BAYESIAN RIDGE REGRESSION: AN OVERVIEW AND COMPARISON TO CLASSICAL REGRESSION.

32102717

Abstract

Classical methods of linear regression model building suffer when the data set is subject to multicollinearity. Ridge regression is one alternative to classical methods that can alleviate this issue. In this paper we aim to explain the theory behind Ridge regression from a Bayesian perspective and suggest why one might use Ridge regression over classical methods. Then, using an exemplar data set on Diabetes provided by Efron et al. (2003) [2], we construct a series of classical and Ridge models and compare their effectiveness, including an extension to a selection of 'hybrid' models. We found that, for this data set, the classical subset models were better at the prediction of new data than the Ridge models, but suggest situations in which the Ridge models may be preferable. We also suggest considering other methods such as LASSO regression, Principle Component regression and Least Angle regression [3;4].

---

1. Introduction

Classical methods of regression model building, such as subset selection, are common place in many fields. They do, however, have their flaws. Classical methods of coefficient estimation suffer greatly when multicollinearity is present in a data set. The coefficient estimates can become unstable, anomalously large, and are subject to extreme changes when covariates are selected to be removed or added, even changing sign in some cases [3;4]. Ridge regression is known as a shrinkage method, and aims to alleviate this issue by applying a penalty to the size of the coefficients [3;4]. The result is that the coefficient estimates are shrunk towards zero and each other, which introduces a bias, but reduces their variance [3;4]. If this relationship is correctly balanced, which is regulated by a shrinkage parameter, λ, then using Ridge regression can lead to a reduction in the Mean Squared Error of the model. The original motivation for Ridge regression when it was first introduced by Hoerl & Kennard (1970) [1] was to make $\boldsymbol{X}^T\boldsymbol{X}$ in the equation for the Ordinary Least Squares coefficient estimates have full rank, even if two covariates were perfectly correlated, allowing it to be inverted. This is done by adding a positive constant, λ, to the diagonal of $\boldsymbol{X}^T\boldsymbol{X}$ before inversion. This simple augmentation gives the Ridge coefficient estimates, $\hat{\boldsymbol{\beta}}^{\text{ridge}}$. In fact, in the case of orthonormal inputs, the Ridge coefficient estimates are just scaled versions of the Ordinary Least Squares estimates, $\hat{\boldsymbol{\beta}}^{\text{ridge}} = \frac{\hat{\boldsymbol{\beta}}}{(1+\lambda)}$ [3]. During this paper we will be considering an exemplar data set to examine the application and effect of Ridge regression compared to classical methods. The data set, as described in Efron et al. (2003) [2], details 10 baseline covariates; age, sex, body mass index, average blood pressure, and 6 blood serum measurements, which relate to a response variable, $\boldsymbol{y}$; a quantitative measure of disease progression one year after baseline. The data set then also includes covariates which represent the quadratic interactions for all these variables, giving a total of 64 covariates. The data set contains observations for 442 unique individuals, with no missing data. The covariates have been centred, and scaled to have ℓ^2-norm. In this paper we aim to explain the concepts, theory and motivation behind Ridge regression from a Bayesian perspective, and then compare it to a range of classical models using our exemplar data set. We wish to compare the models based on their ability to predict new data and the 'Evidence' for each model given the data. In §2 we derive how the Ridge coefficient estimates, $\hat{\boldsymbol{\beta}}^{\text{ridge}}$, are calculated from a Bayesian perspective, and how these are then used to make predictions on new data. We then give a brief overview of some of the tools used to build the classical models, and then a consideration of why we would choose to use Ridge regression in place of classical methods, and suggest some points to consider when performing Ridge regression. We then explain how we will compare the models, including how the 'Evidence' and Mean Squared Error are calculated, and some diagnostics that can be performed on the Ridge regression models. In §3, we present the results of our model building and perform some basic comparisons and analysis. We then offer some extensions of the models, and show how they compare to their associated counterparts. In §4 we discuss the models in more depth, and consider other aspects which could make one model preferable to another, before making suggestions on extensions that could be considered and concluding the paper in §5.

2. Theory and Methodology

Ridge regression is an example of a shrinkage method of model fitting, an alternative to classical methods such as subset selection. Ridge regression works by shrinking the coefficient estimates towards zero, and each other, by applying a penalty to their size [3;4]. From a frequentist point of view, the Ridge coefficients, $\hat{\boldsymbol{\beta}}^{\text{ridge}}$, are chosen to minimise a residual sum of squares [3;4] given by:

$$\text{RSS}(\lambda) = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^T\boldsymbol{\beta}, \tag{1}$$

for which the solution is

$$\hat{\boldsymbol{\beta}}^{\text{ridge}} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda I_p)^{-1}\boldsymbol{X}^T\boldsymbol{y}, \tag{2}$$

where I_p is a $p \times p$ identity matrix, $\boldsymbol{X}$ is a matrix of covariate observations (excluding the intercept), $\boldsymbol{y}$ is a vector of the observed response variable, and $\lambda \geq 0$ is known as the shrinkage parameter. Larger values of lambda lead to greater shrinkage of the Ridge coefficients [3;4], with $\hat{\boldsymbol{\beta}}^{\text{ridge}} = \hat{\boldsymbol{\beta}}$, the Ordinary Least Squares estimates from classical regression, when $\lambda = 0$ and $\hat{\boldsymbol{\beta}}^{\text{ridge}} \to 0$ as $\lambda \to \infty$.

Ridge regression can also be considered from a Bayesian point of view. In this case the estimates of the Ridge coefficients can be derived as the mean or mode of the marginal posterior distribution of $\boldsymbol{\beta}$, when the prior placed on $\boldsymbol{\beta}$ is $\text{MVN}_p(\boldsymbol{0}, \Sigma)$ [3;4], where Σ is a diagonal matrix, so the β_j^{ridge}'s are independent, for j in $(1, \ldots, p)$, where p is the number of parameters in the model. We will now show how this result is derived.

For a given data set, let the response variable be denoted by a $1 \times n$ matrix (column vector) $\boldsymbol{y} = [y_1, \ldots, y_n]^T$, which is explained by an $n \times p$ matrix of covariates $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p]$ and a $1 \times p$ matrix (column vector) of regression coefficients $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_p]^T$. The observations of the response variable can be expressed as a multivariate distribution:

$$\boldsymbol{y}|\tau, \boldsymbol{\beta} \sim \text{MVN}_p(\boldsymbol{X}\boldsymbol{\beta}, \tfrac{1}{\tau} I_n) \tag{3}$$

where $\frac{1}{\tau}$ is the variance of residuals. Thus the likelihood is given by:

$$f(\boldsymbol{y}|\tau, \boldsymbol{\beta}) \propto \tau^{\frac{n}{2}} \exp\left(-\frac{\tau}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})\right). \tag{4}$$

As described, we now place a prior on $\boldsymbol{\beta}$, and also on τ:

$$\boldsymbol{\beta}|\tau \sim \text{N}\left(\boldsymbol{\beta}_0, \frac{1}{\tau}\Sigma_0\right), \qquad \tau \sim Gamma(a_0, b_0), \tag{5}$$

and we let $a_0 = b_0 = 2$ to make the prior on τ uninformative, and set $\boldsymbol{\beta}_0 = \boldsymbol{0}$. Our goal is to find the marginal posterior distributions of $\boldsymbol{\beta}$ and τ, as well as the posterior predictive distribution of the data. To do this, we first need to find the marginal likelihood (conditional on τ) of the data, which can be shown to be:

$$\boldsymbol{y}|\tau \sim \text{N}\left(\boldsymbol{X}\boldsymbol{\beta}_0, \frac{1}{\tau}(I_n + \boldsymbol{X}\boldsymbol{\Sigma}_0\boldsymbol{X}^T)\right) \tag{6}$$

the proof of which is given in Appendix 6.1. Thus, by integrating out τ, we can show that the marginal likelihood for the data (also known as the 'Evidence') can be expressed as:

$$\boldsymbol{y} \sim \text{MVT}_{2a_0}\left(\boldsymbol{X}\boldsymbol{\beta}_0, \frac{b_0}{a_0}(I_n + \boldsymbol{X}\boldsymbol{\Sigma}_0\boldsymbol{X}^T)\right), \tag{7}$$

the proof of which is given in Appendix 6.2.

