Project Report

Bayesian Effect Fusion for Categorical Predictors

A Reproduction of the Method Proposed by Daniela Pauger and Helga Wagner

Date: April 20, 2025

Team 4

Team Members:

Yash Bihany (221216), Rythm Kumar (220925),
Subham Anand (221093), Devansh Gupta (220345)

Abstract

This report presents a reproduction of the paper *Bayesian Effect Fusion for Categorical Predictors* (Pauger and Wagner, 2019). We aim to obtain a sparse representation of categorical predictors in regression models through Bayesian variable selection and effect fusion. Theoretical derivations are provided, and the methodology is demonstrated via simulation studies as described in the original paper.

Contents

1 Introduction

In regression modeling, categorical predictors are commonly encoded using dummy variables, with one level set as the baseline. This conventional approach results in a group of regression effects for each categorical variable, leading to a high-dimensional parameter space. Interpretation becomes particularly challenging when some levels are associated with sparse data, as the corresponding effects are estimated with high uncertainty.

To address the issues of high dimensionality and interpretability, various sparsity-inducing methods have been proposed. Frequentist approaches such as the lasso (Tibshirani, 1996) and the elastic net (Zou and Hastie, 2005) use penalty terms to select relevant predictors, while Bayesian methods employ priors like shrinkage priors (Park and Casella, 2008; Griffin and Brown, 2010) or spike-and-slab priors (Mitchell and Beauchamp, 1988; George and McCulloch, 1997). However, these methods typically focus on selecting individual non-zero effects and do not account for the natural grouping structure of categorical predictors.

Advancements such as the group lasso (Yuan and Lin, 2006), Bayesian group lasso (Raman et al., 2009; Kyung et al., 2010), and sparse group lasso (Simon et al., 2013) have extended sparsity to groups of coefficients, but still do not address the potential for fusing levels with similar effects. For metric predictors, effect fusion has been explored through the fused lasso (Tibshirani et al., 2005) and its Bayesian variants, but these approaches rely on an ordering of levels and are not directly applicable to nominal predictors. Some recent works, including those by Gertheiss and Tutz (Gertheiss and Tutz, 2009; Tutz and Gertheiss, 2016), have begun to address effect fusion for categorical variables, but a comprehensive Bayesian approach remained limited.

Building on these developments, Pauger and Wagner proposed a Bayesian effect fusion method that encourages both variable selection and the fusion of similar level effects for categorical predictors. In this project, we reproduce and extend their main findings, focusing on both nominal and ordinal categorical variables. We provide a detailed theoretical overview, implement the methodology in R, and conduct simulation studies following the setup of Gertheiss and Tutz to evaluate the effectiveness of the approach. Our results are compared with those reported in the original paper to validate our implementation and highlight the practical utility of Bayesian effect fusion in achieving sparse and interpretable regression models.

2 Literature Review

2.1 Model Specification

Following Pauger and Wagner, we consider a linear regression model with a normally distributed response y and p categorical predictors. Each predictor h has $c_h + 1$ levels, with level 0 as the baseline. Dummy variables $X_{h,k}$ represent the effect of level k, and the model is specified as:

$$y = \mu + \sum_{h=1}^{p} \sum_{k=1}^{c_h} X_{h,k}\beta_{h,k} + \varepsilon, \tag{1}$$

where μ is the intercept, $\beta_{h,k}$ is the effect relative to the baseline, and $\varepsilon \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$ is the error.

In matrix form, for response vector $\mathbf{y} = (y_1, \ldots, y_n)'$, the model becomes:

$$\mathbf{y} = \mathbf{1}\mu + \sum_{h=1}^{p} \mathbf{X}_h \boldsymbol{\beta}_h + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}), \tag{2}$$

where $\mathbf{X}_h$ is the design matrix for predictor h, $\boldsymbol{\beta}_h$ the corresponding coefficient vector, and $\mathbf{1}$ and $\mathbf{I}$ denote a vector of ones and the identity matrix, respectively.

2.2 Prior Specification for Effect Fusion

The prior for the model parameters is specified as:

$$p(\mu, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_p, \sigma^2) = p(\sigma^2)p(\mu) \prod_{h=1}^{p} p(\boldsymbol{\beta}_h \mid \tau_h^2, \boldsymbol{\delta}_h) p(\tau_h^2) p(\boldsymbol{\delta}_h), \tag{3}$$

where μ has a normal prior $\mu \sim \mathcal{N}(0, M_0)$, and σ^2 has an inverse Gamma prior $p(\sigma^2) \sim \mathcal{G}^{-1}(s_0, S_0)$. We use the improper prior $p(\sigma^2) \propto 1/\sigma^2$ in practice.

The regression coefficients $\boldsymbol{\beta}_h$ are assigned hierarchical priors, with each $\boldsymbol{\beta}_h$ conditional on τ_h^2 and $\boldsymbol{\delta}_h$:

$$\boldsymbol{\beta}_h \mid \tau_h^2, \boldsymbol{\delta}_h \sim \mathcal{N}(0, \mathbf{B}_{h0}(\boldsymbol{\delta}_h, \tau_h^2)), \tag{4}$$

where $\mathbf{B}_{h0}(\boldsymbol{\delta}_h, \tau_h^2) = \gamma_h \tau_h^2 \mathbf{Q}_h^{-1}(\boldsymbol{\delta}_h)$, and $\tau_h^2 \sim \mathcal{G}^{-1}(g_{h0}, G_{h0})$.

The structure of $\mathbf{Q}_h(\boldsymbol{\delta}_h)$ is controlled by binary indicators $\delta_{h,kj}$, which determine whether effects $\beta_{h,k}$ and $\beta_{h,j}$ are considered distinct or similar. This allows flexible fusion of categories, including shrinkage of level effects. The structure of $\mathbf{Q}_h$ will be discussed in the following sections.

For nominal covariates, all pairs of effects can be fused, while for ordinal covariates, fusion is typically restricted to adjacent levels.

2.2.1 Prior on Non-Restricted Categorical Covariates

To perform unrestricted effect fusion for a categorical covariate with levels $0, \ldots, c$, we introduce a binary indicator δ_{ij} for each pair of effects $0 \leq i < j \leq c$. Thus, the vector $\boldsymbol{\delta}$ subsuming all these indicators is of dimension $d \times 1$ where $d = \binom{c+1}{2}$.

