Title

Prior-Preconditioned Conjugate Gradient Method for Accelerated Gibbs Sampling in Large n, Large p Bayesian Sparse Regression.

Journal

Journal of the American Statistical Association, 118(544)

ISSN

0162-1459

Authors

Nishimura, Akihiko
Suchard, Marc

Publication Date

2023

DOI

10.1080/01621459.2022.2057859

Peer reviewed

HHS Public Access

Author manuscript

J Am Stat Assoc. Author manuscript; available in PMC 2024 March 28.

Published in final edited form as:

J Am Stat Assoc. 2023 ; 118(544): 2468–2481. doi:10.1080/01621459.2022.2057859.

Prior-Preconditioned Conjugate Gradient Method for Accelerated Gibbs Sampling in "Large *n*, Large *p*" Bayesian Sparse Regression

Akihiko Nishimura[a], **Marc A. Suchard**[b]

[a]Department of Biostatistics, Johns Hopkins University, Baltimore, MD

[b]Department of Biomathematics, Biostatistics, and Human Genetics, University of California-Los Angeles, Los Angeles, CA

Abstract

In a modern observational study based on healthcare databases, the number of observations and of predictors typically range in the order of 10^5–10^6 and of 10^4–10^5. Despite the large sample size, data rarely provide sufficient information to reliably estimate such a large number of parameters. Sparse regression techniques provide potential solutions, one notable approach being the Bayesian method based on shrinkage priors. In the "large n and large p" setting, however, the required posterior computation encounters a bottleneck at repeated sampling from a high-dimensional Gaussian distribution, whose precision matrix $\mathbf{\Phi}$ is expensive to compute and factorize. In this article, we present a novel algorithm to speed up this bottleneck based on the following observation: We can cheaply generate a random vector $\boldsymbol{b}$ such that the solution to the linear system $\mathbf{\Phi}\boldsymbol{\beta} = \boldsymbol{b}$ has the desired Gaussian distribution. We can then solve the linear system by the conjugate gradient (CG) algorithm through matrix-vector multiplications by $\mathbf{\Phi}$; this involves no explicit factorization or calculation of $\mathbf{\Phi}$ itself. Rapid convergence of CG in this context is guaranteed by the theory of *prior-preconditioning* we develop. We apply our algorithm to a clinically relevant large-scale observational study with $n = 72{,}489$ patients and $p = 22{,}175$ clinical covariates, designed to assess the relative risk of adverse events from two alternative blood anti-coagulants. Our algorithm demonstrates an order of magnitude speed-up in posterior inference, in our case cutting the computation time from two weeks to less than a day. Supplementary materials for this article are available online.

Keywords

Big data; Conjugate gradient; Markov chain Monte Carlo; Numerical linear algebra; Sparse matrix; Variable selection

CONTACT Akihiko Nishimura aki.nishimura@jhu.edu Department of Biostatistics, Johns Hopkins University, Baltimore, MD.

Supplementary Materials

The supplementary materials include: extensive additional simulation results based on 20,000+ hours of cpu time, discussions on choice of linear algebra libraries and how it affects computational performance, proofs of the theoretical results in Appendix B, the precise definition of the outcome and cohort for the blood anti-coagulant study, among others.

1. Introduction

Given an outcome of interest y_i and a large number of features $x_{i1}, \ldots, x_{ip}$ for $i = 1, \ldots, n$, the goal of sparse regression is to find a small subset of these features that captures the principal relationship between the outcome and features. Such a sparsity assumption is mathematical necessity when p exceeds the sample size n. Even when $n > p$, however, the assumption often remains critical in improving the interpretability and stable estimation of regression coefficients $\boldsymbol{\beta}$. This is especially true under the following conditions, either of which reduces the amount of information the data provides on the regression coefficients: (a) the design matrix $\boldsymbol{X}$ is sparse; that is, only a small fraction of the design matrix contains nonzero entries due to infrequent binary features, and/or (b) the binary outcome $\boldsymbol{y}$ is rare; that is, $y_i = 0$ for most of i's. Sparse design matrices are extremely common in modern observational studies based on healthcare databases; while a large number of potential preexisting conditions and available treatments exist, only a small subset of these applies to each patient (Schuemie et al. 2018). Rare binary outcomes are also common as many diseases of interest have low incidence rates among the population.

A particular application considered in this manuscript is a comparative study of two blood anti-coagulants *dabigatran* and *warfarin*, using observational data from Truven Health MarketScan Medicare Supplemental and Coordination of Benefits Database. The anti-coagulants help prevent blood clot formation among patients with atrial fibrilation but come with risks of serious side effects. The goal of the study is to quantify which of the two drugs has a lower risk of gastrointestinal bleeding. The dataset consists of $n = 72{,}489$ patients and $p = 22{,}175$ clinical covariates of potential relevance.

To induce sparsity in the estimate of regression coefficient $\boldsymbol{\beta}$, an increasingly common approach is the Bayesian method based on *shrinkage priors*. This class of prior is often represented as a scale-mixture of Gaussians:

$$\beta_j \mid \lambda_j, \tau \sim \mathcal{N}\left(0, \tau^2\lambda_j^2\right), \lambda_j \sim \pi_{\text{loc}}(\cdot), \tau \sim \pi_{\text{glo}}(\cdot),$$

where τ and λ_j are unknown *global* and *local scale* parameters with priors $\pi_{\text{loc}}(\cdot)$ and $\pi_{\text{glo}}(\cdot)$ (Carvalho, Polson, and Scott 2010; Polson, Scott, and Windle 2014; Bhattacharya et al. 2015; Bhadra et al. 2019). Compared to more traditional "spike-and-slab" discrete-mixture priors, continuous shrinkage priors are typically more computationally efficient while maintaining highly desirable statistical properties (Datta and Ghosh 2013; Pal and Khare 2014; Bhattacharya et al. 2015). Despite the relative computational advantage, however, posterior inference under these priors still faces a serious scalability issue. In the blood anti-coagulant safety study, for instance, it takes over 200 hr on a modern high-end commodity desktop to run 10,000 iterations of the current state-of-the-art Gibbs sampler, even with optimized implementation (Section 4).

We focus on sparse logistic regression in this article, but our Gibbs sampler acceleration technique applies whenever the likelihood function can be expressed as a Gaussian mixture. The data augmentation scheme of Polson, Scott, and Windle (2013) makes a posterior under

the logistic model amenable to Gibbs sampling as follows. Conditioning on a Polya-Gamma auxiliary parameter $\boldsymbol{\omega}$, the likelihood of a binary outcome $\boldsymbol{y}$ becomes

$$\tilde{y}_i \mid \boldsymbol{X}, \boldsymbol{\beta}, \boldsymbol{\omega} \sim \mathcal{N}\left(\boldsymbol{x}_i^\top \boldsymbol{\beta}, \omega_i^{-1}\right) \text{ for } \tilde{y}_i := \omega_i^{-1}(y_i - 1/2). \tag{1.1}$$

Correspondingly, the full conditional distribution of $\boldsymbol{\beta}$ is given by

$$\boldsymbol{\beta} \mid \boldsymbol{\omega}, \lambda, \tau, \boldsymbol{y}, \boldsymbol{X} \sim \mathcal{N}\left(\boldsymbol{\Phi}^{-1}\boldsymbol{X}^\top \boldsymbol{\Omega}\tilde{\boldsymbol{y}}, \boldsymbol{\Phi}^{-1}\right) \text{ for } \boldsymbol{\Phi} = \boldsymbol{X}^\top \boldsymbol{\Omega} X + \tau^{-2}\boldsymbol{\Lambda}^{-2}, \tag{1.2}$$

where $\boldsymbol{\Omega} = \text{diag}(\boldsymbol{\omega})$, a diagonal matrix with entries $\boldsymbol{\Omega}_{ii} = \boldsymbol{\omega}_i$, and $\boldsymbol{\Lambda} = \text{diag}(\boldsymbol{\lambda})$. (See supplementary materials S1 for a complete description of the conditional updates within the Gibbs sampler.)

The main computational bottleneck of the Gibbs sampler is the need to repeatedly sample from high-dimensional Gaussians of the form (1.2). The standard algorithm requires $O(np^2 + p^3)$ operations: $O(np^2)$ for computing the term $\boldsymbol{X}^\top\boldsymbol{\Omega}\boldsymbol{X}$ and $O(p^3)$ for Cholesky factorization of $\boldsymbol{\Phi}$. These operations remain significant burden even with sparsity in $\boldsymbol{X}$ because computing times of sparse linear algebra operations are dominated not by the number of arithmetic operations but by latency in irregular data access (Dongarra, Heroux, and Luszczek 2016; Duff, Erisman, and Reid 2017).

The "large n, large p" logistic regression problem considered in this article remains unsolved despite the recent computational advances. For $n \ll p$ cases, Bhattacharya, Chakraborty, and Mallick (2016) propose an algorithm to sample from (1.2) with only $O(n^2p + n^3)$ operations. Johndrow, Orenstein, and Bhattacharya (2020) reduce the $O(n^2p)$ cost by replacing the matrix $\mathbf{X}\boldsymbol{\Lambda}^2\boldsymbol{X}^\top$ with an approximation that can be computed with $O(n^2k)$ operations for $k < p$. These techniques offer no reduction in computational cost for $n > p$ cases, however. Hahn, He, and Lopes (2018) propose a sampling approach for linear regression based on an extensive preprocessing of the matrix $\boldsymbol{X}^\top\boldsymbol{X}$—a trick limited in scope strictly to the Gaussian likelihood model.

Proposed in this article is a novel algorithm to rapidly sample from a high-dimensional Gaussian distribution of the form (1.2) through the conjugate gradient (CG) method, using only a small number of matrix-vector multiplications $\boldsymbol{v} \to \boldsymbol{\Phi}\boldsymbol{v}$. Our algorithm requires no explicit formation of the matrix $\boldsymbol{\Phi}$ because we can compute $\boldsymbol{v}$ via operations $\boldsymbol{v} \to \boldsymbol{X}\boldsymbol{v}$ and $\boldsymbol{w} \to \boldsymbol{X}^\top\boldsymbol{w}$, along with element-wise vector multiplications. This is an important feature not only for computational efficiency but also for memory efficiency when dealing with a large and sparse design matrix $\boldsymbol{X}$. The matrix $\boldsymbol{X}^\top\boldsymbol{\Omega}\boldsymbol{X}$ and hence $\boldsymbol{\Phi}$ typically contain a much larger proportion of nonzero entries than $\boldsymbol{X}$, making it far more memory intensive to handle $\boldsymbol{\Phi}$ directly. For example, when $p = 10^5$, it would require 74.5 GB of memory to store a $p \times p$ dense matrix $\boldsymbol{\Phi}$ in double-precision numbers. On the other hand, our algorithm can exploit a sparsity structure in $\boldsymbol{X}$ for both computational and memory efficiency.

Practical utility of CG depends critically on effective *preconditioning*, whose purpose is to speed up the algorithm by relating the given linear system to a modified one. Finding an effective preconditioner is a highly problem-specific task and is often viewed as "a combination of art and science" (Saad 2003). Exploiting fundamental features of sparse regression posteriors, we develop the *prior-preconditioning* strategy tailored toward the linear systems in our specific context. We study its theoretical properties and demonstrate its superiority over general-purpose preconditioners in Bayesian sparse regression applications.

The rest of the article is organized as follows. Section 2 begins by describing how to recast the problem of sampling from the distribution (1.2) as that of solving a linear system $\boldsymbol{\Phi\beta} = \boldsymbol{b}$. The remainder of the section explains how to apply CG to rapidly solve the linear system, developing necessary theories along the way. In Section 3, we use simulated data to study the effectiveness of our CG sampler in the sparse regression context. Also studied is how the behavior of CG depends on different preconditioning strategies. In Section 4, we apply our algorithm to the blood anti-coagulant safety study, demonstrating an order of magnitude speed-up in the posterior computation. Among the 22,175 predictors, the sparse regression posterior identifies age groups as significant source of treatment effect heterogeneity.

Our CG-accelerated Gibbs sampler is implemented as the *bayesbridge* package available from Python Package Index (pypi.org). The source code is available at a GitHub repository https://github.com/ohdsi/bayes-bridge.

2. Conjugate Gradient Sampler

2.1. Generating Gaussian Vector as Solution of Linear System

The standard algorithm for sampling a multivariate-Gaussian requires the Cholesky factorization $\boldsymbol{\Phi} = \boldsymbol{LL}^\mathsf{T}$ of its precision (or covariance) matrix (Rue and Held 2005). When the precision matrix $\boldsymbol{\Phi}$ has a specific structure as in (1.2), however, it turns out we can recast the problem of sampling from the distribution (1.2) to that of solving a linear system. This in particular obviates the need to compute and factorize $\boldsymbol{\Phi}$.

