Simulation Studies

1 Simulation Studies

This chapter investigates the effect of manipulating some of the `mcmc_ridge()` inputs and thereby sampling parameters in a controlled environment, such that changes in the estimation outcome can be directly linked to respective changes in the model inputs. The simulation studies discussed in this chapter can be categorized into three groups:

1. Evaluating the **quality** and **robustness** of the sampler.

   Sections 1.1 and 1.4 create data configurations that can be challenging for estimation. The variation in the input parameters is therefore focused on the function argument `m` or alternatively the combination of `X`, `Z` and `y`. Section 1.2 is dedicated to manipulating the sample size through the input parameter `n`. Here, we are looking for a possible stabilization process with increasing size of the input data hinting at asymptotic/convergence properties.

2. Investigating the **penalty** effect.

   Sections 1.3 and 1.4 replicate real world estimation scenarios, where the `mcmc_ridge()` sampler is primarily chosen for the desired regularization effect. To emphasize the shrinkage effect on the coefficient estimates, the `mcmc_ridge()` model is compared to the `lmls()` and the `mcmc()` models from the `lmls` package, which do not leverage a penalty.

3. Exploring the effect of the **hyperparameters**.

   Sections 1.5 and 1.6 analyze the connection between the input parameters `a_tau`, `b_tau`, `a_xi` and `b_xi` of the Inverse Gamma prior distributions of τ^2 and ξ^2 to the simulation results. The effect of hyperparameters in a hierarchical Bayesian model can be difficult to predict based on pure logical reasoning. Therefore simulations are a useful tool to either confirm prior assumptions or discover unexpected behaviour.

Since the resulting simulation studies serve such diverse purposes (e.g. diagnostic vs. explorative), they demand for different approaches in the simulation settings, the implementation as well as the analysis and presentation of the results. For that reason, we decided against forcing all of the following sections into one common rigid framework. Instead, each section individually motivates, explains and interprets the methods chosen for its particular use case.

In order to keep the analysis in this chapter compact and succinct, there will be almost no code included. It it worth noting though that the `R Markdown` document itself as well as all `R Scripts` used for the simulations are contained in the `simulation-studies` folder of the `asp21bridge` package. Thus, each figure as well as all numerical results are fully reproducible and can be repeated and extended by the reader.

1.1 Correlated Predictor Variables

Up to this point, we have often illustrated the usage and results of the `mcmc_ridge()` sampler with simulated data from the built-in `toy_data` set. As stated in section **??**, each regressor variable is independently sampled from a normal distribution and the outcome variable is simulated based on the correctly specified location-scale regression model introduced in chapter **??**. All these conditions lead to an excellent performance of the `mcmc_ridge()` sampler, but might arguably not represent the most challenging task.

Sections 1.1 and 1.4 analyze the sampler's performance on simulated data, which might be closer to data found in the real world. First, we will induce correlation among the predictor variables, whereas in a later

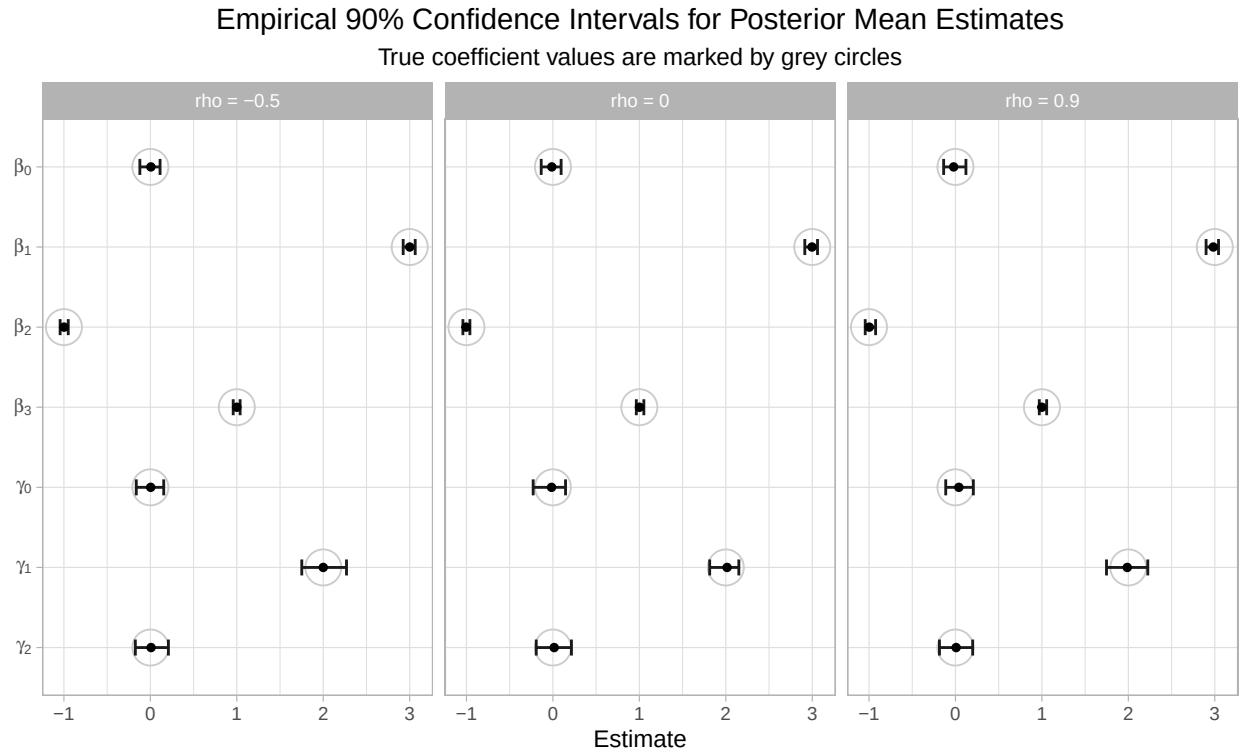

Figure 1: Comparison of Correlation Structures - 50 Simulation Cycles

part the distributional assumptions are considerably changed. Further, the `mcmc_ridge()` performance is compared to the Maximum Likelihood based `lmls()` estimates and the Markov Chain Monte Carlo `mcmc()` sampler.

Simulation Setting

Briefly stated, we simulated data from a three dimensional multivariate normal distribution with three different values for the pairwise correlation ρ. Then, we fitted the three models `mcmc_ridge()`, `mcmc()` and `lmls()`, where the number of simulations was set to 10000 for the Bayesian Samplers.

This simulation study is intended to evaluate the *accuracy* of the three models. Therefore, the same (standardized) values of the covariates are used for both, the generation of the outcome variable $\mathbf{y}$ and the model fit. This entails that the true coefficient values are known and can be compared to the resulting estimates of all three and in particular the `mcmc_ridge()` model. The exact simulation setting is described in full detail in the Appendix.