We specify the structure matrix $\mathbf{Q}(\boldsymbol{\delta})$ as

$$\mathbf{Q}(\boldsymbol{\delta}) = \begin{pmatrix} \sum_{j \neq 1} \kappa_{1j} & -\kappa_{12} & \cdots & -\kappa_{1c} \\ -\kappa_{21} & \sum_{j \neq 2} \kappa_{2j} & \cdots & -\kappa_{2c} \\ \vdots & \vdots & \ddots & \vdots \\ -\kappa_{c1} & -\kappa_{c2} & \cdots & \sum_{j \neq c} \kappa_{cj} \end{pmatrix} \tag{6}$$

with diagonal elements q_{kk} given as

$$q_{kk} = \sum_{j \neq k} \kappa_{kj} = \kappa_{k0} + \cdots + \kappa_{k,k-1} + \kappa_{k,k+1} + \cdots + \kappa_{kc}, \quad k = 1, \ldots, c.$$

For $k > j$, κ_{kj} is defined as

$$\kappa_{kj} = \delta_{kj} + r(1 - \delta_{kj}),$$

and $\kappa_{jk} = \kappa_{kj}$ for $j > k$. The value of δ_{kj} determines whether $\kappa_{kj} = 1$ (for $\delta_{kj} = 1$) or $\kappa_{kj} = r$ (for $\delta_{kj} = 0$). r is a fixed large number, which we call precision ratio for reasons explained below. Finally, we set $\gamma = c/2$.

We discuss this specification now in more detail. The structure matrix $\mathbf{Q}(\boldsymbol{\delta})$ determines the prior precision matrix $\mathbf{B}_0^{-1}(\boldsymbol{\delta}, \tau^2)$ up to the scale factor $1/(\gamma\tau^2)$.

The diagonal elements q_{kk} determine the prior partial precisions and the off-diagonal elements q_{kj} the prior partial correlations of the level effects:

$$\mathrm{Cor}(\beta_j, \beta_k | \boldsymbol{\beta}_{-(j,k)}) = \frac{-q_{jk}}{\sqrt{q_{jj} q_{kk}}} = \frac{\kappa_{jk}}{\sqrt{\left(\sum_{l \neq j} \kappa_{jl}\right)\left(\sum_{m \neq k} \kappa_{km}\right)}} \tag{8}$$

$$\mathrm{Prec}(\beta_k | \boldsymbol{\beta}_{-k}) = \frac{q_{kk}}{\gamma\tau^2}. \tag{9}$$

Thus, depending on the value of the binary indicator δ_{kj}, the prior allows for high (if $\delta_{kj} = 0$) or low (if $\delta_{kj} = 1$) positive prior partial correlation of β_j and β_k.

The prior partial precision $\mathrm{Prec}(\beta_k \mid \beta_{\setminus k})$ depends on the binary indicators associated with level k, and can take up to c different values. Let $\vec{\delta}_k = (\delta_{k0}, \ldots, \delta_{k,k-1}, \delta_{k+1,k}, \ldots, \delta_{c,k})$ collect these indicators. When all elements of $\vec{\delta}_k$ are equal to 1, the precision reaches its minimum value of $\frac{c}{\gamma\tau^2}$; when all are 0, the precision reaches its maximum, given by $\frac{r \cdot c}{\gamma\tau^2}$. Hence, the ratio r directly determines the spread between the largest and smallest possible prior partial precisions.

If we choose $\gamma = \frac{c}{2}$, the prior partial precision varies within the interval $\left[\frac{2}{\tau^2}, \frac{2r}{\tau^2}\right]$, which is independent of the number of levels c.

2.2.2 Prior on Restricted Categorical Covariates

To apply *hard fusion restrictions*, we use a vector of fixed indicators, denoted by $\boldsymbol{\zeta}$. Each element ζ_{kj} corresponds to the difference between two effects θ_{kj}. The value of ζ_{kj} tells us whether this effect difference should be allowed to fuse:

- $\zeta_{kj} = 1$: fusion is allowed,
- $\zeta_{kj} = 0$: fusion is not allowed.

Unlike the unrestricted case, we now define a *stochastic indicator* δ_{kj} only when $\zeta_{kj} = 1$. Hence, the dimension of the $\boldsymbol{\delta}$ vector equals the number of pairs (k, j) where fusion is allowed.

We define the structure matrix $Q(\boldsymbol{\zeta}, \boldsymbol{\delta})$ as follows:

- For off-diagonal elements:

$$q_{kj} = \begin{cases} -\kappa_{kj}, & \text{if } \zeta_{kj} = 1 \\ 0, & \text{if } \zeta_{kj} = 0 \end{cases} \quad \text{and} \quad q_{jk} = q_{kj}.$$

- For diagonal elements:

$$q_{kk} = \begin{cases} \kappa_{k0} - \sum_{j \neq k} q_{kj}, & \text{if } \zeta_{k0} = 1 \\ -\sum_{j \neq k} q_{kj}, & \text{if } \zeta_{k0} = 0 \end{cases}$$

Ordinal Covariate A useful special case is when the categorical covariate is *ordinal.* In this case, fusion is only allowed between adjacent categories, so we define:

$$\zeta_{kj} = \begin{cases} 1, & \text{if } j = k-1 \\ 0, & \text{otherwise} \end{cases}$$

Here, the $\boldsymbol{\delta}$ vector has c elements: $\delta = (\delta_{10}, \delta_{21}, \ldots, \delta_{c,c-1})$, and the matrix $Q(\boldsymbol{\zeta}, \boldsymbol{\delta})$ becomes tri-diagonal:

$$Q(\boldsymbol{\zeta}, \boldsymbol{\delta}) = \begin{pmatrix} \kappa_{10} + \kappa_{21} & -\kappa_{21} & 0 & \cdots & 0 \\ -\kappa_{21} & \kappa_{21} + \kappa_{32} & -\kappa_{32} & \cdots & 0 \\ 0 & -\kappa_{32} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \kappa_{c-1,c-2} + \kappa_{c,c-1} & -\kappa_{c,c-1} \\ 0 & 0 & \cdots & -\kappa_{c,c-1} & \kappa_{c,c-1} \end{pmatrix}$$

In this case, the maximum value of a diagonal entry is $q_{kk} = 2r$, so we set $\gamma = 1$.

2.2.3 Prior for Indicator Variables

In variable selection, each indicator δ_{kj} is often assumed to be conditionally independent, with a prior probability

$$p(\delta_{kj} = 1) = \omega,$$

where ω is either a fixed constant or assigned a Beta hyperprior, $\omega \sim \text{Beta}(v_0, w_0)$.