Proposition 2.1.—The following procedure generates a sample $\boldsymbol{\beta}$ from the distribution (1.2):

1. Generate $\boldsymbol{b} \sim \mathcal{N}\left(\boldsymbol{X}^\mathsf{T}\boldsymbol{\Omega}\tilde{y}, \boldsymbol{\Phi}\right)$ by sampling independent Gaussian vectors $\eta \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_n)$ and $\boldsymbol{\delta} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_p)$ and then setting

$$\boldsymbol{b} = \boldsymbol{X}^\mathsf{T}\boldsymbol{\Omega}\tilde{\boldsymbol{y}} + \boldsymbol{X}^\mathsf{T}\boldsymbol{\Omega}^{1/2}\eta + \tau^{-1}\boldsymbol{\Lambda}^{-1}\boldsymbol{\delta}. \tag{2.3}$$

2. Solve the following linear system for $\boldsymbol{\beta}$:

$$\boldsymbol{\Phi}\beta = \boldsymbol{b} \text{ where } \boldsymbol{\Phi} = \boldsymbol{X}^\mathsf{T}\boldsymbol{\Omega}X + \tau^{-2}\boldsymbol{\Lambda}^{-2}. \tag{2.4}$$

The result follows immediately from basic properties of multivariate Gaussians. The Gaussian vector $\boldsymbol{b}$ has $\mathrm{var}(\boldsymbol{b}) = \boldsymbol{\Phi}$ and is generated with a computational cost negligible compared to computing and factorizing $\boldsymbol{\Phi}$. The solution to (2.4) has the required covariance structure because $\mathrm{var}(\boldsymbol{\Phi}^{-1}\boldsymbol{b}) = \boldsymbol{\Phi}^{-1}\mathrm{var}(\boldsymbol{b})(\boldsymbol{\Phi}^{-1})^{\intercal}$.

Bhattacharya, Chakraborty, and Mallick (2016) propose a related algorithm which reduces the task of sampling a multivariate Gaussian to solving a $n \times n$ linear system. On the other hand, our algorithm reduces the task to solving a $p \times p$ system, which is smaller in size when $p < n$ and, more importantly, amenable to a fast solution via CG as we will show.

2.2. Iterative Method for Solving Linear System

Proposition 2.1 is useful because solving the linear system (2.4) can be significantly faster than the standard algorithm for sampling a Gaussian vector. We achieve this speed-up by applying the CG method (Hestenes and Stiefel 1952; Lanczos 1952). CG belongs to a family of *iterative methods* for solving a linear system. Compared to traditional direct methods, iterative methods are more memory efficient and, if the matrix $\boldsymbol{\Phi}$ has certain structures (Section 2.3), can be significantly faster.

Iterative methods have found applications in Gaussian process models, where optimizing the hyper-parameters of covariance functions requires solving linear systems involving large covariance matrices (Gibbs and MacKay 1997). Significant research has gone into how best to apply iterative methods in this specific context; see Stein, Chen, and Anitescu (2012), Sun and Stein (2016), and Stroud, Stein, and Lysen (2017) for example. Outside the Gaussian process literature, Zhou and Guan (2019) use an iterative method to address the bottleneck of having to solve large linear systems when computing Bayes factors in a model selection problem.

A novel feature of our work is the use of CG as a computational tool for Monte Carlo simulation. A related work is Zhang, Datta, and Banerjee (2019), brought to our attention while we were preparing the first draft of our manuscript. They use the same idea as in Proposition 2.1 to generate a posterior sample from a Gaussian process model. However, they fail to investigate when and how CG delivers practical computational gains. Our work is distinguished by the development—supported by both theoretical analysis and systematic empirical evaluations—of a novel preconditioning technique tailored toward Bayesian sparse regression problems (Sections 2.4 and 2.5). In the process, we also compile a summary of the most practically useful of theoretical results regarding CG (Appendix B), which has previously been scattered across the literature, to facilitate potential applications of CG to a broader range of statistical problems.

The CG method solves a linear system $\boldsymbol{\Phi}\boldsymbol{\beta} = \boldsymbol{b}$ involving a positive definite matrix $\boldsymbol{\Phi}$ as follows. Given an initial guess $\boldsymbol{\beta}_0$, which may be taken as $\boldsymbol{\Phi}\boldsymbol{\beta}_0 = \boldsymbol{0}$ for example, CG generates a sequence $\{\boldsymbol{\beta}_k\}_{k=1,2,\ldots}$ of increasingly accurate approximations to the solution. The convergence of the CG iterates $\boldsymbol{\beta}_k$'s is intimately tied to the *Krylov subspace*

$$\mathscr{K}(\boldsymbol{\Phi}, \boldsymbol{r}_0, k) = \mathrm{span}\left\{\boldsymbol{r}_0, \boldsymbol{\Phi}\boldsymbol{r}_0, \ldots, \boldsymbol{\Phi}^{k-1}\boldsymbol{r}_0\right\},$$

generated from the initial residual $\boldsymbol{r}_0 = \boldsymbol{\Phi}\boldsymbol{\beta}_0 - \boldsymbol{b}$. With $\beta_0 + \mathcal{K}(\boldsymbol{\Phi}, \boldsymbol{r}_0, k)$ denoting an affine space $\{\beta_0 + \boldsymbol{v} : \boldsymbol{v} \in \mathcal{K}(\boldsymbol{\Phi}, \boldsymbol{r}_0, k)\}$, the approximate solution $\boldsymbol{\beta}_k$ satisfies the following optimality property in terms of a weighted l^2 norm $\|\cdot\|_{\boldsymbol{\Phi}}$, often referred to as the $\boldsymbol{\Phi}$-norm:

$$\boldsymbol{\beta}_k = \operatorname{argmin}\{\|\beta' - \beta\|_{\boldsymbol{\Phi}} : \beta' \in \beta_0 + \mathcal{K}(\boldsymbol{\Phi}, \boldsymbol{r}_0, k)\} \quad \text{where } \|\boldsymbol{r}\|_{\boldsymbol{\Phi}}^2 := \boldsymbol{r}^{\mathsf{T}}\boldsymbol{\Phi}\boldsymbol{r}. \tag{2.5}$$

The optimality property (2.5) in particular implies that CG yields the exact solution after p iterations. As evident from the pseudo-code in Section S2 of supplementary materials, the main computational cost of each update $\boldsymbol{\beta}_k \rightarrow \boldsymbol{\beta}_{k+1}$ is a matrix-vector operation $\boldsymbol{v} \rightarrow \boldsymbol{\Phi}\boldsymbol{v}$. Consequently, the required number of arithmetic operations to run p iterations of the CG update is comparable to that of a direct linear algebra method. For a typical precision matrix $\boldsymbol{\Phi}$ in the conditional distribution (1.2), however, we can induce rapid convergence of CG through the preconditioning strategy described in the next section. In our numerical results, we indeed find that the distribution of $\boldsymbol{\beta}_k$ even for $k \ll p$ is indistinguishable from (1.2) for all practical purposes.

2.3. Convergence of CG and its Relation to Eigenvalue Distribution

The iterative solution $\{\boldsymbol{\beta}_k\}_{k=0,1,2,\ldots}$ often displays slow convergence when CG is directly applied to a given linear system. Section 2.4 covers the topic of how to induce more rapid CG convergence for the system (2.4). In preparation, here we describe how the convergence behavior of CG is related to the structure of the positive definite matrix $\boldsymbol{\Phi}$.

CG convergence behavior is partially explained by the following well-known error bound in terms of the *condition number* $\kappa(\boldsymbol{\Phi})$, the ratio of the largest to smallest eigenvalue of $\boldsymbol{\Phi}$.

Theorem 2.2.—Given a positive definite system $\boldsymbol{\Phi}\boldsymbol{\beta} = \boldsymbol{b}$ and a starting vector $\boldsymbol{\beta}_0$, the kth CG iterate $\boldsymbol{\beta}_k$ satisfies the following bound in its $\boldsymbol{\Phi}$-norm distance to the solution $\boldsymbol{\beta}$:

$$\frac{\|\boldsymbol{\beta}_k - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}}{\|\boldsymbol{\beta}_0 - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}} \leq 2\left(\frac{\sqrt{\kappa(\boldsymbol{\Phi})} - 1}{\sqrt{\kappa(\boldsymbol{\Phi})} + 1}\right)^k. \tag{2.6}$$

See Trefethen and Bau (1997) for a proof. Theorem 2.2 guarantees fast convergence of the CG iterates when the condition number is small. On the other hand, a large condition number does not always prevent rapid convergence. This is because CG converges quickly also when the eigenvalues of $\boldsymbol{\Phi}$ are "clustered." The following theorem quantifies this phenomenon, albeit in an idealized situation in which $\boldsymbol{\Phi}$ has exactly $k < p$ distinct eigenvalues.

Theorem 2.3.—If the positive definite matrix $\boldsymbol{\Phi}$ has only $k + 1$ distinct eigenvalues, then CG yields an exact solution within $k + 1$ iterations. In particular, the result holds if $\boldsymbol{\Phi}$ is a rank-k perturbation of an identity that is, $\boldsymbol{\Phi} = \boldsymbol{F}\boldsymbol{F}^{\mathsf{T}} + \boldsymbol{I}$ for $\boldsymbol{F} \in \mathbb{R}^{p \times k}$.

See Golub and Van Loan (2012) for a proof.

Theorems 2.2 and 2.3 are arguably the most famous results on the convergence property of CG, perhaps because their conclusions are clear-cut and easy to understand. These results, however, fall short of capturing the most important aspects of CG convergence behavior in practice. To address this problem, we bring together the most useful of the known results scattered around the numerical linear algebra literature and summarize them as the following rule of thumb. All the statements below are made mathematically precise in Appendix B.

Rule of Thumb 2.4.—Suppose that the eigenvalues $\nu_p(\boldsymbol{\Phi}) \leq \cdots \nu_1(\boldsymbol{\Phi})$ of $\boldsymbol{\Phi}$ are clustered in the interval $[\nu_{p-s}, \nu_r]$ except for a small fraction of them. Then CG effectively "removes" the outlying eigenvalues exponentially quickly. Its convergence rate subsequently accelerates as if the condition number in Equation (2.6) is replaced by the effective value ν_1/ν_{p-s}. The r largest eigenvalues are removed within r iterations, while the same number of smallest eigenvalues tends to delay convergence longer.

2.4. Preconditioning Linear System to Accelerate CG Convergence

A *preconditioner* is a positive definite matrix $\boldsymbol{M}$ chosen so that the *preconditioned system*

$$\widetilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{\beta}} = \tilde{\boldsymbol{b}} \text{ for } \widetilde{\boldsymbol{\Phi}} = \boldsymbol{M}^{-1/2}\boldsymbol{\Phi}\boldsymbol{M}^{-1/2} \text{ and } \tilde{\boldsymbol{b}} = \boldsymbol{M}^{-1/2}\boldsymbol{b} \tag{2.7}$$

leads to faster convergence of the CG iterates. In practice, the algorithm can be implemented so that only the operation $\boldsymbol{v} \rightarrow \boldsymbol{M}^{-1}\boldsymbol{v}$, and not $\boldsymbol{M}^{-1/2}$, is required to solve the preconditioned system (2.7) via CG (Golub and Van Loan 2012). This *preconditioned CG* algorithm still returns a solution $\boldsymbol{\beta}_k = \boldsymbol{M}^{-1/2}\tilde{\boldsymbol{\beta}}_k$ in terms of the original system.

In light of Rule of Thumb 2.4, an effective preconditioner should modify the eigenvalue structure of $\boldsymbol{\Phi}$ so that the preconditioned matrix $\widetilde{\boldsymbol{\Phi}}$ has more tightly clustered eigenvalues except for a small number of outlying ones. Larger outlying eigenvalues are preferable over smaller ones, as smaller ones cause a more significant delay in CG convergence. Additionally, a choice of a preconditioner must take into consideration (a) the one-time cost of computing the preconditioner $\boldsymbol{M}$ and (b) the cost of operation $\boldsymbol{v} \rightarrow \boldsymbol{M}^{-1}\boldsymbol{v}$ during each CG iteration.