Moreover, we compared the performance of the classical `mcmc_ridge()` implementation, which draws $\boldsymbol{\beta}$ from the closed form full conditional (multivariate normal) distribution, with an alternative sampling process that uses a Metropolis-Hastings approach for both, the location parameter $\boldsymbol{\beta}$ as well as the scale parameter $\boldsymbol{\gamma}$.

Simulation Results

All three models perform very similarly for each correlation structure: The estimates for $\boldsymbol{\beta}$ are almost exactly equal to the true values, whereas there are slight deviations for the $\boldsymbol{\gamma}$ vector. These errors, however, are consistent in all three sampling scenarios and, thus, do not depend on the strength nor the direction of the correlation between the covariates. Due to this similarity between the three models, the remaining analysis of this simulation study is dedicated to the `mcmc_ridge()` model.

There is one additional aspect to be noted: The performance of the `mcmc_ridge()` function is considerably

worse, when $\boldsymbol{\beta}$ is sampled by means of the Metropolis-Hastings algorithm. Although the acceptance rates of the proposed values are in acceptable ranges for a random walk proposal (between 35% and 40%), the chains are strongly correlated and did not fully converge even after 10000 iterations. For that reason, we limit the Metropolis-Hastings sampling process for $\boldsymbol{\beta}$ to this one example and will focus on the classical `mcmc_ridge()` implementation in the remaining parts of chapter 1.

The corresponding graphical display for the comparison between the three models and between the two sampling procedures for $\boldsymbol{\beta}$ is omitted here for brevity reasons, since the findings are not that surprising. The code as well as all plots can be found and fully reproduced in the `regressor-correlation.R` file located inside of the `simulation-studies` folder of the project directory.

In order to make any conclusions about bias and variance, the above procedure is repeated 50 times with 1000 simulations each. The black points in Figure 1 represent the mean of these 50 Posterior Mean estimates from the penalized Ridge sampler. For a better visual comparison, the true values for each coefficient are indicated by grey circles. Since we cannot rely on distributional theory for the standard errors, the variability of the estimates is displayed by nonparametric 'confidence' intervals, which are simply given by the range from the empirical 0.05 quantile to the 0.95 quantile of the 50 estimated values.

These intervals are very narrow and centered around the true value in all cases with slightly larger variability for the $\boldsymbol{\gamma}$ vector, which is sampled via the Metropolis-Hastings algorithm. Overall, these results impressively solidify the sampler's stability in presence of correlated covariates. The bias towards zero that is typical for the Ridge penalty cannot be observed in this case; all coefficients are estimated with both high accuracy and high precision.

It is worth noting that the results are partially affected by the standardization of the covariates. The standardization generally leads to a stabilized fit for all three models. Compared to the performance for unstandardized data (as used for the second report) the variability of the Posterior Mean estimates of β_0 in particular is significantly reduced.

1.2 Sample Size

This simulation study analyzes the effect of the sample size `n` on the means of the posterior distribution for the coefficients of $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$. There are two main goals of this simulation study: On the one hand, we want to investigate whether the posterior means of large samples are closer to the true values than the posterior means of small samples. On the other hand, we want to analyze whether the `mcmc_ridge()` penalty affects the location of the posterior means.

Simulation Setting

- The design matrix $\mathbf{X} = \begin{pmatrix} \mathbf{1}_n & \mathbf{x}_1 & \mathbf{x}_2 \end{pmatrix}$ contains two independently sampled regressor variables plus one intercept column:
  - $\mathbf{x}_1 \overset{iid}{\sim} \mathcal{N}(1, 1)$,
  - $\mathbf{x}_2 \overset{iid}{\sim} \mathcal{N}(2, 1)$.
- The design matrix $\mathbf{Z} = \begin{pmatrix} \mathbf{1}_n & \mathbf{z}_1 & \mathbf{z}_2 \end{pmatrix}$ is structured in the same way with the regressor variables:
  - $\mathbf{z}_1 \overset{iid}{\sim} \mathcal{N}(1, 1)$,
  - $\mathbf{z}_2 \overset{iid}{\sim} \mathcal{N}(2, 1)$.
- The true coefficient vectors are given by $\boldsymbol{\beta} = \begin{pmatrix} \beta_0 & \beta_1 & \beta_2 \end{pmatrix}^T = \begin{pmatrix} 1 & -1 & 4 \end{pmatrix}^T$ and $\boldsymbol{\gamma} = \begin{pmatrix} \gamma_0 & \gamma_1 & \gamma_2 \end{pmatrix}^T = \begin{pmatrix} 0 & -0.5 & 1 \end{pmatrix}^T$.
- The posterior means are analyzed with respect to 6 different sample sizes: $\mathtt{n} \in \{0, 50, 100, 200, 300, 500\}$.
- In the next step, the outcome vector $y \in \mathbb{R}^n$ is simulated and passed to the `mcmc_ridge()` function with $\mathtt{nsim} = 500$ simulations.