For effect fusion, we adopt a more convenient prior:

$$p(\delta) \propto |Q(\delta)|^{-1/2} \cdot r^{\sum(1-\delta_{kj})/2} \tag{11}$$

This cancels out the determinant in the joint prior of β and δ, resulting in:

$$p(\beta, \delta \mid \tau^2) = \left(\frac{1}{\gamma\tau^2}\right)^{c/2} \exp\left(-\frac{\beta^\top Q(\delta)\beta}{2\gamma\tau^2}\right) \cdot (\sqrt{r})^{\sum(1-\delta_{kj})} \tag{12}$$

Factorization This prior has the attractive property of factorization:

$$p(\beta, \delta \mid \tau^2) \propto \prod_{k>j} (\sqrt{r})^{1-\delta_{kj}} \cdot \exp\left(-\frac{(\beta_k - \beta_j)^2}{2\tau^2\gamma\left[\delta_{kj} + r(1-\delta_{kj})\right]}\right) \tag{13}$$

Which simplifies to:

$$\prod_{k>j} p(\theta_{kj}, \delta_{kj} \mid \tau^2) \tag{14}$$

So, given τ^2, the effect differences θ_{kj} and indicators δ_{kj} are independent across all pairs $0 \leq j < k \leq c$.

Adaptive Shrinkage Given an effect difference θ_{kj}, the conditional prior for δ_{kj} is:

$$p(\delta_{kj} = 1 \mid \theta_{kj}, \tau^2) = \frac{p(\theta_{kj} \mid \mathcal{N}(0, \gamma\tau^2))}{p(\theta_{kj} \mid \mathcal{N}(0, \gamma\tau^2)) + p(\theta_{kj} \mid \mathcal{N}(0, \gamma\tau^2/r))} \tag{15}$$

Special Case: Ordinal Covariates For ordinal covariates, fusion is restricted to adjacent levels. Then, $\delta = (\delta_{10}, \delta_{21}, \ldots, \delta_{c,c-1})$ and the matrix $Q(\delta)$ is tri-diagonal.

In this case, the prior becomes uniform:

$$p(\delta) \propto 1$$

Special Case: Nominal Covariates For nominal covariates with unrestricted effect fusion, $\delta_{kj} \in \{0, 1\}$ for all $0 \leq j < k \leq c$. The prior $p(\delta)$ in Equation (11) favors sparse models.

Though $Q(\delta)$ does not have a closed-form determinant in general, we can compare:

- Full model: $\delta = 1 \Rightarrow Q(1)$
- Null model: $\delta = 0 \Rightarrow Q(0) = rQ(1)$

Then:

$$|Q(0)| = |rQ(1)| = r^d |Q(1)|, \quad \text{where } d = \frac{c(c-1)}{2}$$

So, the prior ratio becomes:

$$\frac{p(\delta = 0)}{p(\delta = 1)} = \frac{|Q(1)|^{1/2}}{|Q(0)|^{1/2}} \cdot r^{d/2} = \frac{|Q(1)|^{1/2}}{(r^d |Q(1)|)^{1/2}} \cdot r^{d/2} = r^{-d/2} \cdot r^{d/2} = r^{\frac{c(c-1)}{2}}$$

Thus, as the number of categories c or the precision ratio r increases, the prior increasingly favors the null model.

2.3 Posterior Inference

We aim to perform **Bayesian inference** for the parameters in a **linear regression model**:

$$y = X\beta + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2 I)$$

Here,

- y is the response vector,
- X is the design matrix (including a column of 1s for the intercept and other covariates),
- β is the vector of regression coefficients,
- ε is the error term with normal distribution and variance σ^2.

We place a joint prior on the parameters:

$$p(\beta, \delta, \tau^2, \sigma^2) = p(\beta \mid \delta, \tau^2) \cdot p(\delta) \cdot p(\tau^2) \cdot p(\sigma^2)$$

The regression coefficient vector is written as $\beta = (\mu, \beta_1^\top, \ldots, \beta_p^\top)^\top$, which includes the intercept μ and coefficient groups β_h for each covariate $h = 1, \ldots, p$. The binary indicator vector is $\delta = (\delta_1^\top, \ldots, \delta_p^\top)^\top$, and the scale parameter vector is $\tau^2 = (\tau_1^2, \ldots, \tau_p^2)^\top$.

We assume a multivariate normal prior on β, i.e.,

$$\beta \mid \delta, \tau^2 \sim \mathcal{N}(0, B_0(\delta, \tau^2))$$

where $B_0(\delta, \tau^2)$ is a block-diagonal covariance matrix. The first block corresponds to the intercept and is denoted by M_0, while each subsequent block $B_{h0}(\delta_h, \tau_h^2)$ for $h = 1, 2, \ldots, p$ corresponds to the group-specific regression coefficients.

Even with a partially improper prior on the intercept and error variance, given by

$$p(\mu, \sigma^2) \propto \frac{1}{\sigma^2},$$

the resulting posterior distribution is proper under conditions established in a theorem by **Sun et al. (2001)**.

2.4 MCMC Scheme

To perform posterior inference, we use a Markov Chain Monte Carlo (MCMC) algorithm. We start the sampler by choosing initial values for:

- The error variance: σ^2
- The binary fusion indicators: δ
- The scale parameters: τ^2

Using these, we compute the prior covariance matrix for the regression coefficients $B_0(\delta, \tau^2)$. This covariance matrix is block-diagonal, with one block M_0 for the intercept and blocks

$$B_{h0}(\delta_h, \tau_h^2) = \gamma_h \tau_h^2 Q(\delta_h)^{-1}$$

for each categorical covariate h.