In the contexts of Bayesian sparse regression, the linear system (2.4) admits a deceptively simple yet highly effective preconditioner. As it turns out, the choice

$$\boldsymbol{M} = \tau^{-2}\boldsymbol{\Lambda}^{-2}$$

yields a modified system (2.7) with an eigenvalue structure ideally suited to CG. With a slight abuse of terminology, we call it the *prior preconditioner* since it corresponds to the precision of $\boldsymbol{\beta} \mid \tau, \lambda, \boldsymbol{\omega}(\overset{d}{=} \boldsymbol{\beta} \mid \tau, \lambda)$ before observing $\boldsymbol{y}$ and $\boldsymbol{X}$. Most existing preconditioners require explicit access to the elements of $\boldsymbol{\Phi}$ for their constructions (Golub and Van Loan

2012) and are thus useless when computing $\boldsymbol{\Phi}$ itself is a bottleneck. Arguably the only reasonable alternative here is the Jacobi preconditioner $\boldsymbol{M} = \mathrm{diag}(\Phi_{11}, \ldots, \Phi_{pp})$, known as one of the most effective for $\boldsymbol{\Phi}$ with large diagonals. Our numerical results clearly show superior performances of the prior preconditioner, however (Sections 3.3 and 4.4).

Noting that the prior-preconditioned matrix is given by

$$\widetilde{\boldsymbol{\Phi}} = \tau^2 \boldsymbol{\Lambda} \boldsymbol{X}^\top \boldsymbol{\Omega} \boldsymbol{X} \boldsymbol{\Lambda} + \boldsymbol{I}_p, \tag{2.8}$$

we can heuristically motivate the preconditioner as follows. When employing the shrinkage prior, we expect posterior draws of $\tau\boldsymbol{\lambda}$ to satisfy $\tau\lambda_j \approx 0$ except for a relatively small subset $\{j_1, \ldots, j_k\}$ of $j = 1, \ldots, p$. The (i, j)th entry of the matrix $\tau^2 \boldsymbol{\Lambda} \boldsymbol{X}^\top \boldsymbol{\Omega} \boldsymbol{X} \boldsymbol{\Lambda}$ is given by

$$\left(\tau^2 \boldsymbol{\Lambda} \boldsymbol{X}^\top \boldsymbol{\Omega} \boldsymbol{X} \boldsymbol{\Lambda}\right)_{i,j} = (\tau\lambda_i)(\tau\lambda_j)\left(\boldsymbol{X}^\top \boldsymbol{\Omega} \boldsymbol{X}\right)_{ij},$$

which is small when either $\tau\lambda_i \approx 0$ or $\tau\lambda_j \approx 0$. Hence, the entries of $\tau^2 \boldsymbol{\Lambda} \boldsymbol{X}^\top \boldsymbol{\Omega} \boldsymbol{X} \boldsymbol{\Lambda}$ are small away from the $k \times k$ block corresponding to the indices $\{j_1, \ldots, j_k\}$. In general, smaller entries of a matrix have less contributions to the eigenvalue structures of the entire matrix (Golub and Van Loan 2012). This means that the prior-preconditioned matrix (2.8) can be thought of as a perturbation of the identity with a matrix of approximate low-rank structure.[1] As such, $\widetilde{\boldsymbol{\Phi}}$ can be expected to have eigenvalues clustered around 1, except for a small number of larger ones.

Alternatively, we can also motivate the prior-preconditioner as follows. Bayesian sparse regression achieves posterior sparsity because the shrinkage prior dominates the likelihood for all but a small number of coefficients. In other words, the posterior looks a lot like the prior except in a small number of directions. As explained in Section S3 of supplementary materials, this phenomenon translates into the eigenvalues of the prior-preconditioned matrix $\widetilde{\boldsymbol{\Phi}}$ clustering around 1. Since this heuristics is based on expected behavior of a posterior under a strongly informative prior in general, it suggests that prior-preconditioning may be applicable beyond the sparse regression context, for example, to a Gaussian process model like that of Zhang, Datta, and Banerjee (2019).

2.5. Theory of Prior-Preconditioning and Role of Posterior Sparsity

We now formally quantify the eigenvalue structure of the matrix (2.8).

Theorem 2.5.—Let $\lambda_{(k)} = \lambda_{jk}$ denote the kth largest element of $\{\lambda_1, \ldots, \lambda_p\}$. The eigenvalues of the prior-preconditioned matrix (2.8) satisfies

$$1 \le v_k(\widetilde{\boldsymbol{\Phi}}) \le 1 + \tau^2 \lambda_{(k)}^2 v_1\left(\boldsymbol{X}^\top \boldsymbol{\Omega} \boldsymbol{X}\right)$$

[1] It is too naive, however, to deduce that we obtain a good approximation to $\widetilde{\boldsymbol{\Phi}}$ by zeroing out $\tau\lambda_j$'s below some threshold. We show in Section S9 of supplementary materials, that such approximation is typically of a poor quality.

for $k = 1, \ldots, p$. In fact, the following more general bounds hold. Let $\boldsymbol{A}_{(-k)}$ denote the $(p - k) \times (p - k)$ submatrix of a given matrix $\boldsymbol{A}$ corresponding to the row and column indices $j_{k+1}, \ldots, j_p$. With this notation, we have

$$1 \leq v_{k+\ell}(\widetilde{\boldsymbol{\Phi}}) \leq 1 + \tau^2 \lambda_{(k)}^2 v_{\ell+1}\left(\left(\boldsymbol{X}^\mathsf{T}\boldsymbol{\Omega}\boldsymbol{X}\right)_{(-k)}\right) \leq 1 + \tau^2 \lambda_{(k)}^2 v_{\ell+1}\left(\boldsymbol{X}^\mathsf{T}\boldsymbol{\Omega}\boldsymbol{X}\right) \tag{2.9}$$

for any $k \geq 1$ and $\ell \geq 0$ such that $1 \leq k + \ell \leq p$.

Theorem 2.5 guarantees tight clustering of the eigenvalues of the prior-preconditioned matrix—and hence rapid convergence of CG—when most of $\tau\lambda_j$'s are close to zero. We can also relate the prior-preconditioned CG approximation error directly to the decay rate in $\tau\lambda_{(k)}$'s:

Theorem 2.6.—The prior-preconditioned CG applied to (2.4) yields iterates satisfying the following bound for any $m, m' \geq 0$:

$$\frac{\|\boldsymbol{\beta}_{m+m'} - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}}{\|\boldsymbol{\beta}_0 - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}} \leq 2\left(\frac{\tilde{\kappa}_m^{1/2} - 1}{\tilde{\kappa}_m^{1/2} + 1}\right)^{m'} \text{ where } \tilde{\kappa}_m = 1 + \min_{k+\ell=m} \tau^2 \lambda_{(k+1)}^2 v_{\ell+1}\left(\left(\mathbf{X}^\mathsf{T}\boldsymbol{\Omega}\boldsymbol{X}\right)_{(-k)}\right). \tag{2.10}$$

See Appendix A for proofs of Theorems 2.5 and 2.6.

To illustrate the implication of Theorem 2.6 in concrete terms, suppose that a posterior draw $\boldsymbol{\tau}, \boldsymbol{\lambda}, \boldsymbol{\omega}$ satisfies $\tau^2 \lambda_{(m+1)}^2 v_1\left(\boldsymbol{X}^\mathsf{T}\boldsymbol{\Omega}\boldsymbol{X}\right) \leq 100$ for some m. In this case, we have $\left(\tilde{\kappa}_m^{1/2} - 1\right)/\left(\tilde{\kappa}_m^{1/2} + 1\right) \leq -0.086$. So the bound of Theorem 2.6 implies

$$\frac{\|\boldsymbol{\beta}_{m+m'} - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}}{\|\boldsymbol{\beta}_0 - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}} \leq 2 \cdot 10^{-0.086 m'}.$$

After m+100 iterations, therefore, the CG approximation error in the $\boldsymbol{\Phi}$-norm is guaranteed to be reduced by a factor of $2 \cdot 10^{-8.6} \approx 10^{-8.3}$ relative to the initial error.

We have so far stated our theoretical results in purely linear algebraic languages. We now summarize our discussions in a more statistical language, providing a practical guideline on the CG sampler performance in the sparse regression context.

Rule of Thumb 2.7.—The prior-preconditioned CG applied to the linear system (2.4) converges rapidly when the posterior of $\boldsymbol{\beta}$ concentrates on sparse vectors. As the sparsity of $\boldsymbol{\beta}$ increases, the convergence rate of the CG sampler also increases.

The statements above are born out by illustrative examples of Section 3 using synthetic sparse regression posteriors. As we have seen, the statements can be made more precise in terms of the decay rate in the ordered statistics $\tau\lambda_{(k)}$ of a posterior sample $\tau\boldsymbol{\lambda}$ (Rule of Thumb 2.4, Theorems 2.5, and 2.6). We also note that, while our theoretical results hold for any values of $\boldsymbol{\omega}$, τ, $\boldsymbol{\lambda}$, and $\boldsymbol{b}$, these quantities are random within a sparse regression Gibbs sampler. Even with substantial variation in these random quantities, however, we consistently observe fast decay in all $\tau\lambda_{(k)}$ and rapid CG convergence at every iteration. In fact, we rarely observe a deviation of more than 5%–10% from the average number of CG iterations at stationarity—see Section S8 of supplementary materials.

2.6. Computational Complexity of Prior-preconditioned CG

Based on the discussion of Section 2.5, we may crudely quantify the number of prior-preconditioned CG iterations required for updating $\boldsymbol{\beta}$ within a sparse regression Gibbs sampler as $O(s)$, where s is the number of $\tau\lambda_j$'s—and hence of $\boldsymbol{\beta}_j$'s—significantly away from 0. As the cost of each CG iteration is dominated by the operations $\boldsymbol{v} \to \boldsymbol{X}\boldsymbol{v}$ and $\boldsymbol{w} \to \boldsymbol{X}^\top\boldsymbol{w}$, both of which require $O(np)$ floating point operations, the $O(s)$ CG iterations translate to the overall computational complexity of $O(nps)$. The cost of prior-preconditioned CG thus can be far smaller than the $O(np^2 + p^3)$ cost of the standard method as $s \ll p$ in many applications.[2]

2.7. Practical Details on Deploying CG for Sparse Regression

While prior-preconditioning is undoubtedly the most essential ingredient, there remain a few more important details in applying the CG sampler to sparse regression posterior computation. These are (a) a choice of the initial CG vector $\boldsymbol{\beta}_0$, (b) a termination criterion for CG, and (c) handling of regression coefficients with uninformative priors. We discuss them briefly here and defer more thorough discussions to Section S4 of supplementary materials.

A choice of the initial vector has little effect on the eventual exponential convergence rate of CG and, while not to be neglected, is nowhere as consequential as that of the preconditioner (Meurant 2006). In fact, we find that any reasonable choice such as $\boldsymbol{\beta}_0 = \boldsymbol{0}$ works fine in our numerical results, with more elaborate choices resulting in $\lesssim$10% improvement in performance (Section S4.1 of supplementary materials).

In its typical applications, CG is terminated when the ℓ^2-norm of the residual $\boldsymbol{r}_k = \boldsymbol{\beta}_k - \boldsymbol{b}$ falls below some prespecified threshold. Utility of $\|\boldsymbol{r}_k\|$ as an error metric is dubious for the purpose of the CG sampler, however. We instead propose the prior-preconditioned residual $\tilde{\boldsymbol{r}}_k = \widetilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{\beta}}_k - \tilde{\boldsymbol{b}}$ as a more tailored alternative, its squared norm being an approximate upper bound to $\sum_j \xi_j^{-2}(\boldsymbol{\beta}_k - \boldsymbol{\beta})_j^2$ with $\xi_j^2 = \mathbb{E}\left[\beta_j^2 \mid \boldsymbol{\omega}, \lambda, \tau, \boldsymbol{y}, \boldsymbol{X}\right]$ (Section S4.2 of supplementary materials). Specifically, we use and validate the termination criterion $p^{-1/2}\|\tilde{\boldsymbol{r}}_k\|_2 \leq 10^{-6}$ in our numerical studies.

[2]While this is a useful qualitative comparison, we also note that the number of floating point operations is an imperfect proxy for the actual computing time on modern hardware. See Section S7.3 of supplementary materials.

When preconditioning CG, regression coefficients with uninformative priors, such as the intercept, must be handled differently from those under shrinkage. We can accommodate such coefficients by augmenting the prior-preconditioner with another diagonal matrix. We analyze the eigenvalues of the resulting preconditioned matrix and show that, by virtue of CG's ability to quickly remove the outlying eigenvalues (Rule of Thumb 2.4), the convergence rate remains fast and is robust to the precise choice of the diagonal matrix (Section S4.3 of supplementary materials).