We now consider a special case of a fully conjugate prior, known as a Ridge prior. The Ridge prior is given by:

$$\boldsymbol{\beta} \sim \text{MVN}_p(\boldsymbol{0}, \frac{1}{\tau\lambda} I_p), \tag{8}$$

where λ is known as the shrinkage parameter, which is fixed but unknown. In a Bayesian setting we can estimate λ by considering a range of values, and choosing the value that maximises the (log-)likelihood of the data. This is known as empirical Bayes. By combining the likelihood (7) with the Ridge prior on $\boldsymbol{\beta}$ (8) and the prior on τ (5), we show in Appendix 6.3 that the marginal posterior distributions are given by:

$$\boldsymbol{\beta}|\boldsymbol{y} \sim \text{MVT}_{2a_n}\left(\boldsymbol{\beta}_n, \frac{b_n}{a_n}\Sigma_n\right), \qquad \tau|\boldsymbol{y} \sim \text{Gamma}(a_n, b_n), \tag{9}$$

where

$$\boldsymbol{\beta}_n = \left(\boldsymbol{X}^T\boldsymbol{X} + I_p\lambda\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}, \qquad \Sigma_n = \left(\boldsymbol{X}^T\boldsymbol{X} + I_p\lambda\right)^{-1},$$
$$a_n = a_0 + \frac{n}{2}, \qquad b_n = b_0 + \frac{1}{2}\left(\boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{\beta}_n^T\Sigma_n^{-1}\boldsymbol{\beta}_n\right).$$

It can also be shown that the posterior predictive distribution for a vector of predictions $\boldsymbol{y}^*$ given a set of covariates $\boldsymbol{X}^*$ is given by

$$\boldsymbol{y}^*|\boldsymbol{X}^* \sim \mathrm{MVT}_{2a_n}\left(\boldsymbol{X}^*\boldsymbol{\beta}_n, \frac{b_n}{a_n}(I_n + \boldsymbol{X}^*\Sigma_n\boldsymbol{X}^{*T})\right). \tag{10}$$

Thus we can see that if we wish to make a prediction $\boldsymbol{y}^*$ from a set of covariates $\boldsymbol{X}^*$, this would just be the mean of the posterior predictive distribution of the data, $\boldsymbol{X}^*\boldsymbol{\beta}_n$, where $\boldsymbol{\beta}_n$ is the posterior mode (and mean) of $\boldsymbol{\beta}$. Thus the Bayesian approach agrees with the frequentist approach, and the Ridge coefficients are given by:

$$\hat{\boldsymbol{\beta}}^{\text{ridge}} = \boldsymbol{\beta}_n = (\boldsymbol{X}^T\boldsymbol{X} + \lambda I_p)^{-1}\boldsymbol{X}^T\boldsymbol{y}. \tag{11}$$

2.1. **Classical Models.** We wish to compare how well a Bayesian Ridge regression model compares to a range of classical regression models. The first model we wish to fit is the 'full' model. In this model no variable selection is implemented, and the coefficient estimates are computed in the classical sense using iteratively re-weighted least squares [5].

The second classic model we wish to consider is one that uses step-wise variable selection [3;4] using the Akaike Information Criterion (AIC) [3;4], which we will denote the 'AIC' model. The AIC is a measure of the fit and parsimony of a regression model. It is defined as:

$$\mathrm{AIC} = -2\ell(x) + 2d, \tag{12}$$

where $\ell(x)$ is the log-likelihood of the data x, and d is the number of parameters in the model. The model with the lowest AIC is considered to be the most parsimonious best fitting model. The AIC has the benefit that the models do not have to be nested. We use the AIC during step-wise variable selection as the metric by which to judge if a parameter is removed or added to the model.

Our third classic model will again be using step-wise variable selection, this time with the Bayes Information Criterion (BIC) [3;4], and we denote it the 'BIC' model. The BIC is an alternative to the AIC. It is defined as:

$$\mathrm{BIC} = -2\ell(x) + \log(N)d, \tag{13}$$

where $\ell(x)$ is the log-likelihood of the data x, d is the number of parameters in the model, and N is number of observations in the data. Similarly to the AIC, the model with the lowest BIC is considered to be the most parsimonious best fitting model, and has the benefit that the models being compared do not have to be nested. In general, the BIC penalises models with more parameters more than the AIC, since $\log(N) > 2$ in most cases, thus the BIC aims for more parsimonious models.

2.2. **Why use Ridge regression?** Classical linear regression does not fare well when the input data suffers from multicollinearity. Multicollinearity occurs when variables are highly correlated [5], for example, if two variables measure the same thing on two different scales (say height in meters and inches) they will be highly correlated. This means that the design matrix, $\boldsymbol{X}$, will not have full rank (or will be very close to having not full rank). This means that, due to the way the Ordinary Least Squares estimates are calculated, the coefficient estimates will be unstable (also called being poorly determined or defined) and will exhibit high variance [3;4]. For instance, one covariate may have an anomalously large positive coefficient, which in every instance will be cancelled by an anomalously large negative coefficient of the covariate it is highly correlated with. Removing one of these covariates from the model will lead the other's coefficient to change drastically, possibly even changing sign. By applying a penalty to the size of the coefficients, Ridge regression alleviates this problem. This works because a positive constant is added to the diagonal of the design matrix, meaning that it gains full rank (becomes non-singular and has an inverse) [3;4].

However, Ridge regression is not guaranteed to be better than classical regression in every instance. The Ridge coefficients are biased, $\mathbb{E}[\hat{\boldsymbol{\beta}}^{\text{ridge}}] \neq \boldsymbol{\beta}$, whereas the classical coefficient estimates $\hat{\boldsymbol{\beta}}$ are not [4]. Since the Mean Squared Error $= bias^2 + Variance$, for the Ridge model to improve on the accuracy of the classical model, the bias gained must be outweighed by the reduction in the variance. In cases where multicollinearity is not an issue, the Ridge estimates can add bias while making negligible

difference to the variance, and as such the MSE will increase. In addition, where as the Ordinary Least Squares coefficient estimates are invariant under scaling, the Ridge coefficient estimates are not[3;4]. This means that multiplying covariate $\boldsymbol{X}_j$ by a constant c simply scales the OLS estimates by a factor of $\frac{1}{c}$ (equivalently, regardless of what value c takes, $\boldsymbol{X}_j\hat{\beta}_j$ remains the same)[4]. This is not true for the Ridge coefficient estimates, which can change dramatically when the associated covariate is scaled. This is due to the quadratic penalty term in (1), which causes each $\hat{\beta}_j^{\text{ridge}}$ to not only be dependent on the scaling of their associated covariate, but also on λ, and the value of the other ridge coefficients[4]. For this reason it is not uncommon to scale and centre the data before calculating $\hat{\boldsymbol{\beta}}_{\text{ridge}}$[3;4]. Also notice how the intercept coefficient is not included in (1) and is not penalised. After scaling and centring the data, we estimate the intercept by $\beta_0^{\text{ridge}} = \bar{y} = \frac{1}{n}\sum_{i=1}^{N} y_i$[3;4].

2.3. **Comparing models.** One way of comparing models is calculate the 'Evidence' for each one. The model with the greatest 'Evidence' is considered to be the most appropriate for the data. We showed in §2 that the marginal likelihood of the data is given by:

$$\boldsymbol{y} \sim \text{MVT}_{2a_0}\left(\boldsymbol{X}\boldsymbol{\beta}_0, \frac{b_0}{a_0}(I_n + \boldsymbol{X}\boldsymbol{\Sigma}_0\boldsymbol{X}^T)\right), \tag{14}$$

where $\boldsymbol{X}$ a matrix of covariates, $\boldsymbol{\beta}_0 = \boldsymbol{0}$, I_n is an $n \times n$ identity matrix, and $\Sigma_0 = \frac{1}{\lambda}I_p$, where I_p is a $p \times p$ identity matrix. Given a data set $\boldsymbol{X}$ and response data $\boldsymbol{y}$ we can calculate the evidence (marginal likelihood) of the model via:

$$\text{Evidence} = f(\boldsymbol{y}) = \frac{\Gamma(\frac{\nu+p}{2})\Gamma(\frac{\nu}{2})}{|\Sigma|^{\frac{1}{2}}(\nu\pi)^{\frac{p}{2}}}\left[1 + \frac{(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}_0)^T\Sigma^{-1}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}_0)}{\nu}\right]^{-\frac{\nu+p}{2}}, \tag{15}$$

where $\nu = 2a_0$, p is the number of parameters in the data, and $\Sigma = \frac{b_0}{a_0}(I_n + \boldsymbol{X}\Sigma_0\boldsymbol{X}^T)$.

An alternative to the 'Evidence' is to consider how well the model predicts new data, which in many circumstances is often the most important factor in choosing a final model. Many different types of models have many different types of tests for the predictivity of a model. We have chosen to calculate the Mean Squared Error (MSE)[3;4] for prediction for each model, and the model that has the lowest MSE will be the best at predicting. We can calculate the MSE via:

$$\text{MSE} = \frac{\sum_{i=1}^{n}(y_i - y_i^*)^2}{n}, \tag{16}$$

where n is the number of predicted data points, y_i is the known response value for a given set of observations x_i, and y_i^* is the predicted response value for the same set of observations. The MSE calculates an average of the squared distance between the true and predicted response values; the lower the MSE, the closer the predicted and true response values, thus the better the fit of the model.

To calculate the out of sample predictivity we need to test how well our fitted model predicts new data. To did this we will use k-fold cross validation[3;4] with the Mean Squared Error. We first divide the data into k equal sets, where we have chosen $k = 10$, we then choose one set to be a 'test set' and use the other nine sets to train the model. This trained model is then used to predict the test set, and calculate the MSE. We repeat this procedure ten times, selecting a new set to be the 'test set' each time. This can give us a good idea about how well our chosen model predicts for new data.

