

Scalable Bayesian Regression in High Dimensions With Multiple Data Sources

Konstantinos Perrakis, Sach Mukherjee & The Alzheimer's Disease Neuroimaging Initiative

To cite this article: Konstantinos Perrakis, Sach Mukherjee & The Alzheimer's Disease Neuroimaging Initiative (2020) Scalable Bayesian Regression in High Dimensions With Multiple Data Sources, Journal of Computational and Graphical Statistics, 29:1, 28-39, DOI: [10.1080/10618600.2019.1624294](https://doi.org/10.1080/10618600.2019.1624294)

To link to this article: <https://doi.org/10.1080/10618600.2019.1624294>

 View supplementary material 

 Published online: 15 Jul 2019.

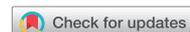
 Submit your article to this journal 

 Article views: 332

 View related articles 

 View Crossmark data 

 Citing articles: 2 View citing articles 



Scalable Bayesian Regression in High Dimensions With Multiple Data Sources

Konstantinos Perrakis, Sach Mukherjee, and The Alzheimer's Disease Neuroimaging Initiative*

Statistics and Machine Learning, German Center for Neurodegenerative Diseases (DZNE), Bonn, Germany

ABSTRACT

Applications of high-dimensional regression often involve multiple sources or types of covariates. We propose methodology for this setting, emphasizing the “wide data” regime with large total dimensionality p and sample size $n \ll p$. We focus on a flexible ridge-type prior with shrinkage levels that are specific to each data type or source and that are set automatically by empirical Bayes. All estimation, including setting of shrinkage levels, is formulated mainly in terms of inner product matrices of size $n \times n$. This renders computation efficient in the wide data regime and allows scaling to problems with millions of features. Furthermore, the proposed procedures are free of user-set tuning parameters. We show how sparsity can be achieved by post-processing of the Bayesian output via constrained minimization of a certain Kullback–Leibler divergence. This yields sparse solutions with adaptive, source-specific shrinkage, including a closed-form variant that scales to very large p . We present empirical results from a simulation study based on real data and a case study in Alzheimer's disease involving millions of features and multiple data sources. Supplementary materials for this article are available online.

ARTICLE HISTORY

Received January 2018
Revised April 2019

KEYWORDS

Bayesian post-processing;
High-dimensional regression;
Multiple data types;
Prediction; Ridge
regularization; Shrinkage
priors

1. Introduction

Advances in data acquisition have meant that studies in many fields now routinely include multiple sources of features, such as different data types, with one or more of the sources being high-dimensional. To fix ideas, consider a biomedical setting in which samples indexed by $i = 1, \dots, n$ each have response y_i and features of several types $k = 1, \dots, K$ (representing say genetic data, imaging, clinical covariates, and so on) with respective dimensionalities p_1, \dots, p_K . We refer to the different types of feature as sources. The p_k 's are the source-specific dimensionalities and $p = \sum_{k=1}^K p_k$ is the total dimensionality. We consider a specific example of this kind below, in the context of Alzheimer's disease (AD).

Constructing regression models using such data is challenging, because the relevance of the sources may be quite unequal (and unknown in advance) and the total dimension p may be large. This motivates a need for methodology that can cope with multiple sources and that scales to high dimensions.

Methods for high-dimensional regression are now well established and include regularized least-squares approaches such as the lasso and extensions (Tibshirani 1996; Tibshirani et al. 2005; Yuan and Lin 2006), elastic net (Zou and Hastie 2005), SCAD (Fan and Li 2001), and Bayesian analogues (see Kyung et al. 2010, for a review). A range of Bayesian approaches have been proposed, notably those based on shrinkage priors, often coupled with variable selection (see, e.g., Yuan and Lin 2005; Park and Casella 2008; Hans 2010; Griffin and Brown 2010;

Carvalho, Polson, and Scott 2010; Armagan, Dunson, and Lee 2013, among others). However, in the very large p case many available methods become computationally cumbersome or intractable and effective source-specific shrinkage remains hard to achieve.

In this article, we put forward an approach to regression in the multiple-source, high-dimensional setting. Specifically:

- We consider a generalized ridge-type prior with shrinkage that adjusts to individual data sources, with the corresponding shrinkage levels estimated from the data.
- We show that estimation (including setting of tuning parameters) can be formulated in a way that renders computation efficient for “wide” data, even for very large p and over multiple sources.
- We introduce sparsifications that achieve competitive prediction performance and that provide a fast yet multivariate technique for discarding non-influential features.

Thus, we consider the case of data from multiple sources with source-specific dimensionalities p_k that could differ by many orders of magnitude, with total p large and a priori unknown source-specific importance. The main strength of our methods is their ability to use source-specific shrinkage to automatically adapt to signals spread across multiple sources.

There has been much interesting work on group selection approaches in regression (reviewed in Huang, Breheny, and Ma 2012). The group lasso (Yuan and Lin 2006) allows specification

CONTACT Konstantinos Perrakis  konstantinos.perrakis@dzne.de  Statistics and Machine Learning, German Center for Neurodegenerative Diseases, Venusberg-Campus 1, 53127, Bonn, Germany.

*Data used in preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. A complete listing of ADNI investigators can be found at: http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf.

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/r/JCGS.

 Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JCGS.

© 2019 American Statistical Association, Institute of Mathematical Statistics, and Interface Foundation of North America

of covariate subsets that can then be selected as groups; however, applying the group lasso in the current setting (by identifying groups with sources) would not be useful because sources would then simply be either included or excluded (without within-source regularization). The sparse group lasso (Simon et al. 2013) permits additional regularization via within-group sparsity but its use here would require a nontrivial extension to source-specific penalties whose tuning would be difficult if not intractable in the very high-dimensional, multiple source setting. Dondelinger, Mukherjee, et al. (2019) consider the case of penalized regression over multiple subgroups of samples; this is quite different from the present setting of sources of covariates (i.e., we focus on the columns, not the rows), also the authors do not tackle the very high-dimensional case.

Ridge-type estimators are among the oldest and best studied regularized regression tools, whether from a penalized likelihood or Bayesian viewpoint. Our results build on these classical tools, developing a variant of the ridge prior to deal with multiple-source, high-dimensional problems. The subsequent sparsification step that we consider is an example of an emerging class of posterior post-processing methods and yields a solution which is similar to the penalized credible region (pCR, Bondell and Reich 2012) and the decoupled shrinkage selection (DSS, Hahn and Carvalho 2015) solutions. In contrast to pCR and DSS we develop our approach via a certain Kullback–Leibler (KL) divergence. Interestingly, we can recover the pCR solution as a special case. Our approach can take advantage of the information from the initial ridge step and thus allows parameter-specific as well as source-specific penalization. In addition, we propose a relaxed variant which leads to a closed-form solution that is immediately applicable to problems involving millions of predictors. The primary motivation for this work is the need for efficient and interpretable predictive models in high-dimensional biomedical applications. We emphasize that the sparse extensions proposed are mainly aimed at achieving parsimonious prediction rather than variable selection per se. However, we also discuss some preliminary empirical results on variable selection using the class of methods proposed here.

As a topical example of a multiple-source, high-dimensional problem, we consider a case study in AD. AD is a neurodegenerative condition in which prediction of future disease course is a central research topic. AD is multifactorial in the sense of being mediated via multiple underlying biological processes and several current and emerging large-scale studies span multiple data types. These include the Alzheimer’s Disease Neuroimaging Initiative (ADNI) (Mueller et al. 2005), the Rhineland study (<http://www.rheinland-studie.de>), and the UK Biobank (<http://www.ukbiobank.ac.uk>) (this is a broader study including neurodegeneration-related data). The data we consider are from ADNI which is a large-scale longitudinal study involving multiple data modalities; we focus specifically on the prediction of future cognitive scores, as described in detail below.

The remainder of this article is organized as follows. In Section 2, we introduce the scalable Bayesian regression (SBR) approach, describing model formulation, prior specification, and tuning of shrinkage levels. Section 3 deals with the sparse extension of the methodology, sparse SBR (SSBR), including a general solution and a relaxed variant for the very large p case. The relationship between SSBR and pCR is discussed.

We further introduce an adaptive approach to regulate induced sparsity. Results and comparisons with standard penalized likelihood approaches from a simulation study are presented in Section 4, while the AD case study appears in Section 5. The article concludes with a discussion in Section 6.

2. Scalable Bayesian Regression

2.1. Model

Let \mathbf{y} be a $n \times 1$ vector of responses and $\mathbf{X}_1, \dots, \mathbf{X}_K$ denote covariate or feature matrices from K data sources. Each \mathbf{X}_k is of size $n \times p_k$ so that the total number of potential predictors is $p = \sum_{k=1}^K p_k$. We consider the normal linear model

$$\mathbf{y} = \mathbf{X}_1\boldsymbol{\beta}_1 + \mathbf{X}_2\boldsymbol{\beta}_2 + \dots + \mathbf{X}_K\boldsymbol{\beta}_K + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim N_n(\mathbf{0}, \mathbf{I}_n\sigma^2),$$

where each $\boldsymbol{\beta}_k$ is a p_k -vector of regression coefficients, $N_n(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes an n -dimensional multivariate normal density with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, and \mathbf{I}_n is the $n \times n$ identity matrix. Without loss of generality we assume throughout that all data are standardized. Let $\mathbf{X} = [\mathbf{X}_1 \dots \mathbf{X}_K]$ and $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^T, \dots, \boldsymbol{\beta}_K^T)^T$ denote the respective global $n \times p$ predictor matrix and p -vector of regression coefficients (here, “global” means with all sources taken together). Then, with prior π the full model under consideration is

$$\mathbf{y} \sim N_n(\mathbf{X}\boldsymbol{\beta}, \mathbf{I}_n\sigma^2), \quad \text{with } \boldsymbol{\beta}|\sigma^2 \sim \pi(\boldsymbol{\beta}|\sigma^2) \text{ and } \pi(\sigma^2) \propto 1/\sigma^2.$$

The improper prior for σ^2 (Jeffreys’ prior) is a common option for linear regression models. The crucial aspect of prior formulation for the multiple-source, high-dimensional setting under consideration is the construction of $\pi(\boldsymbol{\beta}|\sigma^2)$, as we discuss in detail below.

2.2. The Prior on $\boldsymbol{\beta}$

The SBR approach is based on a natural generalization of the standard ridge prior. Specifically, the prior on $\boldsymbol{\beta}$ is

$$\boldsymbol{\beta} \mid \boldsymbol{\lambda}, \sigma^2 \sim N_p(\mathbf{0}, \boldsymbol{\Lambda}^{-1}\sigma^2), \tag{1}$$

where $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_K)^T$, $\boldsymbol{\Lambda} = \text{diag}(\boldsymbol{\Lambda}_1, \dots, \boldsymbol{\Lambda}_K)$, and $\boldsymbol{\Lambda}_k = \lambda_k \mathbf{I}_{p_k}$ with $\lambda_k > 0$, for $k = 1, \dots, K$. Here each λ_k is a source-specific shrinkage level on the corresponding $\boldsymbol{\beta}_k$. The special case $K = 1$ recovers the standard ridge prior with just one shrinkage level (and indeed the solutions presented below could be used to give a scalable implementation of classical ridge with a single λ). However, when dealing with multiple data sources the various data sources may differ in importance. This motivates a need for source-specific penalties that can adjust to account for such differences and additionally provide potentially useful information about the relevance of specific data sources.

