Uniwersytet Wrocławski
Wydział Matematyki i Informatyki
Instytut Matematyczny
specjalność: Analiza danych

Joanna Pokora

Logistic Adaptive Bayesian SLOPE

Praca magisterska
napisana pod kierunkiem
dr. Michała Burdukiewicza
oraz prof. dr hab. Małgorzaty Bogdan

Wrocław 2026

Contents

1 Introduction

Regression models form a broad family of supervised learning methods. They have wide applications in many fields, including medical studies, where they can be used to quantify the association of biological features with the investigated outcome [1][2]. Although basic regression models are simple and powerful tools that allow the estimation of relationship strength and outcome prediction, they also have substantial limitations, especially relevant in modern experiments.

In the biological field, many studies face analytically demanding datasets, particularly involving:

- high dimensionality - the number of features is usually much larger than the number of observations,
- strong correlations between the covariates,
- sparse signals - the majority of the features do not influence the response variable,
- missing values.

These constraints can make standard regression models insufficient or even impossible to apply. Adaptive Bayesian SLOPE (ABSLOPE) [3] addresses these challenges by combining the sorted L_1 penalized estimation (SLOPE) [4] with Bayesian spike-and-slab modeling. It aims to exclude unimportant covariates from the model and simultaneously provide reliable estimation of signal strengths. As many biological datasets contain missing values, ABSLOPE also incorporates adaptive imputation during model fitting.

2 Aim

The aim of this thesis is to extend the existing Adaptive Bayesian SLOPE model, developed for Gaussian regression, to the logistic case. Although the Gaussian model is suitable for problems with continuous response variables, classification problems involving categorical outcomes are equally important and often encountered in biomedical analyses [5][6]. Logistic regression provides a way to model binary classification, broadening the range of possible ABSLOPE applications.

To enable binary modeling using the ABSLOPE estimator, we describe a logistic version of the model and provide its implementation. The original package, developed in Rcpp, was rewritten as a standalone C++ implementation and is available both as an R and C++ package. In addition, the previously used SLOPE solver, alternating direction method of multipliers (ADMM), was replaced with the more computationally efficient hybrid coordinate descent algorithm, implemented in the SLOPE library described in [7].

3 Regression analysis

Regression analysis is a statistical approach focused on modeling the relationship between a response (outcome) variable associated with the phenomenon under study and measured covariates, called explanatory variables or predictors. Many model formulations are possible, depending on the assumptions about the form of this relationship, the properties of the response vector and data complexity. In this section, we first introduce regression models, focusing on linear and logistic regression, then move to regularization methods and Bayesian approaches, including spike-and-slab models.

3.1 Gaussian and logistic regression

Generalized linear models (GLMs) form a class of regression models used to model the relationship between the expected value of a response variable and a set of predictors.

Definition 3.1. Given a sample consisting of n observations $\{y_i, x_{i,1}, \ldots, x_{i,p}\}_{i=1}^n$, let $y_1, \ldots, y_n$ be the realizations of the independent random outcome variables $Y_1, \ldots, Y_n$. We assume that all Y_i belong to the same exponential family, with probability density or mass function given by

$$p(y_i; \theta_i, \phi) = h(y_i, \phi) e^{\frac{\theta_i y_i - A(\theta_i)}{\phi}},$$

where θ_i and $\phi > 0$ are parameters, while $h(\cdot, \cdot)$ and $A(\cdot)$ symbolize known functions and A has a continuous second derivative.

For each $i \in \{1, \ldots, n\}$, a generalized linear model defines the relationship between the predictors $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,p})^T$ and the conditional expected value $\mu_i = \mathbb{E}[Y_i \mid \mathbf{x}_i, \boldsymbol{\beta}]$ as

$$g(\mu_i) = \eta_i = \beta_0 + x_{i,1}\beta_1 + \cdots + x_{i,p}\beta_p,$$

where $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_p)^T$ is a vector of model coefficients, η_i is called a linear predictor and $g(\cdot)$ represents a link function.

Two fundamental regression methods are Gaussian and logistic models. Since they are crucial for this thesis, we introduce them explicitly. First, we define the normal distribution, which underlies Gaussian regression.

Definition 3.2. The random variable Y follows a normal (Gaussian) distribution with mean $\mathbb{E}[Y] = \mu$ and variance $\mathrm{Var}(Y) = \sigma^2$, where $\mu \in \mathbb{R}$ and $\sigma^2 > 0$, if its probability density function takes the form

$$p(y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}.$$

We write $Y \sim N(\mu, \sigma^2)$.

To formulate the Gaussian model in matrix form, we need to extend the univariate normal distribution to the multivariate setting. This requires introducing the covariance matrix, which is a parameter of the multivariate Gaussian distribution.

Definition 3.3. The covariance of random variables X and Y is defined as

$$\begin{aligned}\mathrm{Cov}(X,Y) &= \mathbb{E}[(X-\mathbb{E}[X])(Y-\mathbb{E}[Y])] \\ &= \mathbb{E}[XY]-\mathbb{E}[X]\mathbb{E}[Y].\end{aligned}$$

Using Definitions 3.2 and 3.3, we can now extend the Gaussian distribution to the multivariate case.

Definition 3.4. The random vector $\mathbf{Y} = (Y_1, \ldots, Y_n)^T$ follows a multivariate normal (Gaussian) distribution $N_n(\boldsymbol{\mu}, \Sigma)$, where $\boldsymbol{\mu} \in \mathbb{R}^n$ denotes the mean vector and $\Sigma \in \mathbb{R}^{n\times n}$ is the symmetric and positive-definite covariance matrix, if its joint probability density function is given by

$$p(\mathbf{y}) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{y}-\boldsymbol{\mu})\right\}.$$

Here, $\boldsymbol{\mu} = (\mathbb{E}[Y_1], \ldots, \mathbb{E}[Y_n])^T = \mathbb{E}[\mathbf{Y}]$ and $\Sigma = \mathrm{Cov}(\mathbf{Y}, \mathbf{Y})$.

Gaussian regression is a special case of a generalized linear model under the assumption that the response vector, conditionally on the measured features and model parameters, follows a multivariate Gaussian distribution.

Definition 3.5. Let $\mathbf{Y} = (Y_1, \ldots, Y_n)^T$ be the response (outcome) random vector and let $\mathbb{X} = (\mathbf{1}, \mathbf{X}_1, \ldots, \mathbf{X}_p)$ be the design matrix where $\mathbf{1}$ denotes a vector of ones. Gaussian regression uses the identity link function, leading to the model formulated as

$$\mathbb{E}[\mathbf{Y}|\mathbb{X}, \boldsymbol{\beta}] = \mathbb{X}\boldsymbol{\beta}$$

and assumes

$$\mathbf{Y} \mid \mathbb{X}, \boldsymbol{\beta}, \sigma^2 \sim N_n(\mathbb{X}\boldsymbol{\beta}, \sigma^2\mathbb{I}_n),$$

where σ^2 denotes the variance parameter and $\mathbb{I}_n$ is the identity matrix of size $n \times n$.

Equivalently, Gaussian regression can be expressed as

$$\mathbf{Y} = \mathbb{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{\varepsilon} \mid \sigma^2 \sim N_n(\mathbf{0}, \sigma^2\mathbb{I}_n)$ is a noise vector, with $\mathbf{0}$ symbolizing a vector of zeros. Indeed, this representation leads to

$$\mathbb{E}\left[\mathbf{Y} \mid \mathbb{X}, \boldsymbol{\beta}\right] = \mathbb{E}\left[\mathbb{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \mid \mathbb{X}, \boldsymbol{\beta}\right] = \mathbb{X}\boldsymbol{\beta} + \mathbb{E}[\boldsymbol{\varepsilon}] = \mathbb{X}\boldsymbol{\beta},$$

$$\mathrm{Cov}(\mathbf{Y} \mid \mathbb{X}, \boldsymbol{\beta}, \sigma^2) = \mathrm{Cov}(\mathbb{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \mid \mathbb{X}, \boldsymbol{\beta}, \sigma^2) = \mathrm{Cov}\left(\boldsymbol{\varepsilon} \mid \sigma^2\right) = \sigma^2\mathbb{I}_n.$$

The next model is used for binary outcomes.

Definition 3.6. The random variable $Y \in \{0, 1\}$ follows a Bernoulli distribution with parameter $0 \leq \pi \leq 1$ if

$$P(Y = 1) = \pi = 1 - P(Y = 0).$$

Thus, the probability mass function of the Bernoulli distribution can be written as

$$p(y) = \pi^y(1-\pi)^{1-y}.$$

Then $\mathbb{E}[Y] = \pi$ and $\mathrm{Var}(Y) = \pi(1-\pi)$.

With a binary response, modeling its expected value as a linear function of the measured covariates can lead to values outside the interval $[0, 1]$. Instead, we use a generalized linear model with the logit link function.

Definition 3.7. The function $\text{logit} : (0, 1) \to \mathbb{R}$ is defined as

$$\text{logit}(\pi) = \log\left(\frac{\pi}{1-\pi}\right).$$

Then, the logistic model is defined as follows.

Definition 3.8. Let Y_i, $i \in \{1, \ldots, n\}$, be conditionally independent random variables following Bernoulli distributions $Y_i \mid \mathbf{x}_i, \boldsymbol{\beta} \sim \text{Bernoulli}(\pi_i)$. Logistic regression models the relationship between $\pi_i = \mathbb{E}[Y_i \mid \mathbf{x}_i, \boldsymbol{\beta}]$ and the linear predictor η_i as

$$\text{logit}(\pi_i) = \log\left(\frac{\pi_i}{1-\pi_i}\right) = \eta_i.$$

Hence, the probabilities are given by

$$\pi_i = \frac{1}{1 + e^{-\eta_i}}.$$

In GLMs, the conditional distribution of the response variable given the predictors depends on the coefficient vector $\boldsymbol{\beta}$. We now turn to the estimation of $\boldsymbol{\beta}$, which is one of the fundamental problems in regression analysis.

3.2 Maximum likelihood estimator

To find the estimate of the coefficient vector $\boldsymbol{\beta}$, we can use the maximum likelihood method. First, we define the likelihood function.

Definition 3.9. Let $y_1, \ldots, y_n$ be the realizations of independent random variables $Y_1, \ldots, Y_n$ with probability density or mass functions $p(y_i, \boldsymbol{\omega})$, where $\boldsymbol{\omega} \in \Omega$ denotes a vector of parameters and Ω is the parameter space. The function $L : \Omega \to \mathbb{R}$, defined as

$$L(\boldsymbol{\omega}) = \prod_{i=1}^{n} p(y_i, \boldsymbol{\omega}),$$

is called the likelihood function.

Suppose that in the above definition $\boldsymbol{\omega} = (\boldsymbol{\beta}, \boldsymbol{\varphi})$ denotes a vector of the model parameters, including $\boldsymbol{\beta}$ and any additional model characteristics represented by $\boldsymbol{\varphi}$. Given the realizations $\mathbf{y} = (y_1, \ldots, y_n)^T$ of the outcome vector $\mathbf{Y}$, let $p(y_i, \boldsymbol{\omega})$ be the probability density function of a component Y_i, $i \in \{1, \ldots, n\}$. The idea is to estimate the parameters by maximizing the likelihood function.