3. Simulation Study of CG Sampler Performance

We study the CG sampler performance when applied to actual posterior conditional distributions of the form (1.2). We specifically focus on the prior-preconditioned CG's performance in solving the linear system (2.4) since this directly translates into the performance of the CG-accelerated Gibbs sampler.[3] We simulate data with varying numbers of nonzero coefficients and confirm how sparsity in regression coefficients translates into faster CG convergence as predicted by Theorem 2.5 and Rule of Thumb 2.7. We also illustrate how the convergence rates are affected by different preconditioning strategies and by corresponding eigenvalue distributions of the preconditioned matrices.

3.1. Choice of Shrinkage Prior: Bayesian Bridge

Among existing global-local shrinkage priors, we adopt the Bayesian bridge prior of Polson, Scott, and Windle (2014) as the corresponding Gibbs sampler allows for collapsed updates of τ to improve mixing. The Bayesian bridge Gibbs sampler is in fact uniformly ergodic when the prior tails are properly modified (Nishimura and Suchard 2022).

Under the Bayesian bridge, the local scale λ_j's are given a prior $\pi(\lambda_j) \propto \lambda_j^{-2}\pi_{\mathrm{st}}\left(\lambda_j^{-2}/2\right)$ where $\pi_{\mathrm{st}}(\cdot)$ is an alpha-stable distribution with index of stability $\alpha/2$. The corresponding prior on $\beta_j \mid \tau$, when λ_j is marginalized out, is

$$\pi(\beta_j \mid \tau) \propto \tau^{-1}\exp\left(-|\beta_j/\tau|^{\alpha}\right).$$

The distribution of $\beta_j \mid \tau$ becomes "spikier" as $\alpha \to 0$, placing greater mass around 0 while inducing heavier tails. In typical applications, the data favors the values $\alpha < 1$ but only weakly identifies α (Polson, Scott, and Windle 2014), so in this article we simply fix $\alpha = 1/2$ except when a smaller value seems warranted; see Section 4.3.

3.2. Experimental Set-up

We generate synthetic data of sample size $n = 25{,}000$ with the number of predictors $p = 10{,}000$. In constructing a design matrix $\boldsymbol{X}$, we emulate a model from factor analysis (Jolliffe 2002). We first sample a set of $m = 99$ orthonormal vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m \in \mathbb{R}^p$ uniformly from a Stiefel manifold. We then set the predictor $\boldsymbol{x}_i$ for the ith observation as

[3] We confirm in Section S6 of supplementary materials that samples generated by the CG sampler is statistically indistinguishable from those generated by the direct linear algebra method. Also in Section S6 of supplementary materials, we show how the CG sampler's performance demonstrated here translates into actual gains in terms of computing time.

$$\boldsymbol{x}_i = \sum_{\ell=1}^{99} f_{i,\ell}\boldsymbol{u}_\ell + \epsilon_i \text{ for } f_{i,\ell} \sim \mathcal{N}\left(0, (100 - \ell + 1)^2 - 1\right) \text{ and } \epsilon_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_p). \tag{3.11}$$

This is equivalent to sampling $\boldsymbol{x}_i \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^\mathsf{T}\right)$ for a diagonal matrix $\boldsymbol{D}$ with $\sqrt{D_{\ell\ell}} = \max\{100 - \ell + 1, 1\}$ and orthonormal matrix $\boldsymbol{U}$ sampled uniformly from the space of orthonormal matrices. We then center and standardize the predictors as is commonly done before applying sparse regression (Hastie, Tibshirani, and Friedman 2009).

The above process yields a design matrix $\boldsymbol{X}$ with moderate correlations among the p predictors—the distribution of pairwise correlations is approximately Gaussian centered around 0 with the standard deviation of 0.13. Based on this design matrix $\boldsymbol{X}$, we simulate three different binary outcome vectors by varying the number of nonzero regression coefficients. More specifically, we consider a sparse regression coefficient $\boldsymbol{\beta}_{\text{true}}$ with $\beta_{\text{true},j} = \mathbb{1}\{j \le s\}$ with varying numbers of signals $s = 10$, 20, and 50. In all three scenarios, the binary outcome $\boldsymbol{y}$ is generated from the logistic model as $y_i \mid \boldsymbol{\beta}_{\text{true}}, \boldsymbol{x}_i \sim \text{Bernoulli}(p_i)$ for $\text{logit}(p_i) = \boldsymbol{x}_i^\mathsf{T}\boldsymbol{\beta}_{\text{true}}$.

For each synthetic dataset, we obtain a posterior sample of $\boldsymbol{\omega}, \tau, \boldsymbol{\lambda} \mid \boldsymbol{y}, \boldsymbol{X}$ by running the Polya-Gamma augmented Gibbs sampler with the brute-force direct linear algebra to sample $\boldsymbol{\beta}$ from its conditional distribution (1.2). We confirm the convergence of the Markov chain by examining the traceplot of the posterior log-density of $\boldsymbol{\beta}, \tau \mid \boldsymbol{y}, \boldsymbol{X}$. Having obtained a posterior sample $(\boldsymbol{\omega}, \tau, \boldsymbol{\lambda})$, we sample the vector $\boldsymbol{b}$ as in (2.3) and apply CG to the linear system (2.4). We compare the CG iterates $\{\boldsymbol{\beta}_k\}_{k \ge 0}$ to the exact solution $\boldsymbol{\beta}_{\text{direct}}$ obtained by solving the same system with the Cholesky-based direct method. We repeat this process for eight random replications of the right-hand vector $\boldsymbol{b}$.

3.3. Results

3.3.1. Convergence Rates and Eigenvalue Distributions—Figure 1 shows the CG approximation error as a function of the number of CG iterations, whose cost is dominated by matrix-vector multiplications $\boldsymbol{v} \to \boldsymbol{\Phi}\boldsymbol{v}$. We characterize the approximation error as the relative error $|(\boldsymbol{\beta}_k - \boldsymbol{\beta}_{\text{direct}})_j / (\boldsymbol{\beta}_{\text{direct}})_j|$ averaged across all the coefficients. Each line on the plot shows the geometric average of this error metric over the eight random replications of $\boldsymbol{b}$. The CG convergence behavior observed here remains qualitatively similar regardless of the error metric choice and varies little across the different right-hand vectors; see Section S5.1 of supplementary materials. We also observe there that, while the error $|(\boldsymbol{\beta}_k - \boldsymbol{\beta}_{\text{direct}})_j / (\boldsymbol{\beta}_{\text{direct}})_j|$ varies substantially across the index j, the coefficient-specific errors all decay at roughly uniform rates as a function of the number of CG iterations.

We first focus on the approximation errors under the prior preconditioner, indicated by the lines with circles. After $k \ll p = 10{,}000$ matrix-vector operations, the distance between $\boldsymbol{\beta}_k$ and $\boldsymbol{\beta}_{\text{direct}}$ is already orders of magnitudes smaller than typical Monte Carlo errors.

With additional CG iterations, the distance reaches the machine precision level; notice the eventual "plateaus" achieved under the prior preconditioner in the $s = 10$ and $s = 20$ cases.

Figure 1 also shows the approximation errors under the Jacobi preconditioner $\boldsymbol{M} = \mathrm{diag}(\Phi_{11}, \ldots, \Phi_{pp})$ which, as discussed in Section 2.4, is the only reasonable alternative when using the CG sampler for the applications considered in this article. The prior preconditioner is clearly superior, with the difference in convergence speed more pronounced when true regression coefficients are sparser. Studying the eigenvalue distributions of the respective preconditioned matrices provides further insight into the observed convergence behaviors. Figure 2(a) and (b) show the eigenvalue distributions of the preconditioned matrices based on a posterior sample from the synthetic data with $s = 10$ and $s = 50$. The trimmed version of the histograms highlight the tails of the distributions. The prior preconditioner induces the distribution with a tight cluster around 1 (or 0 in the $\log_{10}$ scale) with a relatively small number of large ones, confirming the theory developed in Section 2.5. On the other hand, the Jacobi preconditioner induces a more spread-out distribution, problematically introducing quite a few small eigenvalues that delay the CG convergence (Rule of Thumb 2.4).

3.3.2. Relationship between Convergence Rate and Posterior Sparsity—

Finally, we turn our attention to the relationship, as seen in Figure 1, between CG convergence rate and sparsity in the underlying true regression coefficients. The convergence is clearly quicker when the true regression coefficients are sparser. To understand this relationship, it is informative to look at the values of $\tau\lambda_j = \mathrm{var}(\boldsymbol{\beta}_j \mid \tau, \boldsymbol{\lambda})^{1/2}$ drawn from the respective posterior distributions. Figure 3 plots the values of $\tau\lambda_j$ for $j = 1, \ldots, 250$ corresponding to the first 250 coefficients. We use two different y-scales for $s = 10$ and $s = 50$, shown on the left and right, respectively, to facilitate qualitative comparison between the two cases. As expected, the posterior sample from the synthetic data with a larger number of signals has a larger number of $\tau\lambda_j$'s away from zero. These relatively large $\tau\lambda_j$'s contribute to the delayed convergence of CG (Theorem 2.5 and Rule of Thumb 2.4).

A more significant cause of the delay, however, is the fact that the shrinkage prior yields weaker shrinkage on the zero coefficients when there are a larger number of signals. With a close look at Figure 3, one can see that $\tau\lambda_1, \ldots, \tau\lambda_s$ corresponding to the true signals are not as well separated from the rest of $\tau\lambda_j$'s when $s = 50$. In fact, the histograms on the left of Figure 4 shows that the distribution of $\tau\lambda_j$'s for $s = 50$ are shifted toward larger values compared to that for $s = 10$. This is mostly due to the posterior distribution of τ concentrating around a larger value—the value of the posterior sample is $\tau \approx 2.0 \times 10^{-3}$ for the $s = 10$ case while $\tau \approx 6.7 \times 10^{-3}$ for the $s = 50$ case.

It is also worth taking a closer look at the tail of the distribution of $\tau\lambda_j$'s. The histograms on the right of Figure 4 show the distribution of the 250 largest $\tau\lambda_j$'s. The figure makes it clear that $\tau\lambda_j$'s corresponding to the true signals are much more well separated from the rest when $s = 10$. Overall, the slower decay in the largest values of $\tau\lambda_j$'s results in the eigenvalues of the preconditioned matrices having a less tight cluster around 1; compare the eigenvalue distributions of Figure 2(a) and (b).

3.3.3. Comments on Generalizability of Conclusions from Simulation Study

—We conclude by noting that the convergence rate of the CG sampler is also a function of signal strengths and correlation among the predictors, both of which affect the posterior sparsity in regression coefficients. For example in the propensity score model application of Section 4, despite 82 regression coefficients having posterior means of substantial magnitudes, the prior-preconditioned CG converges after 107–120 iterations in 95% of the cases. We also confirm that, when using a synthetic design matrix with independent columns, the CG sampler demonstrates much faster rates of convergence for the same numbers of signals (Section S5.2 of supplementary materials). On the other hand, a synthetic design matrix with correlation structure more extreme than (3.11) leads to slower convergence rates for the same numbers of signals (Section S5.3 of supplementary materials). Finally, the posterior sparsity structure, and hence the CG sampler's performance, is also affected by a choice of shrinkage prior. How this choice affects the posterior sparsity is difficult to quantify. The additional simulation studies using different priors (Sections S6.3 and S8.3 of supplementary materials) indicate, however, that the main takeaway holds regardless: the sparser the posterior, the faster the CG sampler's convergence.

4. Application: Comparison of Alternative Treatments

In this section, we demonstrate the magnitude of speed-up delivered by CG-acceleration in modern large-scale applications. We apply Bayesian sparse logistic regression to conduct a comparative study of two blood anti-coagulants *dabigatran* and *warfarin*. The goal of the study is to quantify which of the two drugs have a lower risk of a potential side effect, gastrointestinal bleeding. This question has previously been investigated by Graham et al. (2015) and our analysis yields clinical findings consistent with theirs (Section 4.5).

We are particularly interested in Bayesian sparse regression as a tool for the Observational Health Data Sciences and Informatics (OHDSI) collaborative (Hripcsak et al. 2015). We therefore follow the OHDSI protocol in preprocessing of the data as well as in estimating the treatment effect. In particular, sparse regression plays a critical role in eliminating handpicking of confounding factors and of subgroups for testing treatment effect heterogeneity; this enables the application of a reproducible and consistent statistical estimation procedure to tens of thousands of observational studies (Tian, Schuemie, and Suchard 2018; Schuemie et al. 2020).