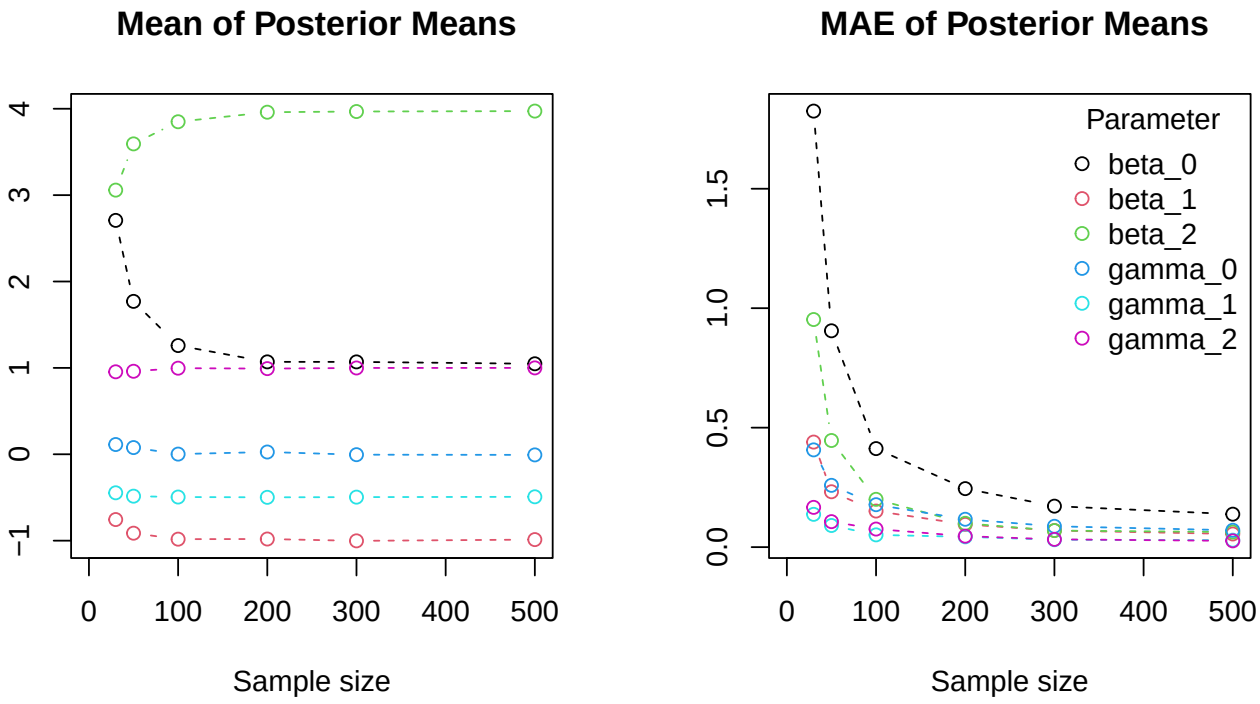

Figure 2: Mean and MAE of 100 Posterior Mean Estimates

- To make the results more stable, the above procedure is repeated 100 times. For each coefficient, the mean value of the Posterior Mean estimates of each coefficient is calculated as well as the Mean Absolute Error (MAE) with respect to the true values of $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$.

Simulation Results

The means of the Posterior Mean estimates are displayed in the left plot of Figure 2. For larger sample sizes ($\texttt{n} \geq 200$) none of the six parameters are extremely biased.

Moreover, for $\texttt{n} = 30$, β_0 and β_2 are significantly biased, which might be caused by the high `mcmc_ridge()` penalty for $\beta_2 = 4$. The significant bias of β_0 might be explained by a counteract of the β_2 bias.

After getting an impression about empirical biases of the coefficients, we now focus on the variability of the posterior means of the coefficients, which are measured by the MAE based on the results of the 100 repetitions. The right plot of Figure 2 points out that the posterior means of β_0 have significantly larger errors than the posterior means of β_2 for $\texttt{n} = 30$. However, this might also be explained by the fact that for $\texttt{n} = 30$, β_0 has a greater empirical bias than β_2 as could be observed in the left panel of Figure 2.

In addition, for increasing sample sizes, the MAE of the Posterior Means tend to zero for all coefficients except β_0. Nevertheless, also the errors of β_0 seem to become smaller with increasing sample size.

1.3 Redundant Covariates

Similar to the simulation study from section 1.1, this section investigates the sampler's behaviour in presence of strong *pairwise* correlation among the covariates. However, in contrast to all other simulation studies, we add additional *redundant* regressors to the model, that were not used for generating the outcome variable $\mathbf{y}$.

The statistical theory of (Ridge) penalization suggests, that the coefficient estimates from the `mcmc_ridge()` function are affected by the shrinkage effect induced by the coefficient's prior distributions in presence of high (compared to the sample size n) number of correlated covariates. Thus, the magnitude of the estimated $\boldsymbol{\beta}$ vector should be smaller compared to models without regularizing penalty component.

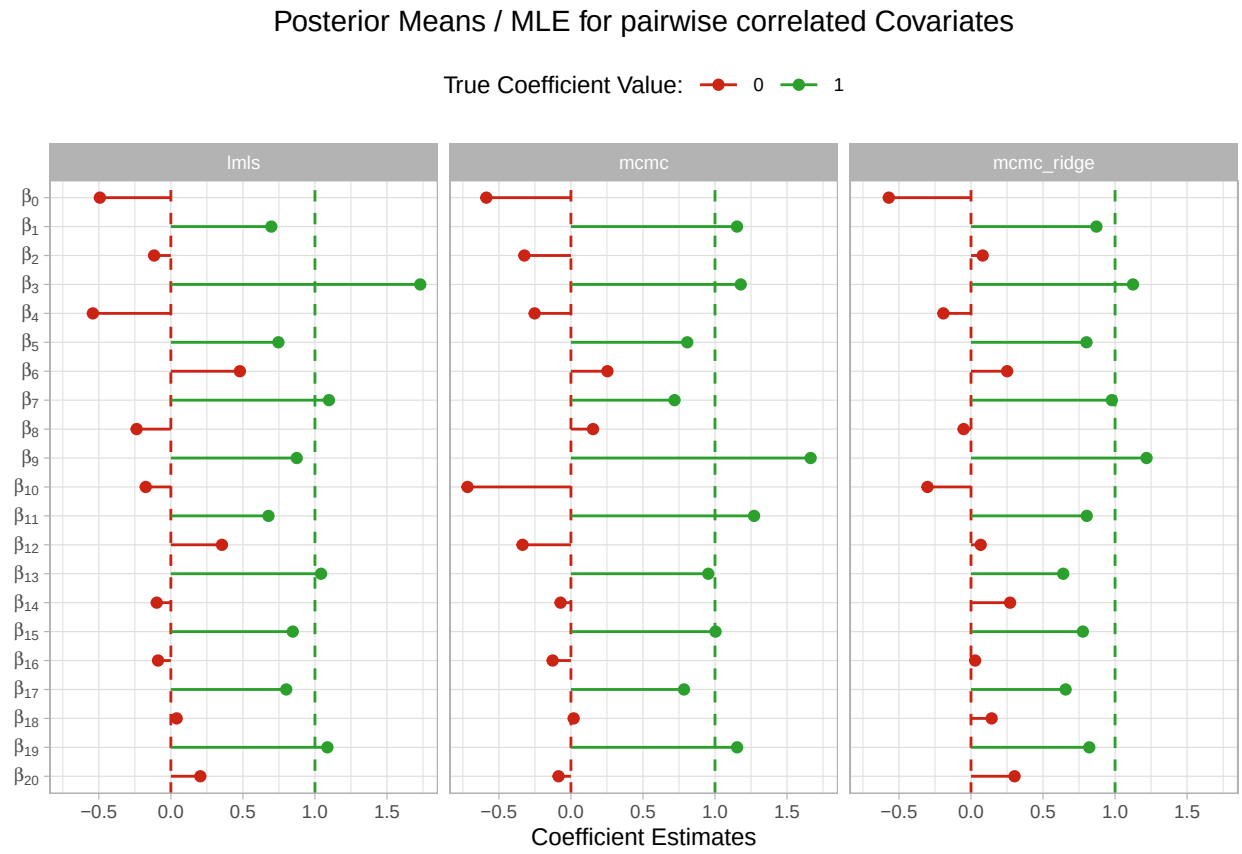

Figure 3: Shrinkage Effect of Ridge Penalty in Presence of Redundant Covariates

Simulation Setting

The model contains 10 pairs of covariates drawn from a multivariate normal distribution with correlation $\rho = 0.9$. For each pair, the first covariate contributes to generating the outcome variable $\mathbf{y}$ with a true coefficient value of 1. The second covariate has no impact on $\mathbf{y}$, but is included nonetheless in the model fitting stage. Similar to section 1.4, the `lmls()`, `mcmc()` and `mcmc_ridge()` functions are used to generate estimates.