At this point it is useful to define the quantity

$$\mathbf{G}_\lambda \stackrel{\text{def}}{=} \sum_{k=1}^K \lambda_k^{-1} \mathbf{X}_k \mathbf{X}_k^T. \tag{2}$$

All formulas presented in the remainder of this section are cast in terms of \mathbf{G}_λ . Importantly, this means that the key computations under SBR can be formulated so as to require only a one-off

computation of these individual inner product (Gram) matrices $\mathbf{X}_k \mathbf{X}_k^T$ of size $n \times n$ (these calculations can be easily implemented in parallel) followed mainly by operations on those matrices. As we show below, for wide data with large p , this gives a practical way to implement SBR.

2.3. Inference

Under the conjugate prior in (1) the posterior distribution of $\boldsymbol{\beta}$ is given by

$$\boldsymbol{\beta} \mid \mathbf{y}, \boldsymbol{\lambda}, \sigma^2 \sim N_p(\hat{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_\beta \sigma^2), \quad (3)$$

where $\hat{\boldsymbol{\beta}} = \boldsymbol{\Sigma}_\beta \mathbf{X}^T \mathbf{y}$ and $\boldsymbol{\Sigma}_\beta = (\mathbf{X}^T \mathbf{X} + \boldsymbol{\Lambda})^{-1}$. Calculating the posterior mode directly involves a $p \times p$ matrix inversion. For $p > n$ we instead use

$$\hat{\boldsymbol{\beta}} = \boldsymbol{\Lambda}^{-1} \mathbf{X}^T \mathbf{w}_\lambda, \quad (4)$$

where $\mathbf{w}_\lambda = [\mathbf{y} - (\mathbf{I}_n + \mathbf{G}_\lambda)^{-1} \mathbf{G}_\lambda \mathbf{y}]$ is an n -vector whose calculation involves only an $n \times n$ matrix inversion. The derivation of (4) is provided in Appendix A of the supplementary materials. For very large problems the computation of the posterior mode can be done in parallel; additionally, we draw attention to the useful expression

$$\hat{\boldsymbol{\beta}}_k = \lambda_k^{-1} \mathbf{X}_k^T \mathbf{w}_\lambda, \quad (5)$$

for $k = 1, \dots, K$. Having obtained the posterior mode, prediction from an available \mathbf{X}^{pred} of dimensionality $m \times p$ is straightforward via $\mathbf{y}^{\text{pred}} = \mathbf{X}^{\text{pred}} \hat{\boldsymbol{\beta}}$. When interest lies solely in prediction the corresponding calculation can be simplified to

$$\mathbf{y}^{\text{pred}} = \mathbf{X}^{\text{pred}} \boldsymbol{\Lambda}^{-1} \mathbf{X}^T \mathbf{w}_\lambda = \left[\sum_{k=1}^K \lambda_k^{-1} \mathbf{X}_k^{\text{pred}} \mathbf{X}_k^T \right] \mathbf{w}_\lambda. \quad (6)$$

Calculating the posterior covariance matrix can also be simplified through the formula

$$\boldsymbol{\Sigma}_\beta = \boldsymbol{\Lambda}^{-1} [\mathbf{I}_p - \mathbf{X}^T (\mathbf{I}_n + \mathbf{G}_\lambda)^{-1} \mathbf{X} \boldsymbol{\Lambda}^{-1}]. \quad (7)$$

For details see Appendix A.

In practice we are not interested in evaluating the entire covariance matrix (for very large p this might in fact be difficult due to memory limitations). However, the methodology considered in Section 3.3 does require the diagonal elements of $\boldsymbol{\Sigma}_\beta$. In this case the formula in (7) facilitates computation as it allows for fast and parallel block matrix computations. Note that $\boldsymbol{\Sigma}_{\beta_k} = \lambda_k^{-1} [\mathbf{I}_{p_k} - \mathbf{X}_k^T (\mathbf{I}_n + \mathbf{G}_\lambda)^{-1} \mathbf{X}_k \lambda_k^{-1}]$, so here the magnitude of each p_k , for $k = 1, \dots, K$, can guide us in determining whether to use block computations or not. To clarify the use of parallel computations here, consider calculating the diagonal elements of a specific large $\boldsymbol{\Sigma}_{\beta_k}$ using B blocks. Consider $\mathbf{X}_k = [\mathbf{X}_k^{(1)} \cdots \mathbf{X}_k^{(B)}]$ where each $\mathbf{X}_k^{(b)}$ has $p_k^{(b)} = \lfloor p_k/B \rfloor$ columns for $b = 1, \dots, B-1$, while $p_k^{(B)} = p_k - (B-1)p_k^{(1)}$ for the last sub-matrix $\mathbf{X}_k^{(B)}$. Denote the variances by $\sigma_{\beta_k}^2 = \text{diag}(\boldsymbol{\Sigma}_{\beta_k})$. The calculation which is then performed in parallel is

$$\sigma_{\beta_k}^{2(b)} = \text{diag} \left(\lambda_k^{-1} \left[\mathbf{I}_{p_k^{(b)}} - \mathbf{X}_k^{(b)T} (\mathbf{I}_n + \mathbf{G}_\lambda)^{-1} \mathbf{X}_k^{(b)} \lambda_k^{-1} \right] \right), \quad (8)$$

for $b = 1, \dots, B$, with the output being the p_k -dimensional vector of variances; namely, $\sigma_{\beta_k}^2 = (\sigma_{\beta_k}^{2(1)}, \dots, \sigma_{\beta_k}^{2(B)})^T$.

Continuing, the posterior distribution of the error-variance parameter is an inverse-gamma distribution; specifically

$$\sigma^2 \mid \mathbf{y} \sim \text{IG}(a, b), \text{ with shape } a = \frac{n}{2} \text{ and scale } b = \frac{\mathbf{y}^T (\mathbf{I}_n + \mathbf{G}_\lambda)^{-1} \mathbf{y}}{2}. \quad (9)$$

The numerator of the scale parameter in (9), whose calculation is again simplified via \mathbf{G}_λ , is important because it will appear throughout; for its derivation see Appendix A in the supplementary materials.

Below we will make use of the marginal likelihood $m(\mathbf{y} \mid \boldsymbol{\lambda})$. This is the likelihood obtained by integrating over the parameter space with respect to the joint prior distribution of $\boldsymbol{\beta}$ and σ^2 . Under our conjugate prior specification we have

$$m(\mathbf{y} \mid \boldsymbol{\lambda}) \propto \int p(\mathbf{y} \mid \boldsymbol{\beta}, \sigma^2) \pi(\boldsymbol{\beta} \mid \boldsymbol{\lambda}, \sigma^2) \frac{1}{\sigma^2} d\boldsymbol{\beta} d\sigma^2 \propto |\mathbf{I}_n + \mathbf{G}_\lambda|^{-\frac{1}{2}} [\mathbf{y}^T (\mathbf{I}_n + \mathbf{G}_\lambda)^{-1} \mathbf{y}]^{-\frac{n}{2}}. \quad (10)$$

2.4. Automatic Setting of Shrinkage Levels $\boldsymbol{\lambda}$

Specification of penalty parameters is often handled through cross-validation (CV) or generalized CV in a frequentist framework (Tibshirani 1996), while Bayesian methods typically rely on empirical Bayes (EB) point estimates or data-dependent hyper-priors (see, e.g., Yuan and Lin 2005; Park and Casella 2008; Balakrishnan and Madigan 2010; Hans 2010; Griffin and Brown 2010). An alternative approach is considered by Lykou and Ntzoufras (2013) who tune the Bayesian lasso penalty based on Pearson correlations at the limit of significance determined by Bayes factors. Furthermore, fully Bayesian shrinkage methods include the horseshoe prior (Carvalho, Polson, and Scott 2010) and the double generalized Pareto (Armagan, Dunson, and Lee 2013).

In our case, the tuning parameter $\boldsymbol{\lambda}$ is vector valued and for the applications we consider we would like fast and efficient approaches by which to set it. To this end we propose three EB approaches for tuning $\boldsymbol{\lambda}$. Here, by EB we refer generically to any procedure that uses the data to estimate hyper parameters. We consider three specific estimators based on (i) minimizing the leave-one-out CV error, (ii) maximizing the marginal likelihood, and (iii) locating the maximum-a-posteriori (MAP) value under a data-dependent prior. All three are free of user input and are computationally fast. We discuss each in turn.

2.4.1. Leave-One-Out Cross-Validation (CV) Estimator

The leave-one-out CV error in our case can be computed as

$$\hat{\boldsymbol{\lambda}}_{\text{CV}} = \underset{\boldsymbol{\lambda}}{\text{argmin}} \mathbf{y}^T (\mathbf{I}_n + \mathbf{G}_\lambda)^{-1} [\text{diag}(\mathbf{I}_n + \mathbf{G}_\lambda)^{-1}]^{-2} (\mathbf{I}_n + \mathbf{G}_\lambda)^{-1} \mathbf{y}. \quad (11)$$

This is similar to the well-known case of ordinary least squares; for proof see supplementary materials, Appendix B.