4.1. Dataset

We extract patient-level data from Truven Health MarketScan Medicare Supplemental and Coordination of Benefits Database. In the database, we find $n = 72{,}489$ patients who became first-time users of either dabigatran or warfarin after diagnosis of atrial fibrillation. Among them, 19,768 are treated with dabigatran and the rest with warfarin. There are $p = 98{,}118$ predictors, consisting of clinical measurements, preexisting conditions, as well as prior treatments and administered drugs—all measured before exposure to dabigatran or warfarin. Following the OHDSI protocol, we screen out the predictors observed in less than 0.1% of the cohort. This reduces the number of predictors to $p = 22{,}175$. The precise definition

of the cohort can be found in supplementary materials as well as at https://github.com/aki-nishimura/anticoagulant-study-cohorts.

Each patient is affected by only a small fraction of the potential preexisting conditions and available treatments. The design matrix $\boldsymbol{X}$ therefore is sparse, with only 4% of the entries being nonzero. Another noteworthy feature of the data is the low incidence rates of gastrointestinal bleeding; the outcome indicator $\boldsymbol{y}$ has nonzero entries $y_i = 1$ for only 713 out of 72,489 patients.

4.2. Statistical Approach: Propensity Score Stratified Regression

To control for covariate imbalances between the dabigatran and warfarin users, we rely on propensity score method in estimating the treatment effect. The procedure involves two logistic models with large numbers of predictors, to deal with which we employ Bayesian sparse regression. We describe the procedure and essential ideas below but refer the readers to Stuart (2010), and the references therein for further details.

Estimation of the treatment effect proceeds in two stages. First, the *propensity score* $\mathbb{P}(T_i = 1 \mid \boldsymbol{x}_i)$ of the treatment assignment to dabigatran is estimated by the logistic model

$$\operatorname{logit}\{\mathbb{P}(T_i = 1 \mid \boldsymbol{x}_i)\} = \beta_0 + \boldsymbol{x}_i^{\mathsf{T}}\boldsymbol{\beta}. \tag{3.12}$$

While not of direct interest within the propensity score method framework, identifying significant predictors of the score is highly relevant in the OHDSI applications. Many of the databases are too small to fit the models with such large numbers of predictors, but the significant heterogeneity among them makes the joint estimation insensible (Hripcsak et al. 2016). Sparse regression provides a tool to screen out the predictors using the larger databases and use only the selected subset to estimate the scores within the smaller databases.

After fitting the model (4.12), the quantiles of the estimated propensity scores are then used to stratify the population into subpopulations of equal sizes. Following a typical recommendation, we choose the number of strata as $M = 5$. Under suitable assumptions, conditioning on the strata indicator removes most of imbalances in the distributions of the predictors between the treatment ($T_i = 1$) and control ($T_i = -1$) groups. After the stratification, we can proceed to estimate the treatment effect via the logistic model without the main effect from the clinical covariate $\boldsymbol{x}_i$ (Tian et al. 2014):

$$\operatorname{logit}\{\mathbb{P}(y_i = 1 \mid \boldsymbol{\alpha}, \boldsymbol{\gamma}, s_i, \boldsymbol{x}_i)\} = \sum_{m=1}^{M} \alpha_m \mathbb{1}\{s_i = m\} + (\alpha_0 + \boldsymbol{x}_i^{\mathsf{T}}\boldsymbol{\gamma})\frac{T_i}{2}, \tag{3.13}$$

where a categorical variable s_i denotes the strata membership of the ith individual. The quantity $\alpha_0 + \boldsymbol{x}^{\mathsf{T}}\boldsymbol{\gamma}$ represents the treatment effect for a patient with covariate $\boldsymbol{x}$, with the

feature x_{ij} contributing to the treatment effect heterogeneity when $\gamma_j \neq 0$. The goal of sparse regression here is to identify such nonzero γj's.

4.3. Prior Choice and Posterior Computation

We fit the models (4.12) and (4.13) using the Bayesian bridge shrinkage prior (Section 3.1). For the main treatment and propensity score strata effects, we place weakly informative $\mathcal{N}(0, 1)$ priors. For the global scale parameter, we use an objective prior $\pi(\tau) \propto 1/\tau$ in the model (4.12) (Berger et al. 2009). For the treatment effect model (4.13), due to the low incidence rate in the outcome, we find the above prior choice to provide insufficient separation of significant predictors from the rest. We therefore use the bridge prior with $\alpha =$ 1/4 and weakly informative conjugate prior $\phi = \tau^{-\alpha} \sim$ Gamma(shape = 339.8, rate = 26.58) so that $\log_{10}(\tau)$ has the prior mean of −1.5 and standard deviation of 0.5.

For posterior computation, we compare two Gibbs samplers that differ only in their methods for drawing $\boldsymbol{\beta}$ from the conditional distribution (1.2). One sampler uses the proposed CG sampler while the other uses a traditional direct method via Cholesky factorization. Sparse Cholesky methods offer no computational benefit here as the precision matrix, despite the sparsity in the design matrix $\boldsymbol{X}$, is almost completely dense (Section S7 of supplementary materials). We refer to the respective samplers as the *CG-accelerated* and *direct* Gibbs sampler. The other conditional updates follow the approaches described in Polson, Scott, and Windle (2014); see Section S1 of supplementary materials for the details.

We implement the Gibbs samplers in Python and run on a 2015 iMac with an Intel Core i7 "Skylake" processor having four cores at 4 GHZ and 32 GB of memory. Linear algebra being the computational bottleneck, both samplers benefit from parallelization and we engage all the four cores. For the linear algebra operations, we interface our Python code with the Intel Math Kernel Library (MKL) implementations of Basic Linear Algebra Subprograms (BLAS) and sparse BLAS, which proved computationally superior to alternatives in our preliminary benchmarking. We use the sparse BLAS for matrix-vector multiplications $\boldsymbol{v} \rightarrow \boldsymbol{X}\boldsymbol{v}$ and $\boldsymbol{w} \rightarrow \boldsymbol{X}^\top \boldsymbol{w}$ within the CG-accelerated Gibbs and for matrix-matrix multiplication $\boldsymbol{X}^\top \boldsymbol{\Omega} \boldsymbol{X}$ within the direct Gibbs. Exploiting the sparsity in $\boldsymbol{X}$ cuts down both computing time and memory usage by an order of magnitude. Details on how we optimized both Gibbs sampler are described in Section S7 of supplementary materials.

We run the Gibbs samplers for 5500 and 11,000 iterations for the propensity score and treatment effect model, discarding the first 500 and 1000 as burn-ins. We confirm their convergences by examining the traceplots of the posterior log-density. We estimate the effective sample sizes (ESS) for all the regression coefficients using the R package coda (Plummer et al. 2006). The smallest ESSs are found among the coefficients with bimodal posteriors, but their traceplots nonetheless indicate reasonable mixing. We find the minimum and median ESS to be 106.2 and 2484 for the propensity score model, and 86.04 and 2496 for the treatment effect model.

4.4. CG Acceleration Magnitudes and Posterior Characteristics

The direct Gibbs sampler requires 106 and 212 hr for the propensity score and treatment effect model. On the other hand, the CG-accelerated sampler finishes in 11.4 and 11.3 hr, yielding 9.3-fold and 18.8-fold speed-ups. For both Gibbs samplers, the total computation times are dominated by the conditional updates of $\boldsymbol{\beta}$. The magnitudes of CG-acceleration thus are determined by the CG convergence rate at each Gibbs iteration.

In agreement with the theory and empirical results of Sections 2.5 and 3.3, the variability in the magnitudes of CG-acceleration can be explained by the posterior sparsity structures of the regression coefficients. For the propensity score model, 82 out of the 22,175 regression coefficients have their posterior mean magnitudes above 0.1, while 18,187 (82.0%) of the coefficients below 0.01. For the treatment effect model, only two of the coefficients have the posterior mean magnitudes above 0.1, while 22,096 (99.6%) below 0.01. We note that the individual posterior samples are much less sparse than the posterior mean. Under the treatment effect model, for example, the number of coefficients with magnitudes above 0.1 typically ranges from 265 to 529 while those below 0.01 from 16,172 to 17,632.

For more in-depth analysis of the CG-acceleration mechanism, in Section S8 of supplementary materials, we examine the CG sampler behavior at each Gibbs iteration. In particular, we verify that the error metric discussed in Section 2.7 works well in deciding when to terminate the CG iteration. We also confirm that the prior preconditioner continues to outperform the Jacobi in this real data example.

4.5. Clinical Conclusions from Dabigatran versus Warfarin Study

The propensity score model finds substantial differences between the patients treated by dabigatran and warfarin. In particular, patients' covariate characteristics are predictive of the treatment assignments as seen in Figure 5. The two most significant predictors are the treatment year and age group. Both predictors have been encoded as binary indicators in the design matrix for simplicity, but the coefficients of categorical and ordinal predictors could have been estimated with shrinkage priors analogous to Bayesian grouped or fused lasso (Kyung et al. 2010; Xu and Ghosh 2015). The posterior mean and 95% credible intervals of the regression coefficients are shown in Figure 6. The figure plots the effect sizes relative to the year 2010 and the age group 65–69; when actually fitting the model, however, we use the most common category as the baseline for categorical variables.

For the treatment effect model, Figure 7(a) shows the posterior distribution of the average treatment effect of dabigatran over warfarin. The posterior indicates an evidence for the lower incidence rate of gastrointestinal bleeding under dabigatran treatment, which is consistent with findings of Graham et al. (2015). Remarkably, our sparse regression model identifies substantial interaction between the treatment and age group 65–69, with effect size potentially large enough to offset the average treatment effect. No other age groups exhibit significant interaction with the treatment. The 65–69 age group being the youngest in our Medicare cohort, our finding suggests a possibility that the relative risk of gastrointestinal bleeding only increases in the older patients. In fact, Graham et al. (2015) reports the risks from dabigatran and warfarin to be comparable for women under 75 and men under 85 years

old. A potential concern with their results is the lack of explanation on their choices of age thresholds. On the other hand, our subgroup detection approach based on sparse regression requires no arbitrary selection of subgroups and thus provides a more datadriven alternative to study treatment effect heterogeneity.

5. Discussion

In this article, we have developed theory and computational techniques to scale Bayesian sparse regression to a typical size of data in modern applications. To our knowledge, our computational approach constitutes the first principled use of CG for the purpose of full Bayesian inference via MCMC. The heuristic described in Section 2.4 suggests that prior-preconditioning may work well in other high-dimensional applications that call for structured, strongly informative priors. For example, the application of CG to a Gaussian process model as explored by Zhang, Datta, and Banerjee (2019) may benefit from prior-preconditioning.

As early as 1997, Gibbs and MacKay emphasized the importance of avoiding expensive linear algebra operations, such as multiplying two matrices or factorizing a matrix, for Bayesian inference to be scalable. Prior to our work, this desiderata had yet to be met for full Bayesian inference of sparse regression models. Moreover, large-yet-sparse design matrices are increasingly common in modern applications; it is thus critical to design computational methods to exploit such sparse structure in the data (Friedman, Hastie, and Tibshirani 2010). Our CG-accelerated Gibbs sampler is an important example to fill these notable gaps in the literature.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

We thank Yuxi Tian and Martijn Schuemie for their help in wrangling the dataset used in Section 4. We also thank Jianfeng Lu and Ilse Ipsen for useful discussions on linear algebra topics.

Funding

This work is partially supported by National Science Foundation grant DMS 1264153, National Institutes of Health grants U19 AI135995 and R01 AI153044, and Food and Drug Administration grant HHS 75F40120D00039.

Appendix A: Proofs

Before we proceed to proving Theorem 2.5, we first derive Theorem 2.6 as its consequence.

Theorem 2.6.

By Theorem B.4, the $(m + m')$th CG iterate $\boldsymbol{\beta}_{m+m'}$ satisfies the bound

$$\frac{\|\boldsymbol{\beta}_{m+m'} - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}}{\|\boldsymbol{\beta}_0 - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}} \leq 2\left(\frac{\sqrt{v_{m+1}/v_p} - 1}{\sqrt{v_{m+1}/v_p} + 1}\right)^{m'},$$

(A.14)

where ν_j denotes the jth largest eigenvalue of $\boldsymbol{\Phi}$. By Theorem 2.5, we know that

$$1 \leq \upsilon_p \leq \upsilon_{m+1} \leq 1 + \min_{k+\ell=m} \tau^2 \lambda^2_{(k+1)} \upsilon_{\ell+1}\left(\left(\boldsymbol{X}^\mathsf{T}\boldsymbol{\Omega}\boldsymbol{X}\right)_{(-k)}\right) = \tilde{\kappa}_m$$

and hence that $\upsilon_{m+1}/\nu_p \leq \tilde{\kappa}_m$. Since the function $\kappa \to (\sqrt{\kappa} - 1)/(\sqrt{\kappa} + 1)$ is increasing in κ, we can upper bound the right-hand side of (A.14) in terms of $\tilde{\kappa}_m$, yielding the desired inequality (2.10). □

Theorem 2.5.