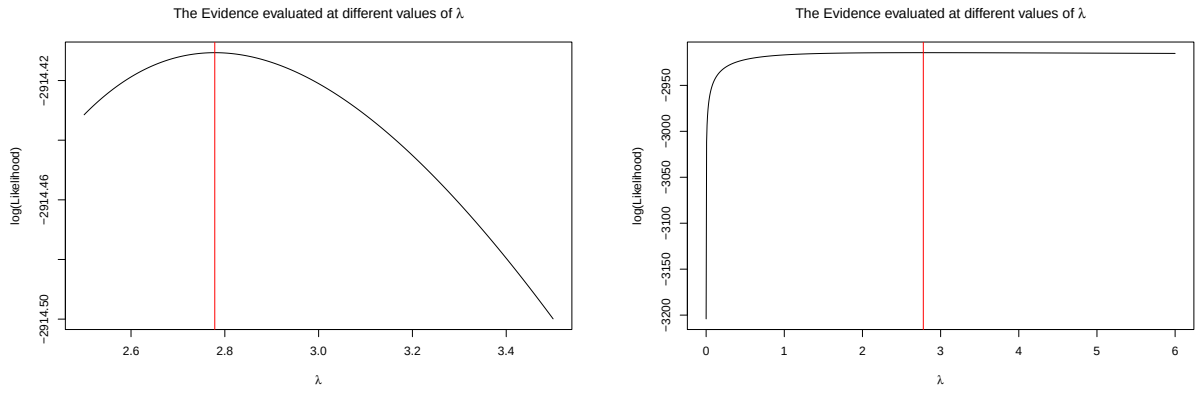
Another aspect of the fit of the Ridge regression model we can consider is how many of the true response values are outliers (in each test set of our 10-fold cross validation) given our assumed posterior predictive distribution, as given in (10). If the probability of seeing the true value of our response variable under the posterior predictive distribution is ≤ 0.05 then this point in an outlier; it is unlikely to be drawn from the given posterior predictive distribution.

3. Results and Analysis

Initially 4 models were produced, the 'full', 'AIC', 'BIC', and 'Ridge' models, as described in §2. We also considered a Ridge regression model where the intercept was penalised, which we denote 'Ridge.int'. Since we have utilised 10-fold cross validation, fitted model parameters and example predictions that are referenced in this section are associated with models fitted with the full data set.

By considering a range of values for the shrinkage parameter, λ, we found the value that gave the strongest evidence for the 'Ridge' regression model (the one which maximised the value of the

marginal likelihood). This value was $\lambda_{\text{opt}} = 2.7786$ which was calculated using all the data, however, a new optimal lambda was calculated for each data set when doing cross validation. We can see from Figure 1 that this was a global maxima. Similarly, for the 'Ridge.int' model $\lambda_{\text{opt}} = 0.1475$.



(A) (B)

FIGURE 1. The Evidence of the 'Ridge' model (the intercept not penalised) evaluated at different values of λ: (A) A zoomed plot to show the value of λ that maximises the Evidence for the 'Ridge' model. (B) A plot that shows the chosen value of lambda is a global maxima.

The models were compared on how large the 'Evidence' for each model was, assessed using the marginal log-likelihood, and how well each model predicted new data, assessed using 10-fold cross validation with Mean Squared Error. As we can see from Table 1, the 'Ridge.int' model (closely followed by the 'Ridge' model) had the highest log(evidence) with a value of -264.92 (-265.76 respectively) and the 'AIC' model had the lowest with a value of -568.23, a substantial difference of almost 300. In contradiction, the 'BIC' model had the lowest the average MSE value with a value of 2788.55, and the 'full' model had the highest with a value of 3903.22. Lower values of the evidence were associated with both the highest MSE (the 'full' model) and the lowest MSE (the 'BIC' model).

Model	Avg. MSE	MSE Std. Dev.	Avg. Log(Evidence)	Log(Evidence) Std. Dev.
Full	3903.22	1088.83	-544.59	5.43
Ridge	3807.29	556.66	-265.76	3.43
Ridge.int	3163.24	587.02	-264.92	5.02
AIC	2856.39	524.83	-568.23	22.77
BIC	2788.55	531.23	-456.31	7.73

TABLE 1. Model comparisons. The summary statistics resulting from 10-fold cross validation utilising the Mean Squared Error and the log of the 'Evidence' for each model.

In our 'Ridge' model, we did not have any outliers in any of our test sets, in our 'Ridge.int' model (where the intercept is penalised) we detected a total of 41 outliers across all test sets.

As an extension of these models, we also consider hybrid models, to see the effect of estimating the parameters of subset models via Ridge methods. We propose an additional four models; 'AIC.Ridge' (the covariates of the 'AIC' model with estimates calculated via Ridge regression), 'AIC.Ridge.int' (equivalently, but the Ridge estimates calculated include penalising the intercept), 'BIC.Ridge' (the covariates of the 'BIC' model with estimates calculated via Ridge regression), and 'BIC.Ridge.int' (equivalently, but the Ridge estimates calculated include penalising the intercept). We can see from Table 2 that the hybrid model with the lowest average MSE is the 'BIC.Ridge.int' model, with an average MSE of 2812.74. The 'AIC.Ridge' model had the highest average MSE of 3281.34. Alternatively, the 'BIC.Ridge' model had the largest average log(Evidence), with a value of -261.54, and the 'BIC.Ridge.int' had the smallest at -267.48. However, there was very little difference in the average log(Evidence) for the four hybrid models, all four falling within 6 points of each other.

By comparing Table 1 and 2, we can see the effect that adapting each model to have Ridge coefficients has. In general, adding Ridge coefficients that are calculated via penalising the intercept have

Model	Avg. MSE	MSE Std. Dev.	Avg. Log(Evidence)	Log(Evidence) Std. Dev.
AIC.Ridge	3281.34	571.70	-262.97	4.15
BIC.Ridge	3005.83	558.15	-261.54	4.44
AIC.Ridge.int	2887.05	577.06	-266.95	5.37
BIC.Ridge.int	2812.74	565.32	-267.48	4.45

TABLE 2. Hybrid model comparisons. The summary statistics resulting from 10-fold cross validation utilising the Mean Squared Error and the log of the 'Evidence' for each hybrid model.

lower Mean squared error then their alternatives, however, neither method of estimating the Ridge coefficients lowers the MSEs beyond that of the associated classical subset models. For instance, the lowest average MSE in the hybrid models is associated with 'BIC.Ridge.int' with 2812.74, but the classical 'BIC' model has an average MSE of 2788.55. If we consider another point of view and look at the hybrid models as selecting a subset of covariates in the 'Ridge' and 'Ridge.int' models, we can see that all four hybrid models in Table 2 have lower average MSEs than their associated Ridge regression model in Table 1. For the log(Evidence) however, adapting each model to have Ridge coefficients drastically increases the log(Evidence) compared to their classical counterparts, with classical models having a value around -500, and the hybrid models having a value around -265. The 'AIC.Ridge' and 'BIC.Ridge' models even improve on the log(Evidence) of their associated model, 'Ridge'.

4. Discussion

It is clear from Table 1 that, for this data set, the classical subset models are better at predicting the response variable, a quantitative measure of disease progression one year after baseline, for new data once the model is trained. In fact the 'Ridge' model which does not penalise the intercept is barely better than the classical 'full' model. It is interesting to note that in all cases, including the hybrid models, the Ridge estimates that penalise the intercept are able to predict new data far better than when the intercept is calculated as the mean of the response variables from the training data, despite the fact that the literature (and common practice) is to not penalise the intercept. This could simply be a random occurrence for this data set that does not occur in general.

By comparing Table 1 and 2, we can see that for out of sample predicitivity, the classical subset models are generally better than their Ridge regression counterparts. Part of the reason to choose Ridge regression and other shrinkage methods over classical subset methods is that they are less computationally intensive, for this reason, for extremely large and complex models, Ridge regression may be used even if the predicitivity is expected to suffer. The hybrid models would require one to run both subset selection and Ridge regression, and so would increase the computational burden, and since they do not improve on the classical subset models' ability to predict, there is very little evidence to suggest they would be worth implementing. It makes sense that these hybrid models are not as effective as the classical subset models, since the subset models are less likely to exhibit multicollinearity, and as of thus their estimates are going to have smaller variance, so the bias added by Ridge coefficients will outweigh the negligible additional reduction in variance they provide. A better method for incorporating variable selection into a shrinkage method is to use LASSO regression [3;4]. LASSO regression works in much the same way that Ridge regression does, except that the prior put on $\boldsymbol{\beta}$ is a multivariate double-exponential distribution (also known as a Laplace distribution) with a mean of $\mathbf{0}$ and a scale parameter that is a function of λ [4]. The LASSO coefficient estimates are given by the posterior mode of $\boldsymbol{\beta}$ (but not the mean) [4]. We suggest that it would be worth performing LASSO regression on this data set, and comparing how effective it is at predicting new data to both the Ridge regression models and the classical subset models.

5. Conclusion

We have given an overview of the theory behind Bayesian Ridge regression, and compared its performance to classical methods of model building. For this particular dataset we found that Ridge regression was not as effective as classical subset methods of regression. Ridge regression is still a very useful tool however, and there will be lots of circumstances where it will perform better than classical methods. In addition there are lots of other regression methods that could be considered, such as LASSO regression, Principle Component regression and Least Angle regression [3;4].