Definition 3.10. The maximum likelihood estimator (MLE) of the regression model parameters, given the values of the outcome variable and predictors, is defined as

$$\hat{\boldsymbol{\omega}} = \arg\max_{\boldsymbol{\omega} \in \Omega} L(\boldsymbol{\omega}).$$

Since the logarithm is a strictly increasing function, maximizing the likelihood is equivalent to maximizing the log-likelihood, denoted by $l(\boldsymbol{\omega})$. Therefore, the estimator given in Definition 3.10 can also be written as

$$\hat{\boldsymbol{\omega}} = \arg\max_{\boldsymbol{\omega}\in\Omega} \log L(\boldsymbol{\omega}) = \arg\max_{\boldsymbol{\omega}\in\Omega} l(\boldsymbol{\omega}).$$

The maximum likelihood method allows for the estimation of all model parameters. In particular, in the case of linear regression, we estimate both the coefficient vector $\boldsymbol{\beta}$ and the variance of errors. To concentrate solely on the estimation of the coefficient vector, we can treat σ^2 as fixed.

Let $\mathbf{x}_i = (x_{i0}, x_{i1}, \ldots, x_{ip})^T$ be the i-th row of the design matrix, $i \in \{1, \ldots, n\}$, where $x_{i0} = 1$ corresponds to the intercept. Then, the MLE of $\boldsymbol{\beta}$ in the Gaussian model can be obtained as

$$\begin{aligned}
\hat{\boldsymbol{\beta}}_{MLE} &= \arg\max_{\boldsymbol{\beta}\in\mathbb{R}^{p+1}} \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2}{2\sigma^2}\right\} \\
&= \arg\max_{\boldsymbol{\beta}\in\mathbb{R}^{p+1}} \sum_{i=1}^{n} \left(-\frac{1}{2}\log\left(2\pi\sigma^2\right) - \frac{(y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2}{2\sigma^2}\right) \\
&= \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^{p+1}} \|\mathbf{y} - \mathbb{X}\boldsymbol{\beta}\|_2^2.
\end{aligned}$$

Thus for each $j \in \{0, \ldots, p\}$, we have

$$\frac{\partial}{\partial\beta_j} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2 = -2\sum_{i=1}^{n} (y_i - \mathbf{x}_i^T\boldsymbol{\beta}) x_{ij} = -2\mathbf{X}_{\cdot j}^T(\mathbf{y} - \mathbb{X}\boldsymbol{\beta}),$$

where $\mathbf{X}_{\cdot j}$ is the j-th column of the design matrix, the gradient of the loss function is given by

$$\nabla_\beta \|\mathbf{y} - \mathbb{X}\boldsymbol{\beta}\|_2^2 = -2\mathbb{X}^T(\mathbf{y} - \mathbb{X}\boldsymbol{\beta}).$$

Hence, the Hessian matrix takes the form

$$\nabla_\beta^2 \|\mathbf{y} - \mathbb{X}\boldsymbol{\beta}\|_2^2 = 2\mathbb{X}^T\mathbb{X}.$$

To introduce the constraints on the design matrix $\mathbb{X}$, we first define the rank of a matrix.

Definition 3.11. The rank of a matrix $\mathbb{X} \in \mathbb{R}^{n\times p+1}$ is the maximum number of linearly independent columns of $\mathbb{X}$ (or equivalently, the maximum number of its linearly independent rows). It satisfies

$$\operatorname{rank}(\mathbb{X}) \leq \min\{n, p+1\}.$$

The matrix has full rank if

$$\operatorname{rank}(\mathbb{X}) = \min\{n, p+1\}.$$

In particular, when $\operatorname{rank}(\mathbb{X}) = p+1$, the matrix $\mathbb{X}$ is said to have full column rank, and if $\operatorname{rank}(\mathbb{X}) = n$, full row rank.

Assuming that the columns of the design matrix are linearly independent, or equivalently that the matrix $\mathbb{X}$ has full column rank, the Hessian $2\mathbb{X}^T\mathbb{X}$ is positive definite, since

$$\forall \mathbf{z} \in \mathbb{R}^{p+1} \setminus \{0\} \qquad \mathbf{z}^T\mathbb{X}^T\mathbb{X}\mathbf{z} = (\mathbb{X}\mathbf{z})^T(\mathbb{X}\mathbf{z}) = \|\mathbb{X}\mathbf{z}\|_2^2 > 0.$$

Thus, the maximum likelihood estimator, that is, the coefficient vector minimizing the loss function, satisfies

$$-2\mathbb{X}^T(\mathbf{y} - \mathbb{X}\hat{\boldsymbol{\beta}}_{MLE}) = 0 \implies \hat{\boldsymbol{\beta}}_{MLE} = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbf{y}.$$

The full column rank assumption on $\mathbb{X}$ implies $p+1 \leq n$. Hence, $\hat{\boldsymbol{\beta}}_{MLE}$ cannot be used in the high-dimensional data setup, where the number of predictors usually exceeds the number of observations. Moreover, when $p+1 = n$, a model becomes saturated with $\boldsymbol{\beta}$ estimated as

$$\hat{\boldsymbol{\beta}}_{MLE} = \mathbb{X}^{-1}(\mathbb{X}^T)^{-1}\mathbb{X}^T\mathbf{y} = \mathbb{X}^{-1}\mathbf{y}.$$

Thus, $\hat{\mathbf{y}} = \mathbb{X}\hat{\boldsymbol{\beta}}_{MLE} = \mathbf{y}$ and the model does not provide any new information.

Remark 1. Because the maximum likelihood estimator of the coefficient vector in linear regression has a closed-form expression, we can determine the conditional distribution of $\hat{\boldsymbol{\beta}}_{MLE}$ given $\mathbb{X}$ and $\boldsymbol{\beta}$. Since $\mathbf{Y} \mid \mathbb{X}, \boldsymbol{\beta}, \sigma^2 \sim N_n(\mathbb{X}\boldsymbol{\beta}, \sigma^2\mathbb{I}_n)$, it follows that

$$\begin{aligned}\mathbb{E}\left[\hat{\boldsymbol{\beta}}_{MLE} \mid \mathbb{X}, \boldsymbol{\beta}\right] &= \mathbb{E}\left[(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbf{Y} \mid \mathbb{X}, \boldsymbol{\beta}\right] = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{E}\left[\mathbf{Y} \mid \mathbb{X}, \boldsymbol{\beta}\right] \\ &= (\mathbb{X}^T\mathbb{X})^{-1}(\mathbb{X}^T\mathbb{X})\boldsymbol{\beta} = \boldsymbol{\beta},\end{aligned}$$

$$\begin{aligned}\mathrm{Cov}(\hat{\boldsymbol{\beta}}_{MLE} \mid \mathbb{X}, \boldsymbol{\beta}, \sigma^2) &= \mathrm{Cov}\left((\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbf{Y} \mid \mathbb{X}, \boldsymbol{\beta}, \sigma^2\right) \\ &= (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathrm{Cov}(\mathbf{Y} \mid \mathbb{X}, \boldsymbol{\beta}, \sigma^2)((\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T)^T \\ &= \sigma^2(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1} \\ &= \sigma^2(\mathbb{X}^T\mathbb{X})^{-1}.\end{aligned}$$

Thus $\hat{\boldsymbol{\beta}}_{MLE} \mid \mathbb{X}, \boldsymbol{\beta}, \sigma^2 \sim N_{p+1}(\boldsymbol{\beta}, \sigma^2(\mathbb{X}^T\mathbb{X})^{-1})$.

For the logistic model, the only parameter is $\boldsymbol{\beta}$. Its maximum likelihood estimator takes the form

$$\begin{aligned}\hat{\boldsymbol{\beta}}_{MLE} &= \operatorname*{arg\,max}_{\boldsymbol{\beta} \in \mathbb{R}^{p+1}} \prod_{i=1}^{n} \pi_i^{y_i}(1-\pi_i)^{1-y_i} \\ &= \operatorname*{arg\,max}_{\boldsymbol{\beta} \in \mathbb{R}^{p+1}} \sum_{i=1}^{n} \left(y_i \log(\pi_i) + (1-y_i)\log(1-\pi_i)\right) \\ &= \operatorname*{arg\,max}_{\boldsymbol{\beta} \in \mathbb{R}^{p+1}} \sum_{i=1}^{n} \left(\log(1-\pi_i) + y_i \log\frac{\pi_i}{1-\pi_i}\right) \\ &= \operatorname*{arg\,min}_{\boldsymbol{\beta} \in \mathbb{R}^{p+1}} \sum_{i=1}^{n} \left(\log(1+e^{\eta_i}) - y_i\eta_i\right).\end{aligned}$$

Unlike in Gaussian regression, the MLE of $\boldsymbol{\beta}$ in the logistic model lacks a closed-form solution. Consequently, it has to be obtained numerically using convex optimization algorithms.

3.3 Regularization methods

With high-dimensional data, using classical model estimators, like MLE, may not be possible or lead to overfitting. Regularization methods try to overcome these limitations by adding a penalty term to the loss function, shrinking coefficient values.

Definition 3.12. The penalized maximum likelihood estimator of regression coefficients is given by

$$\hat{\boldsymbol{\beta}}_{pen} = \underset{\boldsymbol{\beta}\in\mathbb{R}^{p+1}}{\arg\max} \left\{ l(\boldsymbol{\beta}, \boldsymbol{\varphi}) - \text{pen}(\lambda) \right\},$$

where $\text{pen}(\lambda)$ denotes a penalty term and λ is either a single tuning parameter or a tuning parameter vector.

Knowing that some covariates do not influence the outcome variable, we would like to exclude them from the model. This is achievable with an L_1-type penalty term, i.e., a scaled sum of the absolute values of the coefficients. It shrinks the coefficient estimates toward zero and, given sufficiently large tuning parameters, can set some of them exactly to zero. Thus, the corresponding variables will be excluded from the model, with their effect on the outcome variable assumed to be negligible.

One of the most fundamental methods used for variable selection is LASSO (Least Absolute Shrinkage and Selection Operator), first introduced in [8].

Definition 3.13. The LASSO regularization term takes the form

$$\text{pen}(\lambda) = \lambda \sum_{j=1}^{p} |\beta_j|,$$

where $\lambda \geq 0$ is a tuning parameter.

It is worth mentioning that, as shown in [9], a unique LASSO solution has at most $\min\{n, p\}$ nonzero coefficients. Therefore, in high-dimensional settings, when $p > n$, LASSO can select at most n of the p predictors. Since intercept is not penalized, it is not included in this restriction.

A great asset of LASSO is the ability to successfully recognize many important covariates. However, as shown in [10], it turns out that under linear sparsity, meaning that the proportion of variables with nonzero effects remains asymptotically constant, even with strong signals and independent predictors, false discoveries occur early during variable selection. In fact, under this constraint it is unattainable for LASSO to obtain high power and, at the same time, maintain a low false discovery rate [10].

To address these limitations, the SLOPE (Sorted L-One Penalized Estimation) method was proposed in [4].

Definition 3.14. The SLOPE penalty term is given by

$$\text{pen}(\lambda) = \alpha \sum_{j=1}^{p} \lambda_j |\beta|_{(j)},$$

where $\alpha \geq 0$ denotes a scaling parameter. Here, the ordered non-ascending absolute values of the penalized coefficients $|\beta|_{(1)} \geq \cdots \geq |\beta|_{(p)}$ are assigned the respective tuning parameters $\lambda_1 \geq \cdots \geq \lambda_p \geq 0$. The penalty can also be written as

$$\text{pen}(\lambda) = \alpha \sum_{j=1}^{p} \lambda_{r(\boldsymbol{\beta},j)} |\beta_j|,$$

where $r(\boldsymbol{\beta}, j) \in \{1, \ldots, p\}$ is the rank of $|\beta_j|$ among the absolute values $|\beta_1|, \ldots, |\beta_p|$ sorted in non-ascending order.