Step 1: Sampling regression coefficients β

We sample β from its full conditional distribution:

$$p(\beta|\sigma^2, \delta, \tau^2, y) \propto p(y|\beta, \sigma^2) \cdot p(\beta|\delta, \tau^2)$$

Both the likelihood and prior are Normal, so the posterior is also multivariate Normal:

$$\beta \sim \mathcal{N}(b, B)$$

where

$$B^{-1} = B_0(\delta, \tau^2)^{-1} + \frac{1}{\sigma^2} X^\top X$$

$$b = \frac{1}{\sigma^2} B X^\top y$$

Step 2: Sampling the error variance σ^2

The full conditional is:

$$p(\sigma^2|\beta,\delta,\tau^2,y) \propto p(y|\beta,\sigma^2)\cdot p(\sigma^2)$$

With a conjugate Inverse Gamma prior, the posterior is:

$$\sigma^2 \sim \text{IG}(s,S)$$

where

$$s = s_0 + \frac{n}{2}, \quad S = S_0 + \frac{1}{2}(y-X\beta)^\top(y-X\beta)$$

Step 3: Sampling the scale parameters τ_h^2

For each covariate $h = 1,\ldots,p$, we update τ_h^2. The prior is:

$$p(\tau_h^2) \propto (\tau_h^2)^{-g_{h0}-1}\exp\left(-\frac{G_{h0}}{\tau_h^2}\right)$$

And the prior on $\beta_h|\delta_h,\tau_h^2$ is:

$$p(\beta_h|\delta_h,\tau_h^2) \propto (\tau_h^2)^{-c_h/2}\exp\left(-\frac{\beta_h^\top Q_h(\delta_h)\beta_h}{2\gamma_h\tau_h^2}\right)$$

The full conditional is:

$$\tau_h^2 \sim \text{IG}(g_h,G_h)$$

with

$$g_h = g_{h0} + \frac{c_h}{2}, \quad G_h = G_{h0} + \frac{\gamma_h}{2}\beta_h^\top Q_h(\delta_h)\beta_h$$

Step 4: Sampling the hyperparameter G_h

Assume:

$$G_{h0} \sim \text{Exp}(\lambda_h)$$

Then

$$p(G_h|\tau_h^2) \propto p(G_h)p(\tau_h^2|G_h) \propto \exp(-G_h\lambda_h)\cdot G_h^{g_h}\exp(-G_h\tau_h^2)$$

This is a Gamma distribution:

$$G_h \sim \text{Gamma}\left(g_h+1, \frac{1}{\lambda_h}+\frac{1}{\tau_h^2}\right)$$

Step 5: Sampling the fusion indicators δ_h

The conditional posterior is:

$$p(\delta_h|\beta_h,\tau_h^2,\sigma^2,y) \propto p(\beta_h|\delta_h,\tau_h^2)\cdot p(\delta_h)$$

From previous derivations:

$$p(\beta_h, \delta_h|\tau_h^2) \propto \prod_{j<k} r^{1-\delta_{h,kj}} \exp\left(-\frac{(\beta_{h,k} - \beta_{h,j})^2}{2\tau_h^2\gamma_h(\delta_{h,kj} + r(1 - \delta_{h,kj}))}\right)$$

This leads to a Bernoulli form:

$$p(\delta_{h,kj} = 1|\theta_{kj}, \tau_h^2) = \frac{1}{1 + L_{h,kj}}, \quad \text{where } \theta_{kj} = \beta_{h,k} - \beta_{h,j}$$

$$L_{h,kj} = r \cdot \exp\left(-\frac{r-1}{2\gamma_h\tau_h^2}(\beta_{h,k} - \beta_{h,j})^2\right)$$

Step 6: Updating the prior covariance matrix

After sampling δ_h and τ_h^2, update:

$$B_{h0}(\delta_h, \tau_h^2) = \gamma_h\tau_h^2 Q_h(\delta_h)^{-1}$$

Then reconstruct $B_0(\delta, \tau^2)$ as a block-diagonal matrix using the updated blocks.

2.5 Model Selection

The goal is to select a final model (e.g. for making predictions), then in a Bayesian decision-theoretic framework, we need to choose an appropriate **loss function**.

One useful loss function for effect fusion is a special case of **Binder's loss** also used by Lau and Green (2007) in Bayesian model-based clustering. This loss function works by evaluating **pairs of items** and penalizing incorrect groupings:

- If two items *should be in different groups*, but are assigned to the *same group*, that is an error.
- If two items *should be in the same group*, but are assigned to *different groups*, that is also an error.

In effect fusion, this means wrongly classifying whether the effect difference between two categories is *non-zero* (they should be separate) or *zero* (they should be fused).

Binder's Loss Function

Binder's loss is defined as:

$$L(z, z^*) = \sum_{j\neq k} \ell_1 \cdot \mathbb{I}\{z_k = z_j\} \cdot \mathbb{I}\{z_k^* \neq z_j^*\} + \ell_2 \cdot \mathbb{I}\{z_k \neq z_j\} \cdot \mathbb{I}\{z_k^* = z_j^*\}$$

where:

- z is the true clustering,
- z^* is the proposed clustering,
- ℓ_1, ℓ_2 are the misclassification penalties.

If $\ell_1 = \ell_2$, the expected posterior loss becomes:

$$\mathbb{E}[L(z, z^*) \mid y] = \sum_{j \neq k} \left| \mathbb{I}\{z_k^* = z_j^*\} - \pi_{kj} \right|$$

where $\pi_{kj} = \mathbb{P}(z_k = z_j \mid y)$ is the (k, j)-th element of the **posterior similarity matrix**.

The **Bayes optimal clustering** minimizes the following quantity:

$$\sum_{j \neq k} \mathbb{I}\{z_k^* = z_j^*\} \left(\frac{1}{2} - \pi_{kj} \right)$$

Lau and Green (2007) propose solving this minimization using **integer programming**, which is implemented in the R function `minbinder` from the `mcclust` package.

Posterior Similarity Matrix

To select the best effect fusion model for each covariate C_h, we compute the posterior similarity matrix $\pi_{h,kj}$ using M MCMC draws (after burn-in) as follows:

$$\hat{\pi}_{h,kj} = \frac{1}{M} \sum_{m=1}^{M} \delta_{h,kj}^{(m)}$$

Finally, we refit the selected model using dummy-coded regression coefficients under a flat Normal prior:

$$\mathcal{N}(0, IB_0)$$

3 Simulation Study

3.1 Simulation Design

We conducted a simulation study to evaluate the performance of different estimation methods under a controlled setting. Specifically, we generated 100 synthetic datasets, each consisting of $n = 500$ observations from the Gaussian linear regression model

$$y = \mu + X\beta + \varepsilon,$$

where $\mu = 1$, $\varepsilon \sim \mathcal{N}(0, 1)$, and X is a fixed design matrix constructed from dummy-coded categorical predictors.