2.4.2. Marginal Likelihood (ML) Estimator

A common EB approach is to use the marginal likelihood; in our case from Equation (10) the quantity to be maximized is

$$\widehat{\lambda}_{\text{ML}} = \underset{\lambda}{\operatorname{argmax}} |\mathbf{I}_n + \mathbf{G}_\lambda|^{-\frac{1}{2}} [\mathbf{y}^T (\mathbf{I}_n + \mathbf{G}_\lambda)^{-1} \mathbf{y}]^{-\frac{n}{2}}. \quad (12)$$

2.4.3. Maximum-a-Posteriori (MAP) Estimator

We consider a product-exponential data-dependent prior for λ with prior mode at zero, prior mean equal to $\widehat{\lambda}_{\text{CV}}$ as given in (11), and prior variance $\widehat{\lambda}_{\text{CV}}^2$, that is, $\pi(\lambda) \equiv \prod_{k=1}^K \operatorname{Exp}(\widehat{\lambda}_{k\text{CV}}^{-1})$. The rationale is that a smaller individual estimated penalty $\widehat{\lambda}_{k\text{CV}}$ corresponds to a stronger belief that the corresponding \mathbf{X}_k matrix contains useful signal and therefore to a smaller prior variance (especially when $\widehat{\lambda}_{k\text{CV}} < 1$). On the other hand as $\widehat{\lambda}_{k\text{CV}}$ increases we let the quadratic prior variance account for the chance that there is actually some useful signal in \mathbf{X}_k which passes undetected by the leave-one-out CV approach. The resulting posterior mode estimate is

$$\begin{aligned} \widehat{\lambda}_{\text{MAP}} = \underset{\lambda}{\operatorname{argmax}} & |\mathbf{I}_n + \mathbf{G}_\lambda|^{-\frac{1}{2}} [\mathbf{y}^T (\mathbf{I}_n + \mathbf{G}_\lambda)^{-1} \mathbf{y}]^{-\frac{n}{2}} \\ & \times \exp\left(-\sum_{k=1}^K \frac{\lambda_k}{\widehat{\lambda}_{k\text{CV}}}\right). \end{aligned} \quad (13)$$

The optimization problems in Equations (11)–(13) are typically well behaved, with the shape of the optimization landscape depending on relative signal strength. For non-informative sources, it is relevant to note that the magnitude of λ_k does not affect the ridge solution after certain large values (as all coefficients are shrunk to zero); hence, it is reasonable to simply define a large upper bound for the optimization. A key point is that the number of available data sources K will typically not be large, hence the vector λ is low dimensional, allowing the optimizations to be efficiently solved via standard routines.

3. Sparse SBR

The SBR posterior mode in (4) is non-sparse (“dense”) in the sense that the regression coefficients will not be set to exactly zero. In this section, we propose a methodology for “sparsifying” SBR. The idea is to find a sparse approximation to the full (dense) Bayesian solution that is closest to it in a KL sense. To do so, we minimize the KL divergence with respect to the posterior distribution of the regression vector, but subject to a lasso-type ℓ_1 constraint to ensure sparsity. We show first a general solution that is suitable for small to moderate p and then go on to consider a relaxed solution that is applicable to the large p case. The solutions presented below bear a resemblance to other Bayesian post-processing approaches (Bondell and Reich 2012; Hahn and Carvalho 2015) and to frequentist methods in the context of wavelet regression (Antoniadis and Fan 2001) and ridge logistic regression for text categorization (Aseervatham et al. 2011). However, these are rooted in different arguments and not equivalent to the KL-based approach below. The particular connection with the pCR solution of Bondell and Reich (2012) is discussed in Section 3.2.

3.1. Sparsification Using the KL Divergence

Let $f(\boldsymbol{\beta}) \equiv N_p(\widehat{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_\beta \sigma^2)$ denote the true posterior over $\boldsymbol{\beta}$, conditional on σ^2 , with mode and covariance as in Equations (4) and (7), respectively, and let $q(\boldsymbol{\beta}) \equiv N_p(\boldsymbol{\gamma}, \boldsymbol{\Sigma}_\beta \sigma^2)$ denote an approximate conditional posterior where $\boldsymbol{\gamma}$ is the approximate mode (this will provide a sparsification of $\widehat{\boldsymbol{\beta}}$). The idea is to minimize the KL divergence from q to f under an ℓ_1 penalty on vector $\boldsymbol{\gamma}$ to induce sparsity. It is easy to show that the KL divergence from $q(\boldsymbol{\beta})$ to $f(\boldsymbol{\beta})$ is

$$D_{\text{KL}}(f||q) = \frac{1}{2\sigma^2} (\widehat{\boldsymbol{\beta}} - \boldsymbol{\gamma})^T \boldsymbol{\Sigma}_\beta^{-1} (\widehat{\boldsymbol{\beta}} - \boldsymbol{\gamma}). \quad (14)$$

Note that D_{KL} in (14) is a true distance metric (satisfying non-negativity, symmetry, and the triangle inequality). Note also that the presence of the nuisance parameter σ^2 cannot be ignored when the minimization also involves a ℓ_1 penalty on $\boldsymbol{\gamma}$. In principle, one could work with the marginal posterior distribution of $\boldsymbol{\beta}$ (a multivariate t distribution) to avoid consideration of σ^2 . However, in this case working with the KL divergence is not straightforward. Another option would be to use a plug-in posterior point estimate in (14) such as the mode or mean of σ^2 . Instead, here we pursue a tuning-free approach in which σ^2 is integrated out; specifically, we work with

$$\begin{aligned} \mathbb{E}_{\sigma^2|\mathbf{y}} [D_{\text{KL}}(f||q)] &= \int D_{\text{KL}}(f||q) p(\sigma^2|\mathbf{y}) d\sigma^2 \\ &= \frac{c_{n,\lambda}}{2} (\widehat{\boldsymbol{\beta}} - \boldsymbol{\gamma})^T \boldsymbol{\Sigma}_\beta^{-1} (\widehat{\boldsymbol{\beta}} - \boldsymbol{\gamma}), \end{aligned} \quad (15)$$

with the posterior of σ^2 given in (9) and $c_{n,\lambda} = n q_\lambda^{-1}$, where $q_\lambda = \mathbf{y}^T (\mathbf{I}_n + \mathbf{G}_\lambda)^{-1} \mathbf{y}$. Using the posterior mean or mode of σ^2 results in $c_{n,\lambda} = (n-2)q_\lambda^{-1}$ and $c_{n,\lambda} = (n+2)q_\lambda^{-1}$, respectively. Using (15), the general solution including the ℓ_1 penalty is

$$\widehat{\boldsymbol{\gamma}} = \underset{\boldsymbol{\gamma}}{\operatorname{argmin}} \frac{c_{n,\lambda}}{2} (\widehat{\boldsymbol{\beta}} - \boldsymbol{\gamma})^T \boldsymbol{\Sigma}_\beta^{-1} (\widehat{\boldsymbol{\beta}} - \boldsymbol{\gamma}) + \alpha \|\boldsymbol{\gamma}\|_1, \quad (16)$$

where $\alpha > 0$ controls the sparsity of $\widehat{\boldsymbol{\gamma}}$. Clearly, the SSBR solution implies a lasso-type model with the particularity of a saturated design where the analogue to sample size equals p . This means that SSBR can include at most p predictors (unlike classical lasso which for $p > n$ can include at most n predictors). The minimization in (16) can be solved as a lasso problem by setting $\mathbf{y}^* = \sqrt{c_{n,\lambda}} \boldsymbol{\Sigma}_\beta^{-1/2} \widehat{\boldsymbol{\beta}}$ as response variable and $\mathbf{X}^* = \sqrt{c_{n,\lambda}} \boldsymbol{\Sigma}_\beta^{-1/2}$ as design matrix, and using the efficient implementation of `glmnet` in R (R Core Team 2019). Note, however, that this involves first calculating and then performing computations with the inverse covariance matrix, which can be problematic even for moderately large p .

As an alternative, we present a mathematically equivalent representation of (16) that is easier to work with. First, observe that the posterior covariance in (7) can be written as $\boldsymbol{\Sigma}_\beta = \boldsymbol{\Lambda}^{-1/2} (\mathbf{I}_p - \mathbf{M}^T \mathbf{M}) \boldsymbol{\Lambda}^{-1/2}$, where $\mathbf{M} = (\mathbf{I}_n + \mathbf{G}_\lambda)^{-1/2} \mathbf{X} \boldsymbol{\Lambda}^{-1/2}$. Now, consider the singular value decomposition (SVD) $\mathbf{M} = \mathbf{U} \mathbf{D} \mathbf{V}^T$, where $\mathbf{U} \in \mathbb{R}^{n \times n}$, $\mathbf{D} \in \mathbb{R}^{n \times p}$ and $\mathbf{V} = [\mathbf{V}_1 \mathbf{V}_2]$ with $\mathbf{V}_1 \in \mathbb{R}^{p \times n}$ and $\mathbf{V}_2 \in \mathbb{R}^{p \times (p-n)}$. Note that computation of $\mathbf{U}, \mathbf{D}, \mathbf{V}_1$ is $\mathcal{O}(n^2 p)$ and also that the last $p-n$ columns of \mathbf{D} contain only zeros. In addition, the decomposition of \mathbf{M} can be greatly sped up by first computing the SVD of $\mathbf{M} \mathbf{M}^T$. Some

algebra (details provided in Appendix C of the supplementary materials) reveals that the resulting equivalent SSBR solution requires as additional input only the n singular values of \mathbf{M} and matrix \mathbf{V}_1 ; specifically, the solution is

$$\begin{aligned} \hat{\boldsymbol{\gamma}} &= \underset{\boldsymbol{\gamma}}{\operatorname{argmin}} \frac{c_{n,\lambda}}{2} \|\tilde{\mathbf{D}}^{1/2} \mathbf{V}_1^T \boldsymbol{\Lambda}^{1/2} (\hat{\boldsymbol{\beta}} - \boldsymbol{\gamma})\|_2^2 \\ &\quad + \frac{c_{n,\lambda}}{2} \|\boldsymbol{\Lambda}^{1/2} (\hat{\boldsymbol{\beta}} - \boldsymbol{\gamma})\|_2^2 + \alpha \|\boldsymbol{\gamma}\|_1, \end{aligned} \quad (17)$$

where $\tilde{\mathbf{D}}$ is $n \times n$ diagonal with elements $\tilde{d}_i = d_i^2 / (1 - d_i^2)$ and d_i are the singular values. In practice (17) can be solved as an augmented-lasso problem, using

$$\begin{aligned} \mathbf{y}^* &= \sqrt{c_{n,\lambda}} \begin{pmatrix} \tilde{\mathbf{D}}^{1/2} \mathbf{V}_1^T \boldsymbol{\Lambda}^{1/2} \hat{\boldsymbol{\beta}} \\ \boldsymbol{\Lambda}^{1/2} \hat{\boldsymbol{\beta}} \end{pmatrix} \quad \text{and} \\ \mathbf{X}^* &= \sqrt{c_{n,\lambda}} \begin{bmatrix} \tilde{\mathbf{D}}^{1/2} \mathbf{V}_1^T \boldsymbol{\Lambda}^{1/2} \\ \boldsymbol{\Lambda}^{1/2} \end{bmatrix}, \end{aligned}$$

which have respective dimensionality $(n+p) \times 1$ and $(n+p) \times p$. The matrix multiplication involves mainly sparse matrices, as $\tilde{\mathbf{D}}$ and $\boldsymbol{\Lambda}$ are diagonal. An interesting side note is that such SVDs can be generally used in $p > n$ problems for covariances of the form $(\mathbf{X}^T \mathbf{X} + \mathbf{Z})^{-1}$ for some diagonal matrix \mathbf{Z} . Thus, using (17) instead of (16) is advocated when p is large. However, for applications with ultra-large p the approach presented in Section 3.3, which leads to a closed-form expression for $\hat{\boldsymbol{\gamma}}$, is the only practical option.