In the case of linear regression, when $\mathbf{Y} \mid \mathbb{X}, \boldsymbol{\beta}, \sigma^2 \sim N(\mathbb{X}\boldsymbol{\beta}, \sigma^2 \mathbb{I}_n)$, the scaling parameter becomes $\alpha = 1/\sigma$.

In particular, when $\lambda_1 = \cdots = \lambda_p = \lambda$, the SLOPE penalty term simplifies to

$$\text{pen}(\lambda) = \alpha \sum_{j=1}^{p} \lambda |\beta_j| = \alpha\lambda \sum_{j=1}^{p} |\beta_j| = \lambda' \sum_{j=1}^{p} |\beta_j|.$$

Thus, the solution of SLOPE becomes equivalent to the LASSO estimator.

As shown in [4], the use of an ordered non-ascending tuning parameter sequence $\boldsymbol{\lambda}$ in SLOPE is similar to the Benjamini-Hochberg multiple testing procedure, introduced in [11]. Given many statistical tests, it adjusts the discovery threshold by increasing it for the larger p-values, thereby controlling the false discovery rate.

Definition 3.15. Let V be the number of false discoveries, i.e., falsely rejected null hypotheses, and let R denote the number of all discoveries. The false discovery rate is defined as

$$\text{FDR} = \mathbb{E}\left[\frac{V}{\max\{R, 1\}}\right].$$

The tuning parameter sequence arising from the Benjamini-Hochberg critical values is given by

$$\lambda_{BH,i} = z\left(1 - i\frac{q}{2p}\right),$$

where $z(\cdot)$ denotes the quantile of the standard normal distribution. As proven in [4], $\boldsymbol{\lambda}_{BH}$ controls the FDR at level q for an orthogonal design matrix.

An important property of SLOPE, described extensively in [12], is its ability to merge regression coefficients by assigning the same absolute estimated values within the clusters. Consequently, the effective model dimensionality is reduced from the number of predictors to the number of clusters and thus, unlike LASSO, SLOPE can select more than n predictors.

Although combining maximum likelihood estimation with an L_1-type penalty allows for eliminating unimportant predictors from the model, shrinking all coefficients toward zero results in estimation bias. In consequence, it can lead to misrepresentation of the effect sizes. This issue is especially relevant for SLOPE, where relatively large tuning parameters may be required to control the FDR. An intuitive method is a two-step procedure: first, we identify important variables using an L_1-type penalty and then

estimate coefficients corresponding to the selected predictors with unpenalized GLM. Another way of preventing bias, which we present here, is to use the mixture-prior models based on the Bayesian approach. Let us begin with the general form of the Bayesian estimator and its connection to the penalized estimator.

3.4 Bayesian estimator

The objective of a Bayesian approach is to express uncertainty of unobserved quantities, such as model parameters, through a probability distribution. The distribution, called a priori, is chosen on the basis of presumptions about the nature of these quantities. Then, given the evidence data, it is updated according to the following theorem.

Theorem 1 (Bayes' theorem)**.** *Let* $\mathbf{y}$ *denote the observed data and let* $\boldsymbol{\theta}$ *be the unknown parameters. Then*

$$p(\boldsymbol{\theta} \mid \mathbf{y}) = \frac{p(\mathbf{y} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{y})},$$

with $p(\boldsymbol{\theta} \mid \mathbf{y})$ *representing the conditional probability of the parameters given the evidence, called the a posteriori distribution,* $p(\mathbf{y} \mid \boldsymbol{\theta})$ *standing for the likelihood and* $p(\boldsymbol{\theta})$ *denoting the a priori distribution.*

In particular, we can associate $\boldsymbol{\theta}$ in Theorem 1 with the coefficient vector $\boldsymbol{\beta}$, conditioning its distribution on any additional model parameters $\boldsymbol{\varphi}$. Then, the evidence $\mathbf{y}$ represents the observed values of the outcome variable. Thus, using the introduced interpretation and adding conditioning on the design matrix $\mathbb{X}$ and additional model parameters, we can rewrite theorem 1 as follows

$$p(\boldsymbol{\beta} \mid \mathbf{y}, \mathbb{X}, \boldsymbol{\varphi}) \propto p(\mathbf{y} \mid \boldsymbol{\beta}, \mathbb{X}, \boldsymbol{\varphi})p(\boldsymbol{\beta} \mid \mathbb{X}, \boldsymbol{\varphi}).$$

Given that, we aim to obtain the Bayesian estimator of the regression coefficient vector $\boldsymbol{\beta}$.

Definition 3.16. The Bayesian estimator of the regression model parameters is given by

$$\hat{\boldsymbol{\beta}}_{BAYES} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\arg\max}\, p(\boldsymbol{\beta} \mid \mathbf{y}, \mathbb{X}, \boldsymbol{\varphi}) = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\arg\max}\{\log p(\mathbf{y} \mid \boldsymbol{\beta}, \mathbb{X}, \boldsymbol{\varphi}) + \log p(\boldsymbol{\beta} \mid \mathbb{X}, \boldsymbol{\varphi})\}.$$

After comparing this approach with the penalized maximum likelihood estimator introduced in definition 3.12, it becomes clear that the two estimators are equivalent. By choosing the right prior, we can obtain any penalized maximum likelihood coefficient estimator, including the LASSO and SLOPE estimators, using the Bayesian approach.

First, we introduce the Laplace distribution.

Definition 3.17. The probability density function of a Laplace(μ, b) distribution with the location parameter μ and the scale parameter $b > 0$ is given by

$$p(x) = \frac{1}{2b} e^{-\frac{|x-\mu|}{b}}.$$

The mean and variance of a random variable following the Laplace distribution equal μ and $2b^2$, respectively.

Next, we define the LASSO prior.

Definition 3.18. The Bayesian LASSO estimate, introduced in [8], assumes the independent Laplace$(0, 1/\lambda)$ priors for each β_j, $j \in \{1, \ldots, p\}$

$$p(\beta_j) = \frac{\lambda}{2} e^{-\lambda|\beta_j|}.$$

The full LASSO prior density can be written as

$$p(\boldsymbol{\beta} \mid \lambda) = \prod_{j=1}^{p} p(\beta_j) = \left(\frac{\lambda}{2}\right)^p e^{-\lambda \sum_{i=1}^{p} |\beta_j|},$$

and after taking the logarithm, we obtain

$$\log p(\boldsymbol{\beta} \mid \lambda) = p \log \frac{\lambda}{2} - \lambda \sum_{j=1}^{p} |\beta_j|.$$

Since $p \log(\lambda/2)$ does not depend on the coefficient vector, it can be omitted from the final form of the estimator. As a result, after changing the sign, we recover the LASSO penalty defined in Definition 3.13.

The same approach can be used to derive the prior corresponding to the SLOPE penalty. The SLOPE prior on the coefficients in linear regression was defined in [13]. Here, we introduce the generalization of this prior.

Definition 3.19. The prior distribution for the SLOPE estimator is assumed to be

$$p(\boldsymbol{\beta} \mid \boldsymbol{\lambda}, \alpha) = C(\boldsymbol{\lambda}, \alpha) e^{-\alpha \sum_{j=1}^{p} \lambda_{r(\boldsymbol{\beta}, j)} |\beta_j|},$$

where $C(\boldsymbol{\lambda}, \alpha)$ is a normalizing constant and $\alpha \geq 0$ denotes a scaling parameter.

The logarithm of the SLOPE prior takes the form

$$\log p(\boldsymbol{\beta} \mid \boldsymbol{\lambda}, \alpha) = \log C(\boldsymbol{\lambda}, \alpha) - \alpha \sum_{j=1}^{p} \lambda_{r(\boldsymbol{\beta}, j)} |\beta_j|.$$

Removing the logarithm of constant and changing the sign gives us exactly the SLOPE penalty.

In [13], the SLOPE estimator in the linear case was defined with $\alpha = \frac{1}{\sigma}$. Tying the penalty and the noise variance allows for tuning the penalty to the variability of the response variable and guarantees a unimodal posterior for $\boldsymbol{\beta}$. A multi-modal posterior, i.e., with more than one local maximum of the probability density function, poses theoretical and computational challenges. As explained in [13], a single point estimate is not comprehensive in that scenario, as a proper summary should incorporate all modes. A multi-modal posterior can also slow the Markov chain Monte Carlo methods.

3.5 Mixture-prior estimators

To avoid the estimation bias arising from shrinking all coefficients toward zero, we may use a mixture-prior estimator. By combining two distributions, called the "spike" and "slab", this approach aims to achieve both variable selection and accurate coefficient estimation.

Definition 3.20. The general spike-and-slab prior is given by

$$p(\boldsymbol{\beta} \mid \boldsymbol{\gamma}) = \prod_{j=1}^{p} [\gamma_j \psi_1(\beta_j) + (1 - \gamma_j)\psi_0(\beta_j)], \quad \boldsymbol{\gamma} \sim p(\boldsymbol{\gamma}),$$

with $\psi_1(\beta)$ serving as a "slab" distribution for estimating signal strengths, concentrated around zero "spike" distribution $\psi_0(\beta)$, accounting for noise filtering, and $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_p)^T$, $\gamma_i \in \{0, 1\}$, indexing the 2^p possible subset models [14].

In [14] this idea was combined with the Bayesian LASSO prior, resulting in the spike-and-slab LASSO.

Definition 3.21. The spike-and-slab LASSO prior is a mixture of two Laplace distributions with different variances, namely

$$\psi_1(\beta) = \frac{\lambda_1}{2} e^{-\lambda_1 |\beta|}, \quad \lambda_1 \text{ small},$$
$$\psi_0(\beta) = \frac{\lambda_0}{2} e^{-\lambda_0 |\beta|}, \quad \lambda_0 \text{ large}.$$

Assuming

$$p(\boldsymbol{\gamma} \mid \theta) = \prod_{j=1}^{p} \theta^{\gamma_j} (1 - \theta)^{1 - \gamma_j}, \quad \theta \sim p(\theta),$$

where $\theta = P(\gamma_j = 1 \mid \theta)$ is the inclusion parameter representing the presumed fraction of signals, the prior distribution conditional on θ can be written as

$$p(\boldsymbol{\beta} \mid \theta) = \prod_{j=1}^{p} \left[\theta \frac{\lambda_1}{2} e^{-\lambda_1 |\beta_j|} + (1 - \theta) \frac{\lambda_0}{2} e^{-\lambda_0 |\beta_j|} \right].$$

Figure 1 presents example prior densities for the spike-and-slab LASSO. The "spike" distribution is characterized by substantially smaller variance than the "slab" distribution, thereby increasing shrinkage toward zero and promoting model sparsity. On the other hand, smaller λ_1 increases the variance of the Laplace distribution, resulting in higher probabilities of large coefficient values.

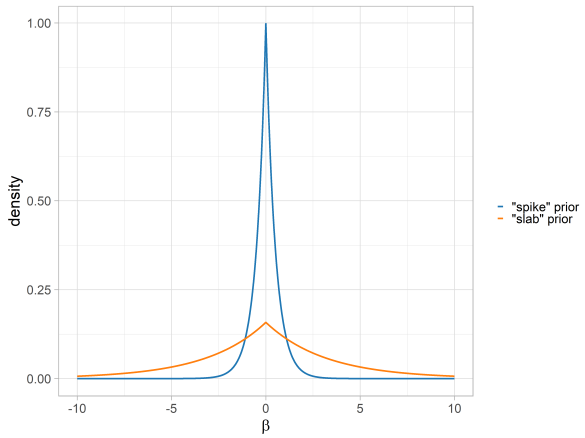

Figure 1: Spike-and-slab LASSO priors - Laplace densities.