The design matrix X is based on eight categorical covariates: four treated as ordinal (C_1–C_4) and four as nominal (C_5–C_8). For each type, two covariates have eight categories and two have four categories. Predictor levels were generated independently from the following discrete distributions:

- For 8-level variables (C_1, C_2, C_5, C_6):

$$\pi_8 = (0.1, 0.1, 0.2, 0.05, 0.2, 0.1, 0.2, 0.05)$$

- For 4-level variables (C_3, C_4, C_7, C_8):

$$\pi_4 = (0.1, 0.4, 0.2, 0.3)$$

The categorical variables were transformed into dummy variables, omitting the first level of each factor. This results in a design matrix with 40 covariate columns. The true regression coefficients β are assigned as follows:

$$\begin{aligned}
\text{Ordinal:}\quad & \beta_1 = (0, 1, 1, 2, 2, 4, 4),\\
& \beta_2 = (0, 0, 0, 0, 0, 0, 0),\\
& \beta_3 = (0, -2, -2),\\
& \beta_4 = (0, 0, 0),\\
\\
\text{Nominal:}\quad & \beta_5 = (0, 1, 1, 1, 1, -2, -2),\\
& \beta_6 = (0, 0, 0, 0, 0, 0, 0),\\
& \beta_7 = (0, 2, 2),\\
& \beta_8 = (0, 0, 0).
\end{aligned}$$

The full coefficient vector β is constructed accordingly by mapping each dummy-coded level to its corresponding true value.

3.2 Benchmarking and Competing Models

We compare our proposed Bayesian effect fusion method against six widely used alternative approaches for handling multi-level categorical predictors. All methods are applied to the same simulated datasets, with tuning performed via cross-validation or default settings in each R package.

1. **Full Model (Full)**
   Implemented using the `effectFusion` package with fusion turned off. This baseline fits a separate dummy variable for every category level, without any regularization.

2. **Penalty Model (Penalty)**
   Based on the fused-lasso ideas of Gertheiss and Tutz (Gertheiss and Tutz, 2009) and implemented via `glmnet`, this method adds an ℓ_1 penalty on all pairwise differences of level coefficients. The penalty encourages adjacent or similar levels to fuse, and its strength is chosen by ten-fold cross-validation.

3. **Bayesian Lasso (BLasso)**
   Using the `monomvn` package, independent Laplace (double-exponential) priors are placed on each coefficient. This prior induces sparsity by shrinking small effects to zero, with posterior means used as final estimates.

4. **Bayesian Elastic Net (BEN)**
   Implemented via `EBglmnet`, this approach employs a hierarchical Normal–Exponential prior that combines ℓ_1 and ℓ_2 penalties. It balances both shrinkage and grouping, allowing coefficients to be set exactly zero or to move in correlated blocks.

5. **Group Lasso (GLasso)**
   Through the `grpreg` package, a group-wise ℓ_2 penalty is applied to all dummy variables belonging to the same covariate. This encourages entire predictors to be selected or dropped as a unit, with the tuning parameter selected by cross-validation.

6. **True Model (True)**
   To serve as an oracle benchmark, we first collapse levels according to the known true coefficient pattern before fitting. For example, for a covariate with true effects $(0, 1, 1, 2, 2, 4, 4)$, levels 1–2 are merged (effect 0), 3–4 merged (effect 1), 5–6 merged (effect 2), and 7–8 merged (effect 4). We then fit this pre-fused data again with `effectFusion` (fusion disabled), extract the posterior mean for each fused group, and finally replicate each group's estimate back to all original dummy variables that were merged.

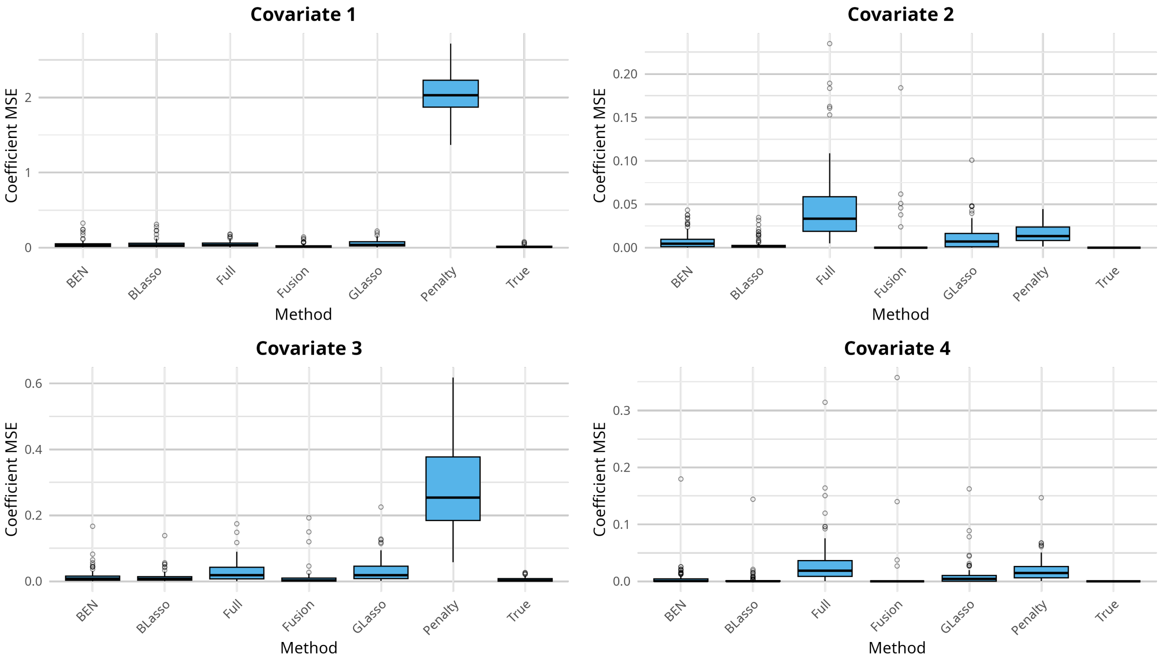
3.3 Simulation Results

From the 100 simulated datasets, we obtained coefficient estimates (betas) for each covariate across all models. For each of the 8 covariates, we computed the mean squared error (MSE) of the estimated coefficients and visualized these in the plots below. The MSE of the estimated coefficients is calculated as:

$$\text{MSE}_h^{(i)} = \frac{1}{c_h} \sum_{k=1}^{c_h} \left(\hat{\beta}_{h,k}^{(i)} - \beta_{h,k} \right)^2$$

The Mean Squared Error $\text{MSE}_h^{(i)}$ quantifies the average squared difference between the estimated and true effects for the h-th covariate under the i-th method. Here, c_h is the number of levels for covariate h, $\hat{\beta}_{h,k}^{(i)}$ is the estimated effect for level k, and $\beta_{h,k}$ is the corresponding true effect. The formula averages these squared differences over all levels to assess estimation accuracy—lower MSE indicates better performance.

