890151578279	1.27848880855716	0.982622511043973	42
0.85	0.9386141663617	1.27475902977512	0.982673206982146	42
0.9	0.937673130972767	1.27204405850098	0.982710109443096	42
0.95	0.936693915686365	1.26933783916539	0.982746892945855	42
1	0.936052266629982	1.26736111022261	0.982773761061666	41

Figure 11: *Frequentist correlated results, highlighted areas show from top to bottom the results from Ridge, Elastic net and LASSO*

When analyzing the results from the highly correlated data set shown in figure 11 the Ridge model once again manages to capture the dependency in the first ten variables but performs predictions on new data in a worse manner than the remaining models. The model also keeps all variables in the data set based on the same arguments as stated before. MAE and MSE are minimized as α increases as in the earlier data set but this time the pure LASSO model outperforms the different Elastic net models by a slight amount. Both models capture the dependency in the first ten predictor variables and the LASSO model manages to eliminate one more irrelevant variable than the best performing Elastic net model where α equals 0.95. The Elastic net model sets 458 variables to zero and keeps the remaining 42 variables in the model. The LASSO model sets 459 variables to zero and keeps the remaining 41 variables in the model. The results show that the LASSO model outperforms the Elastic net model when data is highly correlated, however by only a small amount. All frequentist models perform relatively well since all values for R^2 are above 95%.

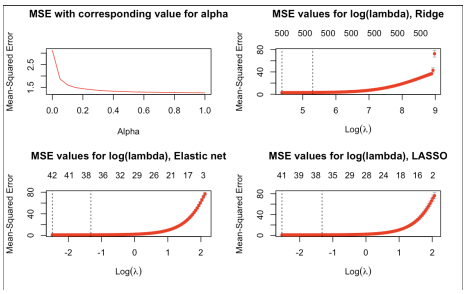

Figure 12: *Mean squared error for frequentist models*

Figure 12 shows as previously which value for $\log \lambda$ that minimizes MSE for each model as well as MSE plotted against values for α which shows a similar pattern where MSE decreases as α increases.

Prior	MAE	MSE	R2	Non_zero_coefficients
Normal	1.51588092757432	3.74541030686405	0.949091594851916	479
Cauchy	1.40673847116648	3.26612195731935	0.955606182969715	474
Horseshoe	0.86636585603358	1.1173506510539	0.984812734793814	102

Figure 13: *Bayesian correlated results*

The Bayesian results in figure 13 show that the Normal model performs the worst out of all three models, obtaining the highest value for MAE, MSE and non-zero coefficients. The model does however capture the dependency in the first ten variables but only manages to set 21 variables below the threshold.

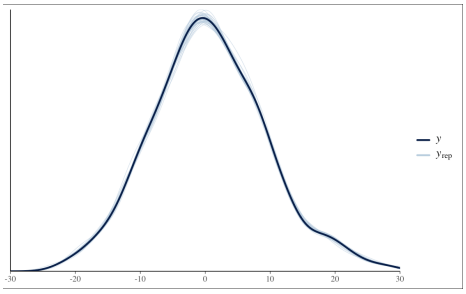

Figure 14: *Posterior distribution of Normal model*

The posterior distribution in figure 14 of the Normal model shows a good fit when sampling data and comparing it to the observed data. This results in a lower MSE and MAE than the previous data set where higher variability could be seen. The Cauchy model lowers the values for MAE and MSE slightly and manages to set 26 variables below the threshold as well as capturing the dependency in the response variable, a somewhat increased performance compared to the Normal model. Both models obtain a relatively high R^2 value which also can be seen in the frequentist models.

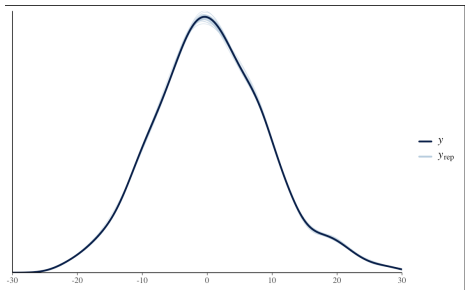

Figure 15: *Posterior distribution of Cauchy model*

The posterior distribution of the Cauchy model in figure 15 exhibits the same promising results as the Normal model showing small variance when compared to the observed values. Lastly the Horseshoe model outperforms the other Bayesian models again by minimizing both MAE, MSE and maximizing R^2 at around 98%. The model captures the dependency in the first ten predictor variables and sets 398 variables below the threshold and 102 variables as non-zero coefficients.