3.2. Relation to Penalized Credible Regions

The pCR approach (Bondell and Reich 2012) seeks solutions of the form $\hat{\boldsymbol{\beta}} = \operatorname{argmin}_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_0$ subject to $\boldsymbol{\beta} \in C_\alpha$, where C_α is the $(1 - \alpha) \times 100\%$ credible set. Under a normal ridge-prior on $\boldsymbol{\beta}$ and an inverse-gamma prior on σ^2 this translates to a feasible set of the form $\{\boldsymbol{\beta} : (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^T \boldsymbol{\Sigma}_\beta^{-1} (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}) \leq C\}$ for some C corresponding to a specific credible region.

To tackle the obvious computational challenges of the above solution, the authors initially relax the ℓ_0 norm to a smooth homotopy between ℓ_0 and ℓ_1 , and subsequently apply a local linear approximation which results in a convex ℓ_1 optimization problem. The resulting solution is very similar to our solution obtained through the KL approach; in fact by setting the penalty in (16) as $\alpha = \frac{c_{n,\lambda}}{2} \|\hat{\boldsymbol{\beta}}\|_1^{-2} \xi$ (now $c_{n,\lambda}/2$ no longer affects the optimization and ξ is the new penalty) we recover exactly the pCR solution. As noted in Bondell and Reich (2012) there is a one-to-one correspondence between C and ξ ; however, it is highly nonlinear. Under this setting we have selection consistency (under mild regularity conditions) when p is fixed or as long as $p/n \rightarrow 0$ for $n \rightarrow \infty$. The authors also demonstrate selection consistency (under stricter conditions) for rates $\log p = \mathcal{O}(n^c)$ for some $c \in (0, 1)$ using univariate thresholding rules on simple ridge estimates.

In practice, the ξ that corresponds to the asymptotically consistent sparse set is not recoverable. Theoretically, the penalty should depend on sample size so that $\xi_n \rightarrow 0$ faster than the posterior distribution concentrates around the ‘‘true’’ parameter value; however, under finite samples the selection of ξ crucially affects the sparsity of $\boldsymbol{\beta}$ (Hahn and Carvalho 2015). Therefore,

tuning of ξ is handled through common grid search and inspection of regularization plots/prediction errors in Bondell and Reich (2012).

3.3. A Relaxed Solution for the Very Large- p Case

Instead of the KL divergence to the posterior used above, consider the KL divergence between the quantities $q^*(\boldsymbol{\beta}) = \mathcal{N}_p(\boldsymbol{\gamma}, \mathbf{V}_\beta \sigma^2)$ and $f^*(\boldsymbol{\beta}) = \mathcal{N}_p(\hat{\boldsymbol{\beta}}, \mathbf{V}_\beta \sigma^2)$, with $\mathbf{V}_\beta = \operatorname{diag}(\boldsymbol{\Sigma}_\beta)$. This amounts to setting as target distribution the product of the marginal posterior densities. The use of independent posterior factorizations is common in various settings; for instance, in marginal likelihood estimation (Botev, L’Ecuyer, and Tuffin 2013; Perrakis, Ntzoufras, and Tsionas 2014), in expectation-propagation algorithms (Minka 2001), and in variational Bayes (Bishop 2006).

Working with the diagonal matrix \mathbf{V}_β leads to the following minimization

$$\begin{aligned} \hat{\boldsymbol{\gamma}} &= \underset{\boldsymbol{\gamma}}{\operatorname{argmin}} \frac{c_{n,\lambda}}{2} (\hat{\boldsymbol{\beta}} - \boldsymbol{\gamma})^T \mathbf{V}_\beta^{-1} (\hat{\boldsymbol{\beta}} - \boldsymbol{\gamma}) + \alpha \|\boldsymbol{\gamma}\|_1 \\ &= \underset{\boldsymbol{\gamma}}{\operatorname{argmin}} \sum_{j=1}^p \frac{c_{n,\lambda}}{2} (\hat{\beta}_j - \gamma_j)^2 v_j^{-1} + \alpha |\gamma_j|, \end{aligned} \quad (18)$$

where v_j is the j th element, for $j = 1, \dots, p$, of the main diagonal of \mathbf{V}_β . Note that the main diagonal elements are feasible to calculate even for very large p ; this can be achieved by calculating Equation (8) in parallel. Moreover, the minimization in (18) has a closed-form solution which is as follows

$$\hat{\gamma}_j = \begin{cases} \hat{\beta}_j - \operatorname{sign}(\hat{\beta}_j) \frac{q_\lambda}{n} v_j \alpha, & \text{if } |\hat{\beta}_j| > \frac{q_\lambda}{n} v_j \alpha \\ 0, & \text{otherwise.} \end{cases} \quad (19)$$

The derivation of (19) is provided as supplementary materials (Appendix D).

Note that for fixed p and $n \rightarrow \infty$ we obtain $\hat{\gamma}_j = \hat{\beta}_j$ which makes sense from an asymptotic perspective. However, when α is a constant not depending on n either directly or indirectly (e.g., through $\hat{\beta}_j$), a ‘‘non-sparsifying’’ effect may be triggered even for moderate sample size, which is in contrast to our initial intent. Setting of α is discussed next.

3.4. Tuning of α

Specification of α can be handled via a grid search with the aim to find the α that minimizes a specific criterion. This is expected to be relatively fast using the general solution for small/moderate p , while under the relaxed solution, once the variances are calculated, the grid search requires only checking a true/false statement. This strategy will typically produce a full path of solutions which can also be used to produce regularization plots. We acknowledge this as a valid common strategy, however, we do not further pursue it here. Instead, we consider a faster, tuning-free, alternative which borrows information from the SBR solutions.

We consider parameter-specific and source-specific adaptive penalties for each $\hat{\gamma}_{jk}$, where $j = 1, \dots, p_k$ and $k = 1, \dots, K$. Specifically, we consider penalties of the form

$$\alpha_{jk} = \left(\frac{1}{|\hat{\beta}_{jk}|} \right)^{w_k}, \quad (20)$$

Table 1. Overview of methods, characteristics, issues, and solutions.

| Method | Characteristics | Issues | Proposed solutions |
|--------------|-------------------------------------------------------------------------|-----------------------------------------------------|-------------------------------------------------------------------------------------------------|
| SBR | Scalable to very large p Very fast Dense solutions | Tuning of λ | CV estimator, Equation (11) ML estimator, Equation (12) MAP estimator, Equation (13) |
| SSBR | Scalable to moderately large p Relatively fast Sparse solutions | Tuning of α | Tune via Equations (20) and (21) |
| Relaxed SSBR | Scalable to very large p Fast Sparse solutions | (1) Tuning of α (2) Control effect of n | (1) Tune via Equations (20) and (21) (2) $f_n = 1$ (no control) or $f_n = \log n$ (cSSBR) |

similar to the adaptive lasso approach (Zou 2006). The rationale in (20) is that the larger the magnitude of $\widehat{\beta}_{jk}$, the smaller the corresponding penalty. In addition, we restrict to $w_j \in (0, 1)$ which leads to reasonable shrinkage when $|\widehat{\beta}_{jk}| > 1$ and avoids extreme shrinkage when $|\widehat{\beta}_{jk}| < 1$. Here, we consider one specific possibility, namely to treat the w_k 's as power-weights, setting them equal to

$$w_k = \frac{\widehat{\lambda}_k}{\sum_{l=1}^K \widehat{\lambda}_l}. \tag{21}$$

The power weight quantifies the ‘‘importance’’ of a data source in relation to the others. Values close to zero and one indicate sources of ‘‘high’’ and ‘‘low’’ importance, respectively. Coefficients with an absolute value smaller than one (the common case) are penalized more in low-importance sources, while shrinkage on large coefficients (greater than one in absolute value) is relatively mild, and at most approximately equal to one (when $w_k \rightarrow 0$). With this approach we take advantage of the available information from the previous SBR step, that is, parameter-specific shrinkage through $\widehat{\beta}_{jk}$ and source-specific shrinkage through $\widehat{\lambda}_k$. Arguably, this strategy may result in undesirable non-sparse solutions, but that will be in the rare, and rather unrealistic, case where K is large and all sources are equally important in the sense that the $\widehat{\lambda}_k$'s will be more or less the same; a setting where in fact a single λ SBR approach is more suitable.

Note that, to implement this approach under the general solution in (16) we find first \boldsymbol{y}^* via `glmnet` (with penalty set to one) using as design matrix $\mathbf{X}^* = \sqrt{c_{n,\lambda}} \boldsymbol{\Sigma}_\beta^{-1/2} \mathbf{A}$, where \mathbf{A} is the diagonal matrix with elements the reciprocals of (20). The solution is then $\widehat{\boldsymbol{y}} = \mathbf{A}\boldsymbol{y}^*$.

As a final comment, we remark that despite the fact that this penalization approach depends indirectly on sample size through the regression coefficients in (20) and the shrinkage parameter in (21), it may still be sensitive to the ‘‘non-sparsifying’’ effect on the relaxed SSBR solution discussed at the end of Section 3.3. Controlling this effect requires scaling the penalty in (20) by a factor $f_n = f(n)$; however, automatic tuning of f_n is not straightforward. Empirical results (see Section 4) suggest that $f_n = \log(n)$ can lead to a reasonable balance between sparsity and predictive performance. We will call this relaxed extension ‘‘controlled’’ SSBR or cSSBR (as it ‘‘controls’’ for sample size). Table 1 provides an overview of the different methods under consideration, their characteristics, the issues that arise under each approach, and our proposed solutions.

4. Simulation Study

In this section, we present a simulation study aimed at mimicking a typical modern biomedical application involving multiple data types. Reflecting the relative ease with which multiple data modalities can now be acquired such designs are becoming common, with examples including the Cancer Genome Atlas (<https://cancergenome.nih.gov>), the ADNI (<http://adni.loni.usc.edu>), and the Rhineland Study (<http://www.rheinland-studie.de>), among many others.

4.1. Setup

4.1.1. The Problem

We consider a regression problem with covariates from three sources, namely clinical (CL), gene-expression (RNA), and genetic (single nucleotide polymorphism or SNP) data with respective (simulated) feature matrices \mathbf{X}_{CL} , \mathbf{X}_{RNA} , and \mathbf{X}_{SNP} . The number of covariates in each data source is set equal to $p_{\text{CL}} = 26$, $p_{\text{RNA}} = 2000$ and $p_{\text{SNP}} = 100,000$. Although the methods we propose can cope with larger p , we restrict total p in this section to facilitate empirical comparison with standard methods.