The plots for covariates C_1 through C_8 can be seen below:



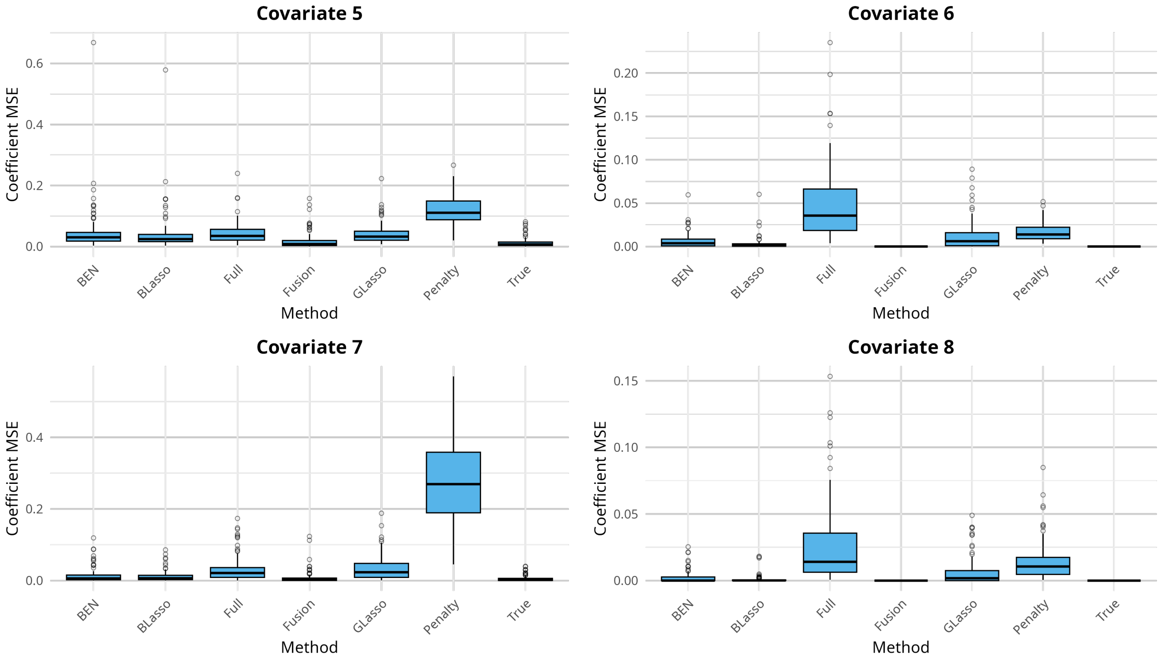


Figure 1: MSE of estimated coefficients for covariates C_1 to C_8 across different models.

We generated an additional testing dataset with 500 samples to evaluate prediction performance. Using the beta values obtained from the simulations, we predicted outcomes on this test dataset and calculated the prediction MSE. Predictions for these new observations are made using the estimated coefficients $\hat{\beta}^{(i)}$ from each of the 100 original datasets. For the i-th dataset, the predicted value for observation j is:

$$\hat{z}_j^{(i)} = \tilde{x}_j \hat{\beta}^{(i)}, \quad i = 1, \ldots, 100$$

The mean squared prediction error (MSPE) for the i-th dataset is calculated as:

$$\text{MSPE}^{(i)} = \frac{1}{n^*} \sum_{j=1}^{n^*} \left(z_j - \hat{z}_j^{(i)} \right)^2, \quad i = 1, \ldots, 100$$

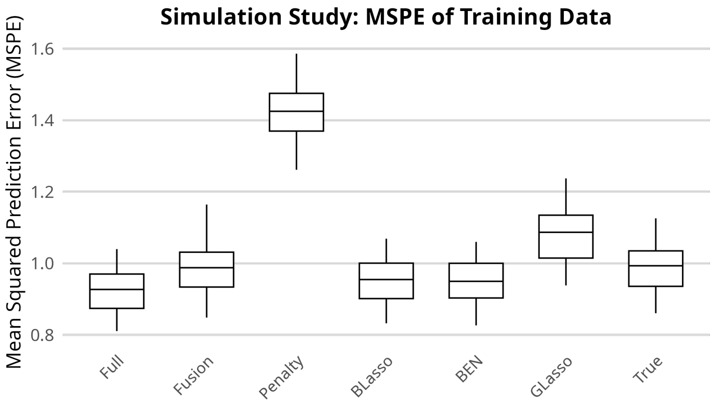


Figure 2: Boxplots of training prediction MSE across models.

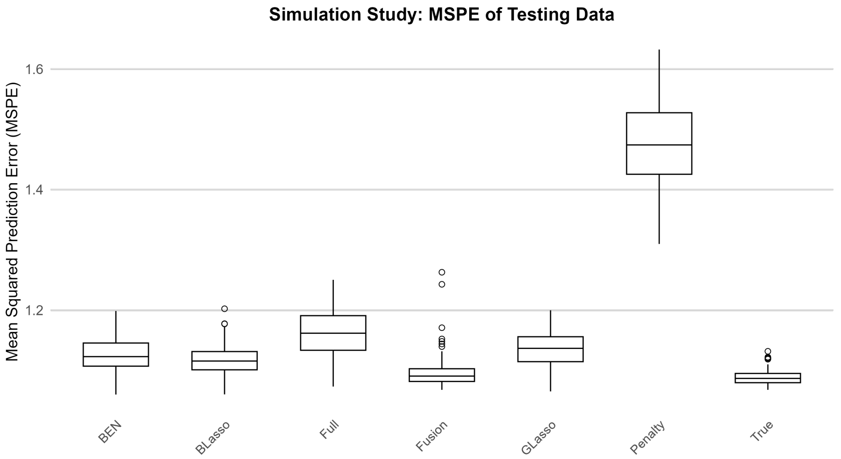


Figure 3: Boxplots of testing prediction MSE across models.