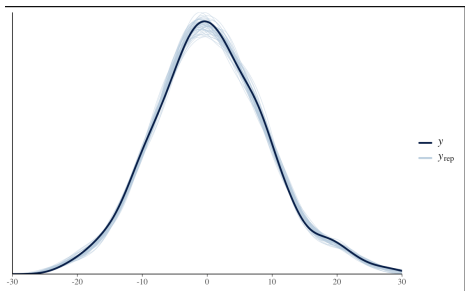

Figure 16: *Posterior distribution of Horseshoe model*

The posterior distribution of the Horseshoe model in figure 16 shows an increased variance from the sampled values compared to the other two models which can be contradictory since the models' loss functions perform with highest rate of success. This is most likely due to the fact that the model shrinks many predictor variables to zero which outweighs the small increase in variance. Between the frequentist and Bayesian models, the LASSO model performs variable selection with highest efficiency and the Horseshoe model performs predictions on new data with the highest efficiency, the results seem to differ between the simulated data sets.

5.3 Data generated from known β coefficients

Alpha	Non_zero_coefficients	MSE	MAE
0	200	2.97775204723204	1.34769086499891
0.05	171	1.44454779916137	0.884943183690418
0.1	163	1.47429450942212	0.872651342232151
0.15	156	1.56899547724423	0.891181866945009
0.2	155	1.67037588775497	0.918865242768405
0.25	150	1.78084964376854	0.946594275095203
0.3	149	1.89556795093857	0.977493199929852
0.35	147	2.00997029985637	1.00558647214529
0.4	145	2.10903344887882	1.02997978492754
0.45	140	2.18486980177255	1.0477321109523
0.5	106	2.43961515023711	1.0425555221786
0.55	140	2.30648407488436	1.07860293600706
0.6	136	2.35999448506026	1.09225614661749
0.65	127	2.44379356537845	1.09685560161684
0.7	133	2.42306174950712	1.10930387251713
0.75	130	2.44689367652543	1.1155536095209
0.8	129	2.45821195977162	1.11871222475243
0.85	116	2.49748328570999	1.09842770374986
0.9	130	2.47716845013073	1.12341222093981
0.95	130	2.48454108152261	1.12525035945992
1	130	2.50526697025316	1.12803034900291

Figure 17: *Frequentist β estimation results, highlighted areas show from top to bottom the results from Ridge, Elastic net (two models in this case) and LASSO*

Lastly the results from the third data set are to be presented, mean square errors and mean absolute errors have been calculated in order to compare the different models accuracy in parameter estimation and can be seen in 17. The MSE is calculated based on the β coefficients for each model and for each value of α (for the frequentist models). This is then repeated for the MAE values in order to compare the models since the test is constructed in a way where the predicted value is compared to the true value. The difference with regular MSE and MAE values in this case is that the loss functions calculated on β coefficients measure accuracy in coefficient estimation instead of accuracy in predicting new values for the response variable based on new data. In theory however both loss functions measure the same thing, how well a model estimates a value compared to its true value.

$$\text{MSE} = \frac{1}{p}\sum_{i=1}^{p}(\beta_i - \hat{\beta}_i)^2 \tag{27}$$

$$\text{MAE} = \frac{1}{p}\sum_{i=1}^{p}|\beta_i - \hat{\beta}_i| \tag{28}$$

Since the true values for the β coefficients are known, the models' estimations of the coefficients can be compared to each other and the scenario where a model

performs perfectly. Since this section purely focuses on the models ability to accurately estimate coefficients the earlier presented loss functions regarding predictability has been ignored. The frequentist models estimate coefficients with differing success, the Ridge regularization model is relatively far from the true values of β with the highest calculated MSE and MAE indicating a poor performance of estimating coefficients. The elastic net model that has greatest accuracy in estimating coefficients is the one where α equals 0.05 and 0.10 both minimizing MSE and MAE respectively, this ratio of both penalty terms performs the best of all the frequentist models and therefore estimate the coefficients at highest accuracy. Lastly the LASSO model's results show that it is performing better than the Ridge model and worse than the Elastic net model with optimized value for α. The LASSO model also sets the highest amount of coefficients to zero, successfully setting 70 coefficients to zero, the optimal value for non-zero coefficients would be 100 since half of the true values are non-zero. To conclude the frequentist models' coefficient estimations, the Elastic net models with α value 0.05 and 0.10 minimizes the error terms and has the best overall accuracy when capturing the underlying β coefficients that were assigned when simulating data.