4 Adaptive Bayesian SLOPE

Adaptive Bayesian SLOPE (ABSLOPE), first introduced in [3], follows the idea of spike-and-slab LASSO. By combining the mixture prior estimator with the Bayesian SLOPE model, it aims to suppress the over-shrinkage of important coefficients while maintaining satisfactory power and FDR levels. Unlike for SLOPE, there is no theoretical proof of ABSLOPE controlling the FDR, even under orthogonal design matrix assumption. Nevertheless, it often does so in practice, which was shown in [3] through extensive simulations across many settings. Moreover, ABSLOPE has the ability to impute the missing values in the design matrix during modeling, treating them as latent variables. This is an important extension, especially for application to biomedical data, which are often affected by missingness.

4.1 Gaussian adaptive Bayesian SLOPE

The original ABSLOPE formulation in [3] was developed solely for Gaussian regression. To present it, we first define a beta distribution, which is later used in the model as a prior for the inclusion parameter θ.

Definition 4.1. A random variable Y is said to follow the beta distribution $\text{Beta}(a, b)$, $a > 0$, $b > 0$, if its probability density function is given by

$$p(y) = y^{a-1}(1-y)^{b-1}\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)},$$

where

$$\Gamma(k) = \int_0^\infty t^{k-1}e^{-t}dt$$

is the gamma function. Then

$$\mathbb{E}[Y] = \frac{a}{a+b}, \quad \text{Var}(Y) = \frac{ab}{(a+b)^2(a+b+1)}.$$

The ABSLOPE mixture prior can be formulated as follows.

Definition 4.2. The adaptive Bayesian SLOPE prior is defined as

$$p(\boldsymbol{\beta} \mid \boldsymbol{\gamma}, c, \sigma^2, \boldsymbol{\lambda}) \propto c^{\sum_{j=1}^{p} \mathbb{1}\{\gamma_j = 1\}} \prod_{j=1}^{p} e^{-\frac{1}{\sigma} w_j \lambda_{r(\mathbb{W}\boldsymbol{\beta}, j)} |\beta_j|}.$$

It consists of several components.

1. The predictor inclusion indicator $\gamma_j \in \{0, 1\}$, determining whether β_j is a signal or noise. Similarly as to the spike-and-slab LASSO, $\gamma = (\gamma_1, \ldots, \gamma_p)^T$ indexes the 2^p possible model configurations and each γ_j is assumed to follow a Bernoulli distribution $Bernoulli(\theta)$, where $\theta \in (0, 1)$ is a mixing (prior inclusion) weight, indicating the level of model sparsity. It can be either fixed or assumed to come from a beta distribution $Beta(a, b)$, where the values of a and b are specified by the user.

2. The parameter $c \in (0, 1)$, controlling the scaling of the penalty between the noise and signal components. The non-informative uniform prior on $[0, 1]$ is assigned to c.

3. The diagonal weighting matrix $\mathbb{W} = \mathrm{diag}(w_1, \ldots, w_p)$, where

$$w_j = c\gamma_j + (1 - \gamma_j) = \begin{cases} c, & \gamma_j = 1 \\ 1, & \gamma_j = 0 \end{cases}.$$

4. If the noise variance is unknown, a non-informative prior $p(\sigma^2) \propto \frac{1}{\sigma^2}$ is assigned.

In the above definition, a mixture distribution is induced with the weighting matrix $\mathbb{W}$ through the scaling parameter c. When $\gamma_j = 1$, i.e., β_j is considered a signal, the penalty is reduced and the model switches to the slab prior. On the other hand, when $\gamma_j = 0$ and β_j is assumed to be a noise, the initial penalty, corresponding to the spike component, is preserved.

The ABSLOPE model was developed on the basis of the SLOPE prior. In fact, it can be shown that the solution of adaptive Bayesian SLOPE can be obtained through the SLOPE estimator with rescaled coefficients. To do this, we first formulate the distribution of a linear transformation of a random vector.

Lemma 1. *Let $\mathbf{Z}$ be a random vector following a distribution with probability density function $p_{\mathbf{Z}}(\mathbf{z})$. Let $\boldsymbol{\beta} = \mathbb{W}^{-1}\mathbf{Z}$, where $\mathbb{W}$ is a real, invertible matrix. Then the probability density of $\boldsymbol{\beta}$ is given by*

$$p(\boldsymbol{\beta}) = |\det(\mathbb{W})| p_{\mathbf{Z}}(\mathbb{W}\boldsymbol{\beta}).$$

We can now move on to the following proposition.

Proposition 1. Let $\mathbf{Z} = (Z_1, \ldots, Z_p)^T$ be a random vector with the SLOPE prior, given in Definition 3.19, with $\alpha = \frac{1}{\sigma}$, i.e.

$$p(\mathbf{z} \mid \sigma^2, \boldsymbol{\lambda}) \propto \prod_{j=1}^{p} e^{-\frac{1}{\sigma}\lambda_{r(\mathbf{z},j)}|z_j|}.$$

Then $\boldsymbol{\beta} = \mathbb{W}^{-1}\mathbf{Z} = \left(\frac{Z_1}{w_1}, \ldots, \frac{Z_p}{w_p}\right)^T$ follows the adaptive Bayesian SLOPE prior, introduced in Definition 4.2.

Proof. Let the random vector $\mathbf{Z} = (Z_1, \ldots, Z_p)^T$ follow the SLOPE prior. We define $\boldsymbol{\beta} = \mathbb{W}^{-1}\mathbf{Z}$, where $\mathbb{W} = \operatorname{diag}(w_1, \ldots, w_p)$ and

$$w_j = c\gamma_j + (1 - \gamma_j) = \begin{cases} c, & \gamma_j = 1 \\ 1, & \gamma_j = 0 \end{cases}, \qquad c \in (0, 1).$$

Using Lemma 1, the prior on $\boldsymbol{\beta}$ can be calculated as

$$\begin{aligned} p(\boldsymbol{\beta} \mid \mathbb{W}, \sigma^2, \boldsymbol{\lambda}) &\propto \prod_{j=1}^{p} w_j \prod_{j=1}^{p} e^{-w_j|\beta_j|\frac{1}{\sigma}\lambda_{r(\mathbb{W}\boldsymbol{\beta},j)}} \\ &= c^{\sum_{j=1}^{p} \mathbb{1}\{\gamma_j = 1\}} \prod_{j=1}^{p} e^{-w_j|\beta_j|\frac{1}{\sigma}\lambda_{r(\mathbb{W}\boldsymbol{\beta},j)}}. \end{aligned}$$

Thus, we obtain the exact adaptive Bayesian SLOPE prior. □

Figure 2 depicts the model variables, parameters and their dependencies in a graph. In addition to the elements described above, it also includes $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, considered solely for the missing values imputation.

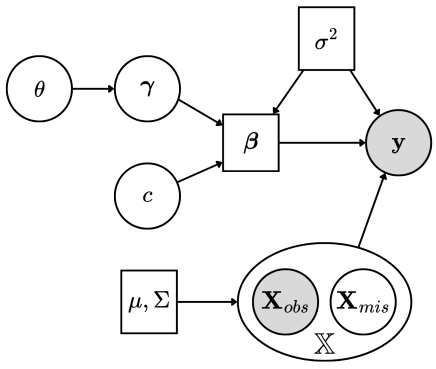

Figure 2: A graphical overview of the ABSLOPE model. White circles contain latent variables, colored circles show observed variables and squares designate parameters. Arrows depict dependencies between the model elements. Source: [3].

As mentioned before, ABSLOPE has the ability to impute values missing from the design matrix. Let $\mathbb{X}$ be a matrix of size $n \times p$, centered (which enables omitting the intercept) and standardized so that all columns have a unit L_2 norm, i.e.,

$$\forall j \in \{1, \ldots, p\} \qquad \sum_{i=1}^{n} X_{ij} = 0 \quad \text{and} \quad \sum_{i=1}^{n} X_{ij}^2 = 1.$$

For each row $i \in \{1, \ldots, n\}$ of the design matrix $\mathbb{X}$ we distinguish two parts: a vector $\mathbf{X}_{i,obs}$, which includes only observed data, and the missing part $\mathbf{X}_{i,mis}$. These two components are separated by denoting $\mathbb{X} = (\mathbf{X}_{obs}, \mathbf{X}_{mis})$. Moreover, all rows are presumed to follow a multivariate normal distribution $N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Both the decomposition of the design matrix and the parameters of its distribution were included in Figure 2.

Missing data can be classified using one of the three probabilistic mechanisms, described in [15]:

- missing completely at random (MCAR) - absence is not related to any values in the data, neither observed nor missing,
- missing at random (MAR) - the probability of the value being missing depends only on the observed data,
- missing not at random (MNAR) - absence depends on the missing values themselves.

In the case of ABSLOPE, the MAR mechanism is assumed, which means that the missingness mechanism is not considered when maximizing the likelihood. Since MCAR has stricter requirements than MAR, it holds also for that mechanism.

To summarize all models described in the thesis so far, we present their main characteristics in Figure 3.

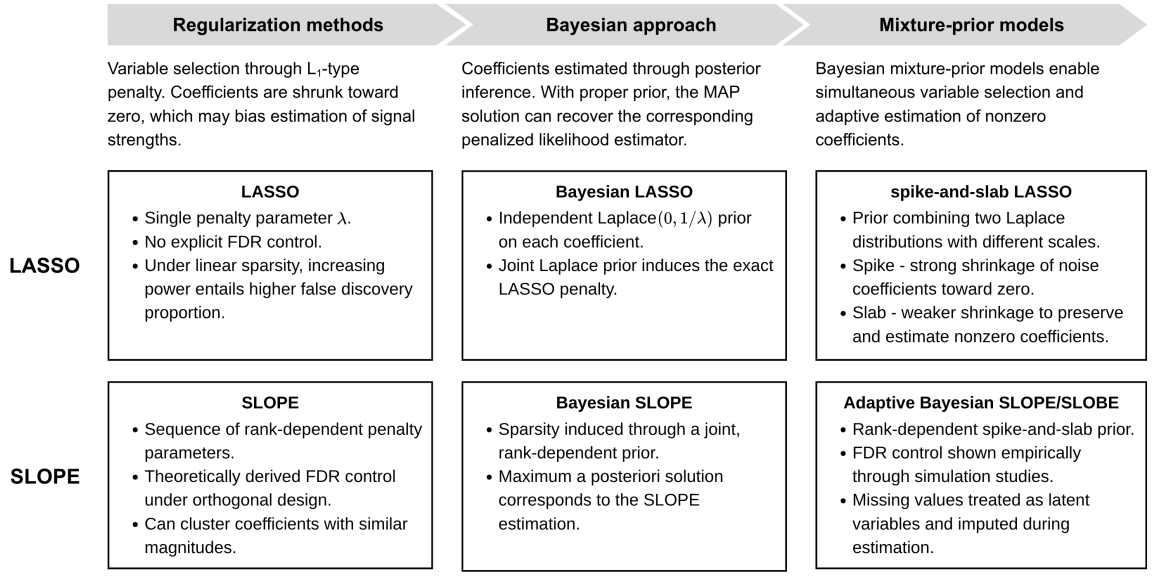

Figure 3: Summary of the introduced estimation methods along with the outlines of corresponding LASSO and SLOPE variants.

4.2 Parameter estimation

We present the algorithm used for ABSLOPE parameter estimation and model selection. It was based on the expectation-maximization algorithm.