4.1.2. Covariates

The covariate matrices for the clinical and gene-expression variables are generated as $\mathbf{X}_{\text{CL}} \sim N_{p_{\text{CL}}}(\mathbf{0}, \boldsymbol{\Sigma}_{\text{CL}})$ and $\mathbf{X}_{\text{RNA}} \sim N_{p_{\text{RNA}}}(\mathbf{0}, \boldsymbol{\Sigma}_{\text{RNA}})$, respectively. Here $\boldsymbol{\Sigma}_{\text{CL}}$ and $\boldsymbol{\Sigma}_{\text{RNA}}$ are covariance matrices estimated from (real) phenotype and gene-expression data from the *Drosophila* Genetic Reference Panel (DGRP) (Mackay et al. 2012) (data available online at <http://dgrp2.gnets.ncsu.edu/data.html>). To simulate the genetic data \mathbf{X}_{SNP} we use a block-diagonal covariance structure. We specify $\boldsymbol{\Sigma}_{\text{SNP}} = \text{diag}(\boldsymbol{\Sigma}_{\text{SNP}}^1, \dots, \boldsymbol{\Sigma}_{\text{SNP}}^B)$, where each $\boldsymbol{\Sigma}_{\text{SNP}}^b$ is of size $S \times S$ (with $S = p_{\text{SNP}}/B$) and is generated from an inverse-Wishart with S degrees of freedom and identity scale matrix, that is, $\boldsymbol{\Sigma}_{\text{SNP}}^b \sim \text{IW}(S, \mathbf{I}_S)$ for $b = 1, \dots, B$. As \mathbf{X}_{SNP} dominates in terms of dimensionality the specification of B essentially controls the overall correlation level. We consider two simulation scenarios: (i) $B = 1000$ corresponding to 1000 blocks of size 100 (‘‘low-correlation scenario’’) and (ii) $B = 100$ corresponding to 100 blocks of size 1000 (‘‘high-correlation scenario’’). We first generate $\mathbf{X}_{\text{SNP}}^c \sim N_{p_{\text{SNP}}}(\mathbf{0}, \boldsymbol{\Sigma}_{\text{SNP}})$ and then discretize in correspondence to the common SNP encoding 0/1/2 (homozygous major allele/heterozygous/homozygous minor allele). The discretization is tuned to give a reasonable empirical distribution of SNPs; specifically, for $j = 1, \dots, p_{\text{SNP}}$

we discretize as

$$X_{j_{\text{SNP}}} = \begin{cases} 0, & \text{if } |X_{j_{\text{SNP}}}^c| < 1.5, \\ 1, & \text{if } |X_{j_{\text{SNP}}}^c| \geq 1.5 \text{ and } |X_{j_{\text{SNP}}}^c| < 2.5, \\ 2, & \text{if } |X_{j_{\text{SNP}}}^c| \geq 2.5. \end{cases}$$

4.1.3. Regression Coefficients and Sparsity

For the (true) regression vectors β_{CL} , β_{RNA} , and β_{SNP} , we consider the following levels of sparsity (fraction of nonzero β 's); $s_{\text{CL}} = 50\%$, $s_{\text{RNA}} = 5\%$, and $s_{\text{SNP}} \in \{1\%, 10\%, 50\%\}$. Varying sparsity of β_{SNP} gives three scenarios for overall sparsity s : (i) $s \approx 1\%$ (sparse scenario), (ii) $s \approx 10\%$ (medium scenario), and (iii) $s \approx 50\%$ (dense scenario). Let p_{CL}^* , p_{RNA}^* , and p_{SNP}^* denote the respective number of elements in the sub-vectors β_{CL}^* , β_{RNA}^* , and β_{SNP}^* containing the nonzero beta coefficients. The nonzero betas are generated from the generalized normal distribution (GND). Following the parameterization in Mineo (2003) the probability distribution function of a GND(μ, σ, u) with location $\mu \in \mathbb{R}$, scale $\sigma > 0$, and shape $u > 0$ is given by

$$f(x) = \frac{1}{2u^{1/u}\sigma\Gamma(1+1/u)} \exp\left(-\frac{|x-\mu|^u}{u\sigma^u}\right).$$

The GND includes as special cases the normal ($u = 2$) and the double exponential ($u = 1$) distributions. To avoid these particular cases (which could potentially bias the simulation toward ridge or lasso, respectively) we set $u = 1.5$ and generate the nonzero effects as $\beta_{j_{\text{CL}}}^* \sim \text{GND}(0, \hat{\sigma}, 1.5)$, for $j = 1, \dots, p_{\text{CL}}^*$, $\beta_{j_{\text{RNA}}}^* \sim \text{GND}(0, \hat{\sigma}, 1.5)$, for $j = 1, \dots, p_{\text{RNA}}^*$, and $\beta_{j_{\text{SNP}}}^* \sim \text{GND}(0, 2\hat{\sigma}/3, 1.5)$, for $j = 1, \dots, p_{\text{SNP}}^*$. The signal strength is controlled via the scale parameter $\hat{\sigma}$ (this is downscaled by a factor of 1.5 for the SNP coefficients to control the total amount of signal in the SNPs). To complete the specification of the simulation we set this scale parameter by considering the finite-sample risk in a simplified CL-only oracle-like setup. Specifically, we consider the correlation induced between predictions $\mathbf{X}_{\text{CL}}^{\text{test}} \hat{\beta}_{\text{CL}}$ (under the OLS estimate using the low-dimensional CL data only) and out-of-sample test data (with the data-generating mechanism being a linear model with conditional mean $\mathbf{X}_{\text{CL}} \beta_{\text{CL}}$ and error variance equal to unity). Specifically, we set $\hat{\sigma} = 0.1$ which results in an average out-of-sample correlation of 0.6 when $n = 100$ and $n_{\text{test}} = 5000$.

Given the above configurations (low/high correlation and sparse/medium/dense scenarios) we generate data from the model

$$y_i = \mathbf{x}_{i_{\text{CL}}}^T \beta_{\text{CL}} + \mathbf{x}_{i_{\text{RNA}}}^T \beta_{\text{RNA}} + \mathbf{x}_{i_{\text{SNP}}}^T \beta_{\text{SNP}} + \varepsilon_i,$$

where $\varepsilon_i \sim N(0, 1)$ and $i = 1, \dots, n_{\text{train}}$ with $n_{\text{train}} \in \{100, 250, 500\}$. The test sample size n_{test} always equals 5000. Each simulation scenario is repeated 50 times.

4.1.4. Methods Under Comparison

We consider SBR and the corresponding sparse extensions (unless otherwise noted, SSBR/cSSBR will refer to the relaxed approach discussed in Section 3.3). Specifically, we consider SSBR with the penalty terms in (20) and the power-weights in (21), as well as cSSBR approaches with $f_n^{(1)} = \sqrt{n}$, $f_n^{(2)} = \log(n)$, and $f_n^{(3)} = \sqrt{\log(n)}$. Results reveal no significant differences

with respect to the EB λ estimates (proposed in Section 2.4) under SBR. However, the MAP estimator resulted in better performance under the sparse approaches and, therefore, here we focus on this approach. Corresponding results under the CV and ML approaches are provided in Appendix E. Furthermore, we present results obtained from $f_n^{(2)}$ as this option led to a good balance between sparsity and predictive performance. We compare to standard ridge, elastic net (enet), and lasso (with λ set to minimize the mean squared error from 5-fold CV using package `glmnet` in R). The enet control parameter α was tuned via grid search over the interval $[0.1, 0.9]$ using a step of 0.1.

4.2. Results

Boxplots of out-of-sample correlations between predictions and test data under the low-correlation and high-correlation simulations are presented in Figure 1. Results for (classical) ridge and lasso are as expected; lasso clearly performs better in the sparse case, while ridge does better in the medium and dense cases. Also, more or less as expected enet performs as well as lasso in the sparse scenario and slightly better than lasso in the medium and dense scenarios; although in the latter two cases it is outperformed by ridge.

SBR performs generally well. Specifically, we see that:

- In the sparse scenario SBR is nearly equivalent to lasso/enet.
- In the medium scenario SBR generally outperforms ridge, lasso, and enet.
- In the dense scenario SBR is nearly equivalent to ridge.

For SSBR approaches we see:

- Under low correlations SSBR/cSSBR are slightly worse than SBR but competitive.
- Under high correlations in the medium/dense cases with large n SSBR/cSSBR are competitive to SBR.

Figure 2 shows the resulting values of $-\log \hat{\lambda}_k$ for $k = \{\text{CL}, \text{RNA}, \text{SNP}\}$ (higher values correspond to lower penalty; i.e., higher estimated importance) in the low-correlation simulations, and provides useful insights concerning the behavior of SBR methods. The estimates appear to adjust well with appropriate source-specific penalization. This adaptation allows SBR to perform well when dealing with multiple data sources. The corresponding plots from the high-correlation scenarios (not shown) are very similar.

As noted above, the SSBR (without or with control) solutions seem to allow equally good predictive performance, in certain cases, as the dense SBR solutions. In addition, they employ fewer parameters; Table 2 shows the average sparsity (over the 50 repetitions) induced by SSBR methods under the various simulations. As seen, the solutions appear to adjust to the true underlying sparsity. In addition, controlling for the effect of sample size yields much sparser models. In contrast, lasso and elastic net produced very sparse models that showed no such adaptation; see Appendix E of the supplementary materials for detailed results. We note that lasso can include at most n predictors and here yielded extremely sparse solutions (ranging from 0.008% to 0.45%). Elastic net can in principle include more