Prior	Nonzerocoeff	MSE	MAE
Normal	196	3.20112263611024	1.37814964347803
Cauchy	197	2.42816769868252	1.16265389462134
Horseshoe	198	1.40234259453136	0.939277665574649

Figure 18: *Bayesian β estimation results*

In the Bayesian framework figure 18 shows that the Normal model has the worst accuracy of all the tested models estimating the coefficients in a poor manner generating relatively high values for both MSE and MAE. The Cauchy model improves the performance somewhat by lowering MSE and MAE and therefore outperforming the Normal model, Ridge model and LASSO model except for the LASSO model's MAE value. Finally the Horseshoe model has the most efficient estimates of coefficients when comparing the Bayesian models, successfully minimizing both MSE and MAE. All models also fail to set approximately half of the coefficients to zero, keeping all but three or four coefficients as non-zero. To summarize the Bayesian results, the Horseshoe model performs the best, capturing the underlying structure of the β values and therefore generating the most accurate estimates. When comparing the two different frameworks the frequentist Elastic net model minimizes the mean absolute error, the Horseshoe model minimizes the mean squared error and the LASSO model regularizes the highest amount of β coefficients with true value equal to zero.

6 Discussion

There was no distinct model that performed the best on all data sets, however there is clear evidence that the Elastic net, LASSO and Horseshoe models perform more efficiently than the other models. Regarding the high dimensional data set the Elastic net model outperformed the other two frequentist models regarding predictability while the LASSO model performed variable selection at highest success. The Horseshoe model outperformed all Bayesian models as well as the frequentist models regarding both predictability and regularizing random noise variables. The Bayesian approach with this certain prior distribution for β coefficients seems to be the most effective model. This applies to a situation where data is approximately normally distributed, where there are more predictor variables than observations and where the majority of predictor variables explain the variance in the response variable poorly.

The highly correlated data set creates similar results for the Bayesian models but changes slightly for the frequentist models. The LASSO model minimizes both MSE and MAE while also selecting the smallest amount of β coefficients of all models, successfully performing variable selection in an effective way. The Bayesian models with Normal and Cauchy distributed coefficients perform worse than all frequentist models in predictability, the Horseshoe model however outperforms the LASSO model when predicting new values by lowering both MSE and MAE slightly. The Horseshoe model does however perform variable selection in a less effective way, the choice is therefore between simplicity and predictability when choosing between frequentist and Bayesian models in a highly correlated setting. As mentioned earlier in the method section, LASSO often performs with less success when multiple variables are highly correlated setting correlated variables to zero while keeping one of the variables. In this case it does not however set any of the ten first predictor variables to zero even though the measured pair wise correlation is around 0.8 between the variables. The constructed dependency in the response variable could therefore have such a big impact that the model does not regularize the variables that explain the variance in the response variable even though the pair wise correlation is high. This works in favour of the LASSO model since important information is not regularized and set to zero, explaining the low mean squared error.

Lastly the third data set results are to be analyzed. Two Elastic net models estimate β coefficients with the greatest accuracy compared to the Ridge and LASSO models depending on which loss function one chooses to prioritize, however both models have an α value with a difference of 0.05 indicating a small LASSO component and a large Ridge component, both models are therefore in practice very similar. The LASSO model performs variable selection with highest success as well which does not come as a surprise. The Horseshoe model outperforms the other two Bayesian models once again and minimizes MSE for all models tested, however the Elastic net models minimize MAE in a more efficient way. The choice of model when valuing accurate estimates of β

therefore depends on how much the presence of outliers affects the choice. MSE squares the difference between the estimate and the true value, larger differences therefore weigh more when squared indicating outliers and their effect on the model. MAE only measures the absolute difference between the true value and the estimated value while not giving outliers a larger weight in the loss function. The optimal model in this setting therefore depends on the data and how larger deviants from the true value should be incorporated in the measurement of estimation accuracy.