Definition 4.3. The expectation-maximization (EM) algorithm, introduced in [16], is an iterative approach to finding maximum likelihood estimates of parameters in models involving unobserved latent variables. Let $\mathbf{t}$ denote the observed data, $\mathbf{z}$ the latent variables and $\boldsymbol{\omega}$ a vector of parameters. The complete-data log-likelihood is given by

$$l_{comp}(\boldsymbol{\omega}; \mathbf{t}, \mathbf{z}) = \log p(\mathbf{t}, \mathbf{z} \mid \boldsymbol{\omega}).$$

We aim to maximize the observed log-likelihood, i.e., the logarithm of the complete-data likelihood integrated over the latent variables

$$l_{obs}(\boldsymbol{\omega}) = \log p(\mathbf{t} \mid \boldsymbol{\omega}) = \log \int p(\mathbf{t}, \mathbf{z} \mid \boldsymbol{\omega})\, d\mathbf{z}.$$

In each iteration, the algorithm consists of two steps.

1. **E step.** Given the estimates of parameters from the previous iteration, we calculate the conditional expected value of the complete-data log-likelihood:

$$Q(\boldsymbol{\omega} \mid \boldsymbol{\omega}^{(t)}) = \mathbb{E}[l_{comp}(\boldsymbol{\omega}) \mid \mathbf{t}, \boldsymbol{\omega}^{(t)}].$$

2. **M step.** The parameter estimates are updated as

$$\boldsymbol{\omega}^{(t+1)} = \arg\max_{\boldsymbol{\omega}} Q(\boldsymbol{\omega} \mid \boldsymbol{\omega}^{(t)}).$$

ABSLOPE uses a modification of EM called stochastic approximation expectation-maximization (SAEM) algorithm. It estimates the latent variables by drawing one realization of them in each iteration and updating the expected complete-data log-likelihood using stochastic approximation. Because the sampling step can be computationally challenging, a faster version, called SLOBE, was proposed in [3]. Instead of a full stochastic approximation, SLOBE estimates the latent variable values with their conditional expected values. This not only decreases computational cost, but also can reduces the variability caused by sampling. The logistic extension was implemented only for SLOBE, therefore we omit the full SAEM procedure and focus only on the updates with conditional expectations. The algorithm consists of the following steps.

1. **Latent variables approximation.** At iteration t, the values of latent variables $\boldsymbol{z}^{(t)} = (\mathbf{X}_{mis}^{(t)}, \boldsymbol{\gamma}^{(t)}, c^{(t)}, \theta^{(t)})$ are estimated using their conditional expectation

$$\mathbb{E}\left[\mathbf{X}_{mis}, \boldsymbol{\gamma}, c, \theta \mid \mathbf{y}, \mathbf{X}_{obs}, \boldsymbol{\beta}^{(t-1)}, \sigma^{(t-1)}, \boldsymbol{\mu}^{(t-1)}, \boldsymbol{\Sigma}^{(t-1)}\right].$$

   In practice, given the quantities from the previous iteration (the superscripts are hidden to simplify notation), they are approximated as follows.

   (a) γ_j is updated as

$$\mathbb{E}[\gamma_j = 1 \mid \boldsymbol{\gamma}_{-j}, c, \boldsymbol{\beta}, \sigma, \theta, \mathbb{W}] = \frac{\theta c \exp\{-c\frac{1}{\sigma}|\beta_j|\lambda_{r(\mathbb{W}\boldsymbol{\beta},j)}\}}{(1-\theta)\exp\{-\frac{1}{\sigma}|\beta_j|\lambda_{r(\mathbb{W}\boldsymbol{\beta},j)}\} + \theta c \exp\{-c\frac{1}{\sigma}|\beta_j|\lambda_{r(\mathbb{W}\boldsymbol{\beta},j)}\}}.$$

   (b) θ is approximated as

$$\mathbb{E}[\theta \mid \boldsymbol{\gamma}, \boldsymbol{\beta}, \sigma, \mathbb{W}] = \frac{a + \sum_{j=1}^{p} \mathbb{1}\{\gamma_j = 1\}}{a + b + p}.$$

   (c) We update c with

$$\mathbb{E}[c \mid \boldsymbol{\gamma}, \boldsymbol{\beta}, \sigma, \mathbb{W}] = \frac{\int_0^1 x^{a'} e^{-b'x} dx}{\int_0^1 x^{a'-1} e^{-b'x} dx},$$

   where

$$a' = 1 + \sum_{j=1}^{p} \mathbb{1}\{\gamma_j = 1\}, \quad b' = \frac{1}{\sigma}\sum_{j=1}^{p} |\beta_j|\lambda_{r(\mathbb{W}\boldsymbol{\beta},j)}\mathbb{1}\{\gamma_j = 1\}.$$

   (d) Missing values are imputed using their conditional expectation

$$\mathbf{X}_{i,mis}^{(t)} = \mathbb{E}\left[\mathbf{X}_{i,mis} \mid y_i, \mathbf{X}_{i,obs}, \boldsymbol{\beta}^{(t-1)}, \sigma^{(t-1)}, \boldsymbol{\mu}^{(t-1)}, \boldsymbol{\Sigma}^{(t-1)}\right].$$

Since in Gaussian regression the conditional distribution of $\mathbf{Y}$ is normal and the prior on missing values is also Gaussian, the posterior distribution, given by

$$\begin{aligned}\mathbf{X}_{mis} &\sim p(\mathbf{X}_{mis} \mid \boldsymbol{\gamma}, c, \mathbf{y}, \mathbf{X}_{obs}, \boldsymbol{\beta}, \sigma, \theta, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ &= p(\mathbf{X}_{mis} \mid \mathbf{y}, \mathbf{X}_{obs}, \boldsymbol{\beta}, \sigma, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ &\propto p(\mathbf{y} \mid \mathbf{X}_{obs}, \mathbf{X}_{mis}, \boldsymbol{\beta}, \sigma) p(\mathbf{X}_{mis} \mid \mathbf{X}_{obs}, \boldsymbol{\mu}, \boldsymbol{\Sigma}).\end{aligned}$$

is Gaussian as well.

2. **Maximization.** Parameter estimates are obtained by maximizing the complete-data log-likelihood.

   (a) Coefficient vector $\boldsymbol{\beta}$ is estimated as

   $$\boldsymbol{\beta}^{(t)} = \arg\max_{\boldsymbol{\beta}} \left\{ -\frac{1}{2(\sigma^{(t-1)})^2} \|\mathbf{y} - \mathbb{X}^{(\mathbf{t})}\boldsymbol{\beta}\|_2^2 - \frac{1}{\sigma^{(t-1)}} \sum_{j=1}^{p} w_j^{(t)} |\beta_j| \lambda_{r(\mathbb{W}^{(t)}\boldsymbol{\beta}, j)} \right\},$$

   where $\mathbb{X}^{(t)} = (\mathbf{X}_{obs}, \mathbf{X}_{mis}^{(t)})$. By Proposition 1, it boils down to obtaining the SLOPE estimates given $\mathbb{W}^{(t)}$, $\mathbb{X}^{(t)}$ and $\sigma^{(t-1)}$.

   (b) In case of unknown noise variance, σ is calculated as

   $$\begin{aligned}\sigma^{(t)} &= \arg\max_{\sigma > 0} \left\{ -n \log(\sigma) - \frac{1}{2\sigma^2} \|\mathbf{y} - \mathbb{X}^{(\mathbf{t})}\boldsymbol{\beta}^{(\mathbf{t})}\|_2^2 - \frac{A^{(t)}}{\sigma} \right\} \\ &= \frac{1}{2n} \left(A^{(t)} + \sqrt{(A^{(t)})^2 + 4n \|\mathbf{y} - \mathbb{X}^{(t)}\boldsymbol{\beta}^{(t)}\|_2^2} \right),\end{aligned}$$

   where

   $$A^{(t)} = \sum_{j=1}^{p} w_j^{(t)} |\beta_j^{(t)}| \lambda_{r(\mathbb{W}^{(t)}\boldsymbol{\beta}^{(t)}, j)}.$$

   (c) Design matrix distribution parameters are $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ are calculated as

   $$\boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(t)} = \arg\max_{\boldsymbol{\mu}, \boldsymbol{\Sigma}} -\frac{1}{2} \log(2\pi |\boldsymbol{\Sigma}|) - \frac{1}{2} (\mathbb{X}^{(t)} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbb{X}^{(t)} - \boldsymbol{\mu}).$$

   When $p << n$, the solution is equivalent to the empirical mean and covariance matrix

   $$\boldsymbol{\mu}^{(t)} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i^{(t)}, \qquad \boldsymbol{\Sigma}^{(t)} = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}_i^{(t)} - \boldsymbol{\mu}^{(t)})(\mathbf{x}_i^{(t)} - \boldsymbol{\mu}^{(t)})^T.$$

   In high-dimensional setting, the covariance matrix is calculated using the Ledoit-Wolf shrinkage estimator

   $$\boldsymbol{\Sigma}^{(t)} = \rho_1 \mathbb{I}_p + \rho_2 S, \quad \text{where} \quad \rho_1, \rho_2 = \arg\min_{\rho_1, \rho_2} \mathbb{E} \|\boldsymbol{\Sigma}^{(t)} - \Sigma\|_2^2,$$

   where S is the sample covariance matrix.

4.3 Logistic adaptive Bayesian SLOPE

We extend the original Gaussian ABSLOPE, allowing the modeling of a binary response. One of the main differences between the two models concerns the scaling. In Gaussian ABSLOPE, the prior depends on the noise standard deviation σ, which does not exist in logistic regression. In [17], simulation involving a binary outcome variable and SLOPE were conducted using the $\boldsymbol{\lambda}_{BH}$ tuning sequence scaled by 0.5. Motivated by this choice, we replace the scaling factor $\frac{1}{\sigma}$ in the prior from Definition 4.2 with 0.5.

Definition 4.4. The prior on the coefficients of the logistic ABSLOPE is given by

$$p(\boldsymbol{\beta} \mid \boldsymbol{\gamma}, c, \boldsymbol{\lambda}) \propto c^{\sum_{j=1}^{p} \mathbb{1}\{\gamma_j = 1\}} \prod_{j=1}^{p} e^{-0.5 w_j \lambda_{r(\mathbb{W}\boldsymbol{\beta}, j)} |\beta_j|}.$$

Other prior components, including the prior distributions and interpretations of $\boldsymbol{\gamma}$, θ, c and $\mathbb{W}$, remain the same. Since the only modification is the scaling factor, Proposition 1 still holds for $\alpha = 0.5$.

The SLOBE algorithm was adapted analogously. Each occurrence of $\frac{1}{\sigma}$ in the conditional expectations of latent variables was replaced by 0.5, while the parts that do not depend on σ remained unchanged. The main modification was made in the coefficient vector estimation through SLOPE, where the Gaussian likelihood was replaced by the logistic likelihood

$$\boldsymbol{\beta}^{(t)} = \arg\max_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} (y_i \eta_i - \log(1 + e^{\eta_i})) - 0.5 \sum_{j=1}^{p} w_j^{(t)} |\beta_j| \lambda_{r(\mathbb{W}^{(t)}\boldsymbol{\beta}, j)} \right\}.$$

Thus, this step corresponds to the logistic version of SLOPE estimator with $\alpha = 0.5$.

An important component of Gaussian ABSLOPE is imputation, which relies on the Gaussian likelihood. In logistic ABSLOPE, the conditional distribution of the missing covariates is no longer normal and the imputation mechanism used in the Gaussian model is not available. Thus, we omit the imputation of missing values in the logistic extension.