References

[1] Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. 12:55–67, 04 2012.

[2] Bradley Efron, Trevor Hastie, Lain Johnstone, and Robert Tibshirani. Least angle regression. 32, 04 2002.

[3] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[4] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014.

[5] P. (Peter) McCullagh. *Generalized linear models*. Monographs on statistics and applied probability. Chapman and Hall, London ; New York, 1983.

6. Appendix

6.1. **Proof 1.** The model can be reformulated such that

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \epsilon_1 \qquad \text{where} \quad \epsilon_1 \sim \text{Normal}(\mathbf{0}, \tfrac{1}{\tau} I_n), \tag{17}$$

$$\boldsymbol{\beta} = \beta_0 + \epsilon_2 \qquad \text{where} \quad \epsilon_2 \sim \text{Normal}(\mathbf{0}, \tfrac{1}{\tau}\Sigma_0). \tag{18}$$

and we can combine these two elements such that

$$\boldsymbol{y} = \boldsymbol{X}\beta_0 + \epsilon_2 + \epsilon_1 \tag{19}$$

thus $\boldsymbol{y}|\tau$ has a Normal distribution, as it is a linear combination of independent normal distributions, with mean given by

$$\begin{aligned}
\mathbb{E}[\boldsymbol{y}] &= \mathbb{E}[\boldsymbol{X}\beta_0 + \epsilon_2 + \epsilon_1] && (20)\\
&= \mathbb{E}[\boldsymbol{X}\beta_0] + \mathbb{E}[\boldsymbol{X}\epsilon_2] + \mathbb{E}[\epsilon_1] && (21)\\
&= \boldsymbol{X}\beta_0 + \mathbb{E}[\boldsymbol{X}\epsilon_2] + \mathbb{E}[\epsilon_1] && (22)\\
&= \boldsymbol{X}\beta_0 + \boldsymbol{X}\mathbf{0} + \mathbf{0} && (23)\\
&= \boldsymbol{X}\beta_0 && (24)
\end{aligned}$$

by the linearity of expectation, and variance given by

$$\begin{aligned}
\text{Var}(\boldsymbol{y}) &= \text{Var}(\boldsymbol{X}\beta_0 + \epsilon_2 + \epsilon_1) && (25)\\
&= \text{Var}(\boldsymbol{X}\beta_0) + \text{Var}(\boldsymbol{X}\epsilon_2) + \text{Var}(\epsilon_1) && (26)\\
&\quad + 2\text{Cov}(\boldsymbol{X}\beta_0, \boldsymbol{X}\epsilon_2) + 2\text{Cov}(\boldsymbol{X}\beta_0, \epsilon_1) + 2\text{Cov}(\boldsymbol{X}\epsilon_2, \epsilon_1) && (27)\\
&= \mathbf{0} + \text{Var}(\boldsymbol{X}\epsilon_2) + \text{Var}(\epsilon_1) + \mathbf{0} + \mathbf{0} + \mathbf{0} && (28)\\
&= \boldsymbol{X}\text{Var}(\epsilon_2)\boldsymbol{X}^T + \text{Var}(\epsilon_1) && (29)\\
&= \boldsymbol{X}(\tfrac{1}{\tau}\Sigma_0))\boldsymbol{X}^T + (\tfrac{1}{\tau} I_n) && (30)\\
&= \tfrac{1}{\tau}(I_n + \boldsymbol{X}\Sigma_0\boldsymbol{X}^T). && (31)
\end{aligned}$$

Thus,

$$\boldsymbol{y}|\tau \sim \text{N}\left(\boldsymbol{X}\boldsymbol{\beta}_0, \frac{1}{\tau}(I_n + \boldsymbol{X}\boldsymbol{\Sigma}_0\boldsymbol{X}^T)\right). \tag{32}$$

6.2. **Proof 2.**

$$f(\boldsymbol{y}|\tau) \propto \frac{1}{\left|2\pi\frac{1}{\tau}(I_n + \boldsymbol{X}\boldsymbol{\Sigma}_0\boldsymbol{X}^T)\right|^{\frac{1}{2}}} \exp\left\{-\tfrac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}_0)^T \left(\tfrac{1}{\tau}(I_n + \boldsymbol{X}\Sigma_0\boldsymbol{X}^T)\right)^{-1}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}_0)\right\} \tag{33}$$

$$\propto \tau^{\frac{n}{2}} \exp\left\{-\tfrac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}_0)^T \left(\tfrac{1}{\tau}(I_n + \boldsymbol{X}\Sigma_0\boldsymbol{X}^T)\right)^{-1}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}_0)\right\}. \tag{34}$$

and we know that the prior on τ is given by

$$\begin{aligned}
f(\tau) &\propto \frac{b_0^{a_0}}{\Gamma(a_0)}\tau^{a_0 - 1}\exp\{-b_0\tau\} && (35)\\
&\propto \tau^{a_0 - 1}\exp\{-b_0\tau\}. && (36)
\end{aligned}$$

thus we can find the joint distribution of the two using Bayes theorem,

$$f(\boldsymbol{y},\tau) \propto f(\boldsymbol{y}|\tau)f(\tau) \tag{37}$$
$$\propto \tau^{\frac{n}{2}} \exp\left\{-\tfrac{1}{2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}_0)^T\left(\tfrac{1}{\tau}(I_n+\boldsymbol{X}\Sigma_0\boldsymbol{X}^T)\right)^{-1}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}_0)\right\}\tau^{a_0-1}\exp\{-b_0\tau\} \tag{38}$$
$$\propto \tau^{\frac{n}{2}+a_0-1}\exp\left\{-\tau\left[\tfrac{1}{2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}_0)^T\left(I_n+\boldsymbol{X}\Sigma_0\boldsymbol{X}^T\right)^{-1}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}_0)+b_0\right]\right\} \tag{39}$$

By integrating out τ from this joint distribution we can find the marginal likelihood for $\boldsymbol{y}$,

$$f(\boldsymbol{y}) = \int f(\boldsymbol{y},\tau)d\tau \tag{40}$$
$$\propto \int \tau^{\frac{n}{2}+a_0-1}\exp\left\{-\tau\left[\tfrac{1}{2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}_0)^T\left(I_n+\boldsymbol{X}\Sigma_0\boldsymbol{X}^T\right)^{-1}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}_0)+b_0\right]\right\}d\tau \tag{41}$$

This is the kernel of a $\text{Gamma}\left(\frac{n}{2}+a_0, \frac{1}{2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}_0)^T\left(I_n+\boldsymbol{X}\Sigma_0\boldsymbol{X}^T\right)^{-1}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}_0)+b_0\right)$ distribution, so

$$f(\boldsymbol{y}) \propto \frac{\Gamma(\frac{n}{2}+a_0)}{\left[\frac{1}{2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}_0)^T\left(I_n+\boldsymbol{X}\Sigma_0\boldsymbol{X}^T\right)^{-1}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}_0)+b_0\right]^{\frac{n}{2}+a_0}} \tag{42}$$
$$\propto (b_0)^{\frac{-(n+2a_0)}{2}}\left[1+\tfrac{1}{2b_0}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}_0)^T\left(I_n+\boldsymbol{X}\Sigma_0\boldsymbol{X}^T\right)^{-1}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}_0)\right]^{\frac{-(n+2a_0)}{2}} \tag{43}$$
$$\propto \left[1+\tfrac{1}{2b_0}\tfrac{a_0}{a_0}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}_0)^T\left(I_n+\boldsymbol{X}\Sigma_0\boldsymbol{X}^T\right)^{-1}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}_0)\right]^{\frac{-(n+2a_0)}{2}} \tag{44}$$
$$\propto \left[1+\frac{(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}_0)^T\left[\frac{b_0}{a_0}\left(I_n+\boldsymbol{X}\Sigma_0\boldsymbol{X}^T\right)\right]^{-1}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}_0)}{2a_0}\right]^{\frac{-(n+2a_0)}{2}} \tag{45}$$

Thus, the marginal distribution of $\boldsymbol{y}$ is given by

$$\boldsymbol{y} \sim \text{MVT}_{2a_0}\left(\boldsymbol{X}\boldsymbol{\beta}_0, \frac{b_0}{a_0}(I_n+\boldsymbol{X}\boldsymbol{\Sigma}_0\boldsymbol{X}^T)\right). \tag{46}$$

6.3. **Proof 3.** We know that

$$\pi(\boldsymbol{y}|\tau,\boldsymbol{\beta}) \propto \tau^{\frac{n}{2}}\exp\left\{-\tfrac{\tau}{2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})\right\} \tag{47}$$
$$\pi(\boldsymbol{\beta}|\tau) \propto (\tau\lambda)^{\frac{p}{2}}\exp\left\{-\tfrac{\tau\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta}\right\} \tag{48}$$
$$\pi(\tau) \propto \tau^{a_0-1}\exp\left\{-b_0\tau\right\}. \tag{49}$$