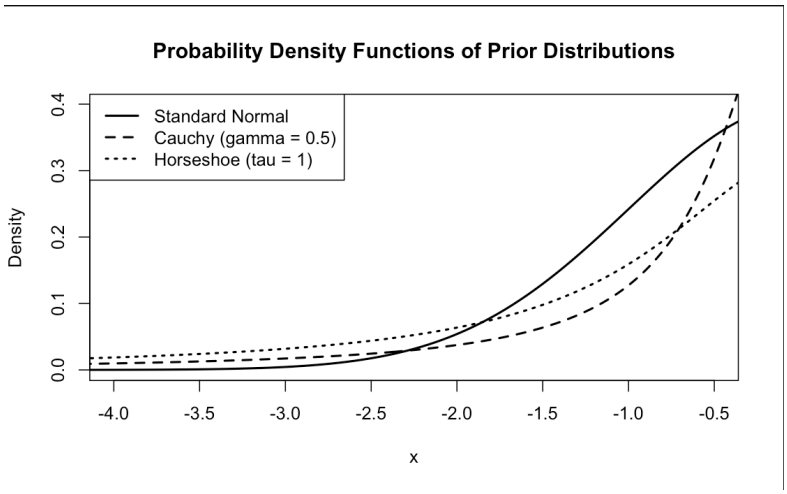

Figure 19: *Tails of prior distributions probability density function*

The Horseshoe models' performance can be explained in figure 19 which is a zoomed in part of figure 3 where the prior distributions tails are plotted against each other. More probability mass exists further out in the tails of the distribution and lesser mass is centered around zero. In practice this encourages β coefficients that are important to be far from zero and β coefficients that explain a small amount of the variance in the response variable to be close to zero. Comparing the tails of the distributions, the Normal distribution have more mass around its mean which enables many coefficients to have a wider span of values around zero even though they optimally would be centered closer to zero. β coefficients with higher values have a smaller amount of probability mass than the Horseshoe model and the Cauchy model which does not encourage sparsity. The Cauchy distribution is similar in practice to the Normal distribution but has a larger portion of its probability mass centered closer to zero while having lower probability mass for higher values of x_i. The amount of density in the distributions tails seems to be the largest factor when Bayesian regression models are applied to high dimensional data. This can be seen as the Horseshoe model encourages the most sparsity when estimating β coefficients and therefore

carries out more accurate predictions on new data.

To conclude the discussion, the Bayesian Horseshoe model shows promising results regarding predictability, outperforming all other models when analyzing mean squared error when predicting new response variable values based on new data as well as estimating β coefficients. The Horseshoe model also regularize the highest amount of noise terms in the first high dimensional data set. The LASSO model regularizes the highest amount of irrelevant predictor variables when data is highly correlated and minimizes the mean absolute value when estimating β coefficients. The Horseshoe model therefore seems to be the most versatile since it performs well on all three data sets. A con that should be addressed is however the computational time for Bayesian models. The smaller data sets regarding high correlation and β estimation are estimated relatively efficiently but for larger data sets such as the high dimensional data set the MCMC sampling method requires strong computational power in order to perform efficiently. For data sets even larger than the simulated data sets that were tested in this thesis the frequentist models might be preferred in order to save computational power and time.

7 Summary

This thesis has compared the frequentist framework with the Bayesian framework by testing three different linear regression models from each framework. Ridge, LASSO and Elastic net was tested against Bayesian regression with Normally distributed, Cauchy distributed and Horseshoe distributed prior distributions. All six models have been evaluated by calculating four different loss functions: mean square error, mean absolute error, R^2 and the amount of non-zero coefficients that the model saves. The models have been trained and tested on three different sets of data, one where the predictor variables exceeds the amount of observations causing a high dimensional setting, one where high correlation is created through a correlation matrix and one where the true values of the β coefficients are known, enabling comparisons between the estimated and the true values between models. Different models perform with differing efficiency depending on what data set that was analyzed. The Bayesian Horseshoe model has the greatest performance overall when comparing predictability and regularization largely due to its prior distributions probability density function. If there are computational restraints the frequentist methods can be preferred, in this case either the Elastic net or the LASSO model are to be chosen as they outperform Ridge regression in all three tests. In high dimensional settings the best choice would be an Elastic net model where several values for α are tested and in a highly correlated setting a LASSO model would be preferred. These are however results derived from purely simulated data and might not generalize as well to real world data which should be kept in mind.