5 Implementation details

The new ABSLOPE implementation was designed to be modular and easy to extend in the future. The main algorithm and core functions, previously developed using Rcpp, were implemented in a standalone C++ package, available in the GitHub repository [18]. Separating the pivotal functionality from the user interface makes it easier to support other programming languages, such as Python, in the future. Moreover, changing ADMM solver to hybrid coordinate descent algorithm, implemented in `libslope` package in C++, required further adaptation of the code by switching from the Armadillo to the Eigen library.

We provide a simple interface through an R package, which can be downloaded from the GitHub repository [19]. The `slobe` package allows users to generate example data,

fit the ABSLOPE model using the SLOBE algorithm and access the estimated model parameters directly from R. A simple example of fitting the model in R is presented below.

```
library(slobe)

dat <- slobe:::generate_data(
    n = 100, p = 100, signals_num = 15, signals_strength = 1,
    rho = 0, p_missing = 0.1, response = "gaussian"
)
fit <- slobe(dat$X, dat$y, family = "gaussian")
fit$coefficients
```

6 Simulation study

We evaluate the new SLOBE implementation and examine the logistic extension through simulations following the simulation studies based on the frameworks in [3] and [17]. All presented analyses were conducted in R and are available in the GitHub repository [20].

6.1 Simulation scheme

We generate artificial datasets with n observations and p predictors as follows.

1. The rows of the design matrix $\mathbb{X}$ are generated independently from the multivariate normal distribution $N_p(0, \frac{1}{n}\mathbb{R})$, with a Toeplitz correlation matrix

$$\mathbb{R} = \begin{pmatrix} 1 & \rho & \cdots & \rho^{p-1} \\ \rho & 1 & \cdots & \rho^{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ \rho^{p-1} & \rho^{p-2} & \cdots & 1 \end{pmatrix},$$

   where ρ is the correlation parameter.

2. For a fixed coefficient vector $\boldsymbol{\beta}$, we calculate the linear predictor

$$\boldsymbol{\eta} = (\eta_1, \ldots, \eta_n)^T = \mathbb{X}\boldsymbol{\beta}.$$

   Then, the response is generated according to the corresponding model. In the Gaussian case, by Definition 3.5, we use

$$\mathbf{y} = \mathbb{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

   where $\boldsymbol{\varepsilon} \sim N(0, \mathbb{I}_n)$.

To generate a binary outcome vector in the logistic case, we first obtain the success probabilities according to the model in Definition 3.8:

$$\pi_i = \frac{1}{1 + e^{-\eta_i}}, \quad i \in \{1, \ldots, n\}.$$

Then, each y_i is independently drawn from the Bernoulli(π_i) distribution.

3. Missing values are introduced into the design matrix according to the missing completely at random (MCAR) mechanism. For each element x_{ij}, a value u_{ij} is drawn from the uniform distribution U[0, 1] and x_{ij} is amputed if u_{ij} is smaller than the specified proportion of missing values.

In each scenario, for both Gaussian and logistic models, we repeat the experiments 200 times. Then, the performance evaluation is based on the power, false discovery rate, relative mean squared error and relative squared prediction error [3]:

- Power = $\frac{TP}{TP+FN}$,
- FDR = $\mathbb{E}\left[\frac{FP}{\max\{FP+TP,1\}}\right]$,
- MSE = $\frac{\|\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}\|_2^2}{\|\boldsymbol{\beta}\|_2^2}$,
- MSP = $\frac{\|\mathbb{X}\hat{\boldsymbol{\beta}}-\mathbb{X}\boldsymbol{\beta}\|_2^2}{\|\mathbb{X}\boldsymbol{\beta}\|_2^2}$.

Here, TP denotes the number of true positives, meaning coefficients correctly classified as nonzero, FP is the number of false positives, i.e., zero coefficients incorrectly identified as signals and FN represents the number of false negatives - true signals identified as noise. The obtained measures values are averaged over 200 repetitions.

6.2 Comparison of SLOBE and ABSLOPE implementations

First, we compare the original and new SLOBE implementations, which are expected to yield similar results, and additionally include ABSLOPE with the full SAEM algorithm. We set $n = p = 100$ and consider the following scenarios:

- 3, 6, 10, 12 and 15 signals,
- no missingness and 10% missing values,
- $\rho \in \{0, 0.5\}$,
- following [3], the nonzero elements of $\boldsymbol{\beta}$ take the form $c\sqrt{2 \log p}$ for $c \in \{1.3, 3\}$, where $c = 1.3$ corresponds to weak signals and $c = 3$ to strong signals.

The $\boldsymbol{\lambda}_{BH}$ tuning sequence was generated using the nominal FDR level $q = 0.05$.

Figure 4 shows the average computation times.

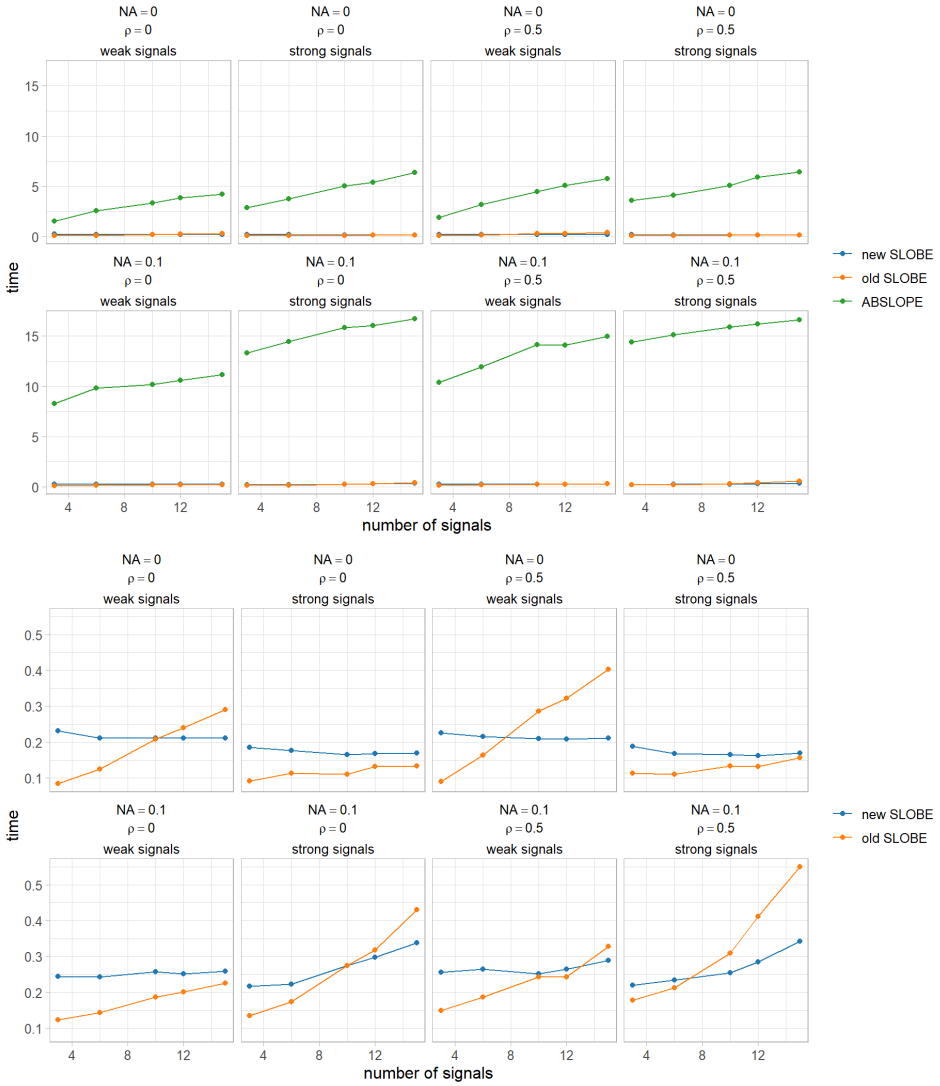

Figure 4: Average computation times for both versions of SLOBE and ABSLOPE depending on the proportion of missing values, predictor correlation, signal strength and number of signals. The second panel shows a closer comparison of the two SLOBE implementations.

As expected, ABSLOPE using the SAEM algorithm required considerably more time than the SLOBE implementations, especially for datasets with missing values. The computation times of both ABSLOPE and old SLOBE increased with the number of signals. In case of the new SLOBE implementation, this trend was less pronounced.

Figure 5 shows the powers obtained by each method in different scenarios.

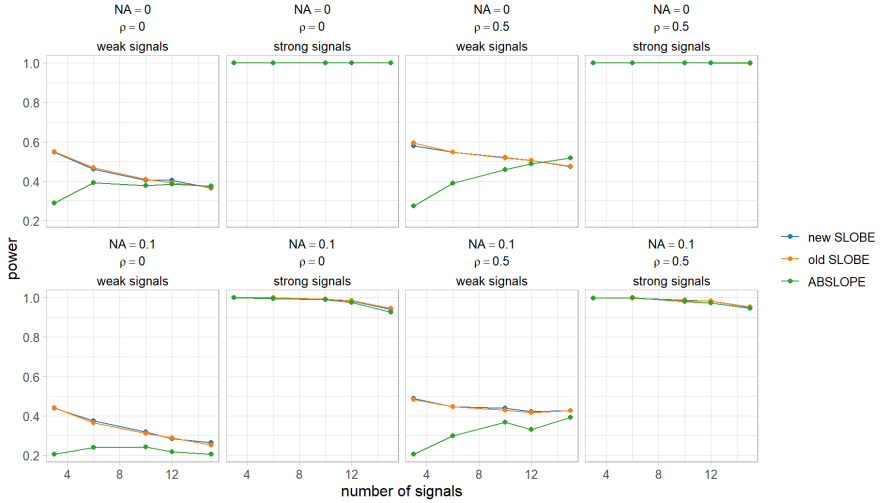

Figure 5: Power of both SLOBE versions and ABSLOPE depending on the proportion of missing values, predictor correlation, signal strength and number of signals.

In the case of strong signals, all implementations yielded similar results, with power close to one. For weak signals, both SLOBE implementations produced very similar results and generally achieved average power higher or comparable to that of ABSLOPE. The difference between SLOBE and ABSLOPE was particularly visible with the small number of signals.

Figure 6 presents the average false discovery rates.

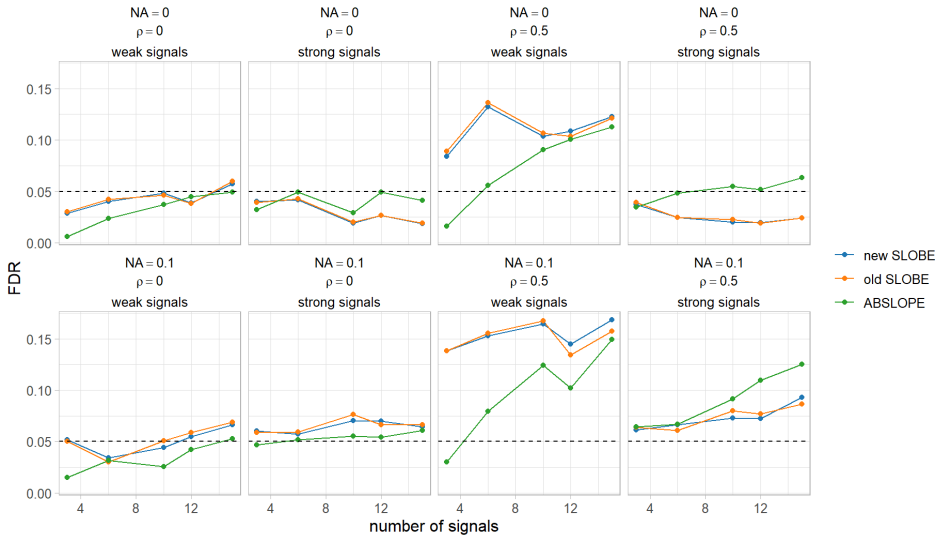

Figure 6: False discovery rates obtained for old and new SLOBE implementations and ABSLOPE across analysed scenarios. Black line shows the nominal FDR level $q = 0.05$.