Covariate	Method	TPR	TNR	PPV	NPV
1	BEN	99.9	16.8	87.8	95.7
	BLasso	100.0	0.0	85.7	–
	Full	100.0	0.0	85.7	–
	Fusion	100.0	96.5	99.4	100.0
	GLasso	100.0	0.0	85.7	–
	Penalty	100.0	0.0	85.7	–
2	BEN	–	58.1	0.0	100.0
	BLasso	–	0.0	0.0	–
	Full	–	0.2	0.0	100.0
	Fusion	–	98.2	0.0	100.0
	GLasso	–	13.0	0.0	100.0
	Penalty	–	0.0	0.0	–
3	BEN	100.0	38.0	76.8	100.0
	BLasso	100.0	0.0	66.7	–
	Full	100.0	0.5	66.8	100.0
	Fusion	100.0	98.0	99.2	100.0
	GLasso	100.0	0.0	66.7	–
	Penalty	100.0	0.0	66.7	–
4	BEN	–	63.0	0.0	100.0
	BLasso	–	0.0	0.0	–
	Full	–	0.0	0.0	–
	Fusion	–	97.8	0.0	100.0
	GLasso	–	22.0	0.0	100.0
	Penalty	–	0.0	0.0	–

Table 1: Performance for Covariates 1–4

Covariate	Method	TPR	TNR	PPV	NPV
5	BEN	100.0	8.5	73.2	98.6
	BLasso	100.0	0.0	71.4	–
	Full	100.0	0.1	71.5	100.0
	Fusion	99.8	98.9	99.6	99.6
	GLasso	100.0	0.0	71.4	–
	Penalty	100.0	0.0	71.4	–
6	BEN	–	60.2	0.0	100.0
	BLasso	–	0.0	0.0	–
	Full	–	0.2	0.0	100.0
	Fusion	–	100.0	–	100.0
	GLasso	–	12.0	0.0	100.0
	Penalty	–	0.0	0.0	–
7	BEN	100.0	40.5	77.5	100.0
	BLasso	100.0	0.0	66.7	–
	Full	100.0	0.5	66.8	100.0
	Fusion	100.0	98.5	99.4	100.0
	GLasso	100.0	0.0	66.7	–
	Penalty	100.0	0.0	66.7	–
8	BEN	–	69.5	0.0	100.0
	BLasso	–	0.2	0.0	100.0
	Full	–	1.0	0.0	100.0
	Fusion	–	100.0	–	100.0
	GLasso	–	27.0	0.0	100.0
	Penalty	–	0.0	0.0	–

Table 2: Performance for Covariates 5–8

The results for the covariates are summarized in Table 1 and Table 2. As seen, the Bayesian effect fusion method consistently achieves high TNR and NPV values across all covariates, with TNR exceeding 96% in most cases and perfect NPV (100%) where defined. This indicates strong performance in correctly identifying zero-effect differences. In contrast, the other methods (BLasso, GLasso, Penalty, Full, BEN) tend to produce TNR values close to zero for most covariates, leading to numerous false positives.

Fusion also achieves very high TPR and PPV values (often close to or above 99%) when these metrics are defined, reflecting its strength in correctly detecting non-zero effect differences. However, this high specificity comes with a slight trade-off: for a few covariates such as Covariate 5, the TPR is marginally below 100% (e.g. 99.8%).

For covariates with all true effects equal (e.g., Covariates 2, 4, 6, 8), TPR and PPV are not defined, but the TNR and NPV results again show Fusion's dominance with values near or at 100%, while other methods fail to distinguish zero-effect differences.

In summary, Bayesian effect fusion outperforms all other methods in terms of both specificity (TNR, NPV) and precision (PPV), with only a minimal reduction in sensitivity (TPR) in a few cases.

4 Real-World Data Study

We apply the **EffectFusion** model to the Airline Dataset on Kaggle. The dataset contains Airlines, Departure city, Arrival city, Departure time, Arrival time, Number of stops, Class, Duration, Days left and Flight prices as columns. The target variable is Flight prices which is a continuous random variable. We combine Departure and Arrival cities to form 30 unique values and drop the Departure and Arrival times to avoid multicollinearity. We also bin the Duration and Days left into bin of 5. Finally, we end up with 6 columns as the independent categorical predictors.

- Airlines (6 levels) = nominal
- Journey (30 levels) = nominal
- Number of stops (3 levels) = ordinal
- Class (2 levels) = ordinal
- Duration (10 levels) = ordinal
- Days left (10 levels) = ordinal

4.1 Results

The results are summarised in Table 3 and Table 4.

Feature	Estimate	Lower Bound	Upper Bound
(Intercept)	-0.679	-0.685	-0.673
stops.one	0.220	0.212	0.227
stops.two_or_more	0.323	0.314	0.333
class.Business	2.006	2.004	2.008
duration_bin.[5,10)	0.159	0.152	0.166
duration_bin.[10,15)	0.159	0.152	0.166
duration_bin.[15,20)	0.159	0.152	0.166
duration_bin.[20,25)	0.159	0.152	0.166
duration_bin.[25,30)	0.159	0.152	0.166
duration_bin.[30,35)	0.261	0.246	0.275
duration_bin.[35,40)	0.261	0.246	0.275
duration_bin.[40,45)	0.261	0.246	0.275
duration_bin.[45,50]	0.261	0.246	0.275
days_left_bin.[5,10)	-0.163	-0.169	-0.158
days_left_bin.[10,15)	-0.163	-0.169	-0.158
days_left_bin.[15,20)	-0.331	-0.336	-0.325
days_left_bin.[20,25)	-0.331	-0.336	-0.325
days_left_bin.[25,30)	-0.331	-0.336	-0.325
days_left_bin.[30,35)	-0.331	-0.336	-0.325
days_left_bin.[35,40)	-0.331	-0.336	-0.325
days_left_bin.[40,45)	-0.331	-0.336	-0.325
days_left_bin.[45,50]	-0.331	-0.336	-0.325