Thus we can use Bayes rule such that

$$\pi(\tau|\boldsymbol{\beta}) \propto \pi(\boldsymbol{\beta}|\tau)pi(\tau) \tag{50}$$
$$\propto (\tau\lambda)^{\frac{p}{2}}\exp\left\{-\tfrac{\tau\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta}\right\}\tau^{a_0-1}\exp\left\{-b_0\tau\right\} \tag{51}$$
$$\propto \tau^{(\frac{p}{2}+a_0-1)}\exp\left\{-\tau\left[b_0+\tfrac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta}\right]\right\} \tag{52}$$

and as such

$$\pi(\tau,\boldsymbol{\beta}|\boldsymbol{y}) \propto \pi(\boldsymbol{y}|\tau,\boldsymbol{\beta})\pi(\tau,\boldsymbol{\beta}) \tag{53}$$
$$\propto \tau^{\frac{n}{2}}\exp\left\{-\tfrac{\tau}{2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})\right\}\tau^{(\frac{p}{2}+a_0-1)}\exp\left\{-\tau\left[b_0+\tfrac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta}\right]\right\} \tag{54}$$
$$\propto \tau^{(\frac{n}{2}+\frac{p}{2}+a_0-1)}exp\left\{-\tau\left[b_0+\tfrac{1}{2}\lambda\boldsymbol{\beta}^T\boldsymbol{\beta}+\tfrac{1}{2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})\right]\right\}. \tag{55}$$

Now, let $Z = \left[b_0+\frac{1}{2}\lambda\boldsymbol{\beta}^T\boldsymbol{\beta}+\frac{1}{2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})\right]$, then

$$Z = b_0+\tfrac{1}{2}\lambda\boldsymbol{\beta}^T\boldsymbol{\beta}+\tfrac{1}{2}\left[\boldsymbol{y}^T\boldsymbol{y}-\boldsymbol{y}^T\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{y}+\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta}\right] \tag{56}$$
$$= b_0+\tfrac{1}{2}\boldsymbol{y}^T\boldsymbol{y}+\tfrac{1}{2}\left[\lambda\boldsymbol{\beta}^T\boldsymbol{\beta}-(\boldsymbol{X}^T\boldsymbol{y})^T\boldsymbol{\beta}-\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{y}+\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta}\right] \tag{57}$$
$$= b_0+\tfrac{1}{2}\boldsymbol{y}^T\boldsymbol{y}+\tfrac{1}{2}\left[-(\boldsymbol{X}^T\boldsymbol{y})^T\boldsymbol{\beta}-\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{y}+\boldsymbol{\beta}^T(\boldsymbol{X}^T\boldsymbol{X}+I_p\lambda)\boldsymbol{\beta}\right]. \tag{58}$$

Now define

$$\boldsymbol{\beta}_n = \left(\boldsymbol{X}^T\boldsymbol{X} + I_p\lambda\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}, \qquad \Sigma_n = \left(\boldsymbol{X}^T\boldsymbol{X} + I_p\lambda\right)^{-1},$$
$$a_n = a_0 + \frac{n}{2}, \qquad b_n = b_0 + \frac{1}{2}\left(\boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{\beta}_n^T\Sigma_n^{-1}\boldsymbol{\beta}_n\right).$$

So that

$$\begin{aligned} Z &= b_0 + \tfrac{1}{2}\boldsymbol{y}^T\boldsymbol{y} + \tfrac{1}{2}\left[-(\Sigma_n^{-1}\boldsymbol{\beta}_n)^T\boldsymbol{\beta} - \boldsymbol{\beta}^T\Sigma_n^{-1}\boldsymbol{\beta}_n + \boldsymbol{\beta}^T\Sigma_n^{-1}\boldsymbol{\beta}\right] && (59)\\ &= b_0 + \tfrac{1}{2}\boldsymbol{y}^T\boldsymbol{y} + \tfrac{1}{2}\left[-\boldsymbol{\beta}_n^T\Sigma_n^{-1}\boldsymbol{\beta} - \boldsymbol{\beta}^T\Sigma_n^{-1}\boldsymbol{\beta}_n + \boldsymbol{\beta}^T\Sigma_n^{-1}\boldsymbol{\beta}\right] && (60)\\ &= b_n + \tfrac{1}{2}\left[\boldsymbol{\beta}_n^T\Sigma_n^{-1}\boldsymbol{\beta}_n - \boldsymbol{\beta}_n^T\Sigma_n^{-1}\boldsymbol{\beta} - \boldsymbol{\beta}^T\Sigma_n^{-1}\boldsymbol{\beta}_n + \boldsymbol{\beta}^T\Sigma_n^{-1}\boldsymbol{\beta}\right] && (61)\\ &= b_n + \tfrac{1}{2}\left[(\boldsymbol{\beta}-\boldsymbol{\beta}_n)^T\Sigma_n^{-1}(\boldsymbol{\beta}-\boldsymbol{\beta}_n)\right]. && (62)\end{aligned}$$

Thus,

$$\begin{aligned} \pi(\tau,\boldsymbol{\beta}|\boldsymbol{y}) &\propto \tau^{(\frac{n}{2}+\frac{p}{2}+a_0-1)}exp\left\{-\tau\left[b_n + \tfrac{1}{2}\left[(\boldsymbol{\beta}-\boldsymbol{\beta}_n)^T\Sigma_n^{-1}(\boldsymbol{\beta}-\boldsymbol{\beta}_n)\right]\right]\right\} && (63)\\ &\propto \tau^{(\frac{p}{2}+a_n-1)}exp\left\{-\tau\left[b_n + \tfrac{1}{2}\left[(\boldsymbol{\beta}-\boldsymbol{\beta}_n)^T\Sigma_n^{-1}(\boldsymbol{\beta}-\boldsymbol{\beta}_n)\right]\right]\right\}, && (64)\end{aligned}$$

which is proportional to a Gamma$\left(\frac{p}{2}+a_n, b_n + \frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\beta}_n)^T\Sigma_n^{-1}(\boldsymbol{\beta}-\boldsymbol{\beta}_n)\right)$. Thus by integrating out τ we can find the marginal distribution of $\boldsymbol{\beta}$, using the fact that $\pi(\tau,\boldsymbol{\beta}|\boldsymbol{y})$ has the form of a Gamma kernel

$$\begin{aligned} \pi(\boldsymbol{\beta}|\boldsymbol{y}) &= \int \pi(\tau,\boldsymbol{\beta}|\boldsymbol{y})d\tau && (65)\\ &\propto \frac{\Gamma(\frac{p}{2}+a_n)}{\left[b_n + \frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\beta}_n)^T\Sigma_n^{-1}(\boldsymbol{\beta}-\boldsymbol{\beta}_n)\right]^{(\frac{p}{2}+a_n)}} && (66)\\ &\propto \left[b_n + \tfrac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\beta}_n)^T\Sigma_n^{-1}(\boldsymbol{\beta}-\boldsymbol{\beta}_n)\right]^{-(\frac{p+2a_n}{2})} && (67)\\ &\propto b_n^{-(\frac{p+2a_n}{2})}\left[1 + \tfrac{1}{2}\tfrac{1}{b_n}(\boldsymbol{\beta}-\boldsymbol{\beta}_n)^T\Sigma_n^{-1}(\boldsymbol{\beta}-\boldsymbol{\beta}_n)\right]^{-(\frac{p+2a_n}{2})} && (68)\\ &\propto \left[1 + \tfrac{1}{2a_n}(\boldsymbol{\beta}-\boldsymbol{\beta}_n)^T(\tfrac{b_n}{a_n}\Sigma_n)^{-1}(\boldsymbol{\beta}-\boldsymbol{\beta}_n)\right]^{-(\frac{p+2a_n}{2})}. && (69)\end{aligned}$$

Thus, the marginal distribution of $\boldsymbol{\beta}$ is given by

$$\boldsymbol{\beta}|\boldsymbol{y} \sim \mathrm{MVT}_{2a_n}\left(\boldsymbol{\beta}_n, \frac{b_n}{a_n}\Sigma_n)\right). \tag{70}$$

Alternatively, by integrating out $\boldsymbol{\beta}$ from $\pi(\tau,\boldsymbol{\beta}|\boldsymbol{y})$ we can find the marginal distribution of τ

$$\begin{aligned} \pi(\tau|\boldsymbol{y}) &= \int \pi(\tau,\boldsymbol{\beta}|\boldsymbol{y})d\boldsymbol{\beta} && (71)\\ &= \int \tau^{(\frac{p}{2}+a_n-1)}exp\left\{-\tau\left[b_n + \tfrac{1}{2}\left[(\boldsymbol{\beta}-\boldsymbol{\beta}_n)^T\Sigma_n^{-1}(\boldsymbol{\beta}-\boldsymbol{\beta}_n)\right]\right]\right\}d\boldsymbol{\beta} && (72)\\ &= \tau^{(a_n-1)}exp\left\{-\tau b_n\right\}\int \tau^{\frac{p}{2}}exp\left\{-\tau\tfrac{1}{2}\left[(\boldsymbol{\beta}-\boldsymbol{\beta}_n)^T\Sigma_n^{-1}(\boldsymbol{\beta}-\boldsymbol{\beta}_n)\right]\right\}d\boldsymbol{\beta}. && (73)\end{aligned}$$