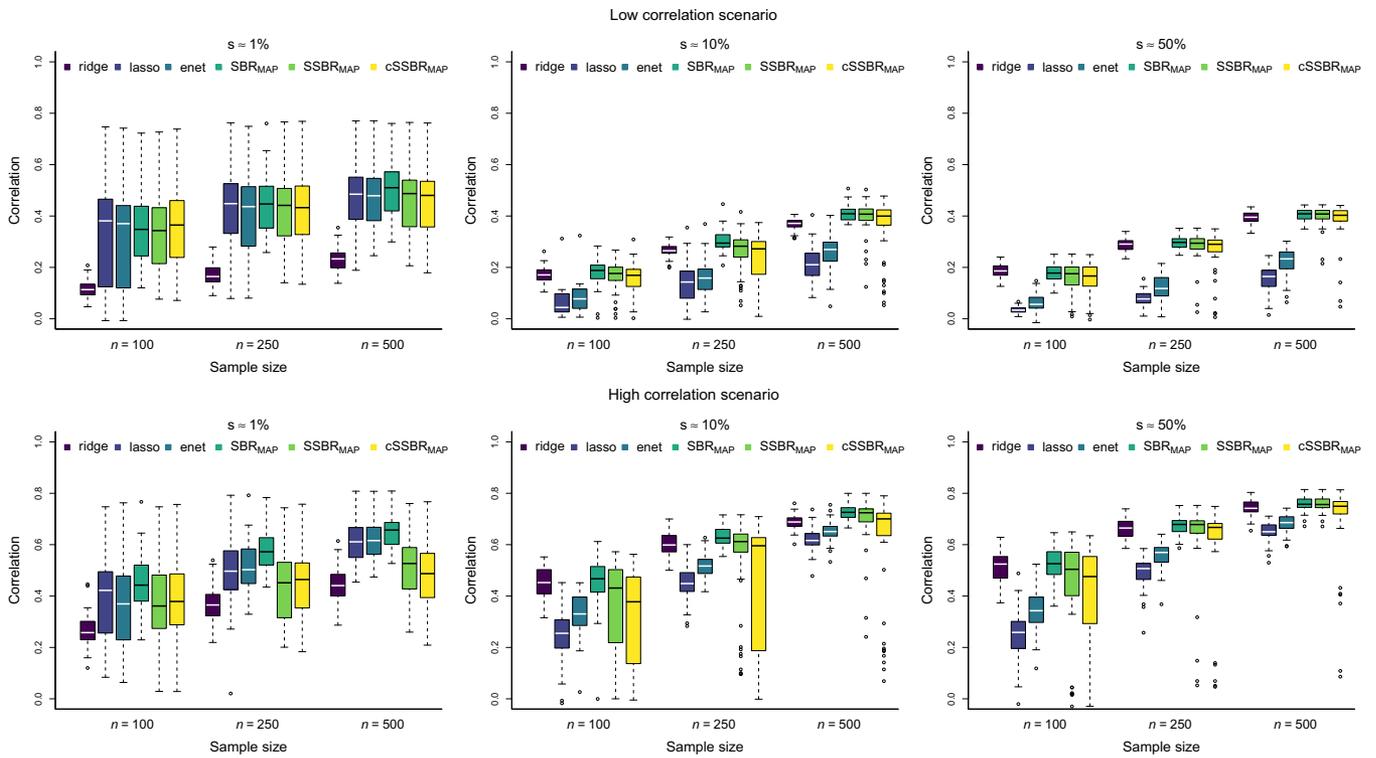


Figure 1. Simulation study. Comparison of ridge, lasso, and elastic net to SBR, SSBR, and cSSBR methods under the MAP estimator for the low correlation (top panels) and high correlation (bottom panels) scenarios. Each panel shows correlations between predictions and held-out test data at various training sample sizes and sparsity levels as indicated. Boxplots are over 50 sampled datasets. The total dimension p is over 100,000 with three (simulated) data sources (see text for details).

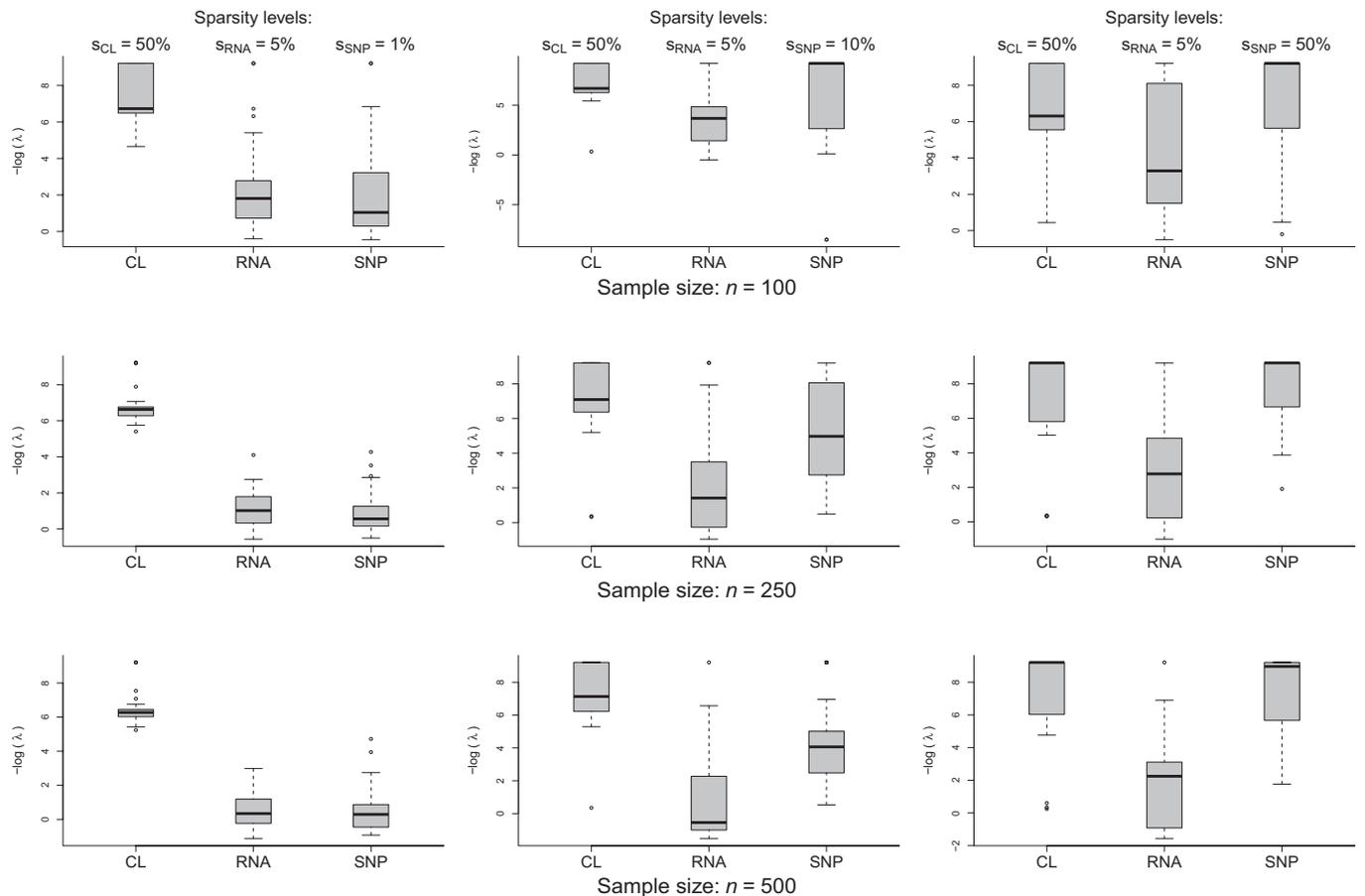


Figure 2. Simulation study, source-specific shrinkage levels. Boxplots showing source-specific $-\log \hat{\lambda}$ values from SBR based on the MAP estimator in the low-correlation simulations under various levels of sparsity for $n = 100$ (top row), $n = 250$ (middle row), and $n = 500$ (bottom row).

Table 2. Simulation study, induced sparsity.

| Method | Sample size | Simulation scenario | | | | | |
|----------------------|-------------|---------------------|------------------|------------------|------------------|------------------|------------------|
| | | Low-correlation | | | High-correlation | | |
| | | $s \approx 1\%$ | $s \approx 10\%$ | $s \approx 50\%$ | $s \approx 1\%$ | $s \approx 10\%$ | $s \approx 50\%$ |
| SSBR _{MAP} | $n = 100$ | 14% | 47% | 57% | 8% | 42% | 53% |
| | $n = 250$ | 10% | 50% | 71% | 4% | 57% | 73% |
| | $n = 500$ | 14% | 69% | 77% | 13% | 70% | 79% |
| cSSBR _{MAP} | $n = 100$ | 4% | 18% | 25% | 2% | 16% | 22% |
| | $n = 250$ | 2% | 24% | 35% | 1% | 25% | 38% |
| | $n = 500$ | 4% | 35% | 43% | 3% | 33% | 42% |

NOTE: Average sparsity induced by the SSBR and cSSBR methods based on the MAP estimator over 50 repetitions of the low and high correlation simulations under varying true sparsity (s).

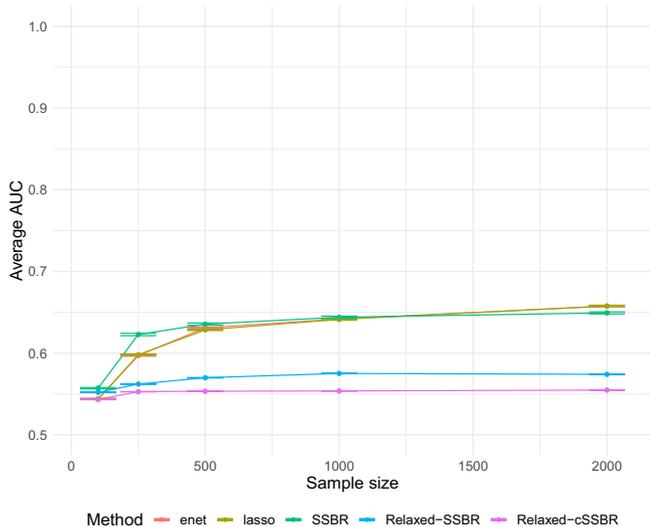


Figure 3. Simulation study, variable selection. One standard error plots of the average area under the ROC curve (AUC), over 20 repetitions, as a function of sample size.

predictors than observations, but in our simulations it did not adjust to the underlying sparsity levels (ranging from 0.06% to 1.2%).

The focus of this article is on prediction, but it is interesting to consider the variable selection behavior of the proposed methods. As already noted, lasso and enet yielded extremely sparse solutions for the problem considered above and are not suitable for selection in this particular $p \gg n$ setting. We therefore considered a smaller problem which includes only the simulated CL and RNA data with $p = 2026$ and $s \approx 6\%$. We summarize our findings via the area under the ROC curve (AUC; calculated using the absolute values of regression coefficients). Results comparing lasso, enet, SSBR, and relaxed-SSBR/cSSBR (all under the MAP estimate) are shown in Figure 3. Results based on the CV and ML estimates (not shown) are similar. We see that SSBR in this case is competitive with the lasso and enet.

4.3. Computational Performance

We conclude by examining computational burden as a function of total dimension p and in comparison with the lasso. To do so we include a fourth “data source” (which is simply Gaussian noise) that adds a variable number of predictors. The number of these additional predictors is set so that the total

$p \in \{5 \times 10^5, 10^6, 2.5 \times 10^6, 5 \times 10^6, 10^7\}$. Sample size is set equal to 100. Computations were carried out on a compute server with 128 cores (2.28GHz) and 1TB of RAM. For lasso we treat the binding of individual matrices into one data matrix (an operation not needed for SBR) as a preprocessing step and do not include this in the reported runtimes. We consider two approaches for tuning the lasso penalty. The first is the most commonly used approach in practice; that is, CV (lasso_{CV}). For this we use the parallel option in `glmnet` for estimation of the penalty parameter via 10-fold CV (the default option). The second approach is to do a grid search (lasso_{grid}) with no CV. This is sometimes used in practice with the purpose of finding a value that maximizes a specific criterion (for instance BIC). We do not consider any particular criterion but report only the time needed to evaluate the lasso over the grid (i.e., we do not include any computational cost for the assessment of any criterion). For the sake of comparison we define a rough grid over the interval $[0.1, 1)$ with a step of 0.2. We note that in a practical application of the lasso the grid might need to be finer and the computational costs of assessing any statistical criterion might be nontrivial given the size of the matrices. These factors would increase the computational time needed for the lasso. For our methods we consider the SBR_{MAP} approach, which requires evaluation of both $\hat{\lambda}_{CV}$ in (11) and $\hat{\lambda}_{MAP}$ in (13) (and is thus in principle the slowest), and also its corresponding relaxed sparse solution. We include the formation of transpose matrices and calculation of Gram matrices in reported runtime (although these could be regarded as preprocessing steps). We do not include ridge because it can be seen as a special case of SBR and hence would be equally fast when implemented as described here. We also do not include elastic net as this method will be slower than lasso due to tuning of the additional parameter α .