We prove the more general inequality (2.9). The lower bound $1 \leq \upsilon_{k+\ell}(\widetilde{\boldsymbol{\Phi}})$ is an immediate consequence of Proposition A.1. For the upper bound, first note that $\upsilon_{k+\ell}(\widetilde{\boldsymbol{\Phi}}) \leq \upsilon_\ell\left(\widetilde{\boldsymbol{\Phi}}_{(-k)}\right)$ by the Poincaré separation theorem (Theorem A.2). From the expression (2.8) for $\widetilde{\boldsymbol{\Phi}}$, we have

$$\begin{aligned}\upsilon_\ell\left(\widetilde{\boldsymbol{\Phi}}_{(-k)}\right) &= \upsilon_\ell\left(\boldsymbol{I}_k + \tau^2 \boldsymbol{\Lambda}_{(-k)}\left(\boldsymbol{X}^\mathsf{T}\boldsymbol{\Omega}\boldsymbol{X}\right)_{(-k)}\boldsymbol{\Lambda}_{(-k)}\right) \\ &= 1 + \tau^2 \upsilon_\ell\left(\boldsymbol{\Lambda}_{(-k)}\left(\boldsymbol{X}^\mathsf{T}\boldsymbol{\Omega}\boldsymbol{X}\right)_{(-k)}\boldsymbol{\Lambda}_{(-k)}\right),\end{aligned}$$

where the second equality follows from Proposition A.1. Applying Lemma A.3 with $\boldsymbol{A} = (\boldsymbol{X}^\mathsf{T}\boldsymbol{X})_{(-k)}$ and $\boldsymbol{B} = \lambda^{-2}_{(k+1)}\boldsymbol{\Lambda}^2_{(-k)}$, we obtain

$$\upsilon_\ell\left(\widetilde{\boldsymbol{\Phi}}_{(-k)}\right) \leq 1 + \tau^2 \lambda^2_{(k+1)} \upsilon_\ell\left(\left(\boldsymbol{X}^\mathsf{T}\boldsymbol{\Omega}\boldsymbol{X}\right)_{(-k)}\right).$$

Thus, we have shown

$$\upsilon_{k+\ell}(\widetilde{\boldsymbol{\Phi}}) \leq 1 + \tau^2 \lambda^2_{(k+1)} \upsilon_\ell\left(\left(\boldsymbol{X}^\mathsf{T}\boldsymbol{\Omega}\boldsymbol{X}\right)(-k)\right) \leq 1 + \tau^2 \lambda^2_{(k+1)} \upsilon_\ell\left(\boldsymbol{X}^\mathsf{T}\boldsymbol{\Omega}\boldsymbol{X}\right),$$

where the inequality $\nu_k((\boldsymbol{X}^\mathsf{T}\boldsymbol{\Omega}\boldsymbol{X})_{(-k)}) \leq \nu_k(\boldsymbol{X}^\mathsf{T}\boldsymbol{\Omega}\boldsymbol{X})$ follows again from the Poincaré separation theorem. □

Proposition A.1.

Given a $p \times p$ symmetric matrix $\boldsymbol{A}$, the eigenvalues of the matrix $\boldsymbol{I}_p + \boldsymbol{A}$ are given by $1 + \nu_k(\boldsymbol{A})$ for $k = 1, \ldots, p$.

Proof.

The result follows immediately from the spectral theorem for normal matrices (Horn and Johnson 2012). □

Theorem A.2 (Poincaré separation theorem).

For a given $p \times p$ symmetric matrix $\boldsymbol{A}$, let $\boldsymbol{A}_{(-k)}$ denote the submatrix with the first k rows and columns removed from $\boldsymbol{A}$. Then the eigenvalues of $\boldsymbol{A}$ and $\boldsymbol{A}_{(-k)}$ satisfies

$$v_{k+\ell}(\boldsymbol{A}) \leq v_{\ell}(\boldsymbol{A}_{(-k)}) \leq v_{\ell}(\boldsymbol{A}) \text{ for } \ell = 1, \ldots, p-k .$$

Since permuting the rows and columns of $\boldsymbol{A}$ does not change its eigenvalues, the above inequality in fact holds for any submatrix of $\boldsymbol{A}$ obtained by removing k rows and columns of $\boldsymbol{A}$ corresponding to a common set of indices $j_1, \ldots, j_k$.

Proof.

See chapter 4.3 of Horn and Johnson (2012). □

Lemma A.3.

Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be $p \times p$ symmetric positive definite matrices and suppose that the largest eigenvalue of $\boldsymbol{B}$ satisfies $\nu_1(\boldsymbol{B}) \leq 1$. Then we have

$$v_k\left(\boldsymbol{B}^{1/2}\boldsymbol{A}\boldsymbol{B}^{1/2}\right) \leq v_k(\boldsymbol{A}) \text{ for } k = 1, \ldots, p$$

where $\nu_k(\cdot)$ denotes the kth largest eigenvalue of a given matrix.

Proof.

The result follows immediately from Ostrowski's theorem (Theorem 4.5.9 in Horn and Johnson 2012). □

Appendix B: Theories of CG convergence behavior

In this section, we provide mathematical foundations behind the claims made in Rule of Thumb 2.4. In essence, Rule of Thumb 2.4 is our attempt at describing a phenomenon known as the *super-linear* convergence of CG in a quantitative yet accessible manner. While this is a well-known phenomenon among the researchers in scientific computing, it is rarely explained in canonical textbooks and reference books in numerical linear algebra.[4] Here we bring together some of the most practically useful results found in the literature and present them in a concise and self-contained manner. Our presentation in Sections B.1 and B.2 is roughly based on section 5.3 of Van der Vorst (2003) with details modified, added, and condensed as needed. More comprehensive treatment of the known results related to CG is found in Meurant (2006). Kuijlaars (2006) sheds additional light on CG convergence behaviors by studying them from the potential theory perspective.

[4]For example, discussions beyond Theorems 2.2 and 2.3 cannot be found in, to name a few, Trefethen and Bau (1997), Demmel (1997), Saad (2003), and Golub and Van Loan (2012).

Section B.1 explains the critical first step in understanding the convergence of CG applied to a positive definite system $\boldsymbol{\Phi\beta} = \boldsymbol{b}$ — relating the CG approximation error to polynomial interpolation error over the set $\{\nu_1, \ldots, \nu_p\}$ comprising the eigenvalues of . From this perspective, one can understand Theorem 2.2 as a generic and crude bound, ignoring the distributions of νj's in-between the largest and smallest eigenvalues (Theorem B.3). Theorem 2.3 similarly follows from the polynomial approximation perspective.

The effects of the largest eigenvalues on CG convergence, as stated in Rule of Thumb 2.4, is made mathematically precise in Theorem B.4. Analyzing how the smallest eigenvalues delay CG convergence is more involved and requires a discussion of how the eigenvalues are approximated in the Krylov subspace. The amount of initial delay in CG convergence is closely related to how quickly these eigenvalue approximations converge. A precise statement is given in Theorem B.5.

The proofs of all the results stated in this section are provided in Section 10 of supplementary materials.

B.1. CG approximation error as polynomial interpolation error

The space of polynomials $\mathscr{P}_k$ as defined below plays a prominent role in the behavior of a worst-case CG approximation error:

$$\mathscr{P}_k = \{Q_k(v) : Q_k \text{ is a polynomial of degree } k \text{ with } Q_k(0) = 1\}.$$

Proposition B.1 below establishes the connection between CG and the space $\mathscr{P}_k$.

Proposition B.1.

The difference between the kth CG iterate $\boldsymbol{\beta}_k$ and the exact solution $\boldsymbol{\beta}$ can be expressed as

$$\boldsymbol{\beta}_k - \boldsymbol{\beta} = R_k(\boldsymbol{\Phi})(\boldsymbol{\beta}_0 - \boldsymbol{\beta}) \text{ for } R_k = \underset{Q_k \in \mathscr{P}_k}{\operatorname{argmin}} \|Q_k(\boldsymbol{\Phi})(\boldsymbol{\beta}_0 - \boldsymbol{\beta})\|_{\boldsymbol{\Phi}}. \tag{B.15}$$

In particular, the following inequality holds for any $Q_k \in \mathscr{P}_k$:

$$\|\boldsymbol{\beta}_k - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}} \leq \|Q_k(\boldsymbol{\Phi})(\boldsymbol{\beta}_0 - \boldsymbol{\beta})\|_{\boldsymbol{\Phi}}. \tag{B.16}$$

Theorem B.2 below uses Proposition B.1 to establish the relation between the CG approximation error and a polynomial interpolation error. We can interpret the result as saying the following: a worst-case CG approximation error can be quantified via how well the set of points $\{(\nu_j, 0)\}_{j=1,\ldots,p}$ can be interpolated by the graph $\nu \to (\nu, Q_k(\nu))$ of a kth degree polynomial Q_k with the constraint $Q_k(0) = 1$.

Theorem B.2.

$$\frac{\|\boldsymbol{\beta}_k - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}}{\|\boldsymbol{\beta}_0 - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}} \leq \max_{Q_k \in \mathscr{P}_k} \max_{j = 1, \ldots, p} |Q_k(v_j)|, \tag{B.17}$$

where v_j denotes the jth largest eigenvalue of $\boldsymbol{\Phi}$. The bound is sharp in a sense that, for each k, there exists an initial vector $\boldsymbol{\beta}_0$ for which the equality holds.

B.2. Bounding CG error via its polynomial characterization

We now derive bounds on the CG approximation error through its characterization as a polynomial interpolation error (Theorem B.2). Minimizing the interpolation error over the entire interval between the largest and smallest eigenvalues yields the following bound.

Theorem B.3.

$$\min_{Q_k \in \mathscr{P}_k} \max_{v \in [v_{\min}, v_{\max}]} |Q_k(v)| \leq 2\left(\frac{\sqrt{v_{\max}/v_{\min}} - 1}{\sqrt{v_{\max}/v_{\min}} + 1}\right)^k. \tag{B.18}$$

Theorems B.2 and B.3 together yield the well-known CG approximation error bound of Theorem 2.2. As the bound of Theorem B.2 depends only on the maximum over a discrete set of the eigenvalues $\{v_p, \ldots, v_1\}$, rather than the entire interval $[v_p, v_1]$, the actual CG convergence rate can be faster.

Theorem B.4 below is a basis of the following claim made in Rule of Thumb 2.4: "the r largest eigenvalues are effectively removed within r iterations."

Theorem B.4.

The following bound holds for all r, $k \geq 0$ with $r < p$:

$$\frac{\|\boldsymbol{\beta}_{r+k} - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}}{\|\boldsymbol{\beta}_0 - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}} = \min_{Q_{r+k} \in \mathscr{P}_{r+k}} \max_{j = 1, \ldots, p} |Q_{r+k}(v_j)| \leq 2\left(\frac{\sqrt{v_{r+1}/v_p} - 1}{\sqrt{v_{r+1}/v_p} + 1}\right)^k, \tag{B.19}$$

where the first equality is given by Theorem B.2.

The smallest eigenvalues affect the CG convergence rate differently from the largest ones due to the constraint $Q_k(0) = 1$ in $\mathscr{P}_k$. Intuitively, this constraint makes the smallest eigenvalues more significant contributers to the polynomial interpolation error because it competes with the objective of minimizing $|Q_k(v)|$ for small v. This is why we state in Rule of Thumb 2.4 that "the same number of smallest eigenvalues tends to delay the convergence longer." Nonetheless, the effects of the smallest eigenvalues on the CG approximation error

becomes attenuated as the CG iterations proceed. To quantify this phenomenon, we need to introduce the notion of *Ritz values* and describe their roles in the CG convergence behavior.