The term inside the integral is the kernel of a multivariate Gaussian distribution, and as such

$$\begin{aligned} \pi(\tau|\boldsymbol{y}) &= \tau^{(a_n-1)}exp\left\{-\tau b_n\right\}\det(2\pi\Sigma_n) && (74)\\ &\propto \tau^{(a_n-1)}exp\left\{-\tau b_n\right\}. && (75)\end{aligned}$$

Thus, the marginal distribution of τ is given by

$$\tau|\boldsymbol{y} \sim \mathrm{Gamma}\left(a_n, b_n\right). \tag{76}$$

6.4. **R code.** The R code that was used to fit all the models presented in this paper is detailed below.

```
#####################################
######## BAYES PROJECT CODE ########
#####################################

library("mvnfast")
library(caret)
library(MASS)

data <- read.csv("diabetes.csv")

y <- data[,2]

X.int <- data[,-2]
X.int[,1] <- 1
colnames(X.int)[1] <- "Int"

X.no.int <- data[,-(1:2)]

a_0 = 2
b_0 = 2



########################
##### Find Lambda #####
########################


llh.y=function(lambda, X, y){

  X=as.matrix(X)
  p=ncol(X)
  n=nrow(X)
  a=2
  b=2
  S=diag(1/lambda,p)

  COV = b/a* ( diag(1,n) + X%*%S%*%t(X) )

  llh = dmvt(t(y),rep(0,n),COV,2*a,log=TRUE)

  return(llh)
}


negllh.y=function(lambda, X, y){

  llh <- llh.y(lambda, X, y)

  return(-llh)
}

S <- optim(0.1,negllh.y, X=X.no.int, y=y, method="Brent",lower=0.001,upper=100,hessian=T)

lambda_opt <- S$par

#plot lambda

```

```r
lambda <- seq(from = 2.5, to = 3.5, length.out = 1000)

loglik = 1
for(i in 1:1000){
  loglik[i] <- llh.y(lambda[i], X.no.int, y)
}

plot(lambda, loglik, type = "l", xlab = expression(lambda), ylab = "log(Likelihood)",
    main =expression(paste("The Evidence evaluated at different values of ", lambda)))
abline(v = lambda_opt, col = "red")


#####################################
##### Find Posterior Parameters #####
#####################################


post.param <- function(a0, b0, lambda, X, y){

  X <- as.matrix(X)

  p <- ncol(X)

  n <- nrow(X)

  beta_n <- solve(t(X)%*%X + diag(lambda,p)) %*% t(X) %*% y

  Sigma_n <- solve(t(X)%*%X + diag(lambda,p))

  a_n <- a0 + n/2

  b_n <- b0 + 0.5 * (t(y) %*% y - t(beta_n) %*% solve(Sigma_n) %*% beta_n)

  params <- list(beta_n, Sigma_n, a_n, b_n)

  return(params)
}

post.params <- post.param(2,2,lambda_opt, X.no.int, y)


#############################################################
##### Find the evidence for a subset of the variables #####
#############################################################


Evidence <- function(X, y, var, a_0 = 2, b_0 = 2, lambda){

  X <- X[,var]
  X <- as.matrix(X)
  p <- ncol(X)
  n <- nrow(X)
  S_0 <- diag(1/lambda, p)
  beta_0 <- as.matrix(rep(0, p))

  mean <- X %*% beta_0
  COV <- (b_0/a_0) * (diag(n) + X%*%S_0%*%t(X))

  llh <- dmvt(X = t(y), mu = mean, sigma = COV, df = 2*a_0, log = TRUE)

```

```
  return(llh)
}


Evidence(X.no.int, y, c(-3), 2, 2, lambda_opt)


#####################################################################
######## BAYES PROJECT: Model Building and Diagnostics ########
#####################################################################


#Randomly shuffle the data
DataMix <- data[sample(nrow(data)),]

yMIX <- DataMix[,2]

XMIX.int <- DataMix[,-2]
XMIX.int[,1] <- 1
colnames(XMIX.int)[1] <- "Int"

XMIX.no.int <- DataMix[,-(1:2)]


#Create 10 equally size folds
folds <- cut(seq(1,nrow(XMIX.no.int)),breaks=10,labels=FALSE)


########## Full Model (L=0) ##########

model.full = lm(y~.,data=as.data.frame(X.no.int)) # model with every covariate,
    including an intercept

MSE.full <- data.frame(Run = 1:10, MSE = NA)

Evidence.full <- data.frame(Run = 1:10, Evidence = NA)

#Perform 10 fold cross validation
for(i in 1:10){
  #Segement your data by fold using the which() function
  testIndexes <- which(folds==i,arr.ind=TRUE)
  testData <- XMIX.no.int[testIndexes, ]
  trainData <- XMIX.no.int[-testIndexes, ]
  testResponse <- yMIX[testIndexes]
  trainResponse <- yMIX[-testIndexes]


  model <- lm(trainResponse~.,data=as.data.frame(trainData))

  testData$pred <- predict(model, testData, type="response")

  MSE.full[i,2] <- (1/length(testResponse)) * sum( (testResponse - testData$pred)^2 )

  Evidence.full[i,2] <- Evidence(X = testData, y = testResponse, c(1:ncol(testData)),
    a_0, b_0, lambda = 0.000000000001)

}
```

```
########## AIC Model (L=0) ##########

model.AIC = step(model.full) # AIC

Names.AIC = names(model.AIC$coeff) # the names of the coefficients chosen by AIC
Covariates.AIC = match(Names.AIC,names(X.no.int))
Covariates.AIC <- Covariates.AIC[-1]

MSE.AIC <- data.frame(Run = 1:10, MSE = NA)

Evidence.AIC <- data.frame(Run = 1:10, Evidence = NA)

#Perform 10 fold cross validation
for(i in 1:10){
  #Segement your data by fold using the which() function
  testIndexes <- which(folds==i,arr.ind=TRUE)
  testData <- XMIX.no.int[testIndexes, ]
  trainData <- XMIX.no.int[-testIndexes, ]
  testResponse <- yMIX[testIndexes]
  trainResponse <- yMIX[-testIndexes]


  model <- lm(trainResponse~.,data=as.data.frame(trainData)[,Covariates.AIC])

  testData$pred <- predict(model, testData, type="response")

  MSE.AIC[i,2] <- (1/length(testResponse)) * sum( (testResponse - testData$pred)^2 )

  Evidence.AIC[i,2] <- Evidence(X = testData, y = testResponse, Covariates.AIC, a_0, b
    _0, lambda = 0.000000000001)

}

########## BIC Model (L=0) ##########

model.BIC = step(model.full,k=log(nrow(X.no.int))) # k=2 is AIC, k= log(n) is the BIC

Names.BIC = names(model.BIC$coeff) # the names of the coefficients chosen by AIC
Covariates.BIC = match(Names.BIC,names(X.no.int))
Covariates.BIC = Covariates.BIC[-1]