Figure 4 shows the average runtimes (in minutes) over five fits on a single simulated dataset. SBR and sparse SBR are considerably faster than lasso_{CV} with the gap increasing with p and this is also the case in comparison to the simple lasso_{grid} approach described above. We note that by adding random noise variables as described above we in a way “favor” the lasso implementation in `glmnet`, as the screening rules that are used by default can relatively easily exclude these covariates. SBR is the fastest method, with the average runtime for $p = 10^7$ being approximately 5 min, net of all steps. Running sparse SBR with suitable parallel block-matrix calculations makes this method also very fast; average runtime was approximately 10 min for $p = 10^7$. We note that our current implementations of the methods are certainly not optimal in terms of computational

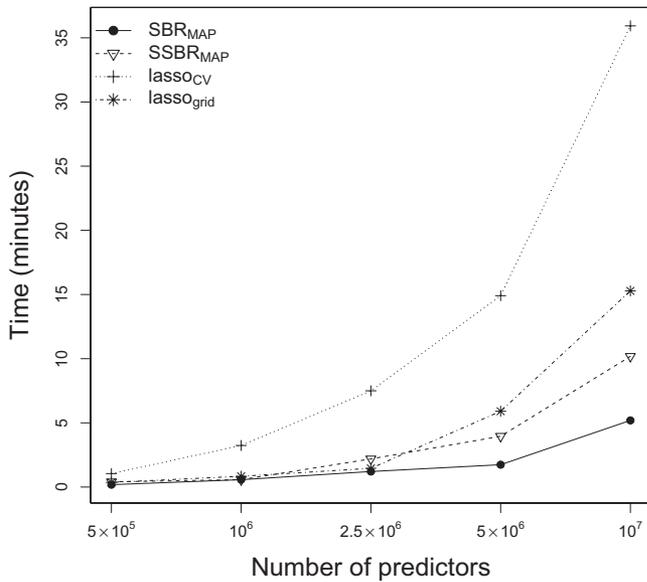


Figure 4. Simulation study, computational efficiency. Average runtime (five runs for one simulated dataset) required for SBR_{MAP}, SSBR_{MAP}, lasso_{CV}, and lasso_{grid}, for increasing p .

efficiency. We note also that our methods can fully use available cores for parallel computation, hence they should continue to gain in runtime if more cores are available even with no increase in clock speed.

5. Alzheimer’s Disease Case Study

5.1. Data

The data we consider are from the ADNI (Mueller et al. 2005), a large scale longitudinal AD study involving multiple data modalities. The specific subset of the ADNI data we use were previously used in a DREAM challenge (Allen et al. 2016) and consists of $n = 759$ baseline and 24 month follow-up records. Similarly to the DREAM challenge and follow-up work (e.g., Dondelinger, Mukherjee, et al. 2019) we consider as response the scores from a cognitive function test called the mini mental state examination (MMSE); in particular, the difference in MMSE between 24 month follow-up and baseline.

We consider three data sources: (1) clinical (CL) data consisting of $p_{CL} = 12$ features (including, among others, diagnosis at baseline, Apolipoprotein E status, gender, age, years of education); (2) structural magnetic resonance imaging (MRI) data consisting of $p_{MRI} = 929$ features; and (3) genetic data in the form of SNP data, consisting of $p_{SNP} \approx 7.3 \times 10^6$ features (this is the number of SNPs available after excluding those with zero variance across subjects and those with more than 10% missing entries). We apply the proposed methods to these data, treating the three data types (1)–(3) as sources.

The aim is to consider a real-world application with data sources of widely differing dimensionality and to investigate whether adding the complex MRI and genetic data to the clinical covariates can improve predictive ability. We emphasize that the goals of the present article are mainly methodological and that the results we present at this stage should be regarded as

illustrative of the capabilities of the methods rather than as candidate AD predictors for practical use.

5.2. Results

Figure 5 shows results using SBR with CV, ML, and MAP estimators applied to CL only, CL and MRI, and finally all of the data (CL, MRI, and SNP). For the latter case we also show results using SSBR and cSSBR with $f_n = \log(n)$. Predictive performance is quantified via the correlation between predicted and observed values in held-out test data. The boxplots show the results of 10 random train/test splits (with $n_{train} = 500$, $n_{test} = 259$ in each split) annotated with the number of variables with nonzero coefficients after fitting the models in each case.

Here we see that the choice of estimator can make a difference: the results from the CV approach are notably worse when considering the CL and MRI datasets together. In this case the CV estimator does not adjust properly, since under the ML and MAP approaches we actually see that the addition of MRI features to clinical covariates improves predictions. In contrast to many studies including MRI data, here we included all available MRI features without any preselection. Adding the SNP features does not increase predictive accuracy further, rather it slightly worsens performance, in line with previous work suggesting that genetic data is not helpful when clinical covariates are already available (Allen et al. 2016). Notably, SSBR_{MAP} yields almost the same predictive accuracy as the regressions that do not include the SNP matrix at all; this is because the sparsified analyses are able to set all SNP coefficients to exactly zero. More generally, the excess risk (over CL and MRI alone) is relatively small in magnitude, despite the vast number of additional covariates. This is due to the fact that the models have a separate tuning parameter for each data source and are therefore able to effectively “switch off” this source, while continuing to regularize the other covariates via source-specific penalties. In addition, cSSBR_{MAP} provides identical predictions with SSBR_{MAP} whilst employing fewer CL and MRI features.

All penalties were estimated directly from the data and not prespecified in any way. To further investigate source-specific penalization, in Figure 6 we show boxplots of the source-specific $-\log \hat{\lambda}_k$ values (when we consider all three sources together), for $k = \{CL, MRI, SNP\}$ and $n_{train} = \{100, 250, 500\}$. Under the CV/ML estimators the MRI dataset is penalized less in comparison to the other two, while under the MAP estimator it is the CL dataset which appears to be the most important. This seems to explain the slight differences between the methods (and their sparse extensions) observed in Figure 5 with respect to predictive accuracy when including all sources.

A referee suggested including results from classical ridge for the combined CL, MRI, and SNP analysis. Due to computational considerations arising from the need to handle the large matrices (unlike in our method, classical ridge cannot decompose the problem) we consider a variant of the problem above with a random subset of half of the SNP variables. Results (again from 10 train/test splits) obtained from 5-fold CV using `glmnet` indicate poor predictive performance; the resulting average correlation is 0.14 (standard deviation of 0.05). This result provides an empirical example of the benefit of source-specific penalization.

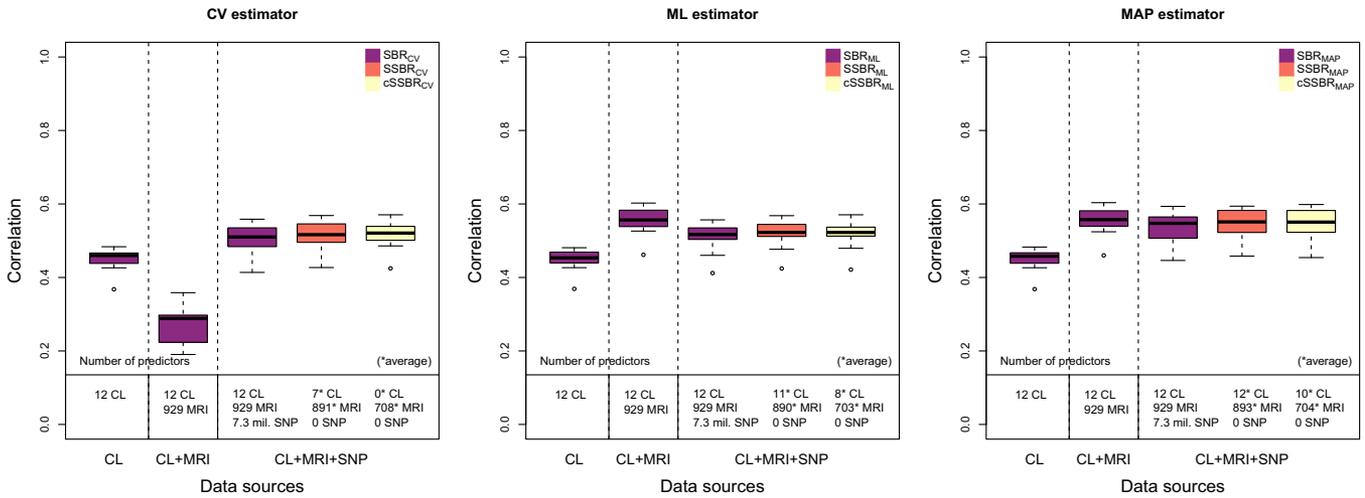


Figure 5. Alzheimer's disease case study, predictive performance. Correlations between predictions and held-out test data from 10 random splits with $n_{\text{train}} = 500$ and $n_{\text{test}} = 259$ under SBR, SSBR, and cSSBR using the CV (left), ML (center), and MAP (right) estimators.

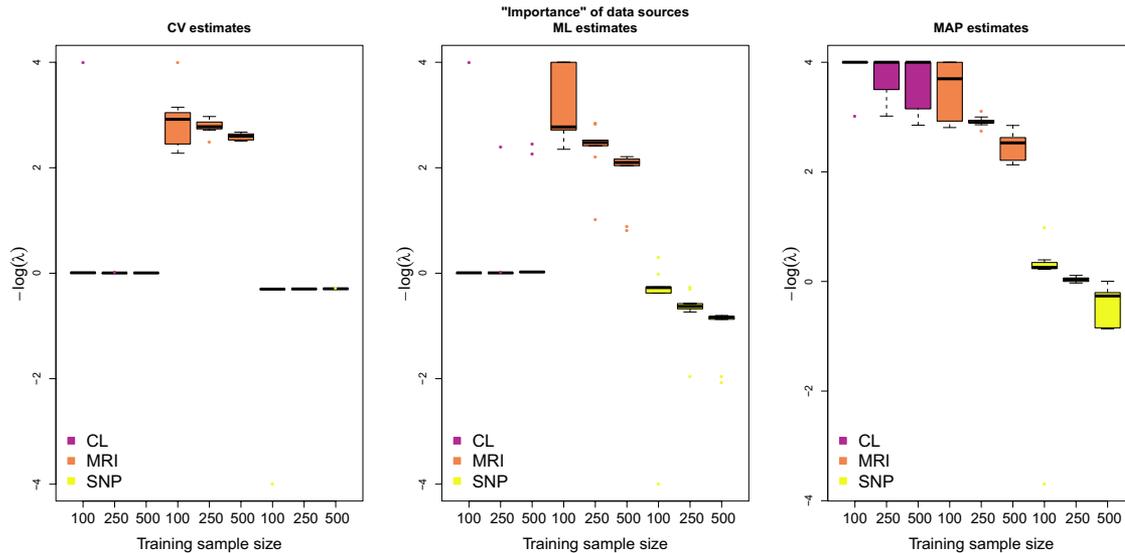


Figure 6. Alzheimer's disease case study, estimated shrinkage levels. Boxplots show source-specific $-\log \hat{\lambda}$ values versus training sample size using the SBR methods with CV (left), ML (middle), and MAP (right) estimation of shrinkage levels.