Since the purpose of this study is to demonstrate the Ridge regularization and not primarily the accuracy of the three models, we used the covariates on their original scale for generating $\mathbf{y}$ and standardized them afterwards before fitting the models. Thus, the true coefficient values are not known and conclusions about the model *quality* are not valid. This scenario resembles a real world application, where the data generating process is latent and estimates are obtained based on purely observational data. The exact design of the simulation study can again be found in the Appendix.

Simulation Results

The following analysis is exclusively focused on the location parameter $\boldsymbol{\beta}$. As mentioned before, there are two aspects of interest we wish to investigate further:

1. Can each model differentiate between the relevant and the redundant covariate for each pair of regressor and, thus, isolate the true effect on $\mathbf{y}$ (without paying too much attention to the *absolute* size of the estimated values)?
2. Due to the high number of 20 covariates compared to the low sample size of 50 observations and the presence of correlated and partially redundant regressors, the setup of this study is prone to

Table 1: $\|\boldsymbol{\beta}\|^2$ excluding β_0

	True Value = 0	True Value = 1
lmls	1.05	10.09
mcmc	1.26	12.13
mcmc_ridge	0.72	7.87

overfitting. Can we therefore observe the hypothesized shrinkage effect on the coefficient estimates for the `mcmc_ridge()` model compared to the models without penalty?

Figure 3 illustrates the results obtained by fitting all three models.

We start with the answer to the first question: For each of the three models and each coefficient pair the absolute value of the coefficient corresponding to relevant regressor is larger than the magnitude of the paired quantity. Thus all three models always contribut the major impact on the outcome variable to the correct covariate. Further, the differences between estimations for those coefficients with true value 1 and those with true value 0 do not differ in an *obvious* manner across the models. Since the variance of the estimates seems to be largest for the `lmls()` model and smallest for the `mcmc_ridge()` model, the ratio (instead of the difference) tends to be largest for the regularized model.

The reduced variance of the `mcmc_ridge()` model can be explained by the Ridge penalty: Since all estimates are shrunken towards zero, fewer 'outlier' estimates that are far away from zero will be observed. This property is beneficial for the red points, since the estimates clutter around their true value in this case. However, the coefficients corresponding to the green points are penalized as well such that a bias towards zero is induced, which is not present for the two unpenalized models. Hence, the hypothesized shrinkage effect can indeed be observed.

It is insightful to compare the observed effect from the Ridge penalty to the expected effect of the *LASSO* penalty. The LASSO penalty encourages *sparse* solutions: each coefficient is assigned the same weight such that shrinkage of large estimates is not preferred over shrinkage of small estimates. Thus, small estimates are often set to exactly zero, while large estimates, although reduced, can still be significantly different from zero.

In contrast, the Ridge penalty generally does not induce sparsity. In fact, quite the opposite can be the case: Small coefficient estimates such as β_{14} in Figure 3 are sometimes *increased* in magnitude compared to the unpenalized models. Large estimates such as β_4 or β_{12}, however, are reduced in size quite heavily. The Ridge penalty tries to put all coefficients on the same scale and, thus, prioritizes shrinkage of large estimates compared to small ones! Note also, that the intercept term is typically excluded from the penalization, wich is the reason why the estimates of β_0 are of approximately equal size for all three models.

In order to quantify the observed shrinkage effect from Figure 3 numerically, we calculate the sum of squared coefficient estimates, i.e. the squared Euclidean norm $\|\boldsymbol{\beta}\|^2$, for each model. Moreover, these magnitudes are compared separately for all coefficients corresponding to relevant regressors (those with an *odd* subscript) and those corresponding to redundant regressors (*even* subscript). Table 1 summarizes the results.

The most interesting and reassuring finding is obtained from the `mcmc_ridge()` vector norms: In agreement with the visual illustration, the Ridge regularization effect can be numerically detected for all coefficient estimates, independent of the true value. While this property induces a bias that is not desired for the second column (underestimating the true values on average), it does indeed prevent overfitting by shrinking the coefficients corresponding to the redundant covariates, which is indicated by the lowest value in the first column.

In that sense, the Ridge penalty increases the model *robustness* in presence of a high number of (correlated) regressors and often leads to an overall lower Mean Squared Error due to the reduction in variance.

1.4 Challenging the Model Assumptions

This simulation study is structured in a very similar way to the study considered in section 1.1. Instead of varying the correlation structure among the regressors in the underlying data set, both the regressors and the outcome variable $\mathbf{y}$ are sampled from distributions that are more challenging for estimation than the normal distribution.

Simulation Setting

More specifically, the covariates of the design matrices $\mathbf{X}$ and $\mathbf{Z}$ are generated by a mix of distributions, including the Normal, Exponential, Uniform, Bernoulli and t - distribution. Given these values, the outcome variable $\mathbf{y}$ is drawn from three different distributions (Normal, t and Uniform) corresponding to three separate scenarios, which are compared in the following analysis. As usual, the full simulation design is stated in the Appendix.

Note that the `lmls()`, `mcmc()` and `mcmc_ridge()` models are built upon the assumption of a Gaussian distribution for $\mathbf{y}$. Hence, all three estimation procedures are expected to perform well under the first outcome specification, which they were designed for. The remaining two cases present mild (in case of the t distribution) and moderately strong (Uniform distribution) violations of this model assumption.

It is worth noting that the results vary heavily across models and outcome distributions if the models are fit with covariates on their original scale, as seen in our Second Report. For this simulation study standardization makes a huge difference, such that coefficient estimates are a lot more consistent for all $3 \cdot 3 = 9$ scenarios, when the models are fit with standardized regressor columns.

Quite surprisingly, the results also considerably depend on the data generation of the outcome $\mathbf{y}$: When covariates are already standardized before contributing to $\mathbf{y}$, MLE and Posterior Mean Estimates are almost exact with very low variability. When the features are standardized after the data generation, the estimates are still accurate (with respect to the scale transformed original true values), but indicate a larger variance. For that reason, we decided to focus this study on the penalization effect rather than precision metrics and therefore will not compare estimates to the (transformed) true coefficient values.

Simulation Results

Figure 4 shows the results for each of the 9 scenarios over 50 simulation cycles. The points indicate the means of the 50 MLE / Posterior Mean estimates, the 'confidence' bounds are given by the empirical 5% and 95% quantiles of those 50 values and, thus, not based on distributional assumptions. As a brief technical note, although the second facet is labeled by $y \sim t$, it is formally sampled from an affine transformation of a t-distributed random variable, which does not follow an exact t distribution.

Overall, the $\boldsymbol{\gamma}$ coefficients seem to be much less affected by the penalty for the given choice of hyperparameters that is used by the `mcmc_ridge()` model. The effect of the hyperparameter values on the Ridge Penalty is discussed in section 1.6, such that we do not investigate this observation further here. Instead we take a closer look at the $\boldsymbol{\beta}$ estimates.

We first examine the results for values close to the dotted zero line: Compared to the `lmls()` and `mcmc()` models without penalty, there is a slight shrinkage effect for β_3 estimates observable in each of the facets. Further, the increased variability for some of the small values such as β_2 in the left plot or β_4 in the right plot is interesting.