The Ritz values at the kth CG iteration refer to the roots $\{\hat{\nu}_1^{(k)}, \ldots, \hat{\nu}_k^{(k)}\}$ of the optimal CG polynomial R_k as defined in (B.15). Unless the eigenvalues $\nu_p, \ldots, \nu_1$ are distributed in a highly unusual manner, the largest and smallest Ritz values have a property that they converges quickly to the largest and smallest eigenvalues of $\mathbf{\Phi}$ (Trefethen and Bau 1997; Driscoll, Toh, and Trefethen 1998; Kuijlaars 2006). More precisely, we have $\hat{\nu}_i^{(k)} \to \upsilon_i$ for $i = 1, \ldots, r$ and $\hat{\upsilon}_{k-i}^{(k)} \to \upsilon_{p-i}$ for $i\,0, \ldots, s$ as $k \to p$. While the convergence rates of the Ritz values can be shown to be exponential, in practice quite a large number of CG iterations may be required to obtain good approximations unless $\max\{r, s\} \ll p$ (Saad 2011).

Theorem B.5 below quantifies how the convergence of the Ritz values are related to the subsequent acceleration of the CG convergence rates.

Theorem B.5.

The CG approximation error of the $(k + \ell)$th iterate relative to the kth iterate satisfies the following bound:

$$\frac{\|\boldsymbol{\beta}_{k+\ell} - \boldsymbol{\beta}\|_{\mathbf{\Phi}}}{\|\boldsymbol{\beta}_k - \boldsymbol{\beta}\|_{\mathbf{\Phi}}} \leq C_{k,r,s} 2\left(\frac{\sqrt{\upsilon_{r+1}/\upsilon_{p-s}} - 1}{\sqrt{\upsilon_{r+1}/\upsilon_{p-s}} + 1}\right)^{\ell}, \tag{B.20}$$

where $C_{k,r,s} = C_{k,r,s}\left(\hat{\upsilon}_1^{(k)}, \ldots, \hat{\upsilon}_k^{(k)}\right) \to 1$ as $k \to p$ for any fixed $r, s \geq 0$ with $r + s < p$. More precisely, $C_{k,r,s}$ tends to 1 as the r largest and s smallest Ritz values converge to the largest and smallest eigenvalues of $\mathbf{\Phi}$.

References

Berger JO, Bernardo JM, and Sun D (2009), "The Formal Definition of Reference Priors," The Annals of Statistics, 37, 905–938. [9]

Bhadra A, Datta J, Polson NG, and Willard B (2019), "Lasso Meets Horseshoe: A Survey," Statistical Science, 34, 405–427. [1]

Bhattacharya A, Chakraborty A, and Mallick BK (2016), "Fast Sampling with Gaussian Scale Mixture Priors in High-Dimensional Regression," Biometrika, 103, 985–991. [2,3] [PubMed: 28435166]

Bhattacharya A, Pati D, Pillai NS, and Dunson DB (2015), "Dirichlet–Laplace Priors for Optimal Shrinkage," Journal of the American Statistical Association, 110, 1479–1490. [1] [PubMed: 27019543]

Carvalho CM, Polson NG, and Scott JG (2010), "The Horseshoe Estimator for Sparse Signals," Biometrika, 97, 465–480. [1]

Datta J, and Ghosh JK (2013), "Asymptotic Properties of Bayes Risk for the Horseshoe Prior," Bayesian Analysis, 8, 111–132. [1]

Demmel JW (1997), Applied Numerical Linear Algebra (Vol. 56), Philadelphia, PA: Society for Industrial and Applied Mathematics. [11]

Devroye L (2006), "Nonuniform Random Variate Generation," in Handbooks in Operations Research and Management Science, eds. Henderson SG, and Nelson BL (Vol. 13), pp. 83–121, Amsterdam: Elsevier.

Dongarra J, Heroux MA, and Luszczek P (2016), "High-Performance Conjugate-Gradient Benchmark: A New Metric for Ranking High-Performance Computing Systems," The International Journal of High Performance Computing Applications, 30, 3–10. [2]

Driscoll TA, Toh K-C, and Trefethen LN (1998), "From Potential Theory to Matrix Iterations in Six Steps," SIAM Review, 40, 547–578. [12]

Duff IS, Erisman AM, and Reid JK (2017), Direct Methods for Sparse Matrices, Oxford: Oxford University Press. [2]

Friedman J, Hastie T, and Tibshirani R (2010), "Regularization Paths for Generalized Linear Models via Coordinate Descent," Journal of Statistical Software, 33, 1–22. [11] [PubMed: 20808728]

Gibbs M, and MacKay D (1997), "Effcient Implementation of Gaussian Processes," (unpublished manuscript). [3,11]

Golub GH, and Van Loan CF (2012), Matrix Computations (Vol. 3), Baltimore, MD: Johns Hopkins University Press. [3,4,11]

Graham DJ, Reichman ME, Wernecke M, Zhang R, Southworth MR, Levenson M, Sheu T-C, Mott K, Goulding MR, Houstoun M, MaCurdy TE, Worrall C, and Kelman JA (2015), "Cardiovascular, Bleeding, and Mortality Risks in Elderly Medicare Patients Treated with Dabigatran or Warfarin for nonvalvular atrial Fibrillation," Circulation, 131, 157–164. [8,10] [PubMed: 25359164]

Guntheroth K (2016), Optimized C++: Proven Techniques for Heightened Performance, Newton, MA: O'Reilly Media, Inc.

Hahn PR, He J, and Lopes HF (2018), "Efficient Sampling for Gaussian Linear Regression with Arbitrary Priors," Journal of Computational and Graphical Statistics, 28, 142–154. [2]

Hastie T, Tibshirani R, and Friedman J (2009), The Elements of Statistical Learning, Springer Series in Statistics, New York: Springer. [6]

Hestenes MR, and Stiefel E (1952), "Methods of Conjugate Gradients for Solving Linear Systems," Journal of Research of the National Bureau of Standards, 49, 409–435. [3]

Horn RA, and Johnson CR (2012), Matrix Analysis, Cambridge: Cambridge University Press. [11]

Hripcsak G, Duke JD, Shah NH, Reich CG, Huser V, Schuemie MJ, Suchard MA, Park RW, Wong ICK, Rijnbeek PR, van der Lei J, Pratt N, Niklas Norén G, Li Y-C, Stang PE, Madigan D, and Ryan PB (2015), "Observational Health Data Sciences and Informatics (OHDSI): Opportunities for Observational Researchers," Studies in Health Technology and Informatics, 216, 574–578. [8] [PubMed: 26262116]

Hripcsak G, Ryan PB, Duke JD, Shah NH, Park RW, Huser V, Suchard MA, Schuemie MJ, DeFalco FJ, Perotte A, Banda JM, Reich CG, Schilling LM, Matheny ME, Meeker D, Pratt N, and Madigan D (2016), "Characterizing Treatment Pathways at Scale Using the OHDSI Network," Proceedings of the National Academy of Sciences, 113, 7329–7336. [9]

Johndrow J, Orenstein P, and Bhattacharya A (2020), "Scalable Approximate MCMC Algorithms for the Horseshoe Prior," Journal of Machine Learning Research, 21, 1–61. [2] [PubMed: 34305477]

Jolliffe IT (2002), Principal Component Analysis, Springer Series in Statistics, New York: Springer. [6]

Kuijlaars ABJ (2006), "Convergence Analysis of Krylov Subspace Iterations with Methods from Potential Theory," SIAM Review, 48, 3–40. [11,12]

Kyung M, Gill J, Ghosh M, Casella G (2010), "Penalized Regression, Standard Errors, and Bayesian Lassos," Bayesian Analysis, 5, 369–411. [10]

Lanczos C (1952), "Solution of Systems of Linear Equations by Minimized Iterations," Journal of Research of the National Bureau of Standards, 49, 33–53. [3]

Meurant GA (2006), The Lanczos and Conjugate Gradient Algorithms: From Theory to Finite Precision Computations, Philadelphia, PA: Society for Industrial and Applied Mathematics. [5,11]

Nishimura A, and Suchard MA (2022), "Shrinkage with Shrunken Shoulders: Gibbs Sampling Shrinkage Model Posteriors with Guaranteed Convergence Rates," Bayesian Analysis. DOI:10.1214/22-BA1308 [6]

Pal S, and Khare K (2014), "Geometric Ergodicity for Bayesian Shrinkage Models," Electronic Journal of Statistics, 8, 604–645. [1]

Plummer M, Best N, Cowles K, and Vines K (2006), "Coda: Convergence Diagnosis and Output Analysis for MCMC," R News, 6, 7–11. [9]

Polson NG, Scott JG, and Windle J (2013), "Bayesian Inference for Logistic Models Using Pólya–Gamma Latent Variables," Journal of the American Statistical Association, 108, 1339–1349. [2]

— (2014), "The Bayesian Bridge," Journal of the Royal Statistical Society, Series B, 76, 713–733. [1,6,9]

Rue H, and Held L (2005), Gaussian Markov Random Fields: Theory and Applications, Boca Raton, FL: CRC Press. [2]

Saad Y (2003), Iterative Methods for Sparse Linear Systems (Vol. 82), Philadelphia, PA: Society for Industrial and Applied Mathematics. [2,11]

— (2011), Numerical Methods for Large Eigenvalue Problems: Revised Edition, Classics in Applied Mathematics, Philadelphia, PA: Society for Industrial and Applied Mathematics. [12]

Schuemie MJ, Ryan PB, Hripcsak G, Madigan D, and Suchard MA (2018), "Improving Reproducibility by Using High-Throughput Observational Studies with Empirical Calibration," Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 376, 20170356. [1]

Schuemie MJ, Ryan PB, Pratt N, Chen R, You SC, Krumholz HM, Madigan D, Hripcsak G, and Suchard MA (2020), "Large-Scale Evidence Generation and Evaluation Across a Network of Databases (LEGEND): Assessing Validity Using Hypertension as a Case Study," Journal of the American Medical Informatics Association, 27, 1268–1277. [8] [PubMed: 32827027]

Stein ML, Chen J, and Anitescu M (2012), "Difference Filter Preconditioning for Large Covariance Matrices," SIAM Journal on Matrix Analysis and Applications, 33, 52–72. [3]

Stroud JR, Stein ML, and Lysen S (2017), "Bayesian and Maximum Likelihood Estimation for Gaussian Processes on an Incomplete Lattice," Journal of Computational and Graphical Statistics, 26, 108–120. [3]

Stuart EA (2010), "Matching Methods for Causal Inference: A Review and a Look Forward," Statistical Science, 25, 1–21. [8] [PubMed: 20871802]

Sun Y, and Stein ML (2016), "Statistically and Computationally Efficient Estimating Equations for Large Spatial Datasets," Journal of Computational and Graphical Statistics, 25, 187–208. [3]

Tian L, Alizadeh AA, Gentles AJ, and Tibshirani R (2014), "A Simple Method for Estimating Interactions between a Treatment and a Large Number of Covariates," Journal of the American Statistical Association, 109, 1517–1532. [9] [PubMed: 25729117]

Tian Y, Schuemie MJ, and Suchard MA (2018), "Evaluating Large-Scale Propensity Score Performance through Real-World and Synthetic Data Experiments," International Journal of Epidemiology, 47, 2005–2014. [8] [PubMed: 29939268]

Trefethen LN, and Bau D (1997), Numerical Linear Algebra, Philadelphia, PA: Society for Industrial and Applied Mathematics. [3,11,12]

Van der Vorst HA (2003), Iterative Krylov Methods for Large Linear Systems (Vol. 13), Cambridge: Cambridge University Press. [11]

Walker AM, Patrick AR, Lauer MS, Hornbrook MC, Marin MG, Platt R, Roger VL, Stang P, and Schneeweiss S (2013), "A Tool for Assessing the Feasibility of Comparative Effectiveness Research," Comparative Effectiveness Research, 3, 11–20. [9]

Xu X, and Ghosh M (2015), "Bayesian Variable Selection and Estimation for Group Lasso," Bayesian Analysis, 10, 909–936. [10]

Zhang L, Datta A, and Banerjee S (2019), "Practical Bayesian Modeling and Inference for Massive Spatial Data Sets on Modest Computing Environments," Statistical Analysis and Data Mining: The ASA Data Science Journal, 12, 197–209. [3,4,11]

Zhou Q, and Guan Y (2019), "Fast Model-Fitting of Bayesian Variable Selection Regression Using the Iterative Complex Factorization Algorithm," Bayesian Analysis, 14, 573–594. [3] [PubMed: 31608133]