References

Bürkner, Paul-Christian (2017). "brms: An R Package for Bayesian Multilevel Models Using Stan". In: *Journal of Statistical Software* 80.1, pp. 1–28. DOI: https://doi.org/10.18637/jss.v080.i01.

Carvalho, Carlos M., Nicholas G Polson, and James G. Scott (2010). "The horseshoe estimator for sparse signals". In: *Biometrika* 97.2, pp. 465–480. DOI: https://dx.doi.org/10.1093/biomet/asq017.

DeBruine, Lisa (2023). *faux: Simulation for Factorial Designs*. R package version 1.2.1. URL: http://doi.org/10.5281/zenodo.2669586.

E. Hoerl, Arthur and Robert W. Kennard (1970). "Ridge Regression: Biased Estimation for Nonorthogonal Problems". In: *Technometrics* 12.1, pp. 55–67.

Fortmann-Roe, Scott (2012). "Understanding the Bias-Variance Tradeoff". In: URL: https://scott.fortmann-roe.com/docs/BiasVariance.html.

Friedman, J, R Tibshirani, and T Hastie (2010). "Regularization Paths for Generalized Linear Models via Coordinate Descent". In: *Journal of Statistical Software* 33.1, pp. 1–22. DOI: https://doi.org/10.18637/jss.v033.i01.

Gabry, J and T Mahr (2022). *bayesplot: Plotting for Bayesian Models*. R package version 1.10.0. URL: https://mc-stan.org/bayesplot/.

Gelman, Andrew et al. (2021). *Bayesian Data Analysis*. CRC press.

James, Gareth et al. (2021). *An Introduction to Statistical Learning: with Applications in R*. Second. Springer-Verlag. URL: https://faculty.marshall.usc.edu/gareth-james/ISL/.

Manyika, James et al. (2010). "Big data: The next frontier for innovation, competition, and productivity". In: URL: https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/big-data-the-next-frontier-for-innovation.

Muth, C., Z. Oravecz, and J. Gabry (2018). "User-friendly Bayesian regression modeling: A tutorial with rstanarm and shinystan". In: *The Quantative Methods for Psychology, Vol. 14, No. 2, pp. 99–119* 14.2, pp. 99–119. DOI: https://dx.doi.org/10.20982/tqmp.14.2.p099.

Qiu, W and H Joe (2023). *clusterGeneration: Random Cluster Generation (with Specified Degree of Separation)*. R package version 1.3.8. URL: https://CRAN.R-project.org/package=clusterGeneration.

Stan-Development-Team, Stan (2023). *RStan: the R interface to Stan*. R package version 2.32.3. URL: https://mc-stan.org/.

Tibshirani, Robert (1996). "Regression Shrinkage and Selection via the Lasso". In: *Journal of the Royal Statistical Society. Series B* 58.1, pp. 266–288. URL: http://www.jstor.org/stable/2346178.

Venables, W. N. and B. D. Ripley (2002). *Modern Applied Statistics with S*. Fourth. ISBN 0-387-95457-0. New York: Springer. URL: https://www.stats.ox.ac.uk/pub/MASS4/.

Wickham, H (2016). *ggplot2: Elegant Graphics for Data Analysis*.

Wickham, H, R François, et al. (2023). *dplyr: A Grammar of Data Manipulation*. R package version 1.1.2. URL: https://CRAN.R-project.org/package=dplyr.

Wickham, H, D Vaughan, and M Girlich (2023). *tidyr: Tidy Messy Data*. R package version 1.3.0. URL: https://CRAN.R-project.org/package=tidyr.

Zhang, Chiyuan et al. (2018). "A Study on Overfitting in Deep Reinforcement Learning". In: eprint: 1804.06893.

Zou, Hui and Trevor Hastie (2005). "Regularization and Variable Selection via the Elastic Net". In: *Journal of the Royal Statistical Society* 67.2, pp. 301–320. URL: http://www.jstor.org/stable/3647580.