One potential explanation was already introduced in section 1.3: Rather than shrinking small coefficients to zero exactly, the Ridge penalty tries to put all parameters (except the intercept) on equal scale, while simultaneously reducing the sum of squared coefficient estimates. Thus, taking β_2 as an example, the original small estimate of β_2 might increase in simulation cycles, where many of the remaining coefficients are large in absolute value in order to close the gap between them. In contrast, β_2 might stay approximately equal in those cases, when other estimates happen to be small as well for this iteration and all estimates shrink synchronously.

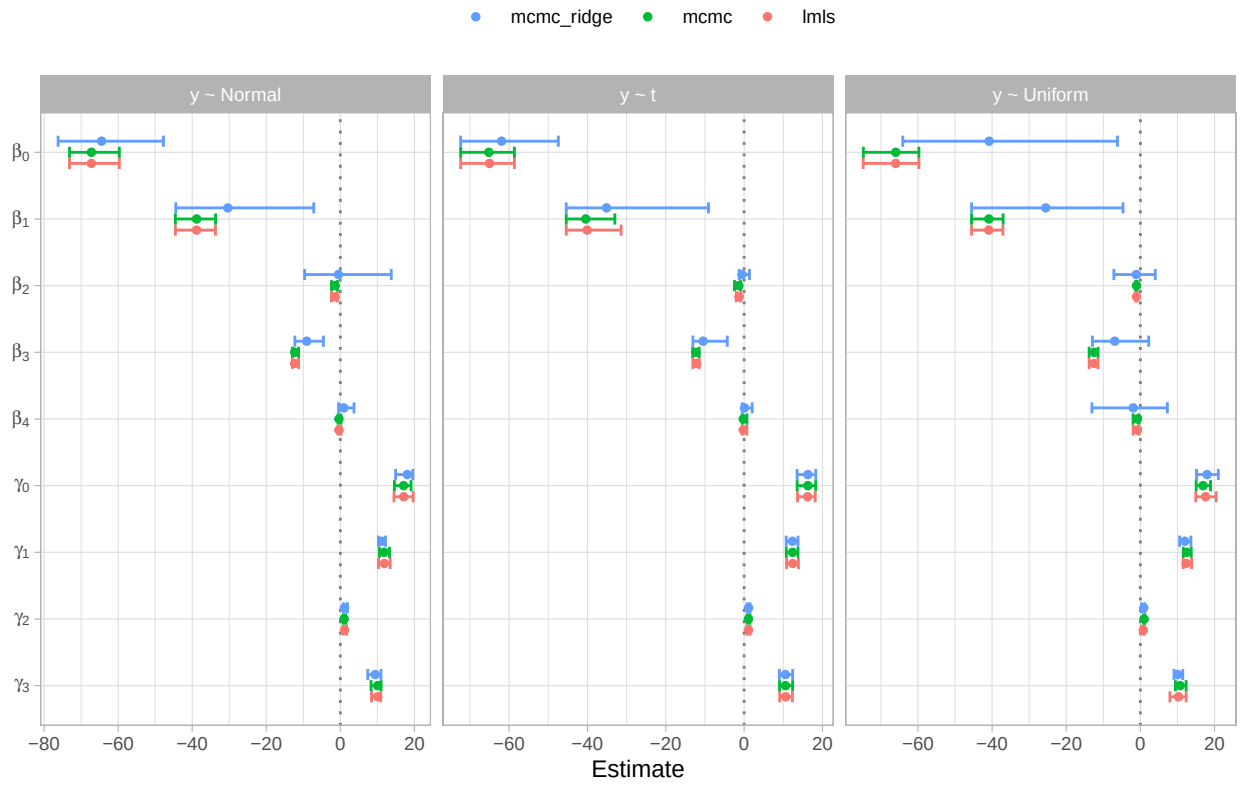

Figure 4: Comparison of different Outcome Distributions - 50 Simulation Cycles

The penalty effect is most obvious for the largest estimate, in this case β_1. For all three outcome distributions, the estimates are consistently closest to zero compared to the unpenalized models. The size of the penalty in each simulation cycle can vary quite heavily, as indicated by the empirical confidence bounds, potentially for the reasons stated above.

In addition to the visual illustration, the penalty effect can be numerically established by computing the (squared) Euclidean Norm of the coefficient vectors $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, since this is the quantity involved in Ridge penalization. Excluding the intercept estimates, the calculated values for this simulation study are shown in Table 2. Confirming the graphical impression, the values of the unpenalized models are almost identical, independent of the outcome distribution, whereas the magnitude of the `mcmc_ridge()` coefficients is significantly reduced in all cases.

Although there are some differences among the three plot facets in Figure 4, the observed patterns appear too random to draw meaningful conclusions about the connection between the strength of violating the distributional assumptions and the regularization. The reduced estimate of the intercept parameter in the right panel is not caused by the Ridge penalty, in which case the observed effect would have been much stronger in all plot facets, but rather by the generally increased estimation variance of the intercept terms in the `mcmc_ridge()` framework. The large magnitude of the intercept term is induced by mean-centering all covariates between data generation and model fitting.

Technical Aspects

As outlined in the previous paragraph, a total of $50 \cdot 9 = 450$ models were fitted to analyze the differences across models. In order to speed up the involved computations of this specific and some of the other simulation studies in this chapter, we used the *parallel computing* capabilities of `R`.

Table 2: $\|\boldsymbol{\beta}\|^2 + \|\boldsymbol{\gamma}\|^2$ excluding β_0 and γ_0

	Normal	t	Uniform
lmls	1906	2024	2088
mcmc	1903	2054	2100
mcmc_ridge	1222	1609	950

There are many options from various packages to choose from. We decided to use the furrr package, which is built on top of the `future` package specialized on parallel processing. As the name suggests, `furrr` provides a convenient way to use many functions from the popular `purrr` package, while using multiple cores at the same time. This *functional programming* based approach (similar to the `apply()` family in 'base `R`') is particularly well suited for simulation studies and provides some structural as well as minor performance advantages compared to the classical `for`-loop approach.

The following (slightly modified) code snippet provides a brief insight into the implementation:

```
plan(multisession, workers = 8)

full_results <- tibble(id = 1:50) %>%
  mutate(samples = future_map(
    .x = id,
    .f = ~ show_results(n = 50, num_sim = 1000),
    .options = furrr_options(seed = 1)
  ))
```

The `plan()` function borrowed from the `future` package initializes the parallel computing process and determines the number of cores/workers available for computation. The `show_results()` helper function fits all three models `mcmc_ridge()`, `mcmc()` and `lmls()` for each outcome distribution in a single simulation cycle.

This entire procedure is repeated 50 times in parallel using the `future_map()` function from the `furrr` package, where the results of all 450 models are saved in a well organized structure inside of a list column. This new column of the data frame contains complete information about all simulations, such that any required element for the further analysis can be easily extracted and post processed.