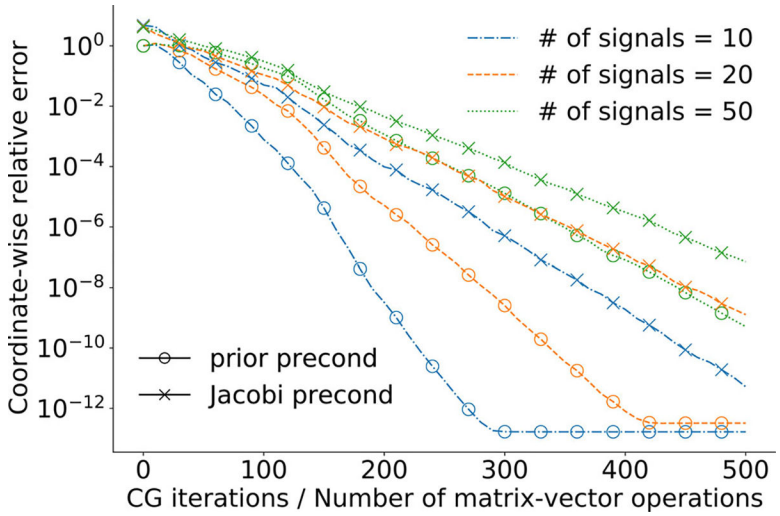
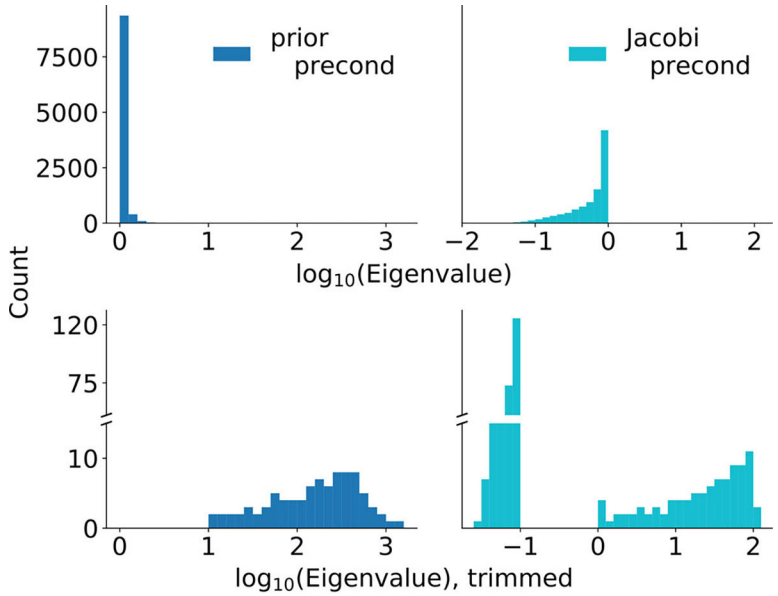
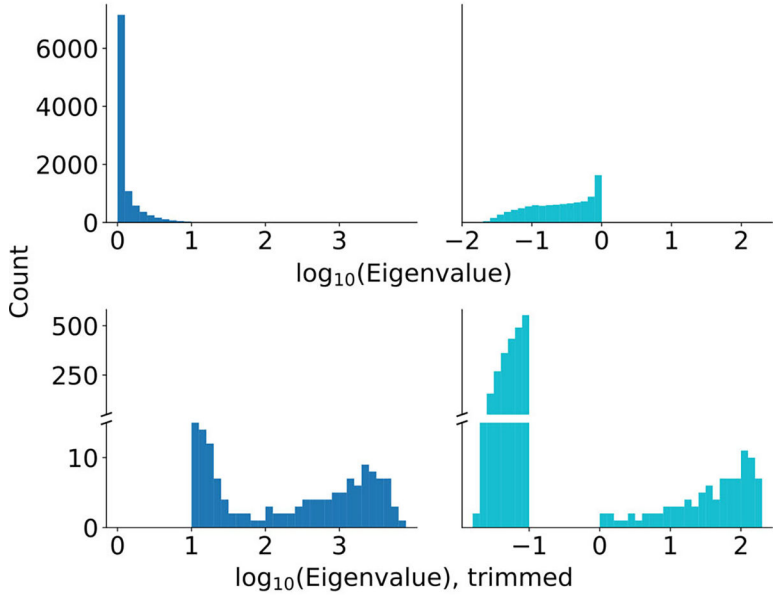


Figure 1.

Plot of the CG approximation error versus the number of CG iterations. The CG sampler is applied to the posterior conditionals based on synthetic data. The different line styles correspond to the different numbers of true signals in underlying data. The circle and cross markers denote the uses of the prior and Jacobi preconditioners.



(a) Based on synthetic data with 10 non-zero coefficients.



(b) Based on synthetic data with 50 non-zero coefficients.

Figure 2.

Histograms of the eigenvalues of the preconditioned matrices. The eigenvalues under the prior preconditioner are shown on the left and those under the Jacobi on the right. Shown on the lower rows are the trimmed versions of the histograms, in which we remove the eigenvalues in the range [0, 1] in the $\log_{10}$ scale for the prior preconditioner and those in the range [−1, 0] for the Jacobi. The y-axes for the trimmed histograms have intermediate values removed to make small counts more visible. The width of the bins are kept constant

throughout so that the *y*-axis values of the bars are proportional to probability densities and thus can be compared meaningfully across the plots.

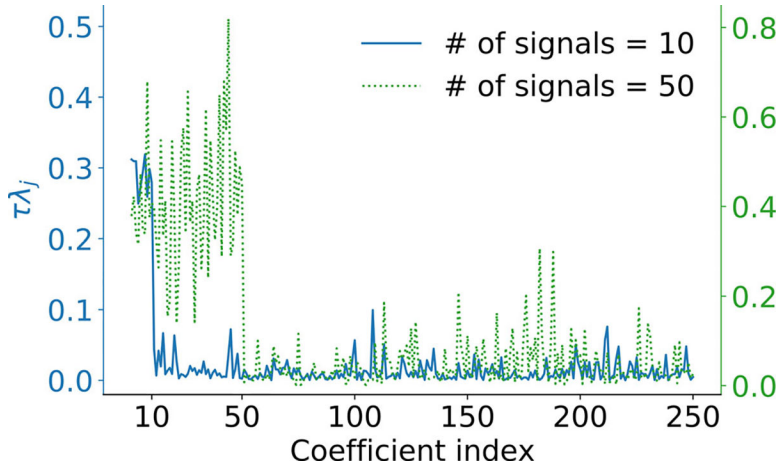


Figure 3.

Plot of the posterior samples of $\tau\lambda_j$'s for $j = 1, \ldots, 250$. The solid blue line and dashed green line correspond to the datasets simulated with $\beta_{\text{true},j} = 1$ for $j \leq 10$ and for $j \leq 50$, respectively.

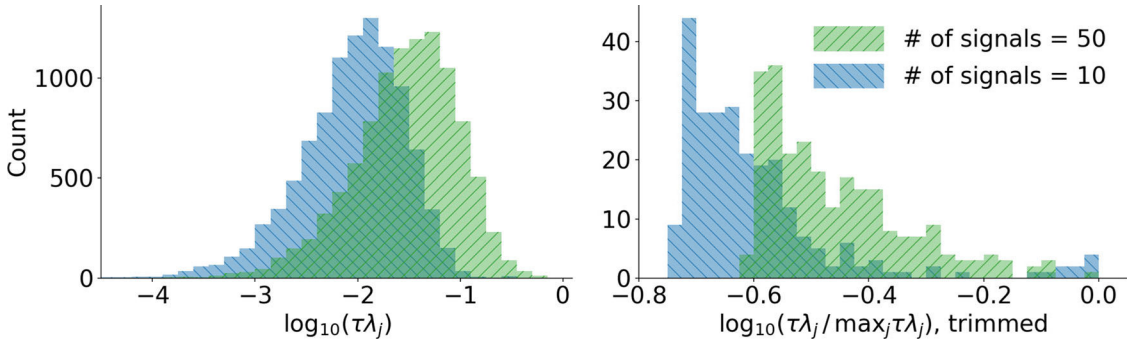


Figure 4.

Histograms of the posterior samples of $\tau\lambda_j$'s. The two different colors indicate the distinct posteriors with 10 and 50 nonzero regression coefficients. To better expose the relative tail behaviors in the two distributions of $\tau\lambda_j$'s, the histograms on the right take the 250 largest values and plot their magnitudes relative to $\max_j \tau\lambda_j$.

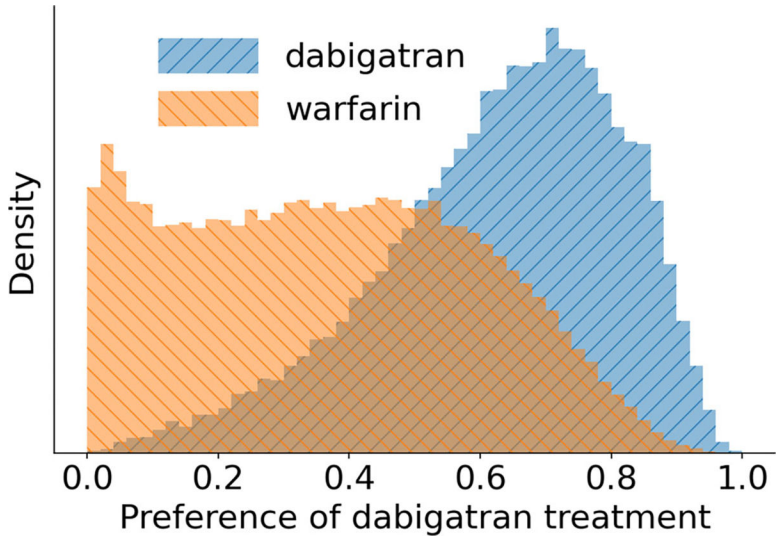


Figure 5.

Normalized histogram of the estimated preference scores for each group. Preference score transforms raw propensity score to make it a more interpretable measure (Walker et al. 2013).

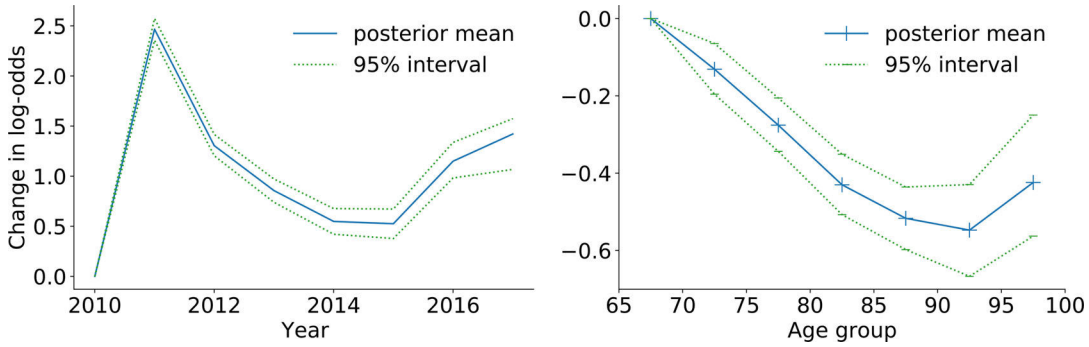
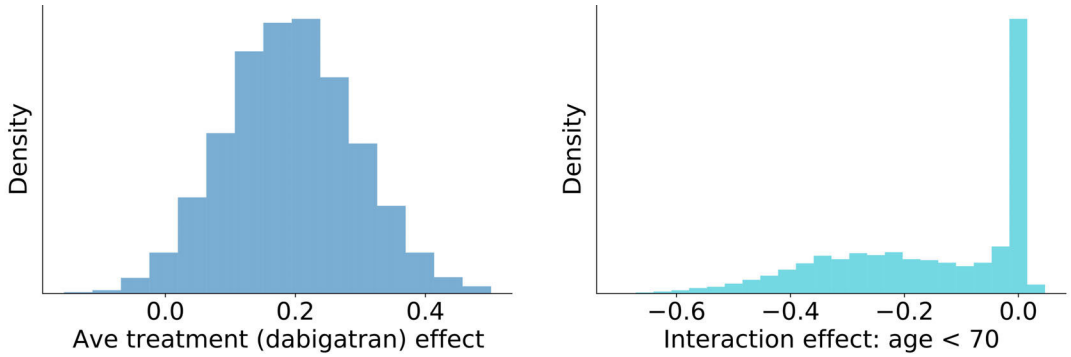


Figure 6.

Posterior means and 95% credible intervals for the regression coefficients of the treatment year and age group indicators. The age groups are divided into 5-year windows.



(a) Average effect of treatment by dabigatran over warfarin on gastrointestinal bleeding.

(b) Coefficient of the interaction between treatment and age group 65–69.

Figure 7.
Posterior distributions from the treatment effect model.