MSE.BIC <- data.frame(Run = 1:10, MSE = NA)

Evidence.BIC <- data.frame(Run = 1:10, Evidence = NA)

#Perform 10 fold cross validation
for(i in 1:10){
  #Segement your data by fold using the which() function
  testIndexes <- which(folds==i,arr.ind=TRUE)
  testData <- XMIX.no.int[testIndexes, ]
  trainData <- XMIX.no.int[-testIndexes, ]
  testResponse <- yMIX[testIndexes]
  trainResponse <- yMIX[-testIndexes]


  model <- lm(trainResponse~.,data=as.data.frame(trainData)[,Covariates.BIC])

  testData$pred <- predict(model, testData, type="response")

  MSE.BIC[i,2] <- (1/length(testResponse)) * sum( (testResponse - testData$pred)^2 )
```

```
235
236   Evidence.BIC[i,2] <- Evidence(X = testData, y = testResponse, Covariates.BIC, a_0, b
        _0, lambda = 0.000000000001)
237
238 }
239
240
241
242 ########## Shrunken Model (Ridge, L = L_opt) ##########
243
244 MSE.shrunk <- data.frame(Run = 1:10, MSE = NA)
245
246 Evidence.shrunk <- data.frame(Run = 1:10, Evidence = NA)
247
248 PPV <- XMIX.no.int[,1:2]
249
250 #Perform 10 fold cross validation
251 for(i in 1:10){
252   #Segement your data by fold using the which() function
253   testIndexes <- which(folds==i,arr.ind=TRUE)
254   testData <- XMIX.no.int[testIndexes, ]
255   trainData <- XMIX.no.int[-testIndexes, ]
256   testResponse <- yMIX[testIndexes]
257   trainResponse <- yMIX[-testIndexes]
258
259
260   L <- optim(0.1,negllh.y, X=trainData, y=trainResponse, method="Brent",lower=0.001,
        upper=10,hessian=T)
261
262   L_opt <- L$par
263
264   params <- post.param(2,2,L_opt, trainData ,trainResponse)
265
266   beta_n <- params[[1]]
267
268   Sigma_n <- params[[2]]
269
270   a_n <- params[[3]]
271
272   b_n <- params[[4]]
273
274   testData$pred <- mean(testResponse) + as.matrix(testData)%*%as.matrix(beta_n)
275
276   MSE.shrunk[i,2] <- (1/length(testResponse)) * sum( (testResponse - testData$pred)^2
        )
277
278   Evidence.shrunk[i,2] <- Evidence(X = testData, y = testResponse, c(1:ncol(testData))
        , a_0, b_0, lambda = L_opt)
279
280   test.ppv = 0
281
282   for(j in 1:length(testResponse)){
283
284     sd = (b_n/a_n * (1 + as.matrix(testData[j,-65]) %*% Sigma_n %*% t(as.matrix(
        testData[j,-65]))))^0.5
285
286     test.ppv[j] = min(pnorm(testResponse[j],testData$pred[j,-65],sd), 1-pnorm(
        testResponse[j],testData$pred[j,-65],sd))
287
288   }
```

```
289
290   PPV[testIndexes,1] <- test.ppv
291
292 }
293
294 names(PPV)[1:2] <- c("P-values", "Outlier?")
295
296 for(i in 1:nrow(PPV)){
297   if(PPV[i,1] < 0.05){PPV[i,2] = TRUE}else{PPV[i,2] = FALSE}
298   }
299 summary(as.factor(PPV[,2]))
300
301 ########## Intercept Ridge Model (Ridge, L = L_opt) ##########
302
303 MSE.ridge.int <- data.frame(Run = 1:10, MSE = NA)
304
305 Evidence.ridge.int <- data.frame(Run = 1:10, Evidence = NA)
306
307 PPV.int <- XMIX.no.int[,1:2]
308
309 #Perform 10 fold cross validation
310 for(i in 1:10){
311   #Segement your data by fold using the which() function
312   testIndexes <- which(folds==i,arr.ind=TRUE)
313   testData <- XMIX.int[testIndexes, ]
314   trainData <- XMIX.int[-testIndexes, ]
315   testResponse <- yMIX[testIndexes]
316   trainResponse <- yMIX[-testIndexes]
317
318
319   L <- optim(0.1,negllh.y, X=trainData, y=trainResponse, method="Brent",lower=0.001,
        upper=10,hessian=T)
320
321   L_opt <- L$par
322
323   params <- post.param(2,2,L_opt, trainData ,trainResponse)
324
325   beta_n <- params[[1]]
326
327   Sigma_n <- params[[2]]
328
329   a_n <- params[[3]]
330
331   b_n <- params[[4]]
332
333   testData$pred <- as.matrix(testData)%*%as.matrix(beta_n)
334
335   MSE.shrunk[i,2] <- (1/length(testResponse)) * sum( (testResponse - testData$pred)^2
        )
336
337   Evidence.shrunk[i,2] <- Evidence(X = testData, y = testResponse, c(1:ncol(testData))
        , a_0, b_0, lambda = L_opt)
338
339   test.ppv = 0
340
341   for(j in 1:length(testResponse)){
342
343     sd = (b_n/a_n * (1 + as.matrix(testData[j,-66]) %*% Sigma_n %*% t(as.matrix(
      testData[j,-66]))))^0.5
344
```

```
    test.ppv[j] = min(pnorm(testResponse[j],testData$pred[j,-66],sd), 1-pnorm(
      testResponse[j],testData$pred[j,-66],sd))

  }

  PPV.int[testIndexes,1] <- test.ppv

}

names(PPV.int)[1:2] <- c("P-values", "Outlier?")

for(i in 1:nrow(PPV.int)){
  if(PPV.int[i,1] < 0.05){PPV.int[i,2] = TRUE}else{PPV.int[i,2] = FALSE}
}
summary(as.factor(PPV.int[,2]))

########## AIC/Ridge Model (Ridge, L = L_opt) ##########

MSE.AIC.R <- data.frame(Run = 1:10, MSE = NA)

Evidence.AIC.R <- data.frame(Run = 1:10, Evidence = NA)

AIC.X.no.int <- XMIX.no.int[,Covariates.AIC]

PPV.AIC.R <- AIC.X.no.int[,1:2]

#Perform 10 fold cross validation
for(i in 1:10){
  #Segement your data by fold using the which() function
  testIndexes <- which(folds==i,arr.ind=TRUE)
  testData <- AIC.X.no.int[testIndexes, ]
  trainData <- AIC.X.no.int[-testIndexes, ]
  testResponse <- yMIX[testIndexes]
  trainResponse <- yMIX[-testIndexes]


  L <- optim(0.1,negllh.y, X=trainData, y=trainResponse, method="Brent",lower=0.001,
    upper=10,hessian=T)

  L_opt <- L$par

  params <- post.param(2,2,L_opt, trainData ,trainResponse)

  beta_n <- params[[1]]

  Sigma_n <- params[[2]]

  a_n <- params[[3]]

  b_n <- params[[4]]

  testData$pred <-  mean(testResponse) + as.matrix(testData)%*%as.matrix(beta_n)

  MSE.AIC.R[i,2] <- (1/length(testResponse)) * sum( (testResponse - testData$pred)^2 )

  Evidence.AIC.R[i,2] <- Evidence(X = testData, y = testResponse, c(1:ncol(testData)),
    a_0, b_0, lambda = L_opt)

  test.ppv = 0