Finally, the `.options()` argument allows the specification of a random seed. Random number generation in the context of parallel computing is slightly more involved compared to the sequential approach. This additional complexity is automatically handled by the `future_map()` function, such that all results are sampled in a statistically valid and fully reproducible manner.

1.5 Hyperparameters: Impact on Estimation Accuracy

In the past, we have been sampling data with the `mcmc_ridge()` function without having a closer look on the effect of the hyperparameters and model inputs `a_tau`, `b_tau`, `a_xi` and `b_xi`. However, they affect the Full Conditional Distributions of τ^2 and ξ^2, as stated in sections **??** and **??** in chapter **??**.

Moreover, the mean vector $\boldsymbol{\mu}_{beta}$ and covariance matrix $\boldsymbol{\Sigma}_{beta}$ of the $\boldsymbol{\beta}$ vector both depend on τ^2 and, thus, implicitly on the hyperparameters a_τ and b_τ (see section **??**). Analogously, section **??** illustrates the direct effect of the Full Conditional distribution of $\boldsymbol{\gamma}$ on ξ^2, which in turn depends on the hyperparameters a_ξ and b_ξ.

Finally, cross effects can be observed, since $f(\boldsymbol{\beta} \mid \cdot)$ depends on $\boldsymbol{\gamma}$ through the quantities $\mathbf{W}$ and $\mathbf{u}$ as defined in chapter **??** and $f(\boldsymbol{\gamma} \mid \cdot)$ directly depends on $\boldsymbol{\beta}$. These dependencies are reflected in the `mcmc_ridge()` sampler by the iterative sampling procedure which is discussed in great detail in the previous sections **??** and **??**.

Thus, the hyperparameter choice of a_τ, b_τ, a_ξ and b_ξ inevitably impacts the result of *all* coefficient estimates contained in the model in a nontrivial way, such that pure analytical reasoning might be misleading. For this reason, the following paragraphs investigate these effects based on a simulation approach.

Section 1.5 is dedicated to the impact of a_τ, b_τ, a_ξ and b_ξ on the Accuracy of the Posterior Mean estimates for $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, whereas section 1.6 examines their effect on the Ridge Penalty that is induced by the Inverse-Gamma distributions, that these quantities parameterize.

Simulation Setting

- The design matrix $\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 \end{pmatrix}$ is simulated from a two dimensional normal distribution $\mathcal{N}_2\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$ with mean vector $\boldsymbol{\mu} = \begin{pmatrix} 1 & 2 \end{pmatrix}^T$ and identity covariance matrix $\boldsymbol{\Sigma} = \mathbf{I}_2$. The same holds true for the design matrix $\mathbf{Z} = \begin{pmatrix} \mathbf{z}_1 & \mathbf{z}_2 \end{pmatrix}$ with mean vector $\boldsymbol{\mu} = \begin{pmatrix} 5 & 3 \end{pmatrix}^T$ and identity covariance matrix.
- In both design matrices intercept columns are added for estimation purposes. The true coefficient vectors are given by $\boldsymbol{\beta} = \begin{pmatrix} \beta_0 & \beta_1 & \beta_2 \end{pmatrix}^T = \begin{pmatrix} 0 & -1 & 4 \end{pmatrix}^T$ and $\boldsymbol{\gamma} = \begin{pmatrix} \gamma_0 & \gamma_1 & \gamma_2 \end{pmatrix}^T = \begin{pmatrix} 0 & -2 & 1 \end{pmatrix}^T$.
- For sampling the location parameter, the full conditional multivariate normal distribution of $\boldsymbol{\beta}$ is chosen, i.e. `mcmc_ridge(..., mh_location = FALSE)` is used. Therefore, the location estimate is directly affected by the hyperparameters.
- For simulating the influence of the hyperparameters, nine different values are chosen: $a_\tau, b_\tau, a_\xi, b_\xi \in \{-1, 0, 0.5, 1, 2, 10, 50, 100, 200\}$. Since for statistical properties like the mean of an Inverse Gamma distribution $\frac{b}{a-1}$ the condition $a > 1$ is required, particular attention is given to larger values. However, it is an aim to inspect the performance of the sampler for smaller hyperparameter values than 1 as well.

Simulation Results

The upper panel of Figure 5 displays the absolute deviations of the Posterior Mean estimates from the true parameters with the stated different values for a_τ and b_τ. For each estimate, the Posterior Mean averages over 1000 simulations of the `mcmc_ridge()` sampler. Note, that location and scale parameters are plotted separately, according to the relationship mentioned above. For a better overview, the dotted line displays the linear trend of all estimate deviations.

The x - axis is transformed by a pseudo logarithm in order to clearly visualize the deviations in the range of -1 to 10, which would not be possible on original scales. Since -1 and 0 are also part of the hyperparameter values, the `pseudo_log_trans()` function of the `scales` package is applied, log-transforming positive values only.

It can be observed, that the intercept estimates in each plot show the largest deviations from their true value. In the left panel of Figure 5, however, the overall deviations of $\boldsymbol{\beta}$ estimates from their corresponding true value are small in absolute value. In contrast, deviations of the $\boldsymbol{\gamma}$ estimates in the right panel are fairly significant, especially for γ_0.

The functional chain that applies to the estimates of $\boldsymbol{\beta}$ can be described by the effect of the mean of the inverse gamma distribution on τ^2: A larger value for b_τ leads to larger values of τ^2, which are again affecting the full posterior parameters of $\boldsymbol{\beta}$ and, thus, potentially increase the absolute deviation of the corresponding estimates from their true values. a_τ causes the opposite effect. This numerically observable effect, however, is hided by the overall small deviation in the upper left plot of Figure 5.

It is remarkable, that the deviation of $\boldsymbol{\beta}$ estimates is smallest when $a_\tau, b_\tau \in \{50, 100\}$. For values of $a_\tau \leq 1$, one obtains wider variances of absolute deviations, since the Posterior Mean requires values larger than one.

In the upper right plots of Figure 5, there is no clear impact of τ^2 and its parameters. Rooted in no direct effect of τ^2 on $\boldsymbol{\gamma}$ according to our underlying mathematical model, one observes cross-effects through the sampling procedure of the `mcmc_ridge()` sampler, where the full posterior $f(\boldsymbol{\gamma} \mid \cdot)$ depends on $\boldsymbol{\beta}$.

Anyway, our sampler produces the lowest deviation of $\boldsymbol{\gamma}$ estimates for $a_\tau, b_\tau \in \{0, 0.5, 200\}$, where 0.5 is chosen by coincidence for a_τ here, since wide variations for $a_\tau \leq 1$ of absolute deviations are observable again.