Once again, the two SLOBE implementations obtained almost identical values. The

largest differences between SLOBE and ABSLOPE occurred for the weak signals combined with correlated predictors, which also yielded the highest false discovery rates. With uncorrelated predictors, the FDR generally remained close to the nominal level $q = 0.05$. ABSLOPE often achieved lower FDR for weak signals, although this was not consistent for strong signals, especially with correlated predictors.

Figure 7 presents the average relative mean squared errors for each method across all scenarios.

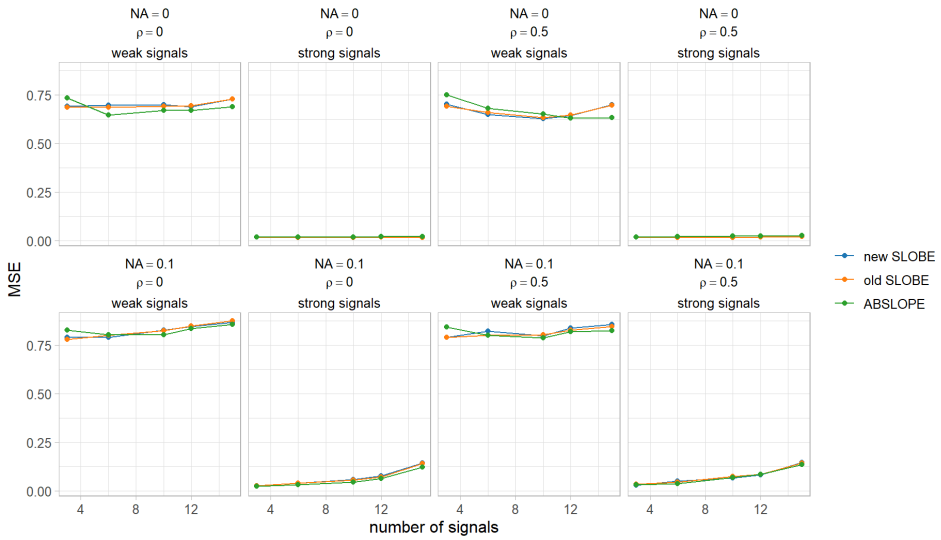

Figure 7: Average MSE values for the old and new SLOBE implementations and ABSLOPE across the considered scenarios.

The three algorithms obtained very similar estimation errors across all settings. Strong signals resulted in substantially lower MSE values and introducing missing values generally increased the error slightly. Correlation had almost no effect on the results.

Figure 8 presents the obtained average MSP values.

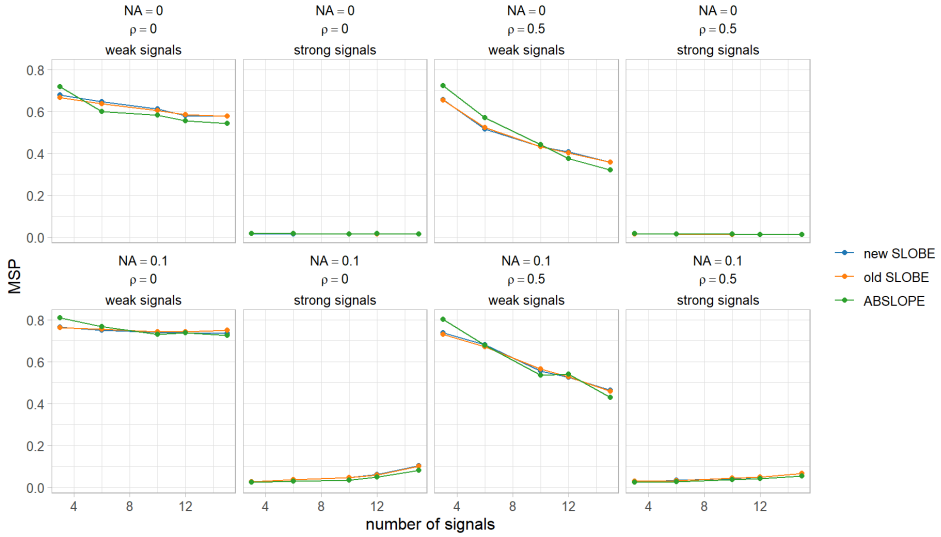

Figure 8: Average MSP values for the old and new SLOBE implementations and ABSLOPE across the considered scenarios.

The MSP results were very similar to those obtained for the estimation errors. The most notable difference was the decrease in prediction errors as the number of nonzero coefficients increased for weak signals.

Overall, the SLOBE implementation produced very similar results to the original algorithm while showing a slower increase in computation time as the number of signals grew.

6.3 Comparison of logistic models

For the logistic simulations, we set $n = 100$ and consider the following settings:

- $p = 100$ signals: 3, 6, 10, 12 and 15 signals,
  $p = 300$ signals: 50, 100, 120, 150 signals,
- $\rho \in \{0, 0.5\}$,
- following [17], the nonzero coefficients are set to 7, 10 or 15.

Since the imputation is not included in the logistic extension, all datasets in this part are generated without missing values. We compare logistic ABSLOPE with the following models:

- LASSO with the tuning parameter selected by cross-validation based on AUC,
- SLOPE with the tuning sequence $\boldsymbol{\lambda}_{BH}$ with $q = 0.05$ and scaling parameter α selected through cross-validation based on AUC,
- spike-and-slab LASSO with spike parameter $\lambda_1 = 0.5$ and slab parameter λ_0 tuned using the cross-validation.

Figure 9 depicts the performance of all models for $p = 100$.

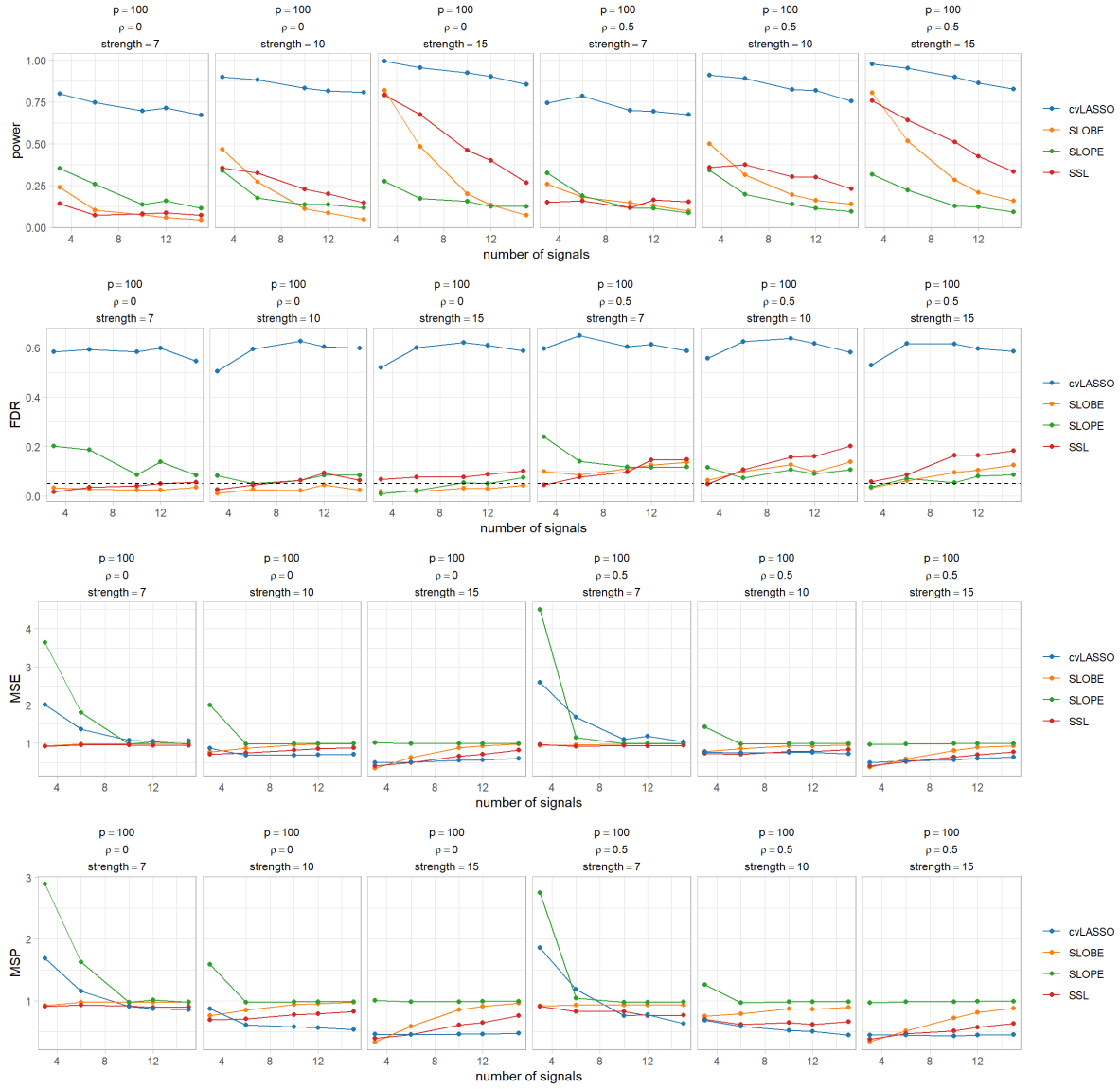

Figure 9: Comparison of logistic SLOBE, cross-validated LASSO, SLOPE and spike-and-slab LASSO for $n = p = 100$ across different signal strengths, predictor correlations and numbers of signals. The dashed line on the FDR plot indicates the nominal level $q = 0.05$.

Cross-validated LASSO achieved the highest power in every scenario, although at the same time it produced considerably higher FDR values. This is consistent with the properties of LASSO discussed previously - high power can come at the cost of a large number of false discoveries. Despite this, LASSO maintained relatively low MSE and MSP values compared with the other models.

SLOBE turned out to be much more conservative, achieving substantially lower power, especially for weaker signals. However, for uncorrelated predictors, it was the

only method that consistently maintained the FDR below the nominal level $q = 0.05$.

Spike-and-slab LASSO achieved better or similar power to SLOBE. Its FDR values were also slightly higher, but at the same time it yielded lower relative prediction and estimation errors, especially for stronger signals.

Unlike the other models, SLOPE did not show an increase in power for stronger signals. Moreover, even for uncorrelated predictors, it did not maintain the FDR control on a level $q = 0.05$, with particularly high values observed for weaker signals. This illustrates the advantage of the adaptive Bayesian SLOPE, which allows a stronger penalty to be assigned to noise coefficients and can therefore reduce the FDR. Moreover, SLOBE improved MSE and MSP compared with SLOPE, which is consistent with the aim of the spike-and-slab approach to reduce estimation bias.

It is also worth noticing that in the logistic setting, correlation between the predictors mainly affected the FDR. Its influence on power, MSE and MSP was much less pronounced.