6. Discussion

The aim of this article was to introduce a framework for high-dimensional regression using multiple data sources that allows efficient and fast computations in the wide data, very large p setting. We introduced SBR, a generalized ridge-type model which can be particularly effective for prediction in the aforementioned setting. We further proposed sparse extensions (SSBR), including a general solution scalable to moderately large p and a relaxed solution scalable to very large p .

Concerning the three EB estimators (CV/ML/MAP), our empirical results suggest that the choice of estimator can affect predictive performance. Specifically, in the case study results under CV were worse for the combined clinical/MRI analysis, likely due to the variance of the CV risk estimate. Overall, using the MAP estimator resulted in better predictive performance both in the simulation study and in the case study; therefore, we recommend using this approach as the default option. With respect to the sparse solutions, we mainly focused

on the relaxed SSBR approaches, demonstrating that they show potential for achieving similar predictive performance to SBR but with explicit sparsity. In addition, using some control for the effect of sample size on sparsity seems desirable as it can lead to enhanced sparsity with no loss in terms of prediction.

It is worth nothing that the proposed sparsification approach based on the KL divergence is quite general in the sense that it is not restricted to the use of a ridge prior which assumes independence a priori. For example, one can consider a g -prior (Zellner 1986) when $n > p$ or the information matrix prior (Gupta and Ibrahim 2009) when $n < p$. The latter approach, which is a combination of the g -prior and the ridge prior, can also be extended to incorporate multiple penalties in a manner similar to the one presented in this article.

As a final remark, we note that this work primarily focused on predictive performance. However, we think that understanding the potential role of the methodologies presented here (and related scalable Bayesian and post-processing methods) for variable selection will be an interesting direction for future work.

Supplementary Materials

Appendices: Mathematical derivations related to Equations (4), (7), (9), (11), (17), and (19) and additional results from Section 4 (pdf file).
 R package and code: Package `sbr` is available at <https://github.com/kperrakis/sbr>. (R installable files and documentation pdf file). Reproducible code for the simulation study of Section 4 available at https://github.com/kperrakis/sbr/tree/master/simulation_paper (Rdata and R files).

Acknowledgments

We would like to thank the editor and anonymous referees for their constructive remarks and suggestions. Furthermore, we want to thank Rajen Shah for his insights into the equivalent representation of the general sparse solution and Heather Cordell for input on the genetic data.

Data collection and sharing for the Alzheimer's data application was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant—NIH funding: U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: AbbVie, Alzheimer's Association; Alzheimer's Drug Discovery Foundation; Araclon Biotech; BioClinica, Inc.; Biogen; Bristol-Myers Squibb Company; CereSpir, Inc.; Cogstate; Eisai Inc.; Elan Pharmaceuticals, Inc.; Eli Lilly and Company; EuroImmun; F. Hoffmann—La Roche Ltd and its affiliated company Genentech, Inc.; Fujirebio; GE Healthcare; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Lumosity; Lundbeck; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Neurotrack Technologies; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Takeda Pharmaceutical Company; and Transition Therapeutics. The Canadian Institutes of Health Research is providing funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (www.fnih.org). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Therapeutic Research Institute at the University of Southern California. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of Southern California.

References

- Allen, G., Amoroso, N., Anghel, C., Balagurusamy, V., Bare, C., Beaton, D., Bellotti, R., Bennett, D. A., Boehme, K. L., Boutros, P. C., and Caberlotto, L. (2016), "Crowdsourced Estimation of Cognitive Decline and Resilience in Alzheimer's Disease," *Alzheimer's and Dementia*, 12, 645–653. [37]
- Antoniadis, A., and Fan, J. (2001), "Regularization of Wavelet Approximations," *Journal of the American Statistical Association*, 96, 939–967. [31]
- Armagan, A., Dunson, D., and Lee, J. (2013), "Generalized Double Pareto Shrinkage," *Statistica Sinica*, 23, 119–143. [28,30]
- Aseervatham, S., Antoniadis, A., Gaussier, E., Burlet, M., and Denneulin, Y. (2011), "A Sparse Version of the Ridge Logistic Regression for Large-Scale Text Categorization," *Pattern Recognition Letters*, 32, 101–106. [31]
- Balakrishnan, S., and Madigan, D. (2010), "Priors on the Variance in Sparse Bayesian Learning: The Demi-Bayesian Lasso," in *Frontiers of Statistical Decision Making and Bayesian Analysis: In Honor of James O. Berger*, eds. M.-H. Chen, P. Muller, D. Sun, and K. Ye, New York: Springer Science and Business Media, pp. 346–359. [30]
- Bishop, C. (2006), *Pattern Recognition and Machine Learning*, Information Science and Statistics, New York: Springer-Verlag. [32]
- Bondell, H., and Reich, B. J. (2012), "Consistent High-Dimensional Bayesian Variable Selection via Penalized Credible Regions," *Journal of the American Statistical Association*, 107, 1610–1624. [29,31,32]
- Botev, Z., L'Ecuyer, P., and Tuffin, B. (2013), "Markov Chain Importance Sampling With Applications to Rare Event Probability Estimation," *Statistics and Computing*, 23, 271–285. [32]
- Carvalho, C., Polson, N., and Scott, J. (2010), "The Horseshoe Estimator for Sparse Signal," *Biometrika*, 97, 465–480. [28,30]
- Dondelinger, F., Mukherjee, S., and the Alzheimer's Disease Neuroimaging Initiative (2019), "The Joint Lasso: High-Dimensional Regression for Group Structured Data," *Biostatistics* (forthcoming). [29,37]
- Fan, J., and Li, R. (2001), "Variable Selection via Nonconcave Penalized Likelihood and Its Oracle Properties," *Journal of the American Statistical Association*, 96, 1348–1360. [28]
- Griffin, J. E., and Brown, P. J. (2010), "Inference With Normal-Gamma Prior Distributions in Regression Problems," *Bayesian Analysis*, 5, 171–188. [28,30]
- Gupta, M., and Ibrahim, J. G. (2009), "An Information Matrix Prior for Bayesian Analysis in Generalized Linear Models With High Dimensional Data," *Statistica Sinica*, 19, 1641–1663. [38]
- Hahn, P., and Carvalho, C. (2015), "Decoupled Shrinkage and Selection in Bayesian Linear Models: A Posterior Summary Perspective," *Journal of the American Statistical Association*, 110, 435–448. [29,31,32]
- Hans, C. (2010), "Model Uncertainty and Variable Selection in Bayesian Lasso Regression," *Statistics and Computing*, 20, 221–229. [28,30]
- Huang, J., Breheny, P., and Ma, S. (2012), "A Selective Review of Group Selection in High-Dimensional Models," *Statistical Science*, 27, 481–499. [28]
- Kyung, M., Gill, J., Ghosh, M., and Casella, G. (2010), "Penalized Regression, Standard Errors, and Bayesian Lassos," *Bayesian Analysis*, 5, 369–412. [28]
- Lykou, A., and Ntzoufras, I. (2013), "On Bayesian Lasso Variable Selection and the Specification of the Shrinkage Parameter," *Statistics and Computing*, 23, 361–390. [30]
- Mackay, T., Richards, S., Stone, E., Barbadilla, A., Ayroles, J., Zhu, D., Casillas, S., Han, Y., Magwire, M. M., Cridland, J. M., and Richardson, M. F. (2012), "The *Drosophila melanogaster* Genetic Reference Panel," *Nature*, 482, 173–178. [33]
- Mineo, A. (2003), "On the Estimation of the Structure Parameter of a Normal Distribution of Order p ," *Statistica*, 63, 109–122. [34]
- Minka, T. (2001), "Expectation Propagation for Approximate Bayesian Inference," in *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pp. 362–369. [32]
- Mueller, S., Weiner, M., Thal, L., Petersen, R., Jack, C., Jagust, W., Trojanowski, J. Q., Toga, A. W., and Beckett, L. (2005), "Ways Toward an Early Diagnosis in Alzheimer's Disease: The Alzheimer's Disease Neuroimaging Initiative (ADNI)," *Alzheimer's and Dementia*, 1, 55–66. [29,37]
- Park, T., and Casella, G. (2008), "The Bayesian Lasso," *Journal of the American Statistical Association*, 103, 681–686. [28,30]
- Perrakis, K., Ntzoufras, I., and Tsonas, E. (2014), "On the Use of Marginal Posteriors in Marginal Likelihood Estimation via Importance Sampling," *Computational Statistics and Data Analysis*, 77, 54–69. [32]
- R Core Team (2019), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, available at <https://www.R-project.org/>. [31]
- Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2013), "A Sparse-Group Lasso," *Journal of Computational and Graphical Statistics*, 22, 231–245. [29]
- Tibshirani, R. (1996), "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society, Series B*, 58, 267–288. [28,30]
- 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, 91–108. [28]
- Yuan, M., and Lin, Y. (2005), "Efficient Empirical Bayes Variable Selection and Estimation in Linear Models," *Journal of the American Statistical Association*, 100, 1215–1225. [28,30]
- (2006), "Model Selection and Estimation in Regression With Grouped Variables," *Journal of the Royal Statistical Society, Series B*, 68, 49–67. [28]
- Zellner, A. (1986), "On Assessing Prior Distributions and Bayesian Regression Analysis Using g-Prior Distributions," in *Bayesian Inference and Decision Techniques: Essays in Honor of Bruno de Finetti*, eds. P. Goel and A. Zellner, Amsterdam: North-Holland, pp. 233–243. [38]
- Zou, H. (2006), "The Adaptive Lasso and Its Oracle Properties," *Journal of the American Statistical Association*, 101, 1418–1429. [33]
- Zou, J., and Hastie, T. (2005), "Regularization and Variable Selection via the Elastic Net," *Journal of the Royal Statistical Society, Series B*, 67, 301–320. [28]