```

```
  for(j in 1:length(testResponse)){

    sd = (b_n/a_n * (1 + as.matrix(testData[j,-ncol(testData)]) %*% Sigma_n %*% t(as.
    matrix(testData[j,-ncol(testData)]))))^0.5

    test.ppv[j] = min(pnorm(testResponse[j],testData$pred[j,-ncol(testData)],sd), 1-
    pnorm(testResponse[j],testData$pred[j,-ncol(testData)],sd))

  }

  PPV.AIC.R[testIndexes,1] <- test.ppv

}

names(PPV.AIC.R)[1:2] <- c("P-values", "Outlier?")

for(i in 1:nrow(PPV.BIC.R)){
  if(PPV.AIC.R[i,1] < 0.05){PPV.AIC.R[i,2] = TRUE}else{PPV.AIC.R[i,2] = FALSE}
}
summary(as.factor(PPV.AIC.R[,2]))


########## BIC/Ridge Model (Ridge, L = L_opt) ##########

MSE.BIC.R <- data.frame(Run = 1:10, MSE = NA)

Evidence.BIC.R <- data.frame(Run = 1:10, Evidence = NA)

BIC.X.no.int <- XMIX.no.int[,Covariates.BIC]

PPV.BIC.R <- AIC.X.no.int[,1:2]

#Perform 10 fold cross validation
for(i in 1:10){
  #Segement your data by fold using the which() function
  testIndexes <- which(folds==i,arr.ind=TRUE)
  testData <- BIC.X.no.int[testIndexes, ]
  trainData <- BIC.X.no.int[-testIndexes, ]
  testResponse <- yMIX[testIndexes]
  trainResponse <- yMIX[-testIndexes]


  L <- optim(0.1,negllh.y, X=trainData, y=trainResponse, method="Brent",lower=0.001,
    upper=10,hessian=T)

  L_opt <- L$par

  params <- post.param(2,2,L_opt, trainData ,trainResponse)

  beta_n <- params[[1]]

  Sigma_n <- params[[2]]

  a_n <- params[[3]]

  b_n <- params[[4]]

  testData$pred <-  mean(testResponse) + as.matrix(testData)%*%as.matrix(beta_n)

  MSE.BIC.R[i,2] <- (1/length(testResponse)) * sum( (testResponse - testData$pred)^2 )
```

```
459
460   Evidence.BIC.R[i,2] <- Evidence(X = testData, y = testResponse, c(1:ncol(testData)),
        a_0, b_0, lambda = L_opt)
461
462   test.ppv = 0
463
464   for(j in 1:length(testResponse)){
465
466     sd = (b_n/a_n * (1 + as.matrix(testData[j,-ncol(testData)]) %*% Sigma_n %*% t(as.
      matrix(testData[j,-ncol(testData)]))))^0.5
467
468     test.ppv[j] = min(pnorm(testResponse[j],testData$pred[j,-ncol(testData)],sd), 1-
      pnorm(testResponse[j],testData$pred[j,-ncol(testData)],sd))
469
470   }
471
472   PPV.BIC.R[testIndexes,1] <- test.ppv
473
474 }
475
476 names(PPV.BIC.R)[1:2] <- c("P-values", "Outlier?")
477
478 for(i in 1:nrow(PPV.BIC.R)){
479   if(PPV.BIC.R[i,1] < 0.05){PPV.BIC.R[i,2] = TRUE}else{PPV.BIC.R[i,2] = FALSE}
480 }
481 summary(as.factor(PPV.BIC.R[,2]))
482
483
484
485 ########## AIC/Ridge.int Model (Ridge, L = L_opt) ##########
486
487 MSE.AIC.R.int <- data.frame(Run = 1:10, MSE = NA)
488
489 Evidence.AIC.R.int <- data.frame(Run = 1:10, Evidence = NA)
490
491 AIC.X.int <- XMIX.int[,c(1,Covariates.AIC+1)]
492
493 PPV.AIC.R.int <- AIC.X.int[,1:2]
494
495 #Perform 10 fold cross validation
496 for(i in 1:10){
497   #Segement your data by fold using the which() function
498   testIndexes <- which(folds==i,arr.ind=TRUE)
499   testData <- AIC.X.int[testIndexes, ]
500   trainData <- AIC.X.int[-testIndexes, ]
501   testResponse <- yMIX[testIndexes]
502   trainResponse <- yMIX[-testIndexes]
503
504
505   L <- optim(0.1,negllh.y, X=trainData, y=trainResponse, method="Brent",lower=0.001,
      upper=10,hessian=T)
506
507   L_opt <- L$par
508
509   params <- post.param(2,2,L_opt, trainData ,trainResponse)
510
511   beta_n <- params[[1]]
512
513   Sigma_n <- params[[2]]
514
```

```
  a_n <- params[[3]]

  b_n <- params[[4]]

  testData$pred <- as.matrix(testData)%*%as.matrix(beta_n)

  MSE.AIC.R.int[i,2] <- (1/length(testResponse)) * sum( (testResponse - testData$pred)^2 )

  Evidence.AIC.R.int[i,2] <- Evidence(X = testData, y = testResponse, c(1:ncol(testData)), a_0, b_0, lambda = L_opt)

  test.ppv = 0

  for(j in 1:length(testResponse)){

    sd = (b_n/a_n * (1 + as.matrix(testData[j,-ncol(testData)]) %*% Sigma_n %*% t(as.matrix(testData[j,-ncol(testData)]))))^0.5

    test.ppv[j] = min(pnorm(testResponse[j],testData$pred[j,-ncol(testData)],sd), 1-pnorm(testResponse[j],testData$pred[j,-ncol(testData)],sd))

  }

  PPV.AIC.R.int[testIndexes,1] <- test.ppv

}

names(PPV.AIC.R.int)[1:2] <- c("P-values", "Outlier?")

for(i in 1:nrow(PPV.AIC.R.int)){
  if(PPV.AIC.R.int[i,1] < 0.05){PPV.AIC.R.int[i,2] = TRUE}else{PPV.AIC.R.int[i,2] = FALSE}
}
summary(as.factor(PPV.AIC.R.int[,2]))



########## BIC/Ridge.int Model (Ridge, L = L_opt) ##########

MSE.BIC.R.int <- data.frame(Run = 1:10, MSE = NA)

Evidence.BIC.R.int <- data.frame(Run = 1:10, Evidence = NA)

BIC.X.int <- XMIX.int[,c(1,Covariates.BIC+1)]

PPV.BIC.R.int  <- BIC.X.int[,1:2]

#Perform 10 fold cross validation
for(i in 1:10){
  #Segement your data by fold using the which() function
  testIndexes <- which(folds==i,arr.ind=TRUE)
  testData <- BIC.X.int[testIndexes, ]
  trainData <- BIC.X.int[-testIndexes, ]
  testResponse <- yMIX[testIndexes]
  trainResponse <- yMIX[-testIndexes]


  L <- optim(0.1,negllh.y, X=trainData, y=trainResponse, method="Brent",lower=0.001, upper=10,hessian=T)
```

```
569
570   L_opt <- L$par
571
572   params <- post.param(2,2,L_opt, trainData ,trainResponse)
573
574   beta_n <- params[[1]]
575
576   Sigma_n <- params[[2]]
577
578   a_n <- params[[3]]
579
580   b_n <- params[[4]]
581
582   testData$pred <- as.matrix(testData)%*%as.matrix(beta_n)
583
584   MSE.BIC.R.int [i,2] <- (1/length(testResponse)) * sum( (testResponse - testData$pred
        )^2 )
585
586   Evidence.BIC.R.int [i,2] <- Evidence(X = testData, y = testResponse, c(1:ncol(
        testData)), a_0, b_0, lambda = L_opt)
587
588   test.ppv = 0
589
590   for(j in 1:length(testResponse)){
591
592     sd = (b_n/a_n * (1 + as.matrix(testData[j,-ncol(testData)]) %*% Sigma_n %*% t(as.
        matrix(testData[j,-ncol(testData)]))))^0.5
593
594     test.ppv[j] = min(pnorm(testResponse[j],testData$pred[j,-ncol(testData)],sd), 1-
        pnorm(testResponse[j],testData$pred[j,-ncol(testData)],sd))
595
596   }
597
598   PPV.BIC.R.int [testIndexes,1] <- test.ppv
599
600 }
601
602 names(PPV.BIC.R.int)[1:2] <- c("P-values", "Outlier?")
603
604 for(i in 1:nrow(PPV.BIC.R.int)){
605   if(PPV.BIC.R.int[i,1] < 0.05){PPV.BIC.R.int[i,2] = TRUE}else{PPV.BIC.R.int[i,2] =
        FALSE}
606 }
607 summary(as.factor(PPV.BIC.R.int[,2]))
608
609
610 ########## Model Comparison ##########
611
612 MSE.compare <- function(){
613   MSE <- data.frame(fold = 1:10, Full = MSE.full[,2], Ridge = MSE.shrunk[,2], Ridge.
        int = MSE.ridge.int[,2], AIC = MSE.AIC[,2],
614                     BIC = MSE.BIC[,2], Ridge.glmnet = MSE.Ridge.auto[,2], Ridge.MASS =
        MSE.Ridge.auto.MASS[,2],
615                     AIC.Ridge = MSE.AIC.R[,2], BIC.Ridge = MSE.BIC.R[,2], AIC.Ridge.int
        = MSE.AIC.R.int[,2], BIC.Ridge.int = MSE.BIC.R.int[,2])
616
617   MSE[11,1] <- "Avg."
618   MSE[11,2] <- mean(MSE[1:10,2])
619   MSE[11,3] <- mean(MSE[1:10,3])
620   MSE[11,4] <- mean(MSE[1:10,4])
```

```
  MSE[11,5] <- mean(MSE[1:10,5])
  MSE[11,6] <- mean(MSE[1:10,6])
  MSE[11,7] <- mean(MSE[1:10,7])
  MSE[11,8] <- mean(MSE[1:10,8])
  MSE[11,9] <- mean(MSE[1:10,9])
  MSE[11,10] <- mean(MSE[1:10,10])
  MSE[11,11] <- mean(MSE[1:10,11])
  MSE[11,12] <- mean(MSE[1:10,12])

  MSE[12,1] <- "SD"
  MSE[12,2] <- sd(MSE[1:10,2])
  MSE[12,3] <- sd(MSE[1:10,3])
  MSE[12,4] <- sd(MSE[1:10,4])
  MSE[12,5] <- sd(MSE[1:10,5])
  MSE[12,6] <- sd(MSE[1:10,6])
  MSE[12,7] <- sd(MSE[1:10,7])
  MSE[12,8] <- sd(MSE[1:10,8])
  MSE[12,9] <- sd(MSE[1:10,9])
  MSE[12,10] <- sd(MSE[1:10,10])
  MSE[12,11] <- sd(MSE[1:10,11])
  MSE[12,12] <- sd(MSE[1:10,12])

  return(MSE)
}

Evi.compare <- function(){
  Evi <- data.frame(fold = 1:10, Full = Evidence.full[,2], Ridge = Evidence.shrunk
    [,2], Ridge.int = Evidence.ridge.int[,2], AIC = Evidence.AIC[,2],
                    BIC = Evidence.BIC[,2], Ridge.auto = Evidence.Ridge.auto[,2],
    Ridge.MASS = Evidence.Ridge.auto.MASS[,2],
                    AIC.Ridge = Evidence.AIC.R[,2], BIC.Ridge = Evidence.BIC.R[,2],
    AIC.Ridge.int = Evidence.AIC.R.int[,2], BIC.Ridge.int = Evidence.BIC.R.int[,2])

  Evi[11,1] <- "Avg."
  Evi[11,2] <- mean(Evi[1:10,2])
  Evi[11,3] <- mean(Evi[1:10,3])
  Evi[11,4] <- mean(Evi[1:10,4])
  Evi[11,5] <- mean(Evi[1:10,5])
  Evi[11,6] <- mean(Evi[1:10,6])
  Evi[11,7] <- mean(Evi[1:10,7])
  Evi[11,8] <- mean(Evi[1:10,8])
  Evi[11,9] <- mean(Evi[1:10,9])
  Evi[11,10] <- mean(Evi[1:10,10])
  Evi[11,11] <- mean(Evi[1:10,11])
  Evi[11,12] <- mean(Evi[1:10,12])

  Evi[12,1] <- "SD"
  Evi[12,2] <- sd(Evi[1:10,2])
  Evi[12,3] <- sd(Evi[1:10,3])
  Evi[12,4] <- sd(Evi[1:10,4])
  Evi[12,5] <- sd(Evi[1:10,5])
  Evi[12,6] <- sd(Evi[1:10,6])
  Evi[12,7] <- sd(Evi[1:10,7])
  Evi[12,8] <- sd(Evi[1:10,8])
  Evi[12,9] <- sd(Evi[1:10,9])
  Evi[12,10] <- sd(Evi[1:10,10])
  Evi[12,11] <- sd(Evi[1:10,11])
  Evi[12,12] <- sd(Evi[1:10,12])

  return(Evi)
```

```
}

MSE <- MSE.compare()
Evi <- Evi.compare()
```