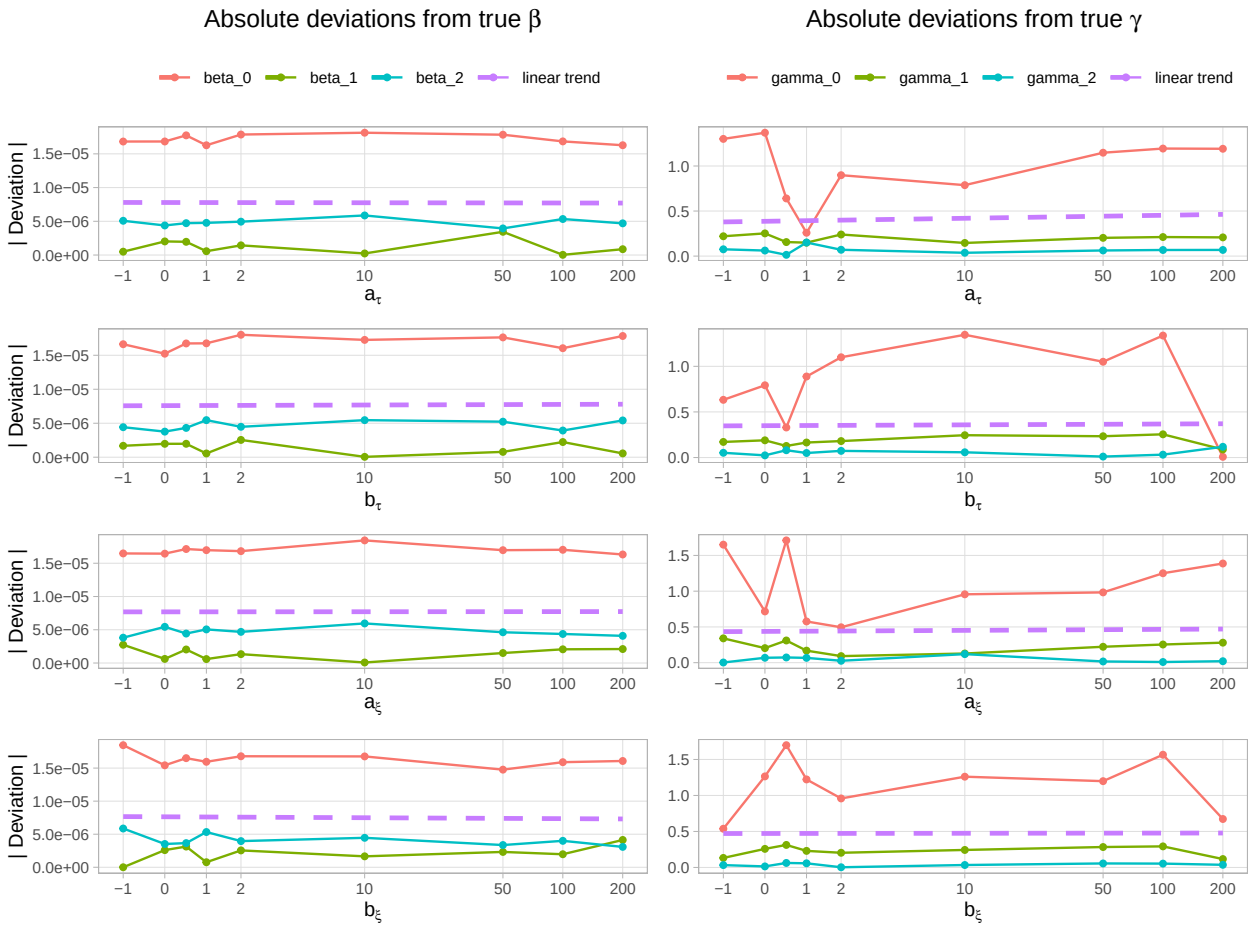

Figure 5: Absolute deviations from true $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$

The lower panel of Figure 5 is constructed analogously, but showing the impact of a_ξ and b_ξ on the location and scale parameters respectively. Again, the overall absolute deviations for the $\boldsymbol{\beta}$ estimates from their true values are small, whereas the deviations for the $\boldsymbol{\gamma}$ estimates are considerably larger. Once again, the intercept estimates display the largest deviations from their true value.

Arguing with the mean of the Inverse Gamma distribution of ξ^2 in a similar way, one obtains larger mean values for b_ξ, while a_ξ lowers them. The impact of ξ^2 on $\boldsymbol{\gamma}$ is assumed to decrease $f(\boldsymbol{\gamma} \mid \cdot)$ according to our underlying theoretical model. This effect is indicated by the linear trend lines in the lower right part of Figure 5.

In general, one obtains smaller deviations for larger values of b_ξ and lower ones of a_ξ, where especially lots of randomness occurs in the deviations of γ_0. Therefore, the impact of a_ξ and b_ξ on the scale intercepts is overshadowed by the randomness induced by the Metropolis Hastings algorithm. The same wide variations exclusively for $a_\xi \leq 1$ cannot be obtained in the same manner as for a_τ.

The sampler exhibits the best results for the *scale* estimates for $a_\xi = 2$ and $b_\xi = 100$. However, due to the wide overall variation, these results must be taken with care.

The effect of a_ξ and b_ξ on $\boldsymbol{\beta}$ can be explained through the cross-effects of the matrix $\mathbf{W}$ and the vector $\mathbf{u}$ introduced at the beginning of this section, both containing $\boldsymbol{\gamma}$. These diminish with increasing values of the $\boldsymbol{\gamma}$ entries.

The matrix $\mathbf{W}$ affects the variance of the full conditional distribution of $\boldsymbol{\beta}$ negatively, while the mean is positively affected. Hence, larger values of a_ξ cause higher Posterior Means of the location parameters. The

positive linear trend in the lower left part of Figure 5 for values of a_ξ is particularly interesting. For values of b_ξ, the trend comes off inferior. The wider variations of deviations for $a_\xi \leq 1$ is again not observable here. Nonetheless, the randomness observable for scale estimates does not show up for location estimates anymore.

The smallest deviations of the *location* estimates can be detected for $a_\xi = 1$ and $b_\xi = 200$.

Shortly noted, the acceptance rates of the Metropolis Hastings algorithm for sampling $\boldsymbol{\gamma}$ are always between 0.31 and 0.53. For the value range of a_τ, b_τ and a_ξ, no distinct pattern is observable in this regard. With growing values of b_ξ, however, acceptance rates are more likely to grow. Since the acceptance rates are in reasonable ranges enabling statistically valid estimation, these results are not further investigated here.

1.6 Hyperparameters: Impact on Ridge Penalty

As introduced in section 1.5, we now continue to relate the values of a_τ, b_τ, a_ξ and b_ξ to the observed Penalty on the coefficients estimates for $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, where we exclude β_0 and γ_0 from the analysis since the intercept terms are not penalized in the Bayesian Ridge Regression model.

Simulation Setting

Many characteristics of the Inverse-Gamma distribution, such as the Mean, Median and Variance, depend in some way on the *ratio* of its two parameters. Thus, we built grids of value pairs for a_τ and b_τ for the prior distribution of $\boldsymbol{\beta}$ and pairs of a_ξ and b_ξ for the prior distribution of $\boldsymbol{\gamma}$.

As noted before, the values of each pair do not only impact the immediate parameter one step above in the hierarchical model, but all other quantities as well due to cross effects in the Full Conditional distributions. For this reason, we evaluate the impact of each hyperparameter pair on both coefficient vectors $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$. As usual, more details about the simulation setting are given in the Appendix.

Simulation Results

Figure 6 shows 4 Heatmaps. On the left, values of $\|\boldsymbol{\beta}\|^2$ are displayed depending on combinations of a_τ and b_τ (top panel) and a_ξ and b_ξ (bottom panel). Analogous observations for $\|\boldsymbol{\gamma}\|^2$ are shown on the right side.

In both cases, the euclidean norms are calculated without the intercept terms β_0 and γ_0 to isolate the Penalty Effect. Larger values (lower Penalty) are indicated by dark red, whereas the color blue is chosen for lower values (larger Penalty). For comparison, the unpenalized euclidean norms based on the true data generating coefficient values are given by $\|\boldsymbol{\beta}\|^2 = 68$ and $\|\boldsymbol{\gamma}\|^2 = 18$.

Starting with the top panel, there are almost no pattern observable. Interestingly, some combinations of a_τ and b_τ that are located in direct neighbourhood in the grid lead to large and small regularization effects at the same time.

However, we have to keep the color scale in mind. Drastically different colors are deceiving in this case, since all euclidean norms are almost equal, indicating that the hyperparameters of τ^2 have very little impact overall on the Ridge Penalty, independent of their size. This finding is somewhat surprising, since e.g. the theoretical Posterior Mean of τ^2 differs wildly throughout the chosen combinations, suggesting a heavily varying prior variance of $\boldsymbol{\beta}$ and therefore different *potential* for larger values in absolute value.

In contrast, the bottom panel shows a huge impact of a_ξ and b_ξ on the regularization. While the left plot does not look incredibly informative at first sight, there is a pattern observable, that is common across both plots: If a_ξ is small and/or b_ξ is large, larger Norms (and thus a lower penalty effect) of the coefficient vectors tend to occur. Only if a_ξ is large *and* b_ξ is small at the same time, there might be a significant Penalty Effect.

While the pattern of this reduction is not super clear in cae of $\boldsymbol{\beta}$ and could potentially be due to chance, the values of $\|\boldsymbol{\gamma}\|^2$ are consistenly lower for $a_\xi \geq 64$ and $b_\xi \leq 4$. This combination leads to low Posterior Means for ξ^2, which again induces a lower / more informative prior variance τ^2.

The results from this simulation study are difficult to connect to their hypothesized immediate effect based on the underlying mathematical model and in some cases even counterintuitive. This clearly illustrates, that

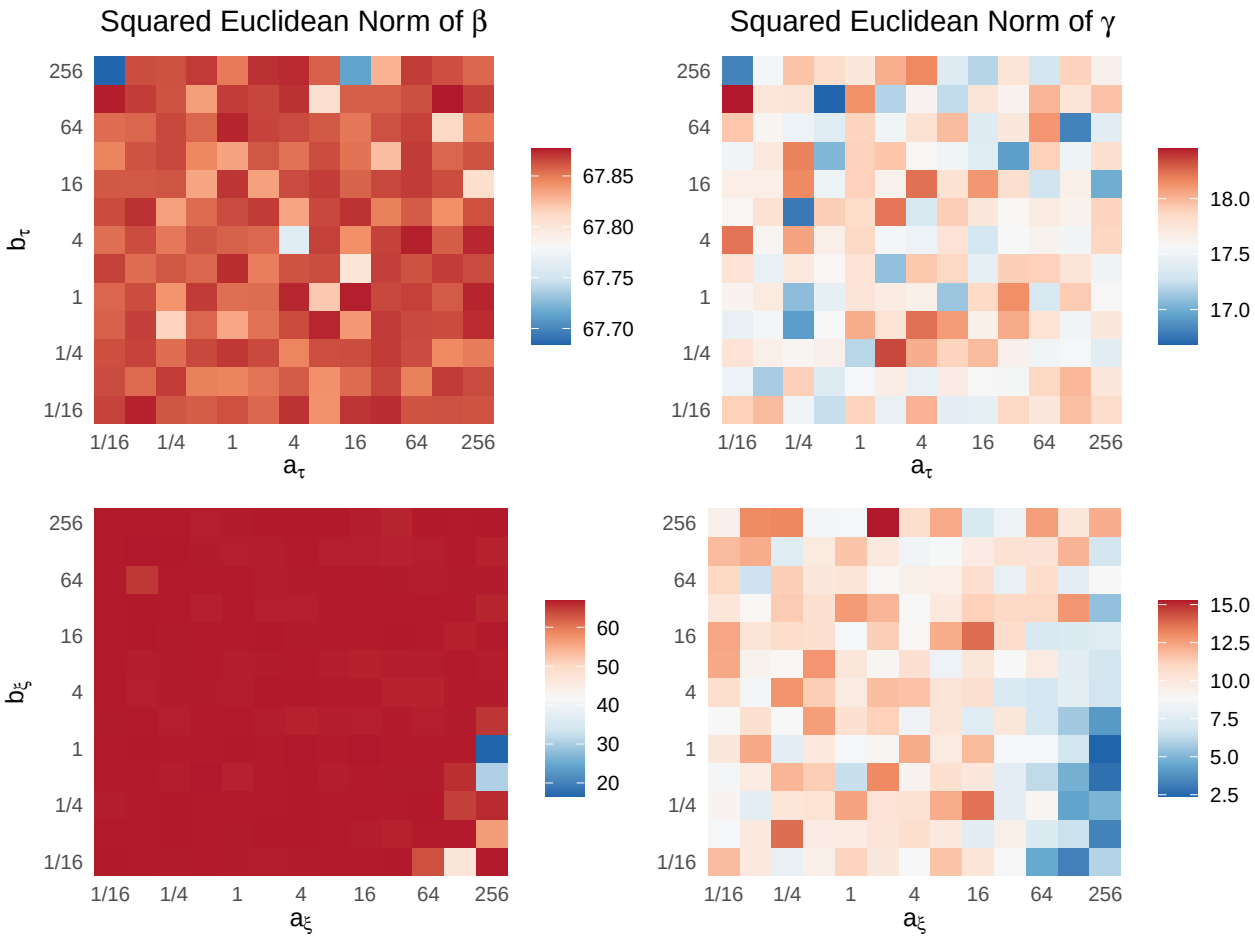

Figure 6: Heatmaps of $\|\boldsymbol{\beta}\|^2$ and $\|\boldsymbol{\gamma}\|^2$ (excluding Intercepts)

tweaking parameters at one end in a Bayesian model of moderate size might not affect the model as a whole as might be intended. Indirect effects through the cross connections in the full Conditional Distributions might dominate the simple relations through the prior distributions. This again emphasizes the value of simulation studies to validate or possibly change the statistical methodology in practice.