Table 3: Summary of the fusion fit model

Feature	Estimate	Lower Bound	Upper Bound
airline.AirAsia	0.000	0.000	0.000
airline.GO_FIRST	0.000	0.000	0.000
airline.Indigo	0.000	0.000	0.000
airline.SpiceJet	0.000	0.000	0.000
airline.Vistara	0.000	0.000	0.000
journey.Bangalore-Delhi	0.000	0.000	0.000
journey.Bangalore-Hyderabad	0.000	0.000	0.000
journey.Bangalore-Kolkata	0.000	0.000	0.000
journey.Bangalore-Mumbai	0.000	0.000	0.000
journey.Chennai-Bangalore	0.000	0.000	0.000
journey.Chennai-Delhi	0.000	0.000	0.000
journey.Chennai-Hyderabad	0.000	0.000	0.000
journey.Chennai-Kolkata	0.000	0.000	0.000
journey.Chennai-Mumbai	0.000	0.000	0.000
journey.Delhi-Bangalore	0.000	0.000	0.000
journey.Delhi-Chennai	0.000	0.000	0.000
journey.Delhi-Hyderabad	0.000	0.000	0.000
journey.Delhi-Kolkata	0.000	0.000	0.000
journey.Delhi-Mumbai	0.000	0.000	0.000
journey.Hyderabad-Bangalore	0.000	0.000	0.000
journey.Hyderabad-Chennai	0.000	0.000	0.000
journey.Hyderabad-Delhi	0.000	0.000	0.000
journey.Hyderabad-Kolkata	0.000	0.000	0.000
journey.Hyderabad-Mumbai	0.000	0.000	0.000
journey.Kolkata-Bangalore	0.000	0.000	0.000
journey.Kolkata-Chennai	0.000	0.000	0.000
journey.Kolkata-Delhi	0.000	0.000	0.000
journey.Kolkata-Hyderabad	0.000	0.000	0.000
journey.Kolkata-Mumbai	0.000	0.000	0.000
journey.Mumbai-Bangalore	0.000	0.000	0.000
journey.Mumbai-Chennai	0.000	0.000	0.000
journey.Mumbai-Delhi	0.000	0.000	0.000
journey.Mumbai-Hyderabad	0.000	0.000	0.000
journey.Mumbai-Kolkata	0.000	0.000	0.000

Table 4: Summary of the fusion fit model (all levels fused to baseline, resulting in zero estimates).

4.2 Observations

On observing the Table 3, we see that for the **Number of stops** predictor, all levels are significant. Hence, there is no fusion. So is the case for the **Class** predictor. However, for **Duration** we can merge all the bins from $[5 - 30)$ and $[30 - 50]$. Hence, we end up with three levels after fusion, $[0, 5), [5 - 30)$, and $[30 - 50]$. In case of Days left, we can fuse levels between $[5 - 15)$ and $[15 - 50]$. Hence, just like before, we end up with three levels after fusion, $[0, 5), [5 - 15)$, and $[15 - 50]$. The remaining estimates are 0 as shown in Table 4, denoting that the levels have no significant effect on the target variable and hence, we can drop columns **Journey and Airlines**, reducing the number of predictors. It is interesting to note that there is no difference in the price pattern among all the different airlines since, all the levels in **Airline** merge into the base level. Also, we speculate that there is no effect of **Journey** because the information may be incorporated in the **Duration** variable.

5 Conclusion

In conclusion, the paper introduces a Bayesian effect fusion method that enables sparse modeling of categorical covariates by both excluding irrelevant predictors and fusing levels with similar effects. The approach employs a mixture of Normal priors with structured precision matrices to encourage effect fusion, and allows for the incorporation of prior information—particularly useful for ordinal covariates. Posterior inference is conducted using MCMC, and the final model is selected by minimizing Binder's loss based on estimated fusion probabilities.

In reference to Figure 3 and 4, our effectFusion model achieves the smallest testing error, indicating better generalization to unseen data. In contrast, the full model minimizes training error, likely because it includes all parameters and thus fits the training data more closely, potentially at the cost of overfitting.

Our results show that the behavior of the method in our experiments is largely in agreement with the findings reported in the original paper. Specifically, we observe that the method tends to exclude non-informative predictors and identifies meaningful level fusions. The performance in terms of model selection, estimation, and prediction is broadly consistent with the published results, supporting the practical effectiveness of Bayesian effect fusion in a variety of regression scenarios.

6 Contribution

The selection of the research paper was a collaborative effort, with all group members coming together to discuss and make the decision collectively. Everyone was actively involved in understanding the theoretical foundations, including priors for nominal and ordinal covariates, as well as the use of indicator variables for both types of covariates.

Together, we worked through the process of drawing posterior inference from linear regression models involving categorical covariates, with a key focus on the MCMC algorithm. This understanding allowed us to explore model selection techniques and the fusion of covariate levels effectively.

Theoretical concepts were divided equally among the four members, and we ensured mutual support so that everyone clearly understood the topics they were responsible for.

For the simulation study, we generated synthetic data and applied the MCMC algorithm to perform inference. We also examined the publicly available code of the effectFusion package, thoroughly understood its methodology, and then implemented the core ideas on real-world data—specifically the airlines dataset.

Rythm, Yash, and Subham focused on the simulation work, while Devansh led the analysis on the real dataset. Subham and Yash created the visualizations and plots, while Rythm and Devansh worked on generating the summary tables. Coding responsibilities were divided accordingly, ensuring that each member contributed meaningfully to their part.

In report, there was equal contribution by everyone in making it. Everyone contributed to writing some section of the Report.

Overall, while specific tasks were distributed for efficiency, the project was a true team effort, with all members contributing equally to both the understanding and implementation phases.

References

George, E. I. and McCulloch, R. E. (1997). Approaches for bayesian variable selection. *Statistica Sinica*, 7(2):339–373.

Gertheiss, J. and Tutz, G. (2009). Penalized regression with ordinal predictors. *International Statistical Review*, 77(3):345–365.

Griffin, J. E. and Brown, P. J. (2010). Inference with normal-gamma prior distributions in regression problems. *Bayesian Analysis*, 5(1):171–188.

Kyung, M., Gill, J., Ghosh, M., and Casella, G. (2010). Penalized regression, standard errors, and bayesian lasso. *Bayesian Analysis*, 5(2):369–411.

Mitchell, T. J. and Beauchamp, J. J. (1988). Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032.

Park, T. and Casella, G. (2008). The bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686.

Pauger, D. and Wagner, H. (2019). Bayesian effect fusion for categorical predictors.

Raman, K. et al. (2009). Bayesian group lasso for nonparametric varying-coefficient models. *Biometrics*, 65(3):738–745.

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2013). A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B*, 67(1):91–108.

Tutz, G. and Gertheiss, J. (2016). Regularization methods for categorical predictors in regression models. *Statistical Modelling*, 16(3):161–200.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, 68(1):49–67.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, 67(2):301–320.