Figure 10 presents the results for the higher dimensional setting with $p = 300$. These results generally confirm the relationships observed for $p = 100$, although the overall results considerably worsened. In particular, all methods achieved substantially lower power, while the relative estimation and prediction errors increased. LASSO once again obtained the highest power, along with high FDR values, whereas SLOBE maintained the low power and FDR values.

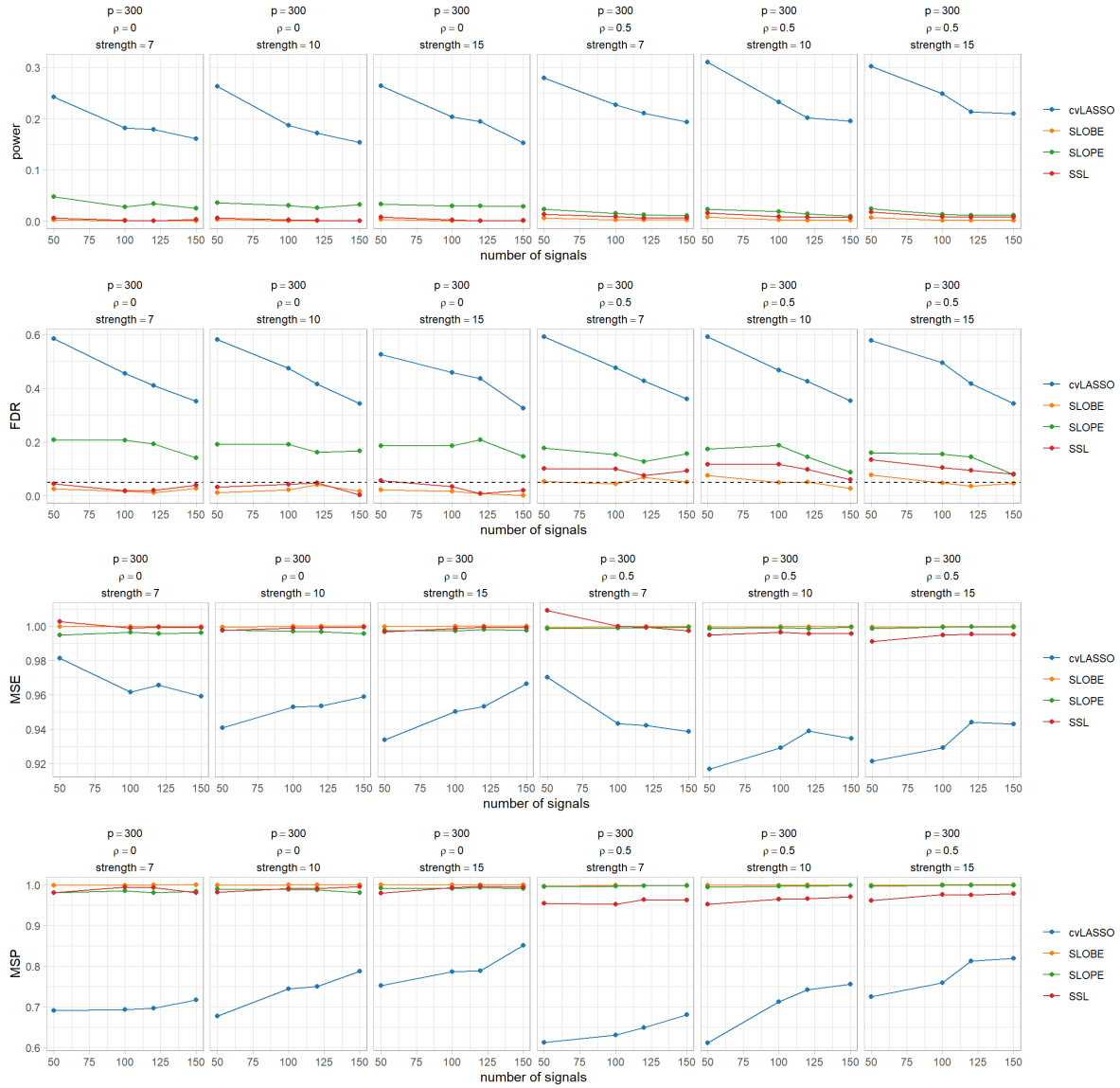

Figure 10: Comparison of logistic SLOBE, cross-validated LASSO, SLOPE and spike-and-slab LASSO for $n = 100$ and $p = 300$ across different signal strengths, predictor correlations and numbers of signals. The dashed line on the FDR plot indicates the nominal level $q = 0.05$

7 Conclusions

Regression methods play an important role in modern data analysis. Classical generalized linear models, although still widely used, may become insufficient for high-dimensional datasets characterized by strong correlations, sparse signals and missing values. In this thesis, starting from basic regression models, we discussed regularization methods that enable variable selection, Bayesian approaches providing greater flexibility and mixture models combining two distributions within a single prior. These concepts

led to adaptive Bayesian SLOPE, which addresses these challenges, often encountered in demanding biomedical experiments.

The main contribution of this thesis was the extension of ABSLOPE from Gaussian to logistic regression. The existing SLOBE implementation was rewritten using a more efficient solver and made available as C++ and R packages. The performance of both the new implementation and the logistic extension was evaluated through simulation studies.

The presented work offers several directions for further development. As mentioned previously, the standalone C++ package containing the core SLOBE algorithm provides a good starting point for integration with other programming languages, such as Python. A crucial extension would be to incorporate the imputation of missing values in the logistic adaptive Bayesian SLOPE. One possible approach is Polya-Gamma augmentation, which through additional latent variables transforms the logistic likelihood into a conditionally Gaussian form [21]. Another possible direction is to extend ABSLOPE to other regression models, for example multinomial regression. Overall, the flexibility of the Bayesian approach and the separation of the SLOBE algorithm from the language-specific interface provide several possibilities for the further development of ABSLOPE.

Notation

μ - mean

σ^2 - variance

Σ - covariance matrix

n - number of observations

p - number of regressors

$\mathbf{Y}$ - response (outcome) random variable

$\mathbf{y}$ - observed response vector

$\mathbb{X}$ - design matrix

$\boldsymbol{\varepsilon}$ - noise vector

$\boldsymbol{\beta}$ - coefficient vector

$\mathbb{I}_n$ - identity matrix of size $n \times n$

π - expected value of a variable following the Bernoulli distribution (probability of success)

η - linear predictor

L - likelihood function

l - log-likelihood function

$\boldsymbol{\omega}$ - a vector of all model parameters

$\boldsymbol{\varphi}$ - a vector of model parameters other than coefficients

$\hat{\boldsymbol{\beta}}_{MLE}$ - maximum likelihood estimator of the coefficient vector

$\mathcal{I}(\boldsymbol{\beta})$ - Fisher information matrix

$\text{pen}(\lambda)$ - penalty term

λ - tuning parameter

α - scaling parameter

$r(\boldsymbol{\beta}, j)$ - rank of $|\beta_j|$ among the absolute values of the coefficient vector sorted in the non-increasing order

$p(\cdot)$ - probability density or mass function

$C(\cdot)$ - normalizing constant

$\boldsymbol{\gamma}$ - vector of signal inclusion indicators

ψ_1 - "slab" distribution

ψ_0 - "spike" distribution

θ - inclusion parameter

c - penalty scaling parameter

$\mathbb{W}$ - diagonal weighting matrix

References

[1] Hamid Abdollahi et al. "Cochlea CT radiomics predicts chemoradiotherapy induced sensorineural hearing loss in head and neck cancer patients: A machine learning and multi-variable modelling study". In: *Physica Medica* (2018), pp. 192–197. DOI: 10.1016/j.ejmp.2017.10.008.

[2] Daniel W Belsky et al. "Quantification of the pace of biological aging in humans through a blood test, the DunedinPoAm DNA methylation algorithm". In: *eLife* (2020). Ed. by Sara Hagg et al. DOI: 10.7554/eLife.54870.

[3] Wei Jiang et al. "Adaptive Bayesian SLOPE: Model Selection With Incomplete Data". In: *Journal of Computational and Graphical Statistics* (2021), pp. 113–137. DOI: 10.1080/10618600.2021.1963263.

[4] Małgorzata Bogdan et al. "SLOPE - Adaptive variable selection via convex optimization". In: *The Annals of Applied Statistics* (2015). DOI: 10.1214/15-AOAS842.

[5] William Hsu et al. "Role of Clinical and Imaging Risk Factors in Predicting Breast Cancer Diagnosis Among BI-RADS 4 Cases". In: *Clinical Breast Cancer* (2019). DOI: 10.1016/j.clbc.2018.08.008.

[6] Ieva Kubiliute et al. "Clinical characteristics and predictors for in-hospital mortality in adult COVID-19 patients: A retrospective single center cohort study in Vilnius, Lithuania". In: *PLOS ONE* (2023). DOI: 10.1371/journal.pone.0290656.

[7] Johan Larsson et al. *Efficient Solvers for SLOPE in R, Python, Julia, and C++*. 2026. DOI: 10.48550/arXiv.2511.02430.

[8] Robert Tibshirani. "Regression Shrinkage and Selection Via the Lasso". In: *Journal of the Royal Statistical Society: Series B (Methodological)* (1996), pp. 267–288. DOI: 10.1111/j.2517-6161.1996.tb02080.x.

[9] Ryan J. Tibshirani. "The lasso problem and uniqueness". In: *Electronic Journal of Statistics* (2013). DOI: 10.1214/13-EJS815.

[10] Weijie Su, Małgorzata Bogdan, and Emmanuel Candes. "False Discoveries Occur Early on the Lasso Path". In: *The Annals of Statistics* (2015). DOI: 10.1214/16-AOS1521.

[11] Yoav Benjamini and Yosef Hochberg. "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing". In: *Journal of the Royal Statistical Society: Series B (Methodological)* (1995), pp. 289–300. DOI: 10.1111/j.2517-6161.1995.tb02031.x.

[12] Małgorzata Bogdan et al. "Pattern recovery by SLOPE". In: *Applied and Computational Harmonic Analysis* (2026). DOI: 10.1016/j.acha.2025.101810.

[13] Amir Sepehri. *The Bayesian SLOPE*. 2016. DOI: 10.48550/arXiv.1608.08968.

[14] Veronika Ročková and Edward George. "The Spike-and-Slab LASSO". In: *Journal of the American Statistical Association* (2018). DOI: 10.1080/01621459.2016.1260469.

[15] Roderick Little and Donald Rubin. “Statistical Analysis with Missing Data, Third Edition”. In: Wiley Series in Probability and Statistics (2019). DOI: 10.1002/9781119482260.

[16] A. P. Dempster, N. M. Laird, and D. B. Rubin. “Maximum Likelihood from Incomplete Data Via the EM Algorithm”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* (1977), pp. 1–22. DOI: 10.1111/j.2517-6161.1977.tb01600.x.

[17] Michał Kos. *Identyfikacja istotnych predyktorów w dużych bazach danych. Własności teoretyczne i zastosowania praktyczne.* 2019.

[18] URL: https://github.com/JoannaPokora/libslobe.

[19] URL: https://github.com/JoannaPokora/slobe.

[20] URL: https://github.com/JoannaPokora/SLOBEsimulations.

[21] Nicholas G. Polson, James G. Scott, and Jesse Windle. “Bayesian Inference for Logistic Models Using Pólya–Gamma Latent Variables”. In: *Journal of the American Statistical Association* (2013), pp. 1339–1349. DOI: 10.1080/01621459.